Rail to Digital Automated up to autonomous Train Operation

D7.1 – Requirements and specifications for Data Factory

Due date of deliverable: 01/04/2024
Actual submission date: 30/04/2024

Leader/Responsible of this Deliverable: Dr. Philipp Neumaier | DB InfraGO AG

Reviewed: Y

Document status		
Revision	Date	Description
01	03/01/2024	First Draft and structure
02	17/03/2024	Second Draft: Requirements added
03	19/03/2024	Third Draft: Conclusion draft added
04	28/03/2024	Final Version for TMT Process
05	06/08/2024	Proposed TMT Changes regarding Ontology added (2.4.3)
06	23/09/2024	Minor changes regarding R2DATO Logo and Footer
07	17/12/2024	Changes regarding feedback from external reviewer

Project funded from the European Union's Horizon Europe research and innovation programme		
Dissemination Level		
PU	Public	X
CO	Confidential, restricted under conditions set out in Model Grant Agreement	

Start date: 01/12/2022 Duration: 18 months

ACKNOWLEDGEMENTS

This project has received funding from the Europe's Rail Joint Undertaking (ERJU) under the Grant Agreement no. 101102001. The JU receives support from the European Union's Horizon Europe research and innovation programme and the Europe's Rail JU members other than the Union.

REPORT CONTRIBUTORS

Name	Company	Details of Contribution
Dr. Philipp Neumaier	DB	Content creator
Sebastian Dubiel	DB	Content creator
Martin Köppel	DB	Content creator
Daniel Obermeyer	DB	Content creator
Patrick Denzler	DB	Content creator
Jochen Lölkes	DB	Content creator
Sascha Geulen	DB	Content creator
Christian C Buchta	DB	Content creator
Martin Jungklaus	DB	Content creator
Martin Boekhoff	DB	Content creator
Waseem Ul Aslam Peer	DB	Content creator
Florian Reiniger	DB	Content creator
Dr.-Ing. Volker Eiselein	DB	Content creator
Alexander Slotta	DB	Content creator
Koraltan Kaynak	DB	Content creator
Betül Sögütlü	DB	Content creator
Christian Schultze	DB	Content creator
Wolfgang Albert	DB	Content creator
Franziska Lange	DB	Content creator
Frederic Antoine	ATSA	Expert Review
Dominik Kevicky	AZD	Expert Review
Saro Thiyagarajan	FT/Wabtec	Expert Review
Lars-Kristian Vognild	NRD	Expert Review
Tom Jansen	NS	Expert Review
Sebastiaan Linssen	NS	Expert Review

Rao Xiaolu	SBB	Expert Review
Oliver Lehmann	SMO	Expert Review
Dr. Thomas Waschulzik	SMO	Expert Review
Philippe David	SNCF	Expert Review
Hardik Jain	THD/Thales	Expert Review
Maik Baehr	Siemens	Expert Review
Bastian Simoni	Alstom	Expert Review
Lars Bergmann	Siemens	Expert Review
Michele Bruzzo	Hitachi	Expert Review
Hájek Jiří	AZD	Expert Review
Michal Novak	AZD	Expert Review

EXECUTIVE SUMMARY

Context and Objectives: The European railway industry is undergoing a significant technological transformation, with many infrastructure managers, railway undertakings and companies aiming for extensive automation and digitization of rail operations. This push is geared towards enhancing capacity and reliability in rail operations.

In this context, the R2DATO project aims to revolutionize the European railway infrastructure through digitalization and automation, leading up to autonomous train operations.

The surge in automation and digitization necessitates a parallel increase in data generation and utilization within the rail system. Particularly with regards to fully automated driving (referred to as automation level 4, or GoA4), the integration of sensors and cameras is crucial for automatic response to hazards in the rail environment through artificial intelligence (AI). Developing AI software for environmental perception, for instance, demands vast amounts of high-quality data. However, it's evident that individual rail entities may struggle to amass sufficient sensor data for adequately training AI systems for fully automated rail operations.

There is therefore a growing consensus that a Data Factory is essential - an ecosystem that enables sensor data to be collected, simulated, processed and shared, as well as AI-models to be generated, certified and implemented in automated rail operations.

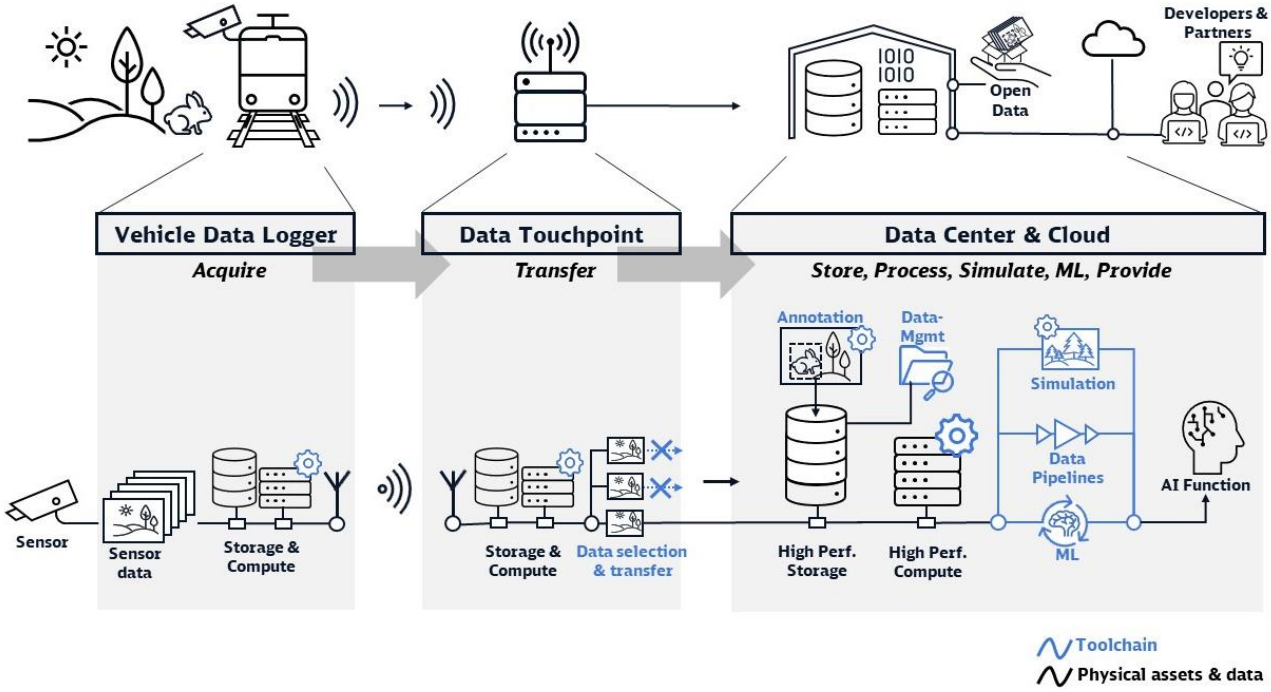


Figure 1: Simplified target configuration of the Data Factory

In Germany, Deutsche Bahn has launched the sector initiative Digitale Schiene Deutschland (DSD) to facilitate this technological transformation and has set up a Data Factory prototype [20] for the development of GoA4 fully automated driving.

In the R2DATO work package WP7, the further development and expansion of the Data Factory was driven forward and stakeholder requirements for the system were compiled. This further sharpened the picture of the target functionalities and system and subsystem architectures were derived.

This document presents the target configuration of the Data Factory and contains the system and subsystem requirements, as well as the requirements for data quality, annotations and security.

This study is closely related to the HADEA project CEF2 Pan-European Railway Data Factory [4], a collaborative ecosystem consisting of several individual Data Factories that enables railway infrastructure managers and railway companies to collect, process and exchange sensor data through a standardized infrastructure.

This document therefore presents the architecture of a Data Factory that can be integrated into a Pan-European Data Factory.

Figure 1 shows the simplified target configuration of the Data Factory with the data flow from the vehicle via the Touchpoint to the Data Centre. The basic functionalities, which will be discussed in more detail later in the document, are described in italics. The physical IT assets and the data are shown in black and the toolchain with the most important platform functionalities is shown in blue.

Vehicle Data Logger: At the vanguard of data acquisition, the Vehicle Data Logger is a designated to record onboard data on trains, capturing sensor data and crucial operational metrics and to transfer the data manually or wirelessly to the track-side (Figure 2 and sections 2.5.1, 2.5.5, 2.7.10).

Data Touchpoint: Strategically placed Data Touchpoints are used for the automatic, wireless and secure transfer of data from the Vehicle to the Data Centre (see Figure 3 and section 2.7.3). The Touchpoints also perform data processing and prioritisation so that only data of high quality and importance needs to be transmitted.

Figure 2: Vehicle Data Logger at DB InfraGO

Figure 3: Data Touch-point at DB InfraGO

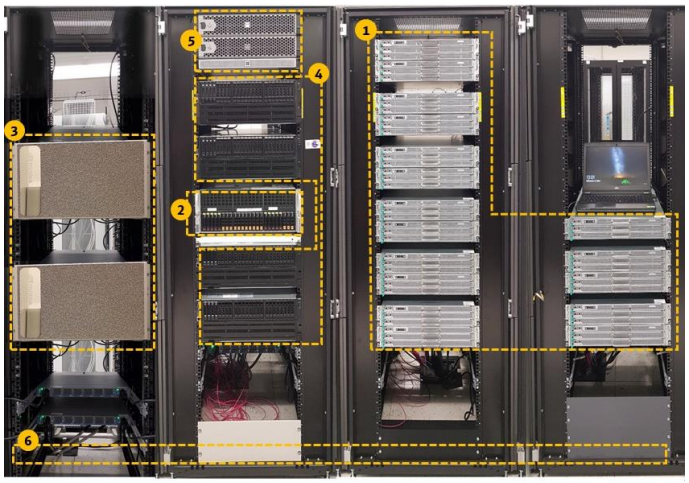


Figure 4: Data Factory HPC infrastructure at DB InfraGO Data Centre

Data Center with HPC infrastructure and Toolchain: This part of the Data Factory forms the core of functionalities to aggregate, process and store huge amounts of data. The infrastructure is scalable and will enable developers and partners to access datasets and to perform calculations.

The current state of the high-performance compute and storage system in the data centre of DB InfraGO is visualized in Figure 4 and the yellow numbers highlight the individual assets:

1.) Storage system with 5 PB

2.) Ingestion system to incorporate data respectively sensor data, also from physical SSDs

3.) HPC cluster for machine learning, training and evaluation of AI-models

4.) HPC cluster for simulation of virtual railway scenarios and synthetic sensor data

5.) Connectivity to cloud services and the provision of the data management

6.) Symbolic representation of the spine-leaf LAN infrastructure to connect all assets

Moreover, the final state of the Data Factory will include a toolchain, which is formed from various subsystems:

- The Simulation Platform subsystem, which generates data for various operational scenarios and synthetic sensor data; see chapter 2.7.8.
- The Data Platform, which is the central subsystem where data is stored, processed and provided to either other subsystems or users; see chapter 2.7.2.
- The ML Platform subsystem, which offers methods to generate model architectures, train and evaluate machine learning models e.g. for object detectors, predictive analytics or automated decision-making; see chapter 2.7.7.
- The Integration Platform, which ensures provision of compute resources to all subsystems and an interactive compute environment to the users; see chapter 2.7.4.
- The IT System Management, which monitors the subsystems and monitors the activities to ensure the integrity and performance of the Data Factory; see chapter 2.7.6.
- The System Access & Security Platform, which provides functions for identification and authentication, use control, system integrity, data confidentiality, restricted data flow, timely response to events and resource availability; see chapter 2.7.9.
- The Accounting Service, which ensures a transparent billing associated to utilization and resource usage; see chapter 2.7.11.
- Interconnect Platform, which ensures the connection to external IT assets and other Data Factories and an integration into a Pan-European Data Factory; see chapter 2.7.5

Main Findings: The development of a Data Factory is about the comprehensive integration of data and data processing technologies with traditional railway systems. Key components such as sensor data acquisition with a Vehicle Data Logger, Data Touchpoints, edge components and IT assets in the Data Centre as well as a uniform toolchain are to be designed in such a way that they form a uniform platform overall.

This platform should in turn be integrated into the concept of the Pan-European Data Factory (PEDF) [5][15], in which individual Data Factories or individual IT assets of the members are connected to each other, realised with a high-speed backbone network [8] and implemented by defining common data standards and interfaces. In the vision, the final expansion stage will comprise a uniform and shared toolchain.

This merger is intended to achieve synergy effects that will enable to master the complex tasks of generating permissible AI functions for automated train operation in accordance with GoA4.

Recommendations for the Future: Building on Deliverable 7.1 as a basis, the Data Factory should be further developed and gradually integrated into the PEDF as described in [13] and [14]. It must be ensured that the Data Factory remains adaptable and compatible with various railroad systems and third-party technologies.

In summary, Report 7.1 is a testament to the collaborative efforts undertaken within the R2DATO project to create a data-centric future for rail transportation. It serves as a strategic dossier for the successful creation of a data and AI platform for the implementation of functions for GoA4 automated train operation.

ABBREVIATIONS AND ACRONYMS

Term	Explanation
ADAS	Advanced Driver Assistance Systems
AI	Artificial Intelligence
ATO	Automatic Train Operation
ATP	Automatic Train Protection
BEV	Bird-eye view
CI/CD	Continuous Integration and Continuous Deployment
CV	Computer Vision
DA	Domain Adaptation
DAFA	Data Factory: Refers to the system responsible for acquiring data on trains, likely an acronym for a specific technology or project within the Data Factory scope.
DB	Deutsche Bahn
DDBB	Database
DL	Deep Learning
DM	Digital Map
DMS	Driver Monitoring System
DOA	Deed of Agreement
DTP	Data Touchpoint
DZSF	Deutsches Zentrum für Schienenverkehrsforschung (German Centre for Rail Traffic)
EBA	Eisenbahnbundesamt (German Federal Railway Authority)
ERA	European Union Agency for Railways
ETCS	European Train Control System
FN	False Negative
FOV	Field of view
FP	False Positive
FRMCS	Future Railway Mobile Communication System
FTP	File Transfer Protocol
FTS	File Transfer Service
GAN	Generative Adversarial Network

GDPR	General Data Protection Regulation
GNSS	Global Navigation Satellite System
GoA	Grade of Automation
HDFS	Hadoop Distributed File System
HiL	Hardware in the Loop
HPC	High-Performance Computing: A big set of CPU and GPU resources available for calculation (mainly deep learning tasks)
HTTPS	HyperText Transfer Protocol Secure
IoT	Internet of Things
IMU	Intertial Measurement Unit
IPM	Incident Prevention Management
IR / NIR	Infra-Red / Near Infra-Red
JSON	JavaScript Object Notation
LAN	Local Area Network
LiDAR	Light Detection and Ranging
LRV	Light Rail Vehicle
mAP	Mean Average Precision
MCG	Mobile Communication Gateway
MEMS	Micro-Electro-Mechanical Systems
ML	Machine Learning
MQTT	Message Queue Telemetry Transport
NLP	Natural Language Processing
OB	On-Board
ODD	Operational Design Domain
OLAP	Online Analytical Processing
ORD	On-board Recording Device
OS	On-Sight
OSDaR23	Open Sensor Data for Rail 2023 is a diverse sensor dataset for railway environment research and AI development
PE	Physical Environment
PEDF	Pan-European Data Factory
PER	PERception

PIR	Passive InfraRed
PIS	Passenger Information System
POC	Proof Of Concept
PRM	Person with Reduced Mobility
PT	Post Trip (ETCS mode)
PTU	Physical Train Unit
R2DATO	Rail to Digital automated up to autonomous train operation
RADAR	RAdio Detection And Ranging
REP	Repository
RGB	Red, Green, Blue
RINF	Register of INFrastructure
ROS	Robot Operating System
RTMaps	Real-Time Multisensor Applications
RTP	Real-time Transport Protocol
RU	Railway Undertaking
SFTP	Secure File Transfer Protocol
SiL	Software in the Loop
SMO	Siemens Mobility
SNCF	Société nationale des chemins de fer français
SR	System Requirement
SSD	Solid State Drive
SSH	Secure Shell
SuC	System under Consideration
T2G	Train to Ground
TAURO	Technologies for the Autonomous Rail Operation
TCN	Train Control Network
TDI	Time Delay Integration
UC	Use Case
UDA	Unsupervised Domain Adaptation
UIC	International Union of Railways
UTC	Universal Time Coordinated
VCD	Video Content Description

VDL	Vehicle Data Logger
WAN	A network that covers a broad area (e.g., across cities, regions, countries) used for computer networking.
WP	Work Package

Term	**Explanation**
Access and Permission Manager	A manager that controls access and permissions within the system.
Acquisition Frequency	The acquisition frequency is the rate at which the sensors trigger. It determines the delta between two consecutive frames, in particular the delta between two Acquisition Timestamps.
Acquisition Timestamp	The acquisition timestamp is the time when a frame is physically captured. The physical event that is stamped with the acquisition time has to be defined for each sensor modality. In terms of camera this can be any time from the opening to the closing of the shutter. In terms of lidar this can be any time from sending the light impulse to detecting (receiving) the reflection. For sensors that record multiple returns (such as LiDAR), each return has its own time stamp or an offset to the time stamp for a set of returns (e.g. a point cloud).
Alert Manager	A manager that handles alerts and notifications within the system.
Annotation	Metadata that aims at providing ground truth for the data it refers to; annotations are the result of manual or automatic labelling.
Annotation Platform	A dedicated environment or set of tools in the Data Platform for annotating and managing workflows related to data annotation.
Asset Manager	A subsystem managing digital or physical assets within the system.
Automated Offload	A subsystem responsible for the automatic offloading of data or processes.
Billing Manager	A subsystem managing billing and financial transactions.
Connect Manager	A manager responsible for overseeing system connections.
Coupled Localisation	The coupled localisation is the localisation information that was calculated based on multiple inputs. A coupled localisation is usually combining input from one or multiple GNSS and IMU devices as well as using perception information in combination with a digital map.
Data Augmentation	Adding information in the data, e.g., for an image, it can be an incrustation (reference to augmented reality)

Data Factory	The Data Factory for Rail is a service to manage necessary data for training ML models for automated train operations. This includes, for example, sensor based perception to monitor the track area. For more details regarding the Data Factory cf. [20]
Data Logger	Device that captures and persists the data streams in the system in a way that allows real time replay of the data.
Data Management	A suite of tools and systems within the Data Platform designed for data querying, permission setting, deletion, and listing available data.
Data pipeline	Data pipelines process the data streams and generate the metadata based on the data.
Data Pipelines	Part of the Data Platform that manages data operations such as ingestion, transformation, storage, and integrity validation.
Data stream	A data stream contains cohesive measurements from a single modality or function over time. A modality can be for example a sensor and a function can be an object detector running on the train. A data stream is part of a recording.
Data Touchpoint	An interface or platform within the Data Factory responsible for offloading and transferring data from the train to the Data Centre.
Dataset	A curated collection of frames and metadata that can be used for ML training. Each dataset has properties that enable training for a certain purpose.
DC Transmitter	A direct current transmitter that sends electrical signals.
Diagnostics	Tools or systems used within the Vehicle Data Logger to diagnose or check the data for integrity and sanity onboard the train.
Digital Twin	A Digital Twin is a real-time virtual representation of a physical object or system, used for simulation, analysis, and optimization to enhance decision-making and operational efficiency.
Digital Twins Manager	A manager responsible for overseeing the digital twins within the system.
Display Manager	A subsystem responsible for managing and controlling display outputs.
Egomotion	The egomotion is the movement of the vehicle at a specific point in time. It is usually provided by an IMU unit and contains velocity and acceleration in all three dimensions.
Ego-vehicle	The train that records the data.

Frame	A single measurement from one sensor like an infrared camera, visual camera, lidar, or RADAR.
Frame Drop	A frame drop is the event of missing the sensor information for a specific point in time at which a message was expected according to the Acquisition Frequency. Frame drops can happen due to a failure in performing the message, as well as any loss of this information on the way to the subsystem expecting the message. A Frame Drop leads to a delta between to consecutive Frames that is twice as large as expected. Multiple Frame Drops can occur successively.
Frame metadata	Describes information about a single frame, such as weather condition, train speed, number or type of detected objects. Everything that can be derived from existing data (e.g., in data pipelines) is also called metadata.
Frame view	An overview to list and search all available frames in the data management system. It is complementary to the recording view.
Generator	A subsystem that generates data or signals.
Hardware Configuration	The hardware configuration describes the set of hardware that is used and affects the sensor data. It consists of serial numbers of sensors and sensor accessories (such as lenses), the lens configuration and identifier of other hardware parts used in the system.
Highest Quality	The highest quality of sensor data is the configuration of the sensors that delivers all information that the sensor is physically able to capture and designed to provide to the user. This means, as an example, when the sensor can capture images with 16bit image depth, choosing 16bit means higher quality that choosing 12bit as the 16bit image contains more information in terms of dynamic range. Parameters that affect that data quality are, among others, the acquisition frequency, the resolution, the point density (LiDAR), the dynamic range (camera). As there are usually technical limitations for the system, there is usually a trade-off to get the highest possible data quality for the respective use-case given the system's performance.
Integration Manager	A subsystem focused on integrating various components and services.
Integration Platform	A platform designed to integrate various applications into the Data Factory ecosystem, providing computation and connection services.

IntegrityGuardian	A specialized tool or system within the Vehicle Data Logger designed to ensure and verify the integrity of the recorded data.
Interactive Computation Environment	An environment allowing for interactive data analysis and computation.
Interface Platform	A system in the Data Factory for managing connections to external IT assets like other data factories, data sources, or cloud services.
IT Service Management	A management layer responsible for accounting tasks like billing and resource logging within the Data Factory ecosystem.
Labeling	Similar to annotation, it refers to adding descriptive tags to datasets to facilitate identification and organization of the data.
Localisation	The localisation is the position of the vehicle in reference to the world coordinate system at a specific point in time. It is usually provided by a GNSS unit and the positions refers to a world coordinate system such as UTM.
Member / DF member	Company (or assimilated to) which gets a software interactivity with the Pan-European Data Factory
Member profile	List of the interfaces and capabilities of a member in the PEDF
Metadata	Metadata is data that provides information about other data, but not the content of the data itself, such as the text of a message or the image itself. E.g. metadata is the time and date on which a picture was taken.
ML Architecture Manager	A manager responsible for the architectural design of machine learning systems.
ML Model Comparer	A subsystem that compares different machine learning models.
ML Model Manager	A subsystem responsible for overseeing machine learning models.
ML Model Trainer	A subsystem involved in training machine learning models.
ML Platform	A platform within the Data Factory focused on machine learning model management, including training, versioning, comparison, and integration.
Modality	In general, available modalities can entail infrared images, visual (RGB) images, point clouds, and radio frequency images (e.g., from a birds-eye view (BEV) RADAR). In our use cases, each sensor is considered its own modality. In OSDaR23, this means there are twelve modalities: three infrared cameras, six visual

	cameras, one merged lidar point cloud, one BEV RADAR sensor, and the localization sensor.
Monitor Manager	A manager overseeing the monitoring processes within the system.
Monitoring collector	A system that monitors and logs the health status and events related to the Vehicle Data Logger's operation.
Multi-modal frame	A set of frames captured at the same or a similar point of time. In OSDaR23, this concept is called Multisensor-Frame.
Multi-modal video	Multi-modal video is a time series that contains synchronized recorded data from all the sensors. The stream can be played similar to a "video".
Object	On every single image, objects such as people, rails or catenary poles can be identified by a person who annotates the images. We use annotations to clearly identify the recognized objects.
PEDF Consortium	The set of members that constitute the PEDF organization
PEDF Organization	Legal Organization of the PEDF. It may be an association, an European Economic Interest Grouping, a Memorandum of Understanding, etc.
Perception Sensor	Perception sensors are all sensors with the purpose of perceiving the environment with the purpose of recognising objects within this environment. This includes visual cameras, infrared cameras, event cameras, stereo cameras, depth cameras, lidars, RADAR and ultrasonic sensors.
Recorder	A component of the Vehicle Data Logger tasked with recording specific data types, such as sensor, functional, and train data.
Recording	A collection of data about a continuous time range that can contain sensor data and metadata. It contains multiple data streams. A recording consists of multiple data streams as depicted in Fig. 4.
Recording metadata	Describes information about a whole recording, such as sensor setup, train identifier, and recording length. This also includes tags and comments.
Recording view	An overview to list and search recordings in the data management system. It is complementary to the frame view.
Reference Time	The reference time is a common time that is used by all subsystems. Usually, it is provided by a reference clock (master clock) within the system. The reference clock provides a protocol for all other clocks in the system to synchronise to the reference time.

Resource Manager	A manager responsible for allocating and managing resources.
Scenario Sampler	A subsystem that samples various scenarios for testing or analysis.
Scene	A continuous time frame which is a subset of one recording.
Sensor Data	Data collected by sensors, reflecting various inputs from the environment, such as temperature, location, or movement.
Sensor Data Stream	A continuous stream of measurements (Sensor Frames) from a sensor.
Sensor Frame	A sensor frame is the captured representation of the environment at a specific point in time. A similar term is sensor measurement. Examples: A camera image is a frame, a lidar point cloud is a frame, a GNSS position is a frame.
Sensor MFrame	A multimodal sensor frame (M-Frame) is the captured representation of multiple sensors at the same point in time (or at a very similar point in time). The synchronicity threshold for a multimodal frame the must be defined for each specific use case.
Sensor Parameters	The sensor parameters describe the configuration of the sensors that is used to capture the sensor data. These are parameters that can be configured and are not fixed for a specific sensor model. The available parameters depend on the sensor type and sensor specifics. Camera parameters are, for example, iso, shutter speed, exposure time, sampling frequency, focal distance, and aperture. Lidar parameters are, for example, field of view, update rate, number of returns and resolution.
Simulation Platform	A component of the Data Factory tasked with generating synthetic data for specific scenarios, managing assets, and validating simulation data.
Software Version	The software versions of all components that affect the sensor data, especially the driver software versions.
Steering committee	Steering committee or comparable group of members able to take decisions in the PEDF regardless of the chosen organization.
System Access & Security Platform	A security layer in the Data Factory that manages user access and permissions according to specific security standards.
System Logging	A subsystem that records system activities and events.
Topic	A data stream channel that contains only data of the same format, type, and purpose.

Transmitter	Part of the Vehicle Data Logger that handles the transfer of recorded data to another location or system.
User Logging	A subsystem that records user activities and interactions.
Validator	A subsystem responsible for validating data or processes.
Vehicle Data Logger	A device or system installed in vehicles (trains, in this case) to records, transfers, and diagnoses various types of data during operation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 ABOUT DELIVERABLE 7.1

This document serves as Deliverable 7.1, titled "Requirements and Specifications on Data Factory" underpinning the development work carried out in the context of the Shift2Rail initiative. Specifically, this deliverable fits within the framework of the Technology Demonstrator (TD) and Work Area (WA) corresponding to Information Technology Solutions for Train Operation.

The R2DATO project contributes to the broader Shift2Rail objectives by targeting advancements in rail system automation, pushing forward the boundaries toward fully autonomous train operations. The work detailed herein contributes to the accomplishment of key objectives set out for TD/WA on Next-Generation Train Control and Communication Systems, achieving a Technology Readiness Level (TRL) that demonstrates a significant step forward from the conceptual stage to the system prototyping in operational environments.

Background and Project Contribution: The rail industry is at a pivotal junction, with digitalization promising to redefine the performance, reliability, and safety standards of rail operations. Within this evolutionary path, the R2DATO project stands out with its specific contribution to the Shift2Rail program. It aims to leverage data-driven methodologies to enhance the control, monitoring, and predictive capabilities of European rail operations.

Objectives and Aim of this Deliverable: The primary objective of this deliverable is to outline the requirements and specifications of a Data Factory and the data itself. The data serves as backbone for analytics and machine learning to develop AI functions for GoA4 automated train operation (ATO). The Data Factory, particularly in combination with the concept of the Pan-European Data Factory (PEDF) is envisioned as a central hub for collecting and storing huge amount of data for Machine Learning tasks, supporting the Shift2Rail's goals of increasing efficiency and advancing toward autonomous train operation.

Motivation: The setup of the Data Factory is motivated by the need to develop fully automated driving with AI-functionalities for environment perception, incidence management, localization and other functionalities.

To achieve this, huge amounts of data needs to be collected, transferred, processed an annotated, also synthetic data needs to be simulated to generate data of rare railway scenarios. With this data machine learning is performed to train AI-models and the data also serves for the evaluation and testing. At the end of the process a platform is needed for certification and homologation of these functions (see Figure 5).

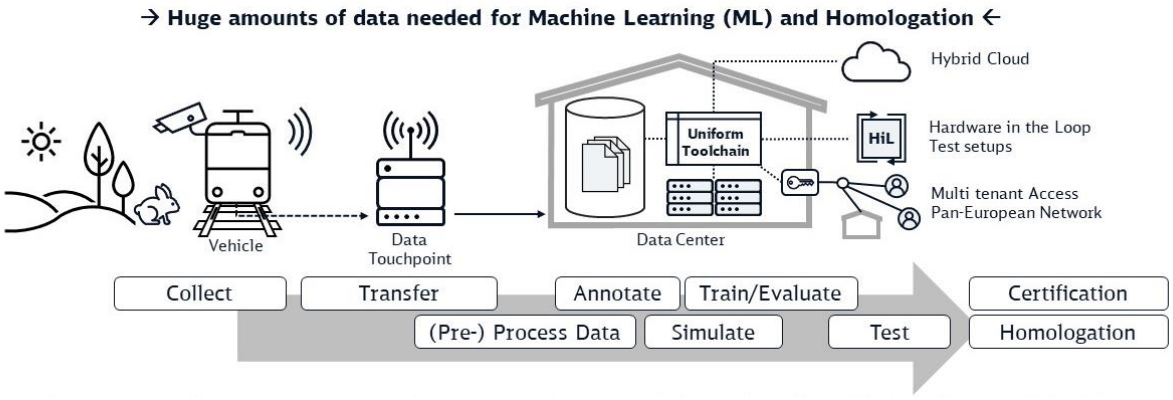


Figure 5: Motivation for a Data Factory

We except that these complex tasks cannot be carried out by the railway suppliers, Infrastructure Managers (IMs) and Railway Undertakings (RUs) independently by themselves.

This document outlines the needs placed on the Data Factory and contains the system and subsystem requirements with its logical architectures.

In a further step multiple individual Data Factories or single assets shall be interconnected to form a Pan-European Data Factory (PEDF, see section 2.3.3) with a consortium setup that also may include safety authorities. Within the PEDF common data standards and interfaces will be defined and a uniform toolchain will leverage synergy effects by greatly increasing the amount of data available, while at the same time preserving the data sovereignty of the members.

Proposed Solution and Document Structure: The proposed solution involves a modular, scalable Data Factory architecture, consisting of train, trackside and Data Centre systems. The deliverable details the stakeholder needs, the derived requirements, functions and logical architectures.

1.2 THE STRUCTURE OF THIS DOCUMENT

Section 1 – Introduction: This section sets out the framework for the project, defines its place within the Shift2Rail initiative and outlines the requirements of the R2DATO project. Section 1.1 explains the objectives and motivation for the Data Factory and the connection with the PEDF.

Section 2 – Main Part: The main body of the document is divided into several subsections, each dealing with one aspect of the development of the Data Factory:

- Section 2.1: Business Requirements and Goals
- Section 2.2: Scenario, including Assumptions, Project Setup Based on Experience, and Exclusions and Legal Considerations
- Section 2.3: Connected Work Packages or Pre-Projects, exploring influences from projects like X2Rail4 and Tauro Shift2Rail
- Section 2.4: Dependencies to other Work Packages, which examines the interrelation with other key WPs such as WP11, WP27, and WP43
- Section 2.5: Data Requirement Sspecification, offering detailed requirements for Sensor Data, Data Models, and Annotation
- Section 2.6: Stakeholder Needs, System Security Requirements, and System Description
- Section 2.7: Subsystems, providing a deep dive into each subsystem of the Data Factory, such as the Dataflow Diagram, Data Platform, and Accounting Services.

Section 3 – Conclusions: This section synthesizes the findings and the journey of the project, summarizing the achievements in line with set objectives and reflecting on the lessons learned. It addresses issues and future recommendations, capturing the essence of the work and its implications for the advancement of Europe’s Rail.

Section 4 – Appendix: This ancillary section contains supplementary information, such as detailed Data Annotation Requirements, which support the main content of the document.

References: The document concludes with references, providing sources and additional reading material that underpins the content presented throughout the deliverable.

Each section of the document is designed to progressively build upon the preceding information, ensuring a logical flow that takes the reader from foundational concepts to detailed technical specifics. The structure facilitates an understanding not only of the Data Factory's current state but also of its trajectory towards enabling a more efficient and technologically advanced rail system.

2 MAIN PART

2.1 BUSINESS REQUIREMENTS AND GOALS

To meet the increasing demand for transportation of both passengers and freight, R2DATO will take the advantages of digitalisation and automation to develop the Next Generation ATO and deliver scalable Digital and Automatic (up to Autonomous) Train Operation (DATO) capabilities in order to enhance the capacity of the existing rail networks.

The goal of the Data Factory is to serve as a platform that provides data, in particular sensor data and simulated synthetic data, as well as machine learning functionality for training and evaluating AI-models. In the future, this platform will also stream data to HiL test stands and will provide data and tools for certification.

The goals of WP7 Data Factory are:

- D7.1: Development of the Data Factory specifications that is compatible with the PEDF concept and the ongoing expansion of the infrastructure
- D7.2: Legal assessment of the Data Factory
- D7.3: Joint simulation of scenarios and production of synthetic sensor data
- D7.4: Joint collection and annotation of sensor data
- D7.5: Generation of sensor-based object detectors for perception using machine learning
- D7.6: Provision of an Open-Data-Set consisting of real sensor data of the railway environment, associated annotations, simulated synthetic data and digital map data

The Data Factory has to be designed in such a way that it can collect and process data from up to 27 countries with different boundary conditions and rules. We assume that only some RUs, IMs and rail suppliers will set up a fully developed Data Factory based on the specifications provided here. Some stakeholders will probably only implement partial aspects or use the services of another stakeholder's data factory.

It must also be ensured that existing and planned Data Factories can be linked together in the future to form a Pan-European Data Factory (PEDF) [4], similar to GaiaX CartenaX [21].

We also assume that once the AI models have reached a certain level of quality, the amount of data to be collected will decrease over the years.

The goal of the task 7.1 is to collect the requirements and specifications for the Data Factory infrastructure and the toolchain for the purpose of data simulation, engineering, and ML/AI model training for GoA3/4 operation. Also, the specifications on the data itself and a sector-wide data ownership model towards the release of an Open-Data-Set will be carried out.

2.2 SCENARIO

The basic scenario is built on following Assumptions, Experiences of similar projects and Exclusions and Legal Considerations.

2.2.1 Assumptions

- Up to 60 TB of raw sensor data will be generated per train per day.
- For a basic data collection, the assumption is that at least 2 trains per country will be used.
- Data will be transmitted to a Data Centre or cloud for a subsequent data ingestion, preferably wirelessly or through other appropriate methods within the respective country or region.
- The Data Factory's setup (cloud-based, on-premise, or hybrid) will be determined after analysing use cases to find the optimal solution.
- It is assumed that the results of the trained machine learning (ML) models will be utilized for the perception systems.
- For certification and homologation huge amounts of data and a certified common toolchain are necessary.

2.2.2 Project Setup Based on Experience

- Data will be collected using specially equipped trains with cameras lidar, radar and localization sensors, with the main data volume coming from high-resolution cameras.
- Data will be transmitted into the Data Factory via network infrastructure.
- In the initial stage, data annotation will be manual and in later stages semi-automatic.
- The training and validation of AI-models will occur within the Data Factory.
- The Data Factory will also be responsible for testing AI-models.
- Regular monthly releases for training, a practice from the automobile sector, are expected.
- Up to 30 training cases may be used in parallel, another insight from the automobile sector.

2.2.3 Exclusions and Legal Considerations

- The rules for certification, to be determined by legal authorities, are not included in the work package WP7, but it is expected that all data used for training must be stored for future audits.
- Operational aspects of the trained ML models and vehicle onboard perception systems are excluded from the WP but have a significant connection with WP11.
- Operational components of the Data Centre and security configuration and operation are beyond the scope of the WP.

2.3 CONNECTED WPS OR PRE PROJECTS

Results of pre projects of the sector were included as far as the Data Factory is concerned. Also, the WP within R2Dato was considered.

2.3.1 X2Rail4

X2Rail-4, part of the Shift2Rail initiative, is focused on developing advanced signalling and automation systems for railways, including specifications for Grade of Automation 3/4 (GoA3/4). This project is crucial for integrating Machine Learning (ML) outcomes from data factories into the train's perception systems. While train-internal perception systems are outside this Work Package's scope, the alignment of ML results with X2Rail-4's architecture is essential for seamless integration and operational efficiency in railway systems. [17]

X2Rail4 delivers the specifications concerning GoA3/4. Results of the Data Factory ML training must fit to the architecture mentioned there.

Assumptions:

- X2Rail-4 is tasked with developing specifications for Grades of Automation 3 and 4 (GoA3/4) to guide the architecture of train perception systems.
- The results from the Data Factory's machine learning (ML) processes are expected to align with X2Rail-4's architectural specifications to ensure system compatibility and integration.

Boundaries:

- The project does not include vehicle internal perception systems within its Work Package (WP), delineating the scope of X2Rail-4's responsibilities.
- Clear interfaces and logical infrastructure alignment with X2Rail-4's standards are mandatory, even though direct involvement with vehicle internal perception systems is outside the project's direct scope.

2.3.2 Tauro Shift2Rail

Tauro is instrumental in advancing Shift2Rail's mission, concentrating on data management and AI integration for railway systems. It facilitates the introduction of sophisticated AI models into railway operation frameworks, enhancing system functionality and data-driven decision-making [3].

Aligning with Shift2Rail Objectives

- Data Platforms & AI-Data basis Role: Tauro provides the foundation for extensive, high-quality datasets, essential for training robust AI models that enhance railway operations and safety.
- Consistency with Shift2Rail's Framework: Tauro's methodology and outputs are designed to seamlessly mesh with Shift2Rail's broader architectural standards, ensuring interoperability and optimization.

Assumptions

- Development and Standardization: Tauro assumes the responsibility for the development and provision of standardized data management practices in line with Shift2Rail's automation goals.
- Machine Learning Integration: It is presumed that Tauro's machine learning outcomes will be consistent with the technical specifications outlined by Shift2Rail to facilitate GoA3/4 automation levels.

Boundaries:

- Scope of Work: While Tauro is pivotal in data and AI aspects, it does not extend its purview to the development of train-internal perception systems, which remains beyond its defined scope.
- Interface Specification: Tauro ensures the definition of clear interfaces and compatibility standards, aligning with Shift2Rail's infrastructure requirements while maintaining its specified boundaries.

Conclusion and Integration Pathway:

Tauro serves as a bridge between data-centric AI development and Shift2Rail's ambitious automation objectives, providing the necessary data infrastructure while adhering to set boundaries and integration specifications.

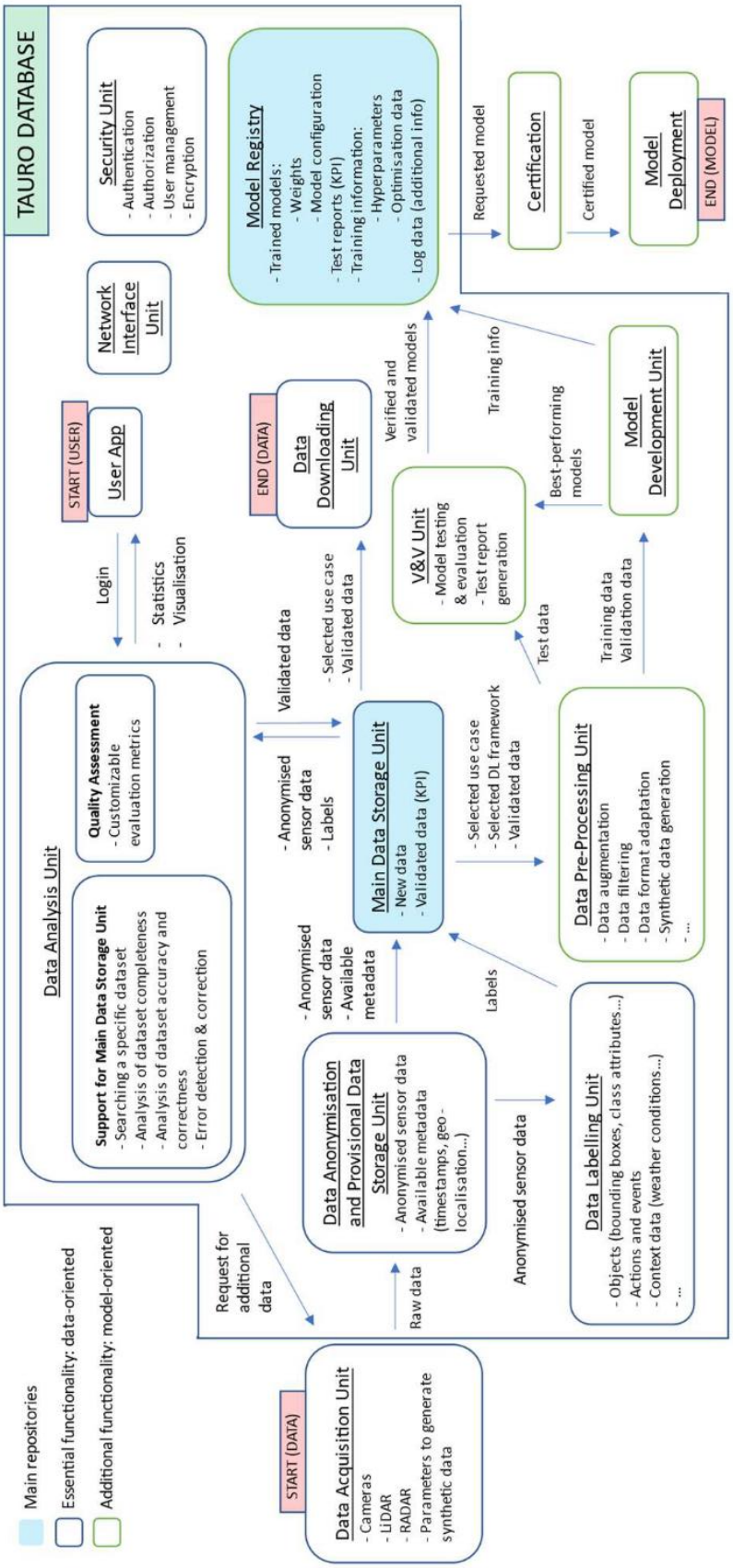


Figure 6: Logical architecture TAURO [3]

2.3.3 CEF 2 RailDataFactory

Railways in Europe are actively working on GoA4 automated rail operations. However, this requires the collection of large amounts of sensor data for AI training - a task that can be challenging for individual railroads or suppliers. To tackle this problem, the concept of a pan-European Rail Data Factory has proven to be a promising solution. The CEF2 RailDataFactory study [4], envisioned as a collaborative infrastructure and partner ecosystem, offers a way forward for RUs, IMs and railway suppliers.

This study was carried out and successfully completed in 2023 as a joint effort between DB, SNCF and NS. Co-funded by the European Health and Digital Executive Agency (HADEA), the study examined various aspects, including technical, operational, commercial, legal and strategic considerations.

The Pan-European Railway Data Factory (PEDF) is conceptualized as a collaborative infrastructure and ecosystem that integrates data from various sources to enable fully automated train operation across Europe [5] by creating a shared infrastructure for extensive sensor data collection, processing and AI development.

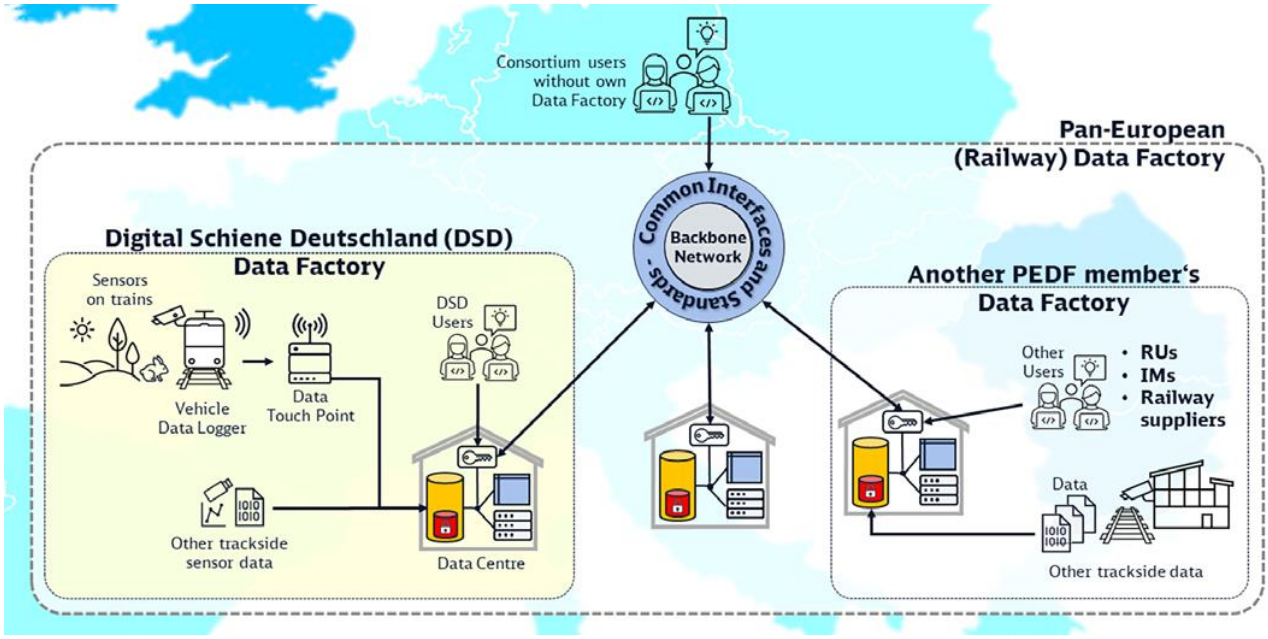


Figure 7: Pan-European Data Factory (PEDF) as union of many individual Data Factories

Figure 7 shows the PEDF as an interconnection and association of several individual data factories or individual IT assets that are to be connected via a high-speed backbone network [8]. The individual Data Factories that are part of the PEDF can exist in different expansion stages.

What is important, however, is the definition of common data standards and interfaces, as well as the prospective creation of a uniform common toolchain. The basic idea of the PEDF is that each member retains sovereignty over its data because the (private) data does not flow out of the respective Data Factory. However, the members can use the partners' data to conduct data analyses or train and evaluate AI models using the uniform toolchain. This construction is similar to GaiaX CartenaX, see section 2.3.5 and [21].

The strategy for building the PEDF is divided into short, mid and long-term time periods as shown in Figure 8. Initially, each stakeholder, whether RU, IM or railway supplier, has its own proprietary solutions. But in the mid-term, the first common standards for data and interfaces should help to

establish data compatibility, so that synergy potentials can be realised in the long-term by greatly increasing the basic data population for all members [14].

These efforts to create common interfaces and a common, uniform toolchain are to be bundled in the interface pillar and the toolchain pillar. These represent working groups and standardisation committees.

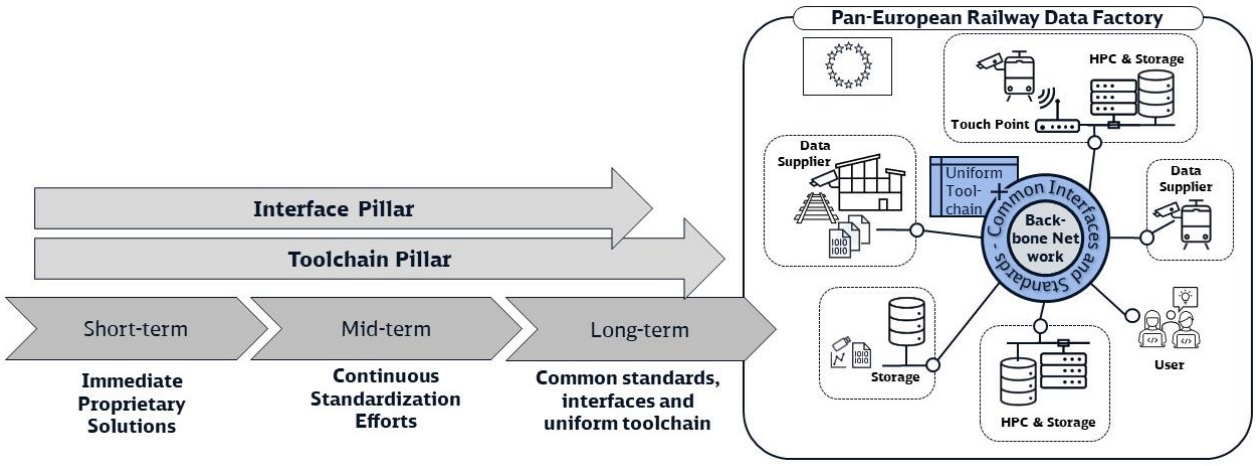


Figure 8: Strategy and time periods to build the PEDF

CEF2 RailDataFactory study carried out some initial use cases and requirements [5] and are the basis for the requirements in this document in section 2.7.5.

The building blocks and components elaborated in the CEF2 study [6] are compatible with those in this document (section 2.7).

The comprehensive cyber security assessment, the IAM concept and the data management concept from CEF2 [7] are complementary to the System Access and Security Platform in section 2.7.9 and the Data Platform in section 2.7.2 presented here.

The CEF2 study also highlighted the bottlenecks for data applications in vehicles [9], carried out a business case analysis [10] and derived mitigation measures for cyber security risks [11].

A legal and regulatory assessment with focus on data privacy was carried out in [12] and the deployment strategy can be found in [13] and [14].

2.3.4 Open Sensor Data for Rail - OSDaR23

Publicly available data sets from the railroad sector are very rare. Therefore, the DB InfraGO AG, within the sector initiative Digitale Schiene Deutschland (DSD), and the German Centre for Rail Transport Research (DZSF) at the Federal Railway Authority (EBA) have created the first publicly available multi-sensor data set, i.e. Open Sensor Data for Rail (OSDaR23) [1][2]. The sensor dataset annotated as part of the project was recorded in several data collection runs in Hamburg by DSD. The dataset contains regular railroad environments and scenes from regular operating as well as some special situations and objects (like flames and smoke), which were posed.

OSDaR23 is a manually annotated open dataset with a multi-sensor setup for the railway environment. The multi-sensor setup includes 3 high resolution cameras, 3 medium resolution cameras, 3 infrared cameras, 3 long-range LiDARs, 1 mid-range Lidar, 2 short-range LiDARs, 1 long-range radar, 4 inertial measurement units, 4 GPS/GNSS sensors, which are usually at the front of the train. An annotation requirements specification was developed that describes the used

annotation types and geometries and the object classes to be annotated. The data is stored in the annotation format ASAM Open LABEL.

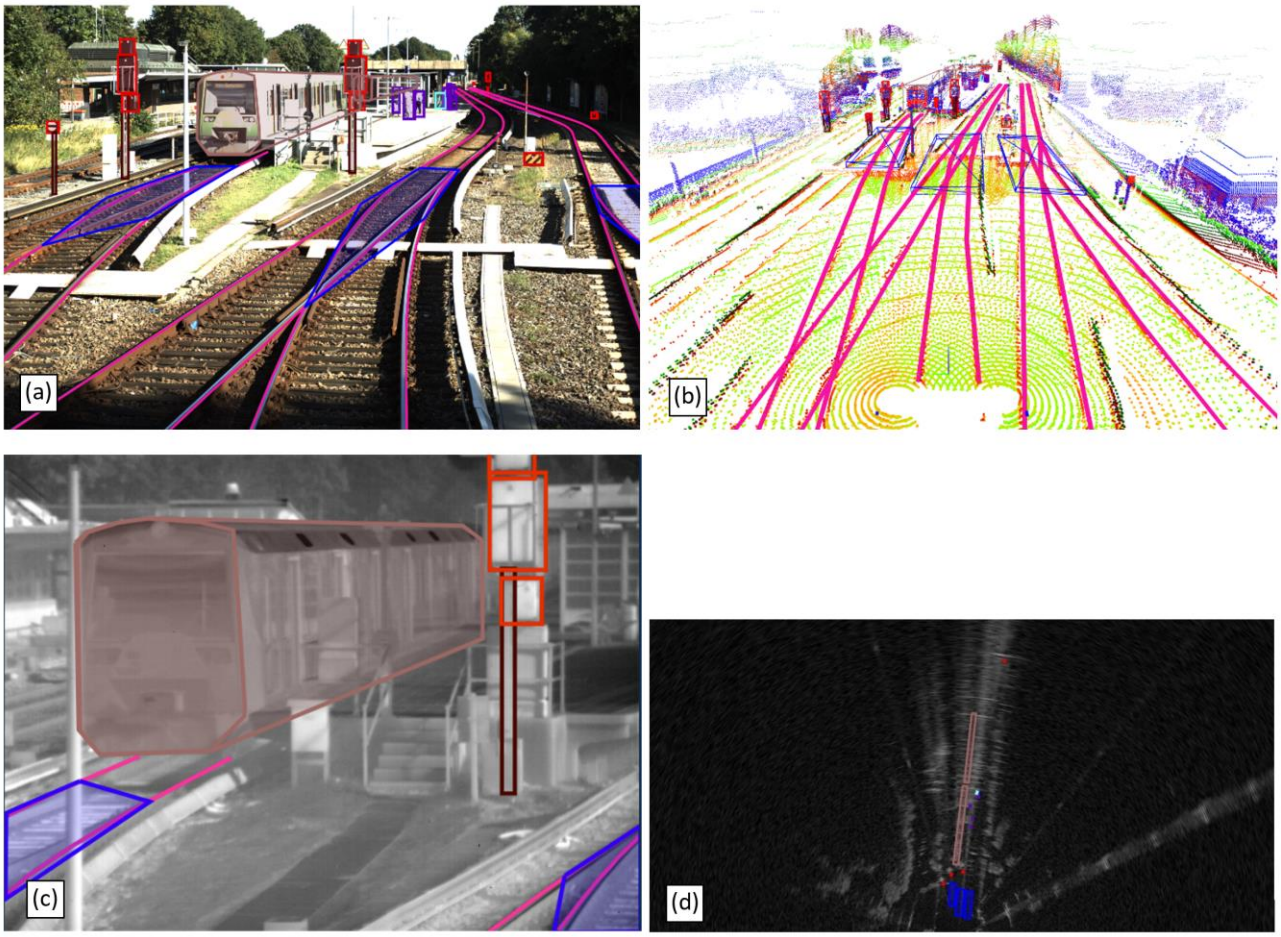


Figure 9: Annotated sensor data in OSDaR23 (a) Camera, (b) Lidar, (c) Infrared, (d) RADAR.

2.3.5 GaiaX CartenaX

The Gaia-X CartenaX project is part of the larger Gaia-X initiative, which aims to establish a secure and sovereign European data infrastructure. Specifically focusing on the automotive industry, the CartenaX project aligns with Gaia-X's mission by creating a framework for secure and collaborative data exchange within the automotive sector. Leveraging the principles of Gaia-X, CartenaX seeks to empower automotive manufacturers and suppliers to share data while ensuring sovereignty and security over their information.

CartenaX aims to develop standards, protocols, and tools tailored to the automotive industry's unique data needs. By promoting interoperability and data sharing, the project aims to facilitate innovation and digital transformation across the automotive value chain. Through collaborative data ecosystems, CartenaX endeavors to enhance decision-making processes, drive competitiveness, and accelerate the development of new products and services in the automotive sector.

In essence, the Gaia-X CartenaX project serves as a vital step toward establishing a trusted and interconnected data infrastructure for the automotive industry, contributing to Europe's broader digital sovereignty objectives [21].

2.4 DEPENDENCIES TO OTHER WPS

Work package WP7 has dependencies to WP11, WP27, WP30 and WP43.

2.4.1 WP11 - Prototype development of perception system

WP11 has as connection to WP7 since the Deliverable D11.1 (Datasets for Perception System) shall be compliant with the here proposed requirements on data quality (section 2.5.4) and annotations (section 2.5.6).

Also there may be a connection between D11.3 (PoC of Perception System) if some prototypical models from D7.5 (Perform ML/AI model training) shall be used.

2.4.2 WP27 - Digital register Specification, Development and Implementation

WP27 has a connection to WP7 since it if planned that the Deliverable D7.3 (Perform simulation with initial implementation of the Fata Factory) includes simulation data at locations that are covered by the digital register.

Also, D7.6 (Release Open-Data-Set) shall include those parts of the digital register, where the sensor data and synthetic data was acquired respectively simulated. This activity is related to D27.5 (Digital Register Object Catalogue for the Data Factory).

2.4.3 WP30 – Conceptual Data Model and semantic dictionary evolution

WP30 has a connection to WP7 as the data factory has to be designed compliant to an overall ERJU data ontology.

This includes both top-down designed data models and bottom-up driven data collections such as the sensor data ontology in section 2.5.3.

2.4.4 WP43 - Freight Demonstrator

WP43 was supposed to deliver sensor data to WP7 that should have been implemented into D7.6 (Release Open-Data-Set). The current status is that no data can be supplied from WP43, which is why mitigation measures have to be taken by DB to generate D7.6.

2.5.1 Sensor Data

This section describes the sensor data using an exemplary sensor setup with 20 sensors mounted at a train front (see Figure 10). The setup includes color and infrared cameras, lidars, a radar, gas and particle detectors and a localization unit as sensor modalities.

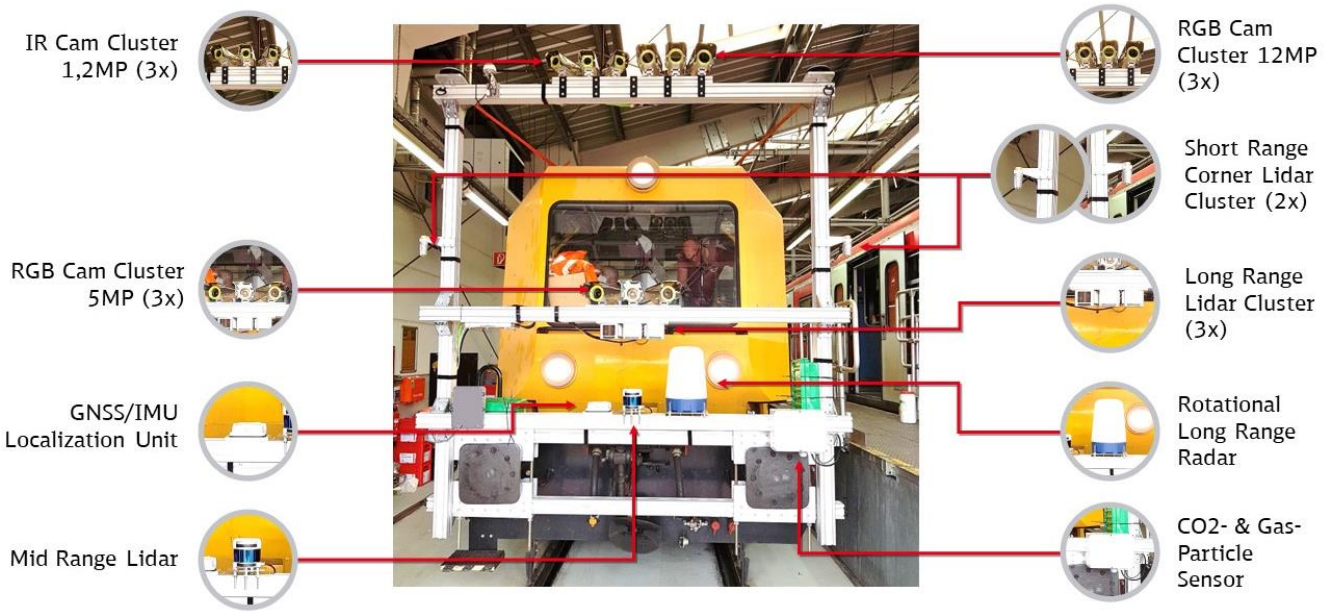


Figure 10: Sensor setup with more than 20 sensors mounted at a train front.

The camera modality comprises 3 medium-resolution, 3 high-resolution color cameras and 3 infrared cameras. The three color cameras (RGB) cover a total horizontal field-of-view (FOV) of 70°, whereby the FOVs between two color cameras overlap by 10°. The three infrared cameras each have a long focal length and cover a total horizontal FOV of 30°, with the FOVs of the second infrared camera overlapping by 1°. The medium-resolution, high-resolution and infrared camera have a respective resolution of 5, 12 and 1.2 megapixel.

The lidar modality comprises 3 long-range lidars, 1 mid-range lidar and 2 corner lidars. The three long-range lidars cover a total horizontal FOX of 16°, with the FOVs of two lidars overlapping by 2°. The mid-range lidar has a horizontal FOV of approximately 180°. The corner lidars are mounted in such a way that the FOV also extends to the rear and each corner lidar thus covers approximately 270°, whereby the entire area in front of the vehicle is covered equally by both lidars. The total resolution of all Lidars in the setup is a maximum of 240.000 points per frame.

The radar modality consists of a long-range radar, which has a horizontal FOV of 180°, a range resolution of approximately 18cm and a maximum detection range of 500 meters.

The gas sensor modality has 2 sensors, 1 gas and 1 particle sensor. The gas sensor detects the concentration of CO2 in air by absorption spectroscopy and the particle sensor detects the concentration of five different particle sizes in air between 0.5 - 10 µm in diameter.

The localization modality records the horizontal localization information or position via GNSS satellite positioning, as well as gyroscope and acceleration information via IMU.

The sensor data in this setup was mapped in a data model, see section 2.5.2. In addition, a strong focus was placed on data quality. The data quality requirements can be found in section 2.5.4 and the data recorded in the described sensor setup meets these data quality requirements.

The sensor data of this setup was then annotated, with great emphasis placed on the quality of the annotations. In-depth annotation requirements were developed, which are presented in section 2.5.6.

The annotation requirements are further developed in various projects together with the German Centre for Rail Traffic Research (DZSF) of the German Federal Railway Authority (EBA).

One of these projects also resulted in the first high-quality multimodal sensor data set from the single-railway sector OSDaR23, see section 2.3.4.

2.5.2 Data Model

2.5.2.1 Ontology

A Sensor data ontology is a formal description of sensor data to enable data exchange between multiple applications across different organizations with a common understanding of the structure and meaning of the information.

The common language provided by a sensor data ontology captures agreements on information and structure and what this information represents, enabling smart data exchange between applications and organizations.

Figure 11 shows the connection and hierarchy between the raw sensor data, capability and entity metadata and the ontology.

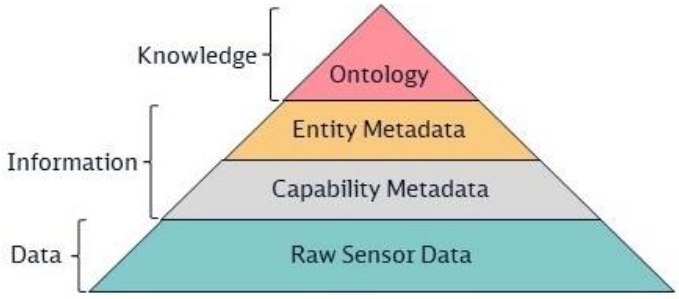


Figure 11: Overview Ontology

Raw Sensor Data - Are the measured values that are generated during the measurements. The data can be structured, semi-structured or unstructured, uncompressed or compressed.

Capability Metadata - Is the metadata that describes the properties of the raw data. On the one hand, this is the data structure of a raw data set including nested data structures and, on the other hand, the data types of the measured values.

Entity Metadata – These are specifications of the sensor and the location where the sensor was installed. The specifications of a sensor include, for example, manufacturer ID, serial number, measurement accuracies, temperature working range and, for cameras, focal length, bend of the lens, and so far. The location where the sensor was installed is, for example, top, bottom, middle, front, back, and so far.

Ontology - A sensor data ontology is a formal description of sensor data to enable data exchange between multiple applications across different organizations with a common understanding of the structure and meaning of the information. Ontology is composed of Capability Metadata and Entity Metadata.

2.5.2.2 Ontology Characteristics

A ontology is a formal representation and must have some structured form and hast to be machine readable. The ontology is an explicit description of a domain and is not like a natural language text description but is an enumeration of the entities that belong to this domain and how they relate to each other.

2.5.2.3 RDF and OQL Ontology Format

The Resource Description Framework (RDF) is a World Wide Web Consortium (W3C) standard originally designed as a data model for metadata. It has come to be used as a general method for description and exchange of graph data. RDF provides a variety of syntax notations and data serialization formats, with Turtle (Terse RDF Triple Language) currently being the most widely used notation.

The W3C Web Ontology Language (OWL) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is a computational logic-based language such that knowledge expressed in OWL can be exploited by computer programs, e.g., to verify the consistency of that knowledge or to make implicit knowledge explicit. OWL documents, known as ontologies, can be published in the World Wide Web and may refer to or be referred from other OWL ontologies.

2.5.2.4 Generation of Data Factory Sensor Ontologies

From the sensor data described in section 2.5.1 in total 37 ontologies were created for the cameras (camera, infrared camera and 5G camera), Lidars, Localization sensors (GNSS and IMU), Gas-Particle sensors and diagnosis data (Robot). Figure 12 shows the five steps how to derive a graphical representation of the Ontology RDF model from the sensor data. The graphical representations are shown in the following sections.

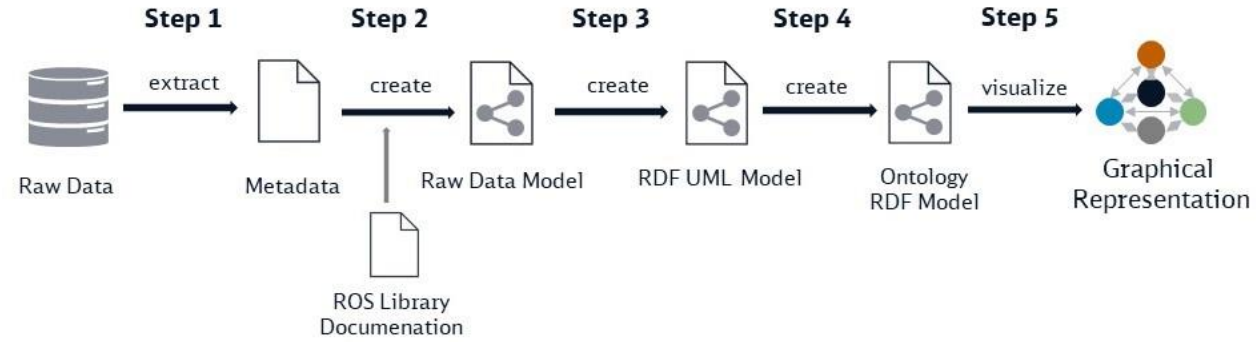


Figure 12: Generate a graphical representation of ontologies from Data Factory sensor data

2.5.2.5 UML Model

Figure 13 shows the abstract model of a sensor in UML notation, which consists of the parts Message, Specification and Attachment.

- Message is the data structure including the data types of the measured values.
- Specification is the specification of the sensor, for example Vendor ID, serial number, measurement accuracies, and so far. If extensions or additions are attached to the sensor, such as the camera lens, which has its own specification, it becomes an attachment.
- Attachment is an extension or addition to the sensor and has its own specification. For example with the camera sensor, the lens with VendorID, serial number, focal length, aperture.

A real sensor can be described exactly by instantiating the abstract model.

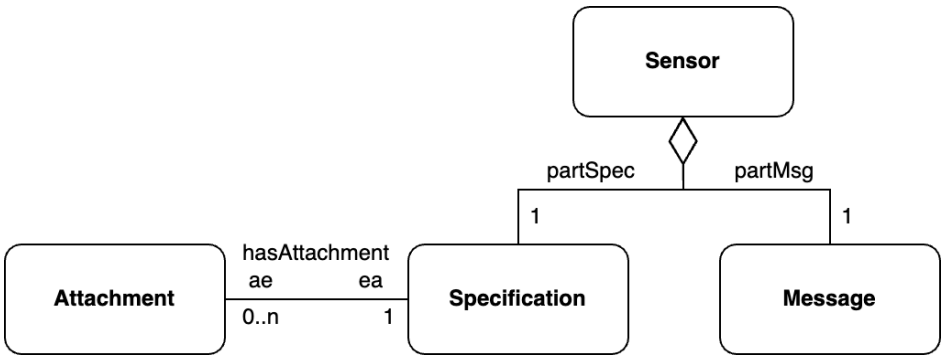


Figure 13: UML Model of sensors

2.5.3 Sensor Model

A sensor has an entity map which describes all properties of the sensor and a capability map which comprises all the values from measuring. Figure 14 exemplarily represents the data model of a camera sensor.

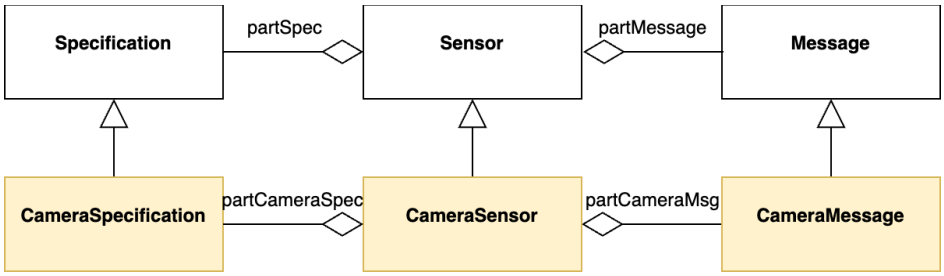


Figure 14: Data model, exemplarily for camera sensor

2.5.3.1 Message

A message of a sensor is a container that contains the relevant data and metrics of a measurement (see Figure 15).

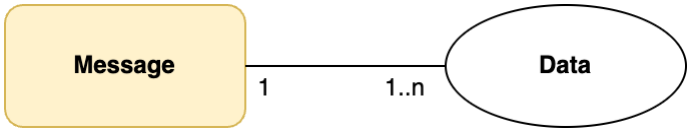


Figure 15: Data model message

2.5.3.2 Camera modality

For the 9 cameras (high-resolution, medium-resolution and infrared, mounted in front left, middle and right position), the models can be specified according the criteria in Figure 16:

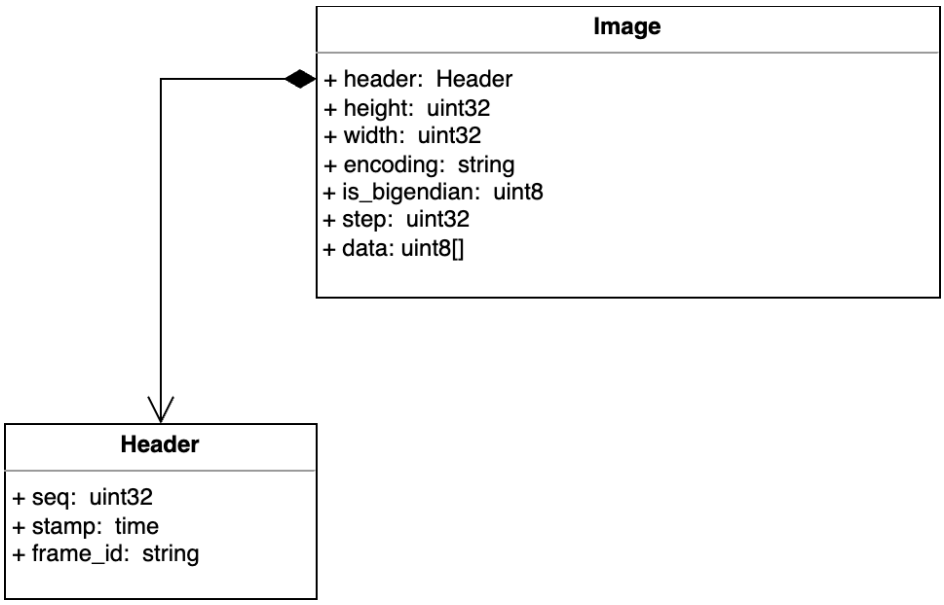


Figure 16: Data model camera sensor data

2.5.3.3 Gas sensor modality

For the 2 gas sensors (CO2 concentration and particle concentration sensors), the models can be specified according the criteria in Figure 17 and Figure 18.

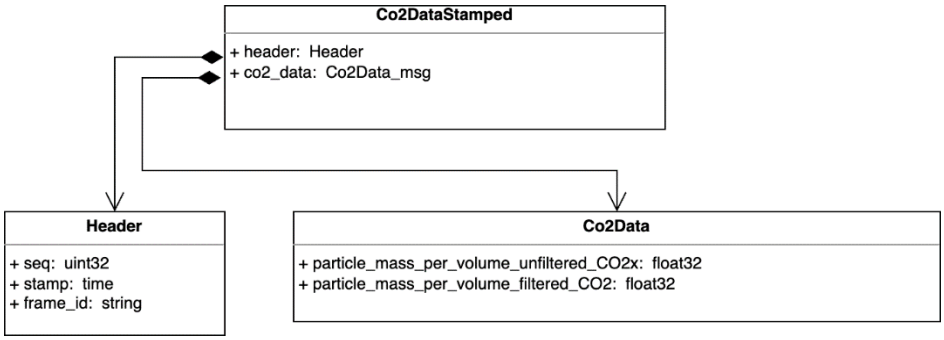


Figure 17: Data model for CO2 sensor data

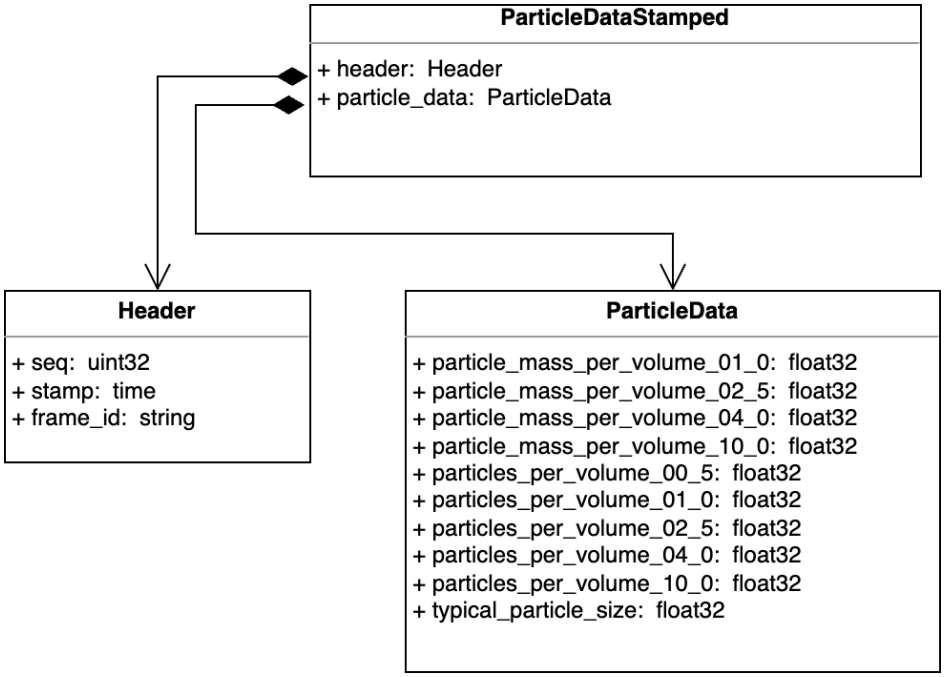


Figure 18: Data model for particle density sensor data

2.5.3.4 Localization sensor modality (GNSS/IMU)

The localization sensor modality delivers the horizontal localization information or position via GNSS satellite positioning, as well as gyroscope and acceleration information via IMU. Strictly speaking, this modality consists of a number of sensors which are not explained in detail here, namely BESTPOS, BESTUTM, BESTVEL, CORRIMU, GPSFix, HEADING2, IMU, INSPVA, INSPVAX, INSSTDEV, NavSatFix.

Exemplarily we only consider BESTPOS and IMU sensor data here, which represents the horizontal localization information and the gyroscope and acceleration information, the models can be specified according the criteria in Figure 19 and Figure 20.

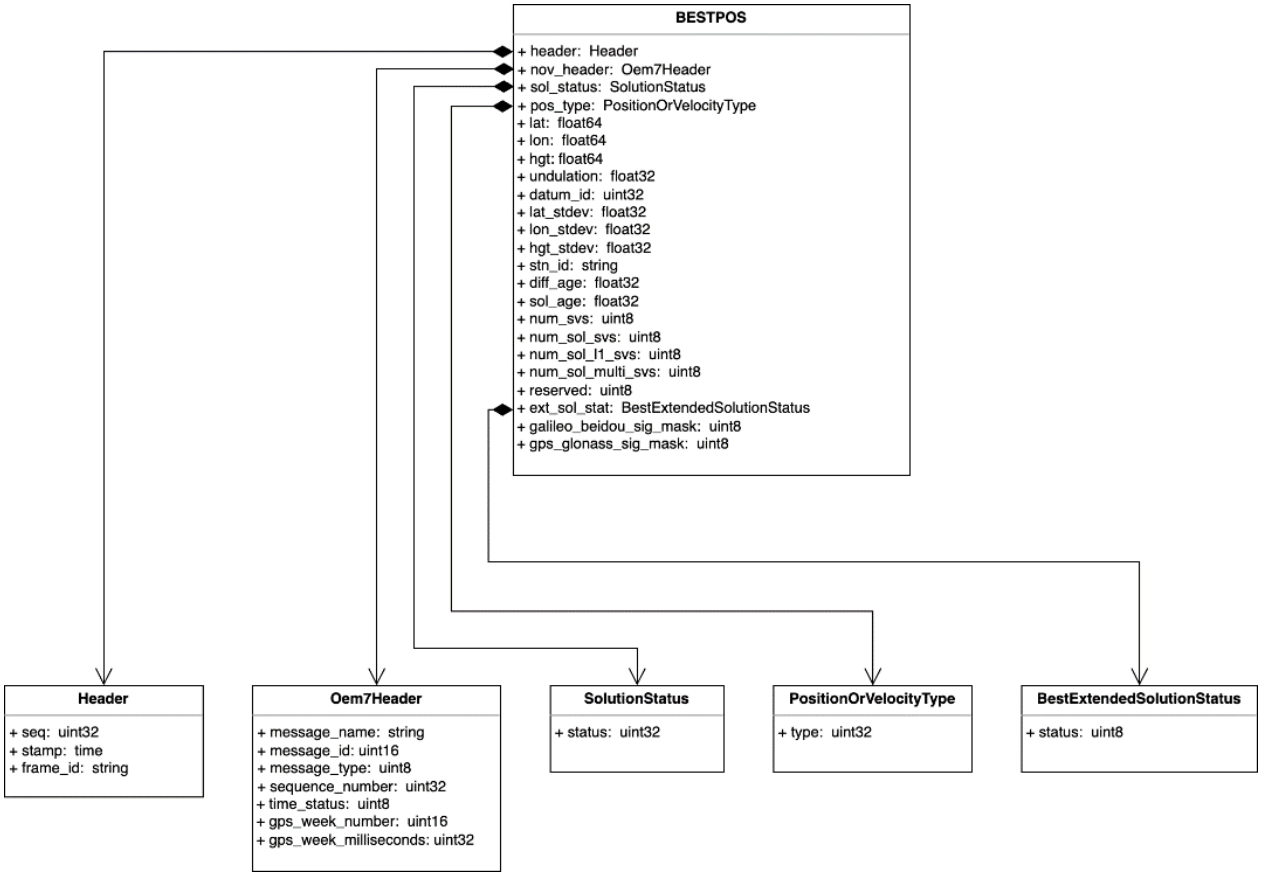


Figure 19: Data model for BESTPOS sensor data

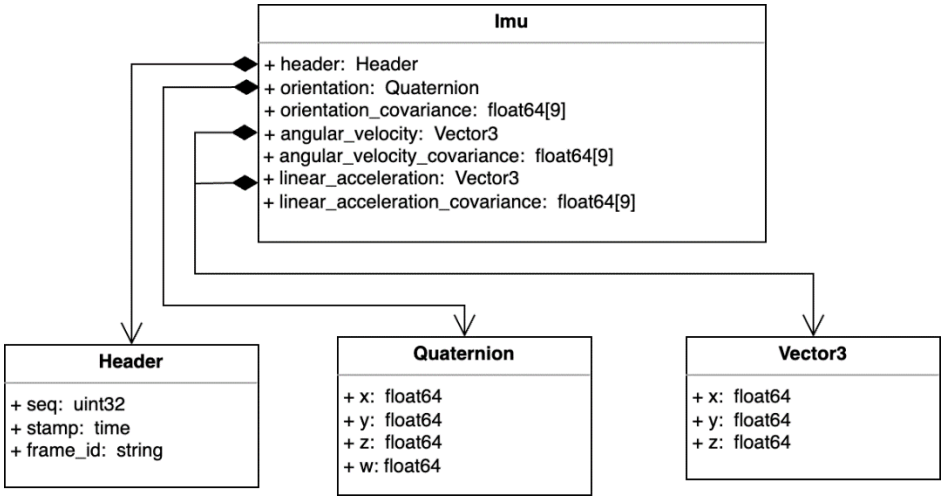


Figure 20: Data model for IMU sensor data

2.5.3.5 Lidar modality

The Lidar modality consists of three long-range, one mid-range and two corner Lidars, and the long-range lidar additionally has a built-in IMU. The model for the IMU can be found in Figure 20 und the model for the Lidars are depicted in Figure 21. The message holds a collection of N-dimensional points, which may contain additional information such as intensities, etc. and the point data is stored as a binary blob and is described by an array.

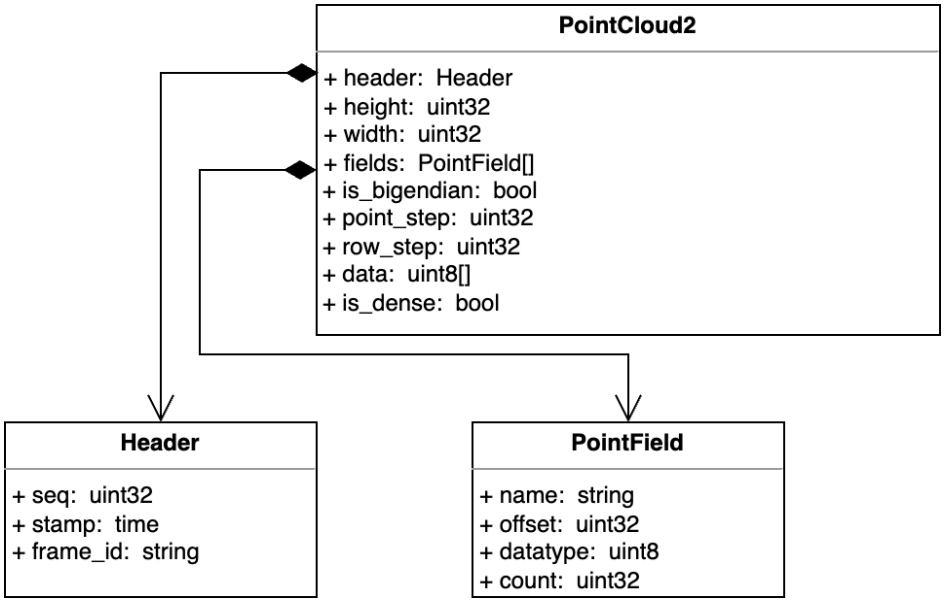


Figure 21: Data model for Lidar sensor data

2.5.3.6 Radar modality

This modality consists of one long-range radar and the message contains an uncompressed image, while (0, 0) is the top-left corner of the image. The model is depicted in Figure 22.

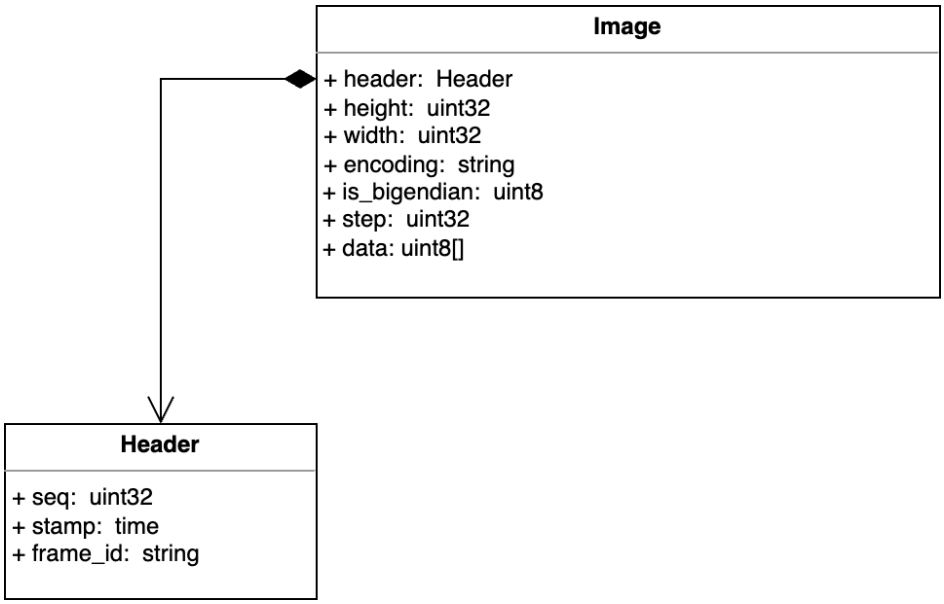


Figure 22: Data model radar sensor data

2.5.4 Data Hierarchy

Figure 23 shows the hierarchy of various categories and subcategories of data and models, related to a technological or IT framework, in the context of machine learning, data management, or autonomous systems.

2.5.4.1 Data Categories:

A dataset is the compilation of data, which can be signed, processed, selected and curated.

- Data: It includes both synthetic and sensor data
    - Synthetic Data: This could be artificially generated data such as camera images, LiDAR point clouds, and radar images.
    - Sensor Data: Actual data captured by sensors, which might include camera images, LiDAR and radar data, localization and temperature sensor information.
    - Metadata: Contains simulation metadata and file metadata.
        - Simulation Metadata: Details that describe the context or environment of a simulation, like 3D assets, digital twins, and scenarios.
        - File Metadata: Could include signatures, data sanity, integrity information, checksums, and version numbers.
        - Functional Data: Outputs from certain functions or processes like perception detectors, localization functions, and triggers.
        - Annotations: Data that has been labelled or annotated, such as with bounding boxes (2D BB, 3D BB), splines, or polygons.
        - Map Data: Geographic or spatial data such as topographic maps and object maps.
        - Vehicle Data: Information related to vehicles, including speed, mode, and identifiers.
        - IT System Data: Data about IT systems, including resource info, user activities, notifications, system info, monitoring data, and alerts.

2.5.4.2 Model Categories:

- Models:
    - DL Models: Deep Learning models that might include Generative Adversarial Networks (GANs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer models.
    - ML Models: Machine Learning models, which can be supervised, unsupervised, or reinforcement models.

The data is to be structured to feed into the model categories, suggesting a workflow from data collection and processing to the application of machine learning and deep learning models.

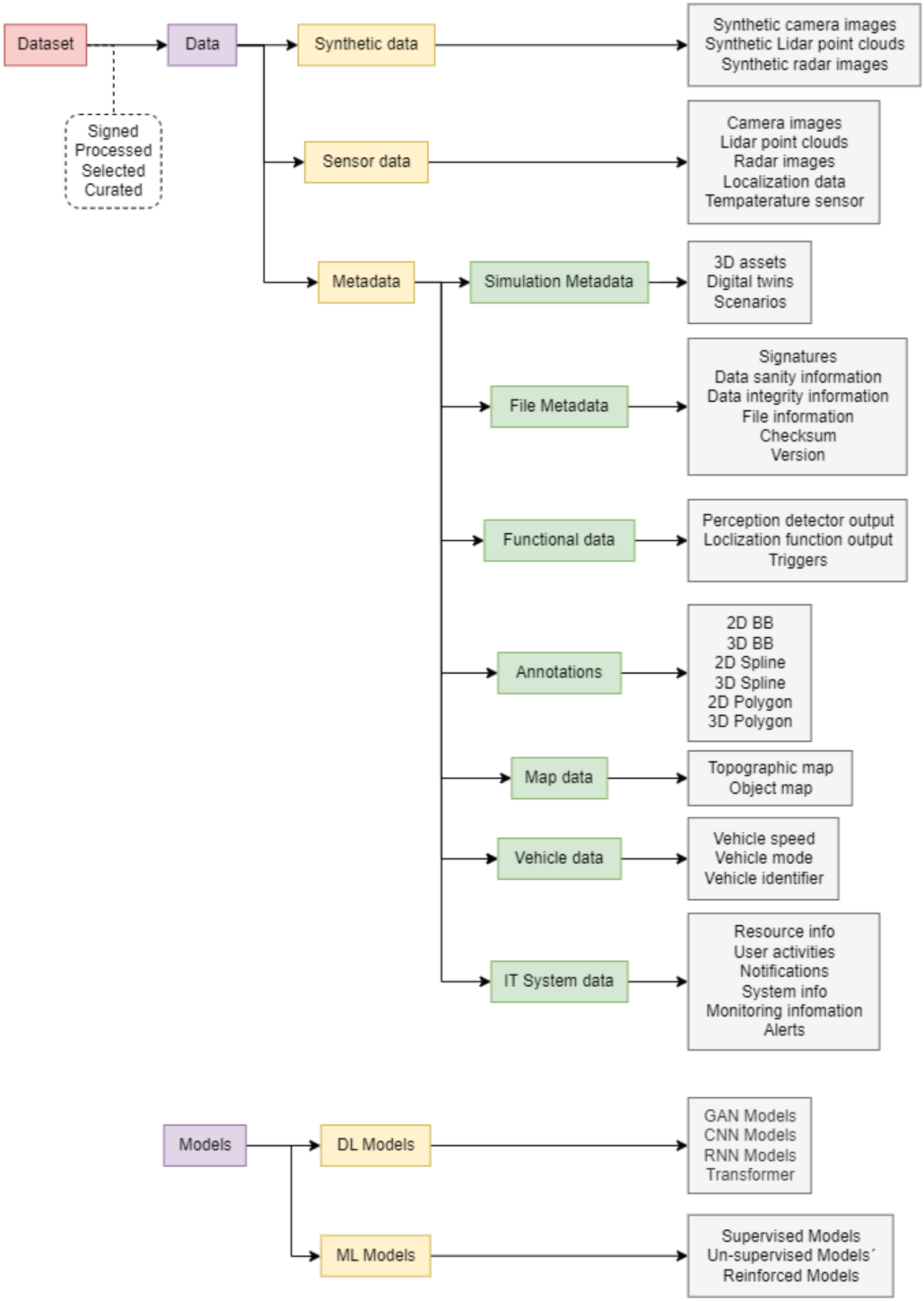


Figure 23: Data hierarchy defintion

2.5.5 Data Quality Requirements

The Data Factory provides sensor data for the development and validation of machine learning applications. In order to provide sensor data in a sufficient quality, the incoming sensor data already need to fulfil a set of quality criteria. These criteria can only be satisfied by the onboard system that produces the sensor data. The following Data Quality Requirements are therefore targeting the onboard system that generates the sensor data and refer to it as the system under consideration (SuC). The requirements are structured thematically, starting with the requirement targeting the sensor configuration. These requirements ensure that the configuration of the onboard system with its sensors is known for each point in time. The second section about data integrity ensures the integrity of the sensor data in the whole chain from the sensors to the Data Factory. That means that the sensor data is not being modified or corrupted while being stored or transferred. The data content section assures that all available sensor information is attached to the sensor data. The next three sections target the multi-modal use of the sensor data and allows the synchronisation of multiple modalities. This includes requirements regarding time stamping, frequency and calibration of the sensor data. The last two section target the ego-motion of the train. The GNSS and IMU information section defines the required information to determine the position and motion of the train. The ego motion compensation section defines requirements that allow to correct or minimize artefacts in the sensor data that are introduced by the motion of the train during the acquisition of sensor data.

2.5.5.1 Sensor Configuration

- The SuC shall provide the sensor data in the Highest_Quality that is technically possible without violating the system's performance requirements.
    - Rationale: The data shall be used for the development of GoA4 train operation. One of the main use cases is the environmental perception with its multiple tasks. In order to fulfil the requirements of these tasks, an overall high data quality is required. A goal is the creation of multi-modal sensor datasets. This requires highly synchronised sensor data with a precise calibration and sensor data that allows to detect and recognise objects as good as possible.
- The SuC shall provide the sensor configuration and parameters for each Sensor_Frame.
    - Rationale: The sensor configuration and parameters shall be provided so that each Sensor_Frame can be associated with the respective sensor's configuration and parameters. This is required in order to be able to evaluate sensor configurations and compare different configuration. In addition, it helps to evaluate artefacts in the sensor data based on the configuration. For example, the rolling shutter effects depends on the exposure time. In order to track down errors, the serial numbers and software versions need to be known.
- The SuC shall provide the Hardware_Configuration for each Sensor_Frame.
    - Rationale: Knowing the hardware configuration of the system is essential to detect misconfigured or faulty hardware components. Especially for the hardware configuration it is not necessary to physically attach it to each Sensor_Framebut instead make transparent which Sensor_Frame are recorded with which hardware.
- The SuC shall provide the Software_Versions for each Sensor_Frame.
    - Rationale: Knowing the software version of each software component in the system is necessary to track down issues in specific versions of the software that are detected

when evaluating the quality of the sensor data. Similar to the hardware version it is not necessary to physically attach the software versions to each Sensor_Frame but instead make transparent which Sensor_Frame are recorded with which software versions.

- The SuC shall provide the Sensor_Parameters for each Sensor_Frame.
    - Rationale: Knowing the sensor parameters for each Sensor_Frameis necessary to compare the sensor parameters at different environment conditions and evaluate the effects of different sensor parameters on the data quality. For example, the exposure time can influence the rolling shutter artefacts of a camera.

2.5.5.2 Data Integrity

- The SuC shall provide the unmodified Sensor_Data_Streams to the Data_Logger for recording.
    - Rationale: The Sensor_Frame shall not be modified by any subsystem in order to avoid that other systems only having data in reduced quality. For certification of the system it is necessary to ensure that the data is not being modified. This includes that all Sensor_Frame are forwarded to the Data_Logger and no information is removed from the data.
- The SuC shall publish the Acquisition_Timestamp that is provided by the sensor.
    - Rationale: The goal is to know the acquisition timestamp when the Sensor_Frame is physically being captured by the sensor. This requires to have a precisely synchronised sensor and that the timestamp provided by the sensor is not overridden with the timestamp when the sensor arrives at the driver or any other subsystem.
- The SuC shall allow to verify the authenticity and integrity of each Sensor_Frame.
    - Rationale: It shall be possible to verify that the message is not being modified. This verification shall be possible at each stage in the system and in the Data Factory. Therefore, the SuC needs to add information to each message (or provide another method) that allows to verify the integrity and authentication of each Sensor_Frame. There might be sensors that do not support adding these information within the sensor. In these cases the information should be added at the earliest possible stage and resource demands and computation time may need to be considered and weighted with the benefits of support an authenticity and integrity verification at that stage.

2.5.5.3 Data Content

- The SuC shall attach all information to each Sensor_Frame that are provided by the manufacturer's sensor driver.
    - Rationale: The Data_Logger shall be able to record all sensor information that is available in the system unless technically infeasible. This means, that each sensor information that is available in the system shall arrive at the Data_Logger.
- The SuC shall make processed Sensor_Data_Stream available to the Data_Logger , if the processing adds information to the Sensor_Data_Stream.

- Rationale: In case sensor data is processed to improve its quality or to add additional information, the system shall be designed to enable the Data_Logger to record the processed as well as the unprocessed data streams, if technically feasible.

2.5.5.4 Time Stamping

- The SuC shall stamp all Sensor_Frames with the Acquisition_Timestamp.
  - Rationale: In order to precisely fuse frames that are recorded at the same / a similar point in time, the physical acquisition time is required. Sensor fusion will likely be a key concept for the environmental perception in GoA4 train operation and is required for multi-modal data annotation.
- The SuC shall provide a definition of what part of the sensor's actual physical acquisition process the Acquisition_Timestamp is referring too.
  - Rationale: Different sensors have different means of acquisition. In practice, acquisition can have multiple possible reference or "anchor" points in time, each of which might be equally appropriate, as long as it is clearly defined, which one is used. Examples: For global shutter cameras, possible references are start time of exposure or end time of exposure. For rolling shutter cameras, this shall additionally include the line of reference, as different lines of a rolling shutter sensor are exposed at different times. It shall also include the line time, i.e., the time delta between the exposure start of successive lines.
- The SuC shall provide the Reference_Time for the system.
  - Rationale: It is important to have the same understanding of time in the system in order to fuse Sensor_Frames from different sensors to a Sensor_MFrame using the acquisition time.
- The SuC shall provide the Reference_Time in UTC.
  - Rationale: Avoid leap seconds errors in the data by configuring all systems to use UTC instead of the atomic time.
- The SuC shall make the Reference_Time available within 30s after system boot is completed.
  - Rationale: To allow all systems to synchronise their time within 60s after system boot completed, the reference time has be available in advance.
- The SuC shall provide timing signals for clock synchronisation in the system.
  - Rationale: The reference clocks has to provide a time synchronisation service with a protocol. The most common one is PTP.
- The SuC shall ensure clocks deviate less than 1 ms from the Reference_Time.
  - Rationale: This is necessary to allow precise stamping of acquisition times and an accurate fusion of sensors. Common protocols such as PTP allow an accuracy up to less than a microsecond.
- The SuC shall ensure clocks are synchronised to the Reference_Time within 60s after system boot is completed.
  - Rationale: As one of the tasks in Data Factory is to ensure that the driving path is free before the train starts moving, the data while the train is still on the parking position

is of interest. In order to capture and use this data, the reference time in the system shall be available within 60s after the system boot is completed.

- The Acquisition_Timestamp shall deviate less than 1 ms from the physical acquisition time of the sensor.
    - Rationale: In order to allow a precise fusion of sensor data, the acquisition timestamp shall be as close as possible to the physical acquisition timestamp. The physical acquisition time is the reference time in the the system at the point in time when the hardware in the sensor registers the object. Common protocols such as PTP allow an accuracy up to less than a microsecond. With a speed of 100km/h, the train moves around 3cm per millisecond.

2.5.5.5 Frequency & Synchronisation

- The SuC shall ensure a stable and constant Acquisition_Frequency for each sensor.
    - Rationale: A stable and constant acquisition frequency is required to reliably produce synchronised sensor data that can be fused.
- The SuC shall ensure that the Acquisition_Timestamps between two consecutive Sensor_Frames have a maximum deviation of 5% or 2.5ms, whatever is lower, to the expected delta which is calculated based on the target frequency (delta = 1/f).
    - Rationale: This requirement defines the upper bounds for the requirement above by providing concrete values for what is the lower bound for a stable and constant acquisition frequency. In the extreme (on sensor is -2.5 ms and the other one is +2.5 ms) this leads to an offset of 5ms within a Sensor_MFrame. With a speed of 100km/h this will lead to an offset of 15cm in the worst case.
- The SuC shall ensure that each Perception_Sensor uses an Acquisition_Frequency that is an integer multiple of the Perception_Sensor with the lowest Acquisition_Frequency.
    - Rationale: This in mandatory in order to have multiple synchronised sensors with a stable and constant acquisition frequency.
- The SuC shall ensure that the maximum percentage of Frame_Drops per Sensor_Data_Stream is lower than one per thousand (1/1000).
    - Rationale: This requirement tolerates one frame drop at a maximum occurrence of once every 1000 frames.
- The SuC shall ensure an Acquisition_Frequency of the Perception_Sensors of at least 10 Hz.
    - Rationale: A frequency of 10 Hz allows to reevaluate the situation every 100ms and therefore 10 times a second. It is necessary to increase the accuracy of tracking algorithms by limiting the distance the train and objects move between two frames.
- The SuC shall ensure an Acquisition_Frequency for the Coupled_Localisation of at least 50 Hz.
    - Rationale: A high update frequency of the coupled localisation is important to get the most accurate egomotion information for each sensor frame. As the coupled localisation consists of multiple modalities, such as GNSS, IMU, map matched localisation, not all modalities need to have an update frequency of 50 Hz. It is

possible to update the position at 50 Hz using the IMU measurements and correct the drift using GNSS position updates at a lower frequency.

- The SuC shall trigger the acquisition of the Perception_Sensors synchronously.
    - Rationale: In order to fuse sensor modalities and create multi-modal frames (Sensor_MFrames), all Perception_Sensors have to capture their data at the same point in time. For technical reasons, there might be sensor modalities that benefit from a slightly shift the acquisition time of these sensors to avoid artefacts (distortions), such as, e.g. multiple RADARs. When having a setup with multiple RADAR sensors it is common to trigger the central RADAR together with the other perception sensors and shift the left and right RADARs slightly in terms of acquisition time to reduce distortion by the FMCW ramp.
- The SuC shall allow a maximum difference between the Acquisition_Timestamps of the same Sensor_MFrame of less than 5 ms for each Perception_Sensor.
    - Rationale: When the train or objects are in motion, a large time offset of the acquisition time of different sensor modalities of the same Sensor_MFrame will introduce errors in the sensor fusion. With a speed of 100km/h this will lead to an offset of 15cm in the worst case. This requirement implies that all synchronously triggered sensors have to trigger within a time period of 5 ms.

2.5.5.6 Sensor Calibration

- The SuC shall provide precise calibration information for each sensor.
    - Rationale: In order to fuse the output of different sensors, a precise calibration is necessary. In the context of the Data Factory this is for example necessary to create a multi-modal sensor dataset.
- The SuC shall provide the extrinsic calibration for each sensor.
    - Rationale: The extrinsic calibration parameters are required to get a precise overall calibration.
- The SuC shall provide the extrinsic calibration with an accuracy of at least 1 cm for each translational axis.
    - Rationale: From the experience of different projects this is a good lower bound. This measurement accuracy is easily achievable and limits the effort during annotation.
- The SuC shall provide the extrinsic calibration with an accuracy of at least 0.2° for each rotational axis.
    - Rationale: From the experience of different projects this is a good lower bound. Rotational inaccuracy leads to an increasing positional error with increasing distance from the train. This measurement accuracy is easily achievable and limits the effort during annotation. Calculation for error with 0.2°: 100m * tan(0.2°) ≈ 0,35m which is already a lot of offset for multi-modal annotation.
- The SuC shall provide the intrinsic calibration for each sensor that uses intrinsic calibration parameters.
    - Rationale: The intrinsic calibration parameters are required to get a precise overall calibration.

2.5.5.7 GNSS and IMU Information

- The SuC shall provide precise GNSS and IMU information for a specified reference point.
    - Rationale: The position of the train is required for various use cases. In terms of data annotation it is required to match the position of the train on the digital map. Precise IMU information is required to perform egomotion compensation. GNSS and IMU information are used to generate metadata to search for specific scenarios such as the train being at a specific area (geofence), the train drives below/above/within a specific speed or accelerates below/above/with a specific threshold.
- The SuC shall provide the GNSS position in WGS84 with the following values and units.
        - latitude [deg]
        - longitude [deg]
        - height/altitude [m]
        - In case the values and units differ from the required, the steps and equations on how to convert to the required values and units shall be provided.
    - Rationale: The GNSS position is required to match the sensor data with the digital map. Providing localisation information in standardised format simplifies the processing of the data and reduces the risk of misinterpretation. The position of the train is used to generate metadata that allow to search for data at a specific area (geofence).
- The SuC shall provide the GNSS velocity in the ENU reference system with the following values and units.
        - east velocity [m/s]
        - north velocity [m/s]
        - and up velocity [m/s]
        - In case the values and units differ from the required, the steps and equations on how to convert to the required values and units shall be provided.
    - Rationale: The velocity of the train is required for egomotion compensation. Providing localisation information in standardised format simplifies the processing of the data and reduces the risk of misinterpretation. The velocity is used to generate metadata that allow the user to search for data at which the train is driving above/below/within a specific velocity threshold.
- The SuC shall provide the IMU sensor orientation according to the right-hand-rule with the following values and units. The orientation angles shall follow the roll-pitch-yaw Tait-Bryan convention: z-y'-x'' (intrinsic) or x-y-z (extrinsic) according to DIN 70000.
        - roll [rad] in body frame according to DIN 70000 (x forward, y to the left, z up)
        - pitch [rad] in body frame according to DIN 70000 (x forward, y to the left, z up)
        - yaw [rad] in ENU coordinate system (with 0 = east)
        - In case the values and units differ from the required, the steps and equations on how to convert to the required values and units shall be provided.

- Rationale: The sensor orientation is required for egomotion compensation. Providing localisation information in standardised format simplifies the processing of the data and reduces the risk of misinterpretation. The orientation is used to generate metadata that allow the user to search for data at which the train is driving in a specific direction (e.g. to evaluate the effects of specific sun angles in relation to the driving direction of the train).

- The SuC shall provide the IMU sensor orientation rate with the following values and units.
  - roll rate [rad/sample]
  - pitch rate [rad/sample]
  - yaw rate [rad/sample]
  - In case the values and units differ from the required, the steps and equations on how to convert to the required values and units shall be provided.
  - Rationale: The sensor orientation rate is required for precise egomotion compensation. Providing localisation information in standardised format simplifies the processing of the data and reduces the risk of misinterpretation.
- The SuC shall provide the INS acceleration with the following values and units
  - lateral acceleration [m/s/sample] (along x-axis)
  - longitudinal acceleration [m/s/sample] (along y-axis)
  - vertical acceleration [m/s/sample] (along z-axis)
  - In case the values and units differ from the required, the steps and equations on how to convert to the required values and units shall be provided.
  - Rationale: The INS acceleration is required for precise egomotion compensation. Providing localisation information in standardised format simplifies the processing of the data and reduces the risk of misinterpretation. The acceleration is used to generate metadata that allow the user to search for data at which the train is accelerating/braking above/below/within a specific threshold (e.g. to identify situation where the train is braking).
- The SuC shall ensure a precision of the train localisation of at least 30cm.
  - Rationale: A precise train localisation is required in order to be able match the sensor data with the digital map. GNSS precision will be limited within tunnels and it shall be determined if map matched localisation can compensate in those situations or if the precision can be worse in areas with limited GNSS connection.

2.5.5.8 Ego-Motion Compensation

- The SuC shall provide ego-motion compensated or ego-motion compensable sensor data using the provided Egomotion information.
  - Rationale: In order to use the sensor data for annotation and object fusion tasks, the LiDAR point clouds likely need to be ego-motion compensated. Therefore, the information to perform this compensation must be available.
- The SuC shall provide the ego-motion compensated sensor data for recording to the Data_Logger, if available.

- Rationale: If there is a real-time ego-motion compensation performed, the results shall be available for the Data Logger

- The SuC shall provide the Acquisition_Timestamp for each point within the point cloud.
  - Rationale: This is often implemented by providing the offset of each point to the acquisition timestamp of the point cloud. It is necessary to apply ego-motion compensation to point clouds and avoid objects that are stretched in driving direction of the train.
- The SuC shall provide the Egomotion information at the mounting position of each LiDAR Perception_Sensor.
  - Rationale: In order to have an accurate egomotion estimation for egomotion compensation of the LiDAR point clouds. The egomotion shall either be measured at the positions of the LiDARs or measured at a reference position that allows to calculate the egomotion at the positions of the LiDARs with a comparable precision by providing the necessary transformation information.
- The SuC shall reduce the rolling shutter effect for cameras with a rolling shutter to a minimal level.
  - Rationale: The rolling shutter effect introduces distortion (curvature) of objects. This reduces the precision of sensor fusion and increases the difficulty and effort for data annotation. It also introduces risks for matching objects to the digital map and detecting changes in the environment compared to the digital map.
- The SuC shall ensure that the curvature of objects is less than 2° for speeds up to 100 km/h compared to the object recorded with a steady train.
  - Rationale: A big distortion of objects makes it difficult to fuse objects between different sensors as the appearance and geometries of this objects differ. The example image (Figure 1 - in the document below) shows the effects of an angle of 4.5°. The distortion of the rolling shutter effect is always a trade-off with the exposure time. It might be necessary to exceed the distortion limit in situations of low light.

2.5.6 Data Annotation Requirements

The ensuing section delineates the format for annotating the dataset. Detailed information is systematically organized in Appendix 4.1, which readers are encouraged to consult for comprehensive understanding.

Context of Machine Learning Annotations: In machine learning contexts, annotations are indispensable for preparing datasets that machine learning algorithms can efficiently interpret. These annotations encapsulate critical information regarding elements within data, such as the identification and classification of objects in imagery. Typically, this intricate process is carried out by trained annotators whose task is to ensure that the scenario depicted in the data is rendered comprehensible for the algorithm. This document sets forth the requisite guidelines for annotating a variety of sensor modalities, including but not limited to IR/RGB cameras, RADAR, and LiDAR. The specification herein describes distinct annotation types applicable to each sensor modality.

Sensor Modality	**Annotation Types**
IR/RGB Camera	2D Bounding Box, 2D Polygon, 2D Polyline

Lidar	3D Bounding Box; 3D Semantic Segmentation
RADAR	2D Bounding Box, 2D Polygon

Table 1: Sensor Modality and Annotation Types

Annotation Diversity: Objects subject to annotation fall into two primary categories: dynamic and static. Additionally, a subset of objects pertinent to railway operations has been identified, with a selection tailored to reflect real-world interactions and their potential implications for railway safety and operation.

Annotatable Entities and Artifacts: The dataset encompasses a diversified array of entities and artifacts necessitating precise annotations. These annotations are pivotal for machine learning algorithms to accurately interpret and learn from visual data. Enumerated below are the categories and their specific inclusions for which annotations are mandated:

- Human Elements:
  - Person: Every individual human figure captured in the dataset.
  - Personal Item: Objects associated with individuals, such as bags, umbrellas, and portable devices.
- Assemblages:
  - Crowd: Congregations of individuals exhibiting collective behavior or movement.
- Mobility Apparatus:
  - Bicycle: Single rider bicycles, including variations such as tandem and electric bicycles.
  - Train: All forms of trains, including locomotives, passenger trains, and freights.
  - Wagon: Freight wagons, tankers, and other rail-bound cargo carriers.
- Automotive and Powered Units:
  - Motorcycle: Two-wheeled motor vehicles, including scooters and mopeds.
  - Road Vehicle: Four-wheeled vehicles such as cars, trucks, and buses, excluding motorcycles.
- Fauna:
  - Animal: All animals larger than a domestic cat in size.
  - Group of Animals: Clusters of animals observed in a collective space or behavior pattern.
- Assistive Technologies:
  - Wheelchair: Manual and electric wheelchairs, including mobility scooters.
- Rail Infrastructure Components:
  - Drag Shoe: Devices used to secure stationary trains or as emergency brakes.
  - Track: Rails including the ties and ballast.
  - Transition: Connective sections enabling the shift of rolling stock from one track to another.

- Switch: Railway switches, enabling trains to be guided from one track to another.
- Catenary Pole: Structures supporting overhead power lines for electric trains.

- Railway Signaling and Safety Devices:
  - Signal Pole: Posts holding signal lights and signs.
  - Signal: Traffic signs and signal devices dictating train movements.
  - Signal Bridge: Overhead structures from which signals are suspended.
  - Buffer Stop: Devices at the end of tracks to prevent railway vehicles from going beyond the end of the track.
- Miscellaneous Objects:
  - Flame: Manifestations of fire relevant to safety scenarios.
  - Smoke: Visual evidence of smoke, indicative of fire or combustion.

Attribute Annotations: Each object and its respective annotation is complemented by a set of attributes that furnish additional details, enhancing the dataset's richness and utility for algorithm training. These attributes elucidate properties such as object dimensions, orientation, and context-specific characteristics. For illustrative purposes, Figure **25** and Figure **25** provide visual representations of annotated objects with their associated attributes.

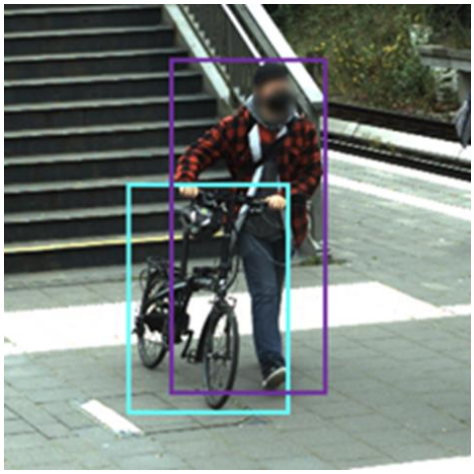


Figure 24: Annotation example for the object classes “bicycle” and “person” in an RGB image

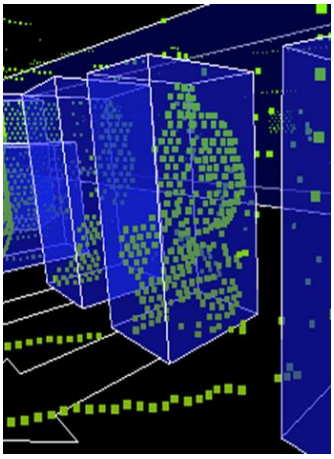


Figure 25: Annotation example for a person sitting in lidar data

2.5.7 Annotation Format Requirements

The annotations within this project adhere to a standardized format, ensuring consistency and compatibility across the entire dataset. The designated format leverages the JSON file structure, conforming to the "RailABEL" JSON schema definition, which aligns with the broader "OpenLABEL" standard. Details on the schema, including its validation and structure, are accessible through the following resources:

RailABEL Schema: An extension of the OpenLABEL standard, the RailABEL schema integrates domain-specific requirements for railway data annotation. The schema and its documentation are maintained on GitHub:

- RailABEL JSON Schema on GitHub.

OpenLABEL Standard: Established by the Association for Standardization of Automation and Measuring Systems (ASAM), OpenLABEL provides a comprehensive framework for data annotation in the realm of automation:

- Official ASAM OpenLABEL Standard.

To facilitate understanding and implementation, an example JSON file conforming to the RailABEL schema is provided:

- Example RailABEL Format JSON.

Users are encouraged to review these resources to ensure adherence to the annotation standards and to foster seamless data integration within the project framework.

2.5.8 Data Governance

Data Governance is the collective term for the practices, processes, and frameworks that ensure the effective and responsible management of an organization's data assets. It involves the clarity of roles, decision rights, and accountability, ensuring that data is used both efficiently and securely. The primary aim is to create a well-structured environment where data quality, accessibility, integrity, and security are always maintained. See the related section of the CEF2 project in [6] and [7].

2.5.8.1 Governance Policies and Automation

Governance within the context of a Data Factory involves establishing clear policies and procedures to guide all data-related activities. It entails setting up roles like Data Owners, Data Stewards, and Data Consumers, each with specific responsibilities to ensure that data is handled correctly throughout its lifecycle. Governance covers everything from how data is collected and stored, to how it is archived or purged. Effective governance means that all data handling is transparent, predictable, and in line with established rules.

- Define Purge Policy
    - Define Purge Policy
        - Automated processes for archiving and purging data shall be in place.

          Rationale: In order to maintain high data quality and ensure the principles of privacy.
- Audit Logging
    - Log Data
        - The system shall maintain comprehensive audit logs for all data-related activities, including access, modification, and deletion. This includes system data, error logging and access logging.

Rationale: The system shall maintain comprehensive audit logs for all data-related activities, including access, modification, and deletion.

Rationale: This includes system data, error logging and access logging.

Rationale: This is done in order to ensure all data-related activities on the system follow the requirements of protection and data governance.

- Log Analysis
  - Logs shall capture user IDs, timestamps, and actions performed to facilitate forensic analysis.

    Rationale: In order to facilitate forensic analysis.

- Manage Data Governance Policies
  - Enforce Policies
    - The system shall define and enforce data governance policies, including adherence to data classification, access controls, and quality standards.

      Rationale: In order to make sure the principles of data governance are being fulfilled.
  - Define Policy Alerts
    - Policy violations should trigger alerts and notifications to appropriate stakeholders.

      Rationale: In order to ensure the right measures are taken to avoid consequences and restore policies.

- Governance Roles
  - Roles for every object
    - Every object and every subsystem of DAFA shall have a Data Owner, Data Steward and Data Consumer.

      Rationale: In order to ensure the modification of data follows the principles of data governance.

2.5.8.2 Compliance

Compliance, refers to adhering to laws, regulations, and policies that apply to data. For Data Factory, this means meeting the requirements of GDPRand other data protection regulations. Compliance ensures that the Data Factory operates legally and ethically, respecting the privacy and rights of individuals. Regular auditing and reporting are part of this, ensuring that the system continually meets these external standards

In order to maintain a suitable balance of interest between data processing entities and data owners, GDPR implements a negotiation-based approach which aims at including all relevant interests by different stakeholder groups into a compromise solution fitting for all of them.

In order to do so in the context of GoA4 development in the rail sector, these groups firstly have to be identified. Afterwards the possibly revealed personally identifiable information (PII) has to be

determined and a risk analysis/balance w.r.t. the legitimate interests of a railway undertaking developing GoA4 trains has to be done.

Analysis of people affected: Relevant groups are

- rail customers (e.g. on platforms)
- rail service employees (in the trackbed or on platforms)
- 3rd party individuals, e.g. people in the general public being visible while a train passes on a track next to them. This also refers to e.g. parked cars of which the license plate could be visible by a camera on the train.

Risk analysis for people affected: The data protection risk for individuals can generally be considered low. The potential risks associated with the processing of personal data, such as discrimination, identity theft, financial losses, and reputational damage, can be entirely or at least largely excluded here. Effects of processing the relevant data on the affected individuals are minimal overall.

From the perspective of the affected individuals, the content and significance of the camera images, insofar as any personal data can be recognized on them, are minimal in relation to the actual purpose of processing these images using algorithms and thus their conversion into a machine-readable environmental representation.

This is especially valid for cameras attached to the front of trains because these cover a limited area surrounding the vehicle, there are no audio recordings or live transmissions of the data. The recordings are thus limited in scope and functionality to what is necessary for creating environmental representations, obstacle detection, and collision avoidance.

Access to the stored data must be kept strictly limited to a few authorized individuals and is controlled with an access authorization concept within the implemented technical and organizational measures (see below). An identification of individuals in the data is not necessary for GoA4 development and should thus be excluded during data processing. Data containing personal content must generally be deleted after the end of the development (achieving the purpose of development). However, some data might have to be kept for safety audits/safety argumentation according to the respective norms.

Balance of interests between stakeholders:

- *Rail customers and people on a platform or other railway areas* are usually already subject to video surveillance measures to ensure their safety. They are informed by respective posters or signs and accept these recordings as a part of their travel. The video surveillance recordings are mostly long-term recordings, potentially live-viewed and cover the whole area of a train station or railway area. It can thus be concluded that an additional (very short!) recording by camera on a train passing a platform does not constitute a huge change or additional risk for a customer.
- *Individuals in the trackbed* can be considered employees of a rail company or a contractor because due to safety requirements, areas around tracks are not public and may not be accessed by 3rd party individuals. Therefore, the focus on data protection needs here can be seen with this employee/employer relationship in mind. It is thus important to integrate all

relevant committees (e.g. workers' council, labour unions etc.) in the definition of a suitable data protection process to ensure compliance and acceptance.

While it can e.g. be assumed that persons working in a track bed have an inherent interest that an automated GoA4 train is able to detect them and brake accordingly, another point in this discussion could be the employer's duty of care to ensure a safe and riskless work environment. Considering these arguments (and many more depending on the specific implementation of a project!) on the table, it can be assumed that a suitable solution in compliance with GDPR can be negotiated between the stakeholder groups of a railway undertaking company. It could e.g. be a solution to only anonymize employees of the company in certain areas while the data from other areas remains unchanged or to limit the data usage in other ways.

However, these considerations cannot be done generally but only within the scope of a specific project and development.

- 3rd party individuals could be affected by recordings when the train runs close to public areas such as roads, level crossings etc. It can again be assumed that on the one hand, public area are often subject to video surveillance measures and also the recording on-board the train will most likely only be very short and not capturing a longer trajectory or path of a person. Also e.g. license plates visible in public areas are a common thing and are not considered an additional risk if they appear on a video recording. Limiting access to the data and not using it for identification of individuals is a natural requirement here which should not interfere with the needs for development of a GoA4 train.
- On the other hand, railway undertakings (and their research & development partners) interested in GoA4 development also have legitimate interests to develop such systems. In order to extend their business and make it sustainable for the future, they have a legitimate interest to focus on new technologies and implement them for their business. These needs can be assumed to be aggravated e.g. by the climate crisis and related interest to focus on climate-friendly traffic technologies or also by a lack of train drivers that e.g. DB expects for the future.

  The general public is also expected to benefit from the development of GoA4 trains as e.g. an increased level of service, higher punctuality and more advantages are expected with GoA4 automation of railways as well as an overall technology development in Europe which is favourable for further future developments. All the aforementioned points are expected to constitute very important interests in favour of developing GoA4 technologies.

Expected technical/organisation measures to be taken according to GDPR: While the exact measures to be taken for ensuring a suitable balance of interest regarding privacy must be defined according to a specific project need, we want to outline at least a few general remarks in this document:

- Persons affected must be informed about potential video recordings. This can e.g. happen using respective signs or posters once people enter an area in which the recording could take place and must meet the respective GPRD requirements.
- Generally, it has to be said that the occurrence of obstacles around rail tracks appears rare during a train ride as trains are one of the safest measures of transport. The events of interest during the processing are thus sparse and require larger times of recording, but on the other

hand, much data can also be deleted after processing if it is identified to be less relevant. Therefore, a suitable selection and prioritization procedure is recommended as it can support the removal of irrelevant data and only data containing interesting events can be kept.

It should be noted that only a triggerless, indiscriminatory recording enables the recording of data for detection of obstacles during operation. If the scope is restricted, for example through anonymization or limitation of the range or the field of view of the cameras, there is a significant risk that the obstacle detection will be limited in its development and later may not reliably detect people in certain situations and trigger an automated emergency braking.

- Anonymization of data is generally **not possible**. An anonymization would adversely affect the training and evaluation of algorithms for the detection and classification of persons and objects, compromising the achievement of the development goals.

  The reasons for this are:

  - Reduced image quality: Anonymization techniques often involve blurring or pixelating parts of images containing personal information, which can affect image quality. This impairment can impact the ability of image recognition and object detection algorithms to accurately recognize and analyze visual features, leading to suboptimal performance.
  - Loss of context: Anonymization techniques can remove or alter context-related information in images. Example techniques include removing objects or parts of objects (e.g., blackening faces). Furthermore, the environment surrounding the objects might also be altered. This loss of context can affect the ability of image recognition algorithms to correctly understand and interpret the visual data.
  - Possible bias and generalization problems: Anonymization can create bias that affects the ability of image recognition algorithms to generalize from training data to real-world scenarios. For example, if certain facial features are consistently covered, the algorithm may struggle to recognize or classify individuals with similarly covered features.
  - Replacing personal data is problematic because, for instance, "replacement faces" must represent a realistic image of all recorded individuals. This includes characteristics such as gender, skin colour, and age. However, evaluating precisely these characteristics requires processing special categories of personal data according to Article 9 of the GDPR.

  Although a general anonymization is not feasible, in the specific project setting, anonymization should of course be implemented where possible.

- Storing of data should be done with the homologation procedure in mind. Although there is not yet a clear path on homologation of GoA4 systems, it is likely that regulation will require all data used during the development to be stored for future reference or audits in case of potential safety issues. The data to be stored for such longer times must be selected carefully and the process must be accompanied by data protection experts to ensure compliance to both technical norms and GDPR.

With these measures taken and a careful, considerate approach implemented together with close collaboration by data protection experts and authorities, we conclude that the development of GoA4

trains can be realized in accordance with GDPR and related national regulation. An individual risk assessment for a specific project has to be done regardless of the summary presented here.

Regulatory Compliance

- Data protection and privacy compliance
  - The system shall implement measures ensuring compliance with GDPR, including technical and organizational safeguards such as pseudonymization, encryption, and access controls. A Data Protection Impact Assessment (DPIA) shall be performed for all systems processing PII..

    Rationale: In order to ensure the regulations are followed.
- Define Compliance Auditing Periods
  - Regular audits shall be conducted to ensure ongoing compliance, and reports should be generated for regulatory reporting.

    Rationale: In order to ensure the rules of compliance are followed.

2.5.8.3 Data Security

Data Security is a key component of Data Governance, comprising the tools and methods used to protect data from unauthorized access or breaches. This encompasses security protocols for data at rest and in transit, such as encryption, and measures to prevent data leaks or theft. Security is a proactive stance to guard the organization's most valuable asset - its data.

- Define Security Protocols
  - Unauthorized Access Protection
    - Security protocols and standards shall be followed to protect against unauthorized access and data breaches.

      Rationale: In order to ensure the security according to accepted standards.
- Data Encryption
  - Data Encryption Mechanisms.
    - The system shall implement encryption mechanisms for data in transit and at rest.

      Rationale: In order to avoid unauthorized access to protected data.

2.5.8.4 User Training and Documentation

User Training and Documentation are the final pillars of robust Data Governance. By providing comprehensive training and clear documentation, stakeholders are made aware of the governance policies, compliance obligations, and security protocols. This training is crucial for ensuring that everyone who interacts with the system understands their role in safeguarding data.

- Applicable User Training
  - Training Materials
    - The system shall provide user training materials and documentation to ensure that stakeholders understand and adhere to data governance policies.

Rationale: In order to ensure that stakeholders understand and adhere to data governance policies.

- Regular Training
    - Information about Updates
        - Regular training sessions shall be conducted for users.

          Rationale: In order to stay informed about updates and best practices.

In DAFA, governance, compliance, and security are not just operational requirements; they are commitments to excellence in data management, forming the bedrock of trust and operational efficiency.

2.6 SYSTEM

In order to specify the system and derive the architecture, the stakeholder needs were collected and put into a standardised notation (section 2.6.1) and the system description can be found in the following section 2.6.3.

Section 2.6.2 lists the system security requirements, that are derived from security standards. Parts of these requirements leads to functional security requirement in the respective subsystem (section 2.7.9).

2.6.1 Stakeholder Needs

In this section, the needs of the stakeholder are listed by the respective actor or connected system in the respective sub-groups. The stakeholder needs are derived from a demand, a necessity of personas and/or entities, which will raise expectations for the system to be operated. In total, the system Data Factory provides the actors to the system with the underlying entities such as Onboard Data, Testing Data, Data, Data Factory Operation and invoice creation, Customers and Data Factory Teams.

Subsequently, the addressed needs are grouped into the Stakeholder Laboratory, Train Onboard System, Train Onboard System, External Data Factories, Data Factory Operator, Customer and Data Factory Teams.

2.6.1.1 Laboratory

- As a Laboratory, I need to replay sensor data.
    - Rational: in order to perform HiL and SiL tests.
- As a Laboratory, I need to access simulation scenarios.
    - Rational: in order to perform HiL and SiL tests.
- As a Laboratory, I need to download neural network models.
    - Rational: in order to validate the neural network models.

2.6.1.2 Train Onboard Systems

- As a GoA4 Sensor Onboard System, I need to store sensor data.

- Rational: store data of new or rare situations in order to improve the neural network performance within the onboard system.

2.6.1.3 External Data Factories

- As a Data Factory EU Partner, I need to connect my data sources with the Data Factory.
  - Rational: in order to exchange data.

2.6.1.4 Data Factory Operators

- As a Data Factory Operator, I need to operate the Data Factory cost covering.
  - Rational: in order to operate the Data Factory economically.
- As a Data Factory Operator, I need industry partners to develop their machine learning models for GoA4 operation within the Data Factory.
  - Rational: in order to keep sovereignty over my sensor data and allow all partners to train certifiable network models. In addition, it might not be economically possible for each industry partner to operate their own Data Factory which is certified.
- As a Data Factory Operator, I need to charge the Data Factory users based on their resource consumption.
  - Rational: in order to apply a fair business model for the data factory.
- As a Data Factory Operator, I need to share sensor data across Data Factories within Europe.
  - Rational: in order to operate trains across country borders.
- As a Data Factory Operator, I need to ingest sensor data from my trains.
  - Rational: in order to improve the network models and improve the reliability of the trains.
- As a Data Factory Operator, I need to restrict the access to sensor data and assets for specific users or user groups.
  - Rational: in order to comply with regulations (and contracts).
- As a Data Factory Operator, I need to persist the versions of my assets that are used in operation.
  - Rational: as regulations for certification may require ensuring reproducibility.
- As a Data Factory Operator, I need to monitor the assets within the Data Factory.
  - Rational: in order to get notified when defined thresholds are exceeded.
- As a Data Factory Operator, I need to log activities within the Data Factory.
  - Rational: in order to get notified about failures, unexpected behavior or security incidents.
- As a Data Factory Operator, I need to redeploy the infrastructure automatically to recover from system failures/crashes.
  - Rational: in order to get system back up running
    - Backups

  - IaC
  - Monitoring
  - Logging

- As a Data Factory Operator, I need to provide documentation of the services within the Data Factory to the users.
  - Rational: in order to enable users to use the Data Factory in the intended way ensuring that the certification stays valid.
- As a Data Factory Operator, I need a central identity access management platform.
  - Rational: in order to have consistent user permissions across the Data Factory.
- As a Data Factory Operator, I need to set access policies for sensor data and datasets.
  - Rational: in order to protect sensitive information.
- As a Data Factory Operator, I need to anonymize sensor data.
  - Rational: in order to provide GDPR compliant data insights.

2.6.1.5 Customers

- As a Customer, I need to access (sensor) data.
  - Rational: in order to get an overview of possibilities.
- As a Customer, I need to train machine learning models.
  - Rational: in order to build autonomous trains, predictive maintenance applications.
- As a Customer, I need to develop and improve machine learning model architectures. (ML Sub-Sys)
  - Rational: in order to train and improve machine learning models.
- As a Customer, I need to create and manage datasets.
  - Rational: in order to train machine learning models for my use-case.
- As a Customer, I need to evaluate the performance of my models.
  - Rational: in order to find weaknesses and potential for improvement of the model.
- As a Customer, I need to validate the performance of my machine learning models.
  - Rational: in order to test if all certification requirements can be fulfilled by ensuring that it is working under multiple environmental conditions.
- As a Customer, I need to improve the performance of my machine learning models. (ML Sub-Sys)
  - Rational: in order to improve reliability and ensure the quality of the service.
- As a Customer, I need to search for specific data samples based on metadata information.
  - Rational: in order to find suitable data for my datasets.
- As a Customer, I need to use synthetic data.
  - Rational: in order to cover scenarios that cannot be recorded.

- As a Customer, I need to request synthetic data for a specific scenario. (Sub-Sys)
    - Rational: in order to get training and validation data for my specific use-case.
- Assumption: It is not intended to provide access for customers to directly access the scenario builder.
- As a Customer, I need synthetic data to include my custom 3D assets.
    - Rational: As an example, own train models, custom environmental assets, ...
- As a Customer, I need synthetic data generated using my custom sensor models.
    - Rational: Synthetic sensors with my sensor properties (sensor technology, shutter speed, noise)
- As a Customer, I need labels for the sensor data.
    - Rational: in order to train machine learning models.
- As a Customer, I need to share assets with my colleagues.
    - Rational: in order to collaborate.
- As a Customer, I need to store and version my machine learning models.
    - Rational: in order to compare different versions of my models
- As a Customer, I need to monitor sensor data and machine learning models.
    - Rational: in order to improve the model training.
- As a Customer, I need to understand characteristics of available data.
    - Rational: in order to ensure that the diversity of the data is sufficient.
        - Which objects appear in our datasets at all
        - How often do certain scenarios occur (animal on track or similar)
        - Are level crossings sufficiently secured (e.g. picture of level crossing, or statistics about objects on level crossing)
        - Do I have all relevant scenarios in the database / sensor data (day/night, seasons, city, country, tunnel, bridge, ...
- As a Customer, I need to prioritize my jobs.
    - Rational: in order to favor urgent tasks when available resources are limited
- As a Customer, I need to homologate my machine learning models.
    - Rational: in order to deploy machine learning models in a safety critical environment.
- As a Customer, I need to search for experiments and network models.
    - Rationale: in order to find the respective experiments or network models
- As a Customer, I need to import local assets into the Data Factory.
    - Rational: For example, I want to use pre-trained models or my custom architecture or my custom 3d models for simulation and everything as well (phil).
- As a Customer, I need to export assets from the Data Factory to my machine.

- As a Customer, I need to use the toolchain within the Data Factory to develop machine learning based software for GoA4 train operation
    - Rational: I don't want or cannot afford to operate my own Data Factory.
- As a Customer, I need to restrict the access to my assets.
    - Rational: In order to protect my intellectual property.
- As a Customer, I need to have a private space within the Data Factory.
    - Rational: In order to share data only within my team of employees.
- As a Customer, I need to manage the access my employees to the toolchain.
    - Rational: In order to give new colleagues access or remove old ones.
- As a Customer, I need to homologate the machine learning based software for GoA4 train operation.
    - Rational: in order to be allowed to implement it to trains.
- As a Customer, I need to have a test environment to do system integration tests.
    - Rational: in order to do Sil, HiL and MiL tests.
- As a Customer, I need to avoid the loss of my stored or consumed data.
    - Rational: import when models run on trains for certification regulations
- As a Customer, I need training material on how to use the Data Factory.
    - Rational: to train my employees find a name for users that are associated to a specific customer (currently "my employees")
- As a Customer, I need a point of contact for getting help.
- As a Customer, I need an overview over my currently billing amount.
    - Rational: in order to estimate my costs.
- As a Customer, I need an overview over my system resource usage.
    - Rational: in order to verify the bill.
- As a Customer, I need to generate statistics of stored data.
    - Rational: in order to generate KPIs.
- As a Customer, I need to analyze data in an automated way.

2.6.1.6 Data Factory Teams

- As a Data Factory Team, I need to transfer data from trains to the Data Center.
    - Rational: in order to provide real sensor data for every operational situation that was recorded.
- As a Data Factory Team, I need to transform data on import into standardized formats.
    - Rational: in order to have all sensor data in a common format.
- As a Data Factory Team, I need to ensure sufficient data quality.

  - Rational: in order to indicate whether data is usable for further processing (e.g. labelling, training network models).
- As a Data Factory Team, I need to ensure the integrity of stored sensor data.
  - Rational: in order to prove that data is not manipulated and is safe to use for training certifiable network models.
- As a Data Factory Team, I need to generate metadata for each sensor data frame.
  - Rational: in order to allow querying data for specific use cases.
- As a Data Factory Team, I need to process data in an automated way.
  - Rational: in order to apply algorithms and extract data from large amounts of stored data with minimal effort.
- As a Data Factory Team, I need to delete faulty data.
  - Rational: in order to reduce operation costs as this data cannot be used for any purpose
- As a Data Factory Team, I need to delete unnecessary data.
  - Rational: in order to reduce operation costs as the data is unlikely to be useful for training and validating neural network models.
- As a Data Factory Team, I need to automate detection of objects in Frames.
  - Rational: in order to reduce costs and effort.
- As a Data Factory Team, I need to add my own tags and comments to datasets.
  - Rational: in order to expand and detail metadata.
- As a Data Factory Team, I need to visualize frames including annotations.
  - Rational: in order to get an overview of the frame of the label.
- As a Data Factory Team, I need to optimize the storage space consumption of the data.
- As a Data Factory Team, I need to run simulations.
  - Rational: in order to generate synthetic sensor data.
- As a Data Factory Team, I need to create and store simulation scenarios.
  - Rational: in order to run simulations.
- As a Data Factory Team, I need to use 3D objects within my scenarios.
  - Rational: in order to build realistic scenarios.
- As a Data Factory Team, I need to create and store 3D objects.
  - Rational: in order to create representations for objects that are relevant in the railway environment and build my scenarios as close to reality as possible.
- As a Data Factory Team, I need to build a representation of my sensor that matches the physical properties.
  - Rational: in order to generate realistic synthetic sensor data.
- As a Data Factory Team, I need to store my synthetic sensor data within the Data Factory.

- Rational: in order to make it available for Data Scientists.

- As a Data Factory Team, I need to run simulations to optimize specifications.
  - Rational: in order to find the optimal parameters and settings for sensor ranges and opening angles.
- As a Data Factory Team, I need to do assessments between real world data and simulated data.
  - Rational: in order to assure high quality of simulated data by simulations.

2.6.2 System Security Requirements

In the context of the R2DATO project, the security of the system and its data is paramount in ensuring the integrity and reliability of the Data Factory. The System Security Requirements are designed to safeguard both the system and its data against a myriad of threats emanating from the environment. These threats can be both intentional, such as hacking or theft, and accidental, such as malfunctions or natural disasters.

The System Security segment is dedicated to outlining the necessary defences and protocols to maintain the confidentiality, integrity, and availability of the system and its data. These elements are crucial in mitigating the risk of operational disruptions, financial losses, and potential safety impacts on stakeholders. Our approach is aligned with best practices and standards, including the adoption of an Information Security Management System (ISMS) in accordance with ISO 27001 [18] and a risk-based security management strategy guided by ISO 27002 [19] controls.

Furthermore, considering the Data Factory's pivotal role in developing algorithms for future railway systems, we integrate security principles from the IEC 62443 [20] standard, which focuses on industrial automation and control systems. This section elucidates our commitment to technical security measures while assuming operational security is managed by the Data Factory operator, who also provides essential security services such as asset management and network security.

The System Security Requirements are structured to address various aspects of security, aiming to ensure the confidentiality, integrity, and availability of the system and its data. Here is an overview of the structure based on the details provided:

2.6.2.1 Technical Security Measures

- Identity and Access Management: Ensuring only authorized users can access the system.
- Privileged Access Management: Special controls for users with elevated access rights.
- Data Confidentiality: Protecting data from unauthorized access and disclosure.
- System Integrity: Ensuring data and system functionality are not improperly modified.
- System Availability: Ensuring the system is accessible and usable when needed.
- System Hardening: Implementing measures to reduce system vulnerabilities.
- Asset and Patch Management: Keeping software updated and managing hardware assets.
- Backup and Restore: Ensuring data can be recovered in the event of loss or corruption.
- Logging and Monitoring: Keeping records of system activities for security analysis.

- Network Security: Protecting the system from network-based threats.
- Physical Security: Preventing unauthorized physical access to system components.

2.6.2.2 General non-functional system security requirements

The following three requirements consider the three primary goals of security confidentiality, integrity and availability. The general approach to ensure security within a system is to ensure that a loss of these goals is avoided based on a risk management approach. Especially that means that an acceptable level of risk has to be defined and that all risks do not exceed this risk level.

The Data Factory shall ensure that the risk of a loss of confidentiality is acceptable according to internal and external regulations.

Confidentiality is one of the primary goals of security and risks have to be acceptable.

The Data Factory shall ensure that the risk of a lost of integrity is acceptable according to internal and external regulations.

Integrity is one of the primary goals of security and risks have to be acceptable

The Data Factory shall ensure that the risk of a lost of availability is acceptable according to internal and external regulations.

Availability is one of the primary goals of security and risks have to be acceptable

2.6.2.3 System security requirements derived from IEC 62443-3-3

There are 7 groups of foundational security requirements coming from IEC 62443-3-3 standard:

- Identification and Authentication Control
- Use Control
- System Integrity
- Data Confidentiality
- Restricted Data Flow
- Timely Response to Events
- Resource Availability

For each of the seven groups system requirements (SR) for security for the Data Factory were derived. The following section lists these system requirements and the first number in the numbering indicates to which foundational security group the requirement belongs to.

SR 1.1 - Human user identification and authentication

- The Data Factory shall provide the function to identify natural users.
- The Data Factory shall provide the function to authenticate natural users.

SR 1.2 – Software process and device identification and authentication.

- No requirements were derived from SR1.2

SR 1.3 – Account management

- The Data Factory shall provide the function to add accounts of authorized users.

- The Data Factory shall provide the function to active accounts of authorized users.
- The Data Factory shall provide the function to modify accounts of authorized users.
- The Data Factory shall provide the function to disable accounts of authorized users.
- The Data Factory shall provide the function to remove accounts of authorized users.

SR 1.4 – Identifier management

- The Data Factory shall implement identification by natural user.
- The Data Factory shall implement identification by group.
- The Data Factory shall implement identification by role.
- The Data Factory shall implement identification by control system interface.
- The Data Factory shall provide the function to initialize authenticator content. Info: Authenticators include credentials like password etc.

SR 1.5 – Authenticator management

- The Data Factory shall provide the function to change all default authenticators after Data Factory installation and setup.
- The Data Factory shall provide the function to change all authenticators.
- The Data Factory shall protect the storage of all authenticators from unauthorized disclosure.
- The Data Factory shall protect the transmission of all authenticators from unauthorized disclosure.
- The Data Factory shall protect the storage of all authenticators from unauthorized modification.
- The Data Factory shall protect the transmission of all authenticators from unauthorized modification.

SR 1.6 – Wireless access management

- The Data Factory shall identify all natural users engaged in wireless communication.
- The Data Factory shall authenticate all natural users engaged in wireless communication.
- The Data Factory shall identify all software processes engaged in wireless communication.
- The Data Factory shall authenticate all software processes engaged in wireless communication.
- The Data Factory shall identify all devices engaged in wireless communication.
- The Data Factory shall authenticate all devices engaged in wireless communication.

SR 1.7 – Strength of password-based authentication

- The Data Factory shall enforce passwords strength based on minimum length.
- The Data Factory shall enforce passwords strength based on variety of character types.

SR 1.8 – Public Key Infrastructure (PKI) certificates

- No requirements were derived from SR 1.8

SR 1.9 – Strength of public key authentication

- No requirements were derived from SR 1.9 – Strength of public key authentication.

SR 1.10 – Authenticator feedback

- The Data Factory shall obscure feedback of authentication information during authentication process.

SR 1.11 – Unsuccessful login attempts

- The Data Factory shall enforce a configurable number of consecutive invalid access attempts by any natural user.
- The Data Factory shall enforce a configurable number of consecutive invalid access attempts by any software process.
- The Data Factory shall enforce a configurable number of consecutive invalid access attempts by any device.
- The Data Factory shall deny access for specified period of time, when this limit has been exceeded.
- The Data Factory shall deny access until unlocked by an administrator, when this limit has been exceeded.

SR 1.12 – System use notification

- The Data Factory shall implement a system use notification message before authentication. Info: This may include Terms of Use
- The Data Factory shall ensure that the system use notification message can be configured by authorized personal.

SR 1.13 – Access via untrusted networks

- The Data Factory shall monitor all methods of access via untrusted networks.
- The Data Factory shall control all methods of access via untrusted networks.

SR 2.1 – Authorization enforcement

- The Data Factory shall implement segregation of duties and least privilege.
- The Data Factory shall enforce authorizations assigned to all natural users on all interfaces.

SR 2.2 – Wireless use control

- The Data Factory shall authorize usage restrictions for wireless connectivity according to commonly accepted security industry practices.
- The Data Factory shall monitor usage restrictions for wireless connectivity according to commonly accepted security industry practices.
- The Data Factory shall enforce usage restrictions for wireless connectivity according to commonly accepted security industry practices.

SR 2.3 – Use control for portable and mobile devices

- The Data Factory shall provide the function to configure usage restrictions.
- The Data Factory shall implement the option to configure usage restrictions to prevent the usage of portable devices.
- The Data Factory shall implement the option to configure usage restrictions to prevent the usage of mobile devices.
- The Data Factory shall implement the option to configure usage restrictions to require context specific authorization.
- The Data Factory shall implement the option to configure usage restrictions to restrict code transfer to/from portable devices.
- The Data Factory shall implement the option to configure usage restrictions to restrict code transfer to/from mobile devices.
- The Data Factory shall implement the option to configure usage restrictions to restrict data transfer to/from portable devices.
- The Data Factory shall implement the option to configure usage restrictions to restrict data transfer to/from mobile devices.
- The Data Factory shall enforce configured usage restrictions.

SR 2.4 – Mobile code

- The Data Factory shall implement the option to configure usage restrictions for mobile code technologies. Info: Mobile code includes scripting language etc.
- The Data Factory shall implement the option to configure usage restrictions to prevent the execution of mobile code.
- The Data Factory shall implement the option to configure usage restrictions to require authentication and authorization for the origin of the mobile code.
- The Data Factory shall implement the option to configure usage restrictions to restrict mobile code transfer to/from the system.
- The Data Factory shall implement the option to configure usage restrictions to monitor the use of mobile code.

SR 2.5 – Session lock

- The Data Factory shall implement the option to initiate a session lock after configurable time period of inactivity.
- The Data Factory shall implement the option to initiate a session lock by manual initiation.
- The Data Factory shall deny user interaction during session lock.
- The Data Factory shall implement the reactivation of a locked session by authorized human user.

SR 2.6 – Remote session termination.

- No requirements were derived from SR 2.6.

SR 2.7 – Concurrent session control

- No requirements were derived from SR 2.7.

SR 2.8 Auditable Events

- The Data Factory shall generate security audit records for access control.
- The Data Factory shall generate security audit records for request errors.
- The Data Factory shall generate security audit records for operating system events.
- The Data Factory shall generate security audit records for backup events.
- The Data Factory shall generate security audit records for restore events.
- The Data Factory shall generate security audit records for configuration changes.
- The Data Factory shall generate security audit records for potential reconnaissance activity.
- The Data Factory shall generate security audit records for audit log events.
- The Data Factory shall export audit records in industry standard format.
- The Security Audit Record shall include the time stamp.
- The Security Audit Record shall include the source activity.
- The Security Audit Record shall include the category.
- The Security Audit Record shall include the type.
- The Security Audit Record shall include the event id.
- The Security Audit Record shall include the event result.

SR 2.9 – Audit storage capacity

- The Data Factory shall provide sufficient storage capacity for audit records.

SR 2.10 – Response to audit processing failure

- The Data Factory shall alert personal to prevent the loss of essential services, in the event of an audit processing failure.
- The Data Factory shall implement appropriate actions in response to audit processing failure.

SR 2.11 – Timestamps

- No requirements were derived from SR 2.11.

SR 2.12 – Non-repudiation

- No requirements were derived from S2.12.

SR 3.1 – Communication integrity

- The Data Factory shall protect the integrity of transmitted information.

SR 3.2 – Malicious code protection

- The Data Factory shall prevent the execution of malicious code.
- The Data Factory shall detect the malicious code.
- The Data Factory shall report the malicious code.
- The Data Factory shall mitigate the effects of malicious code.

- The Data Factory shall prevent the execution of unauthorized software.
- The Data Factory shall detect the unauthorized software.
- The Data Factory shall report the unauthorized software.
- The Data Factory shall mitigate the effects of unauthorized software.
- The Data Factory shall provide the function to update the malicious code protection system.

SR 3.3 – Security functionality verification

- The Data Factory Team shall verify the intended operation of security functions.
- The Data Factory Team shall report anomalies discovered during FAT, SAT and scheduled maintenance. FAT: Factory Acceptance Test, SAT: Site Acceptance Test.

SR 3.4 – Software and information integrity

- No requirements were derived from SR 3.4.

SR 3.5 – Input validation

- The Data Factory shall validate the syntax and content of any input.

SR 3.6 – Deterministic output

- If an attack prevents normal operation, the Data Factory shall set all output to a predetermined state.

SR 3.7 – Error handling

- No requirements were derived from SR 3.7.

SR 3.8 – Session integrity

- No requirements were derived from SR 3.8.

SR 3.9 – Protection of audit information

- No requirements were derived from SR 3.9

SR 4.1 – Information confidentiality

- The Data Factory shall protect the confidentiality of information at rest.
- The Data Factory shall protect the confidentiality of information in transit.

SR 4.2 – Information persistence

- No requirements were derived from SR 4.2 .

SR 4.3 – Use of cryptography

- The Data Factory shall use cryptographic algorithms, key sizes and mechanisms for key establishment and management according to commonly accepted security industry practices and recommendations.

SR 5.1 – Network segmentation

- The Data Factory shall logically segment control system networks from non-control system networks.

- The Data Factory shall logically segment critical control system networks from other control system networks.

SR 5.2 – Zone boundary protection

- The Data Factory shall monitor the communication at zone boundaries.
- The Data Factory shall control the communication at zone boundaries.
- The Data Factory shall enforce the compartmentalization defined in the risk-based zones and conduits model.

SR 5.3 – Zone boundary protection

- The Data Factory shall prevent the usage of general purpose person-to-person messaging system from users externals to the control system. Info: General purpose person-to-person messaging system include E-Mail, Facebook, Twitter, etc.
- The Data Factory shall prevent the usage of general purpose person-to-person messaging system from systems externals to the control system.

SR 5.4 – Application partitioning

- The Data Factory shall partition the data, applications and services based on their criticality.

SR 6.1 – Audit log accessibility

- The Data Factory shall provide the function to access audit logs by authorize humans on a read-only basis.
- The Data Factory shall provide the function to access audit logs by authorize tools on a read-only basis.

SR 6.2 – Continuous monitoring

- No requirements were derived from SR 6.2.

SR 7.1 – Denial of service protection

- The Data Factory shall operate in a degraded mode during a DoS event.

SR 7.2 – Resource management

- The Data Factory shall provide the function to limit the use of resources (to prevent resource exhaustion).

SR 7.3 – Control system backup

- The Data Factory shall conduct user-level backups of the critical files and directories.
- The Data Factory shall conduct system-level backups of the critical files and directories including system state information.

SR 7.4 – Control system recovery and reconstitution

- If a disruption or failure occur, the Data Factory shall recover to a known secure state.
- If a disruption or failure occur, the Data Factory shall reconstitute to a known secure state.

SR 7.5 – Emergency power

- If the normal power supply is disrupted, the Data Factory shall switch to emergency power supply.
- If the normal power supply is disrupted, the Data Factory shall continue operation in the current security state or a documented degraded mode.
- If the normal power supply is restored, the Data Factory shall switch back to normal power supply.
- If the normal power supply is restored, the Data Factory shall stay in the documented degraded mode or continue operation in the normal mode.

SR 7.6 – Network and security configuration settings

- The Data Factory shall be configured according to the recommended network and security configuration as described in the guidelines provided by the supplier.
- The Data Factory shall provide an interface to the currently deployed network and security configuration settings.

SR 7.7 – Least functionality

- If the prevention of the usage of unnecessary functions is possible, the Data Factory shall prevent the usage of unnecessary functions.
- If the prevention of the usage of unnecessary ports is possible, the Data Factory shall prevent the usage of unnecessary ports.
- If the prevention of the usage of unnecessary protocols is possible, the Data Factory shall prevent the usage of unnecessary protocols.
- If the prevention of the usage of unnecessary services is possible, the Data Factory shall prevent the usage of unnecessary services.
- If the prevention of the usage of unnecessary functions is not possible, the Data Factory shall restrict the usage of unnecessary functions.
- If the prevention of the usage of unnecessary ports is not possible, the Data Factory shall restrict the usage of unnecessary ports.
- If the prevention of the usage of unnecessary protocols is not possible, the Data Factory shall restrict the usage of unnecessary protocols.
- If the prevention of the usage of unnecessary services is not possible, the Data Factory shall restrict the usage of unnecessary services.

SR 7.8 – Control system component inventory

- No requirements were derived from SR 7.8.

2.6.3 System description

From the stakeholder needs (section 2.6.1) the black box system description was derived and is depicted in Figure 26. It shows the respective stakeholders and external systems from section 2.6.1.

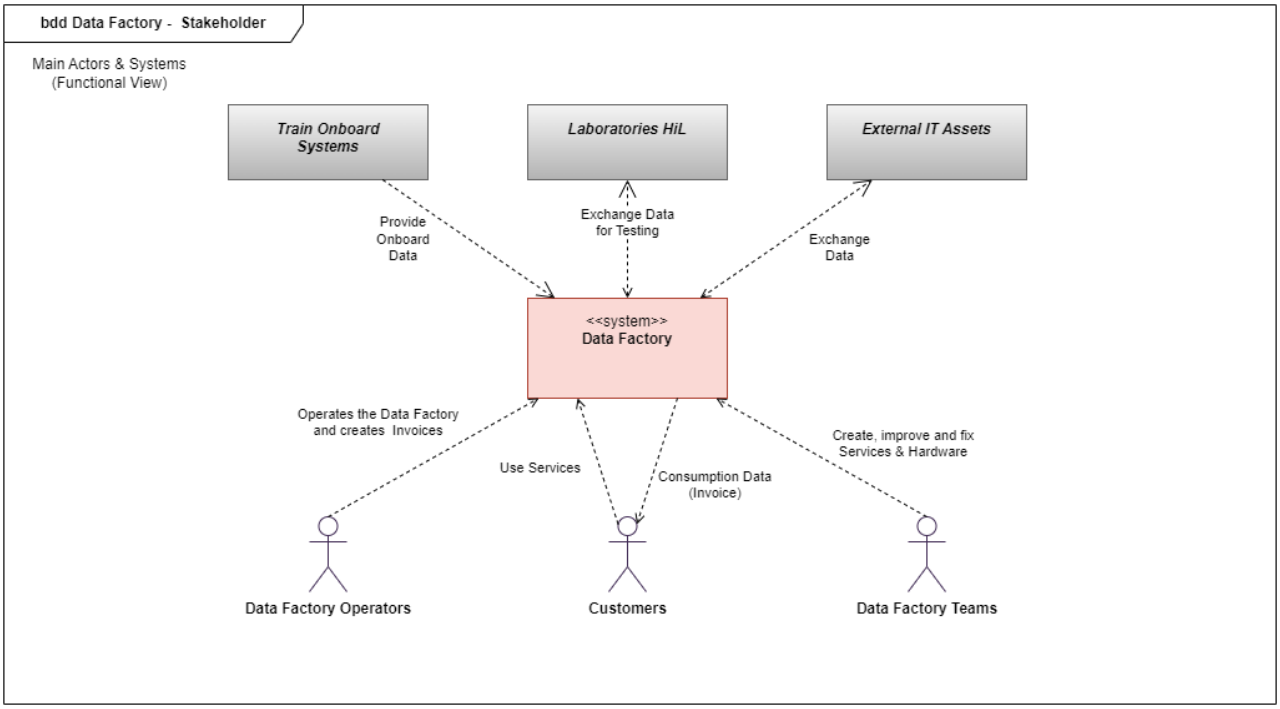


Figure 26: System description as black box

The Data Factory as black box can be described as a system which as input stores onboard data from trains, including perception sensor data. As output to the customer data for GoA4 ATO systems is provided. This data includes sensor data, annotations, trained ML models etc., compare section 2.7.1.

The data input into the Data Factory is coming from Train Onboard Systems and External IT Assets and contains:

- Camera data
- LiDAR data
- RADAR data
- Localization data
- Other sensor data
- Annotations
- Vehicle data
- Data sanity information
- Diagnostics data
- Resource information
- System health information
- 3D assets
- Digital twins, including sensor models

The output of the Data Factory goes to Laboratories HiL, External IT assets and the customers and contains the same type of data as mentioned before and additionally the following data types:

- Synthetic sensor data
- Datasets
- Trained models including architecture and weights
- Model configurations
- Test reports
- Training information
- Log data
- Invoices

The following section 2.7 describes the functions and subsystems necessary to achieve the here described data flows and puts the stakeholders and external systems into context.

The following processes or functions are out of scope and are not considered in section 2.7.

- Any on-train operations
- Any model deployment to trains / vehicles
- Operational parts of the data centre
- Certification of training results
- Homologation processes an functionalities

2.7 SUBSYSTEMS

The subsystems constitute the foundational elements of our Data Factory, serving as critical components that underpin the operational framework and enable the realization of the system's overarching functionality. Each subsystem is designed to fulfil a specific set of roles within the larger ecosystem, functioning as an integral piece of the puzzle that drives the Data Factory towards its strategic objectives.

The purpose of detailing the subsystem requirements is to provide a comprehensive understanding of the individual and collective contributions of these components to the system's performance. By delineating the specifications and requirements of each subsystem, we establish clear expectations for functionality, interoperability, and performance standards that are necessary for the seamless integration and optimization of the Data Factory.

In this section, we will explore the intricate makeup of each subsystem, elucidating their purposes, the specific needs they address, and the mechanisms by which they contribute to the data lifecycle within the Data Factory. From data acquisition and management to processing and analytics, each subsystem plays a pivotal role in ensuring data flows efficiently and securely, allowing for robust and scalable solutions tailored to meet our stakeholders' diverse requirements.

Through this exploration, we will gain insight into how each subsystem's design and requirements align with our strategic goals, ensuring that the Data Factory operates not just as a collection of independent modules, but as a cohesive and harmonized unit, driving innovation and value creation.

A note on the interactions: When an interaction is described as bi-directional (both in the text and the images), then the exchanged data are regarded on a high level such that some specific types of data are transferred in one direction, others in the other.

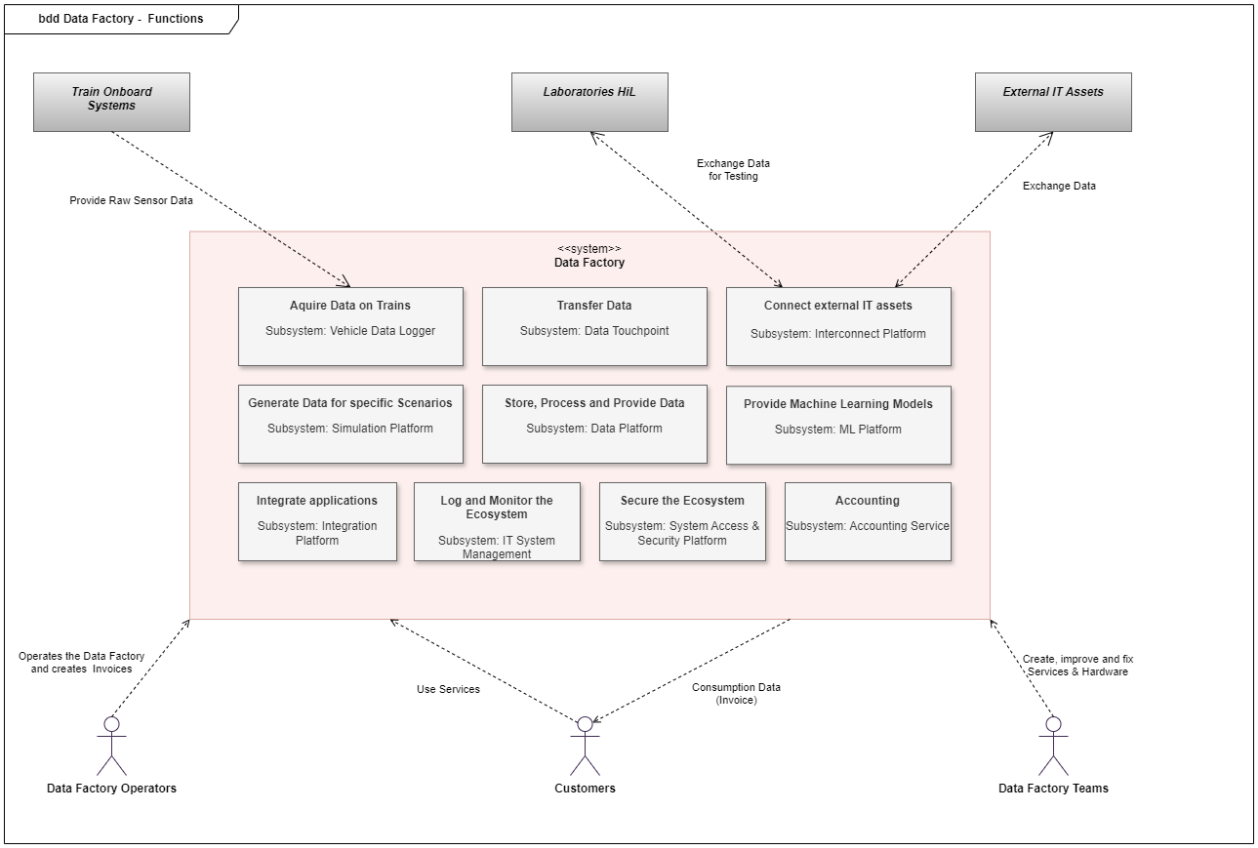


Figure 27: Whitebox view Data Factory

- Train Onboard Systems
  - Subsystem: Vehicle Data Logger
  - Interaction: Gathers and provides raw sensor data to the Data Factory, constituting the first crucial step in the data lifecycle.
- Laboratories HIL
  - Interaction: Receives processed data for testing, including synthetic sensor data, annotations, and ML models from the Data Factory.
- External IT-Assets
  - Interaction: Exchanges data with the Data Factory, providing external data sources and receiving data, enhancing the richness of the Data Factory's datasets.
- Data Factory Functions
  - Subsystem Function: Acquire Data on Trains
    - Interaction: Processes the raw sensor data from Train Onboard Systems, performing initial tasks such as filtering and encoding, to ready it for further use.
    - Dependency: Relies on the constant and reliable inflow of raw data from the Train Onboard Systems.
  - Subsystem Function: Generate Data for Specific Scenarios

        - Interaction: Generates refined datasets for simulations and models, tailored to specific analytical requirements of the Data Factory.
        - Dependency: Builds on the pre-processed data from the Vehicle Data Logger subsystem.
    - Subsystem Function: Integrate Applications
        - Interaction: Integrates various applications with the Data Factory, enabling the utilization of processed data across multiple platforms and services.
        - Dependency: The quality and accessibility of data processed by the Data Factory are crucial for successful integration.
    - Subsystem Function: Store, Process and Provide Data
        - Interaction: Manages the storage, further processing, and provisioning of data to stakeholders, ensuring data is ready for access and use.
        - Dependency: Works closely with both the acquisition and application integration stages of the data lifecycle.
    - Subsystem Function: Log and Monitor the Ecosystem
        - Interaction: Logs system activities and monitors the health and performance of the Data Factory, providing important feedback and alerts.
        - Dependency: Dependent on the integration and smooth operation of all subsystems to accurately monitor and log their activities.
    - Subsystem Function: Provide Machine Learning Models
        - Interaction: Develops and delivers machine learning models, which enhance the functionality and analytical capabilities of the Data Factory.
        - Dependency: Requires a steady supply of well-curated data from the Data Platform for training and refining models.
    - Subsystem Function: Secure the Ecosystem
        - Interaction: Implements and manages security protocols to protect data integrity and factory operations.
        - Dependency: Must remain synchronized with all subsystems to maintain comprehensive security measures.
    - Subsystem Function: Accounting
        - Interaction: Tracks the consumption of Data Factory resources, potentially for invoicing or resource management.
        - Dependency: Utilizes data from the monitoring subsystem for accurate accounting.
- Data Factory Roles
    - Data Factory Operators
        - Interaction: Oversee the operation of the Data Factory, ensuring that all subsystems function harmoniously and efficiently.

- Dependency: Reliant on the seamless interaction between subsystems to maintain operational continuity.

- Customers
  - Interaction: Utilize the Data Factory's services, which can include anything from accessing processed data to utilizing insights from machine learning models.
  - Dependency: Customer needs influence the development and provision of services, driving the focus of data processing and analysis.
- Data Factory Teams
  - Interaction: Responsible for creating, improving, and fixing services and hardware within the Data Factory ecosystem.
  - Dependency: Their work is dependent on feedback from system monitoring and user interactions to prioritize development and maintenance tasks effectively.

In summary, the Data Factory is a complex network of interdependent subsystems, each playing a specific role in the data's journey from raw input to valuable output. The successful operation of this ecosystem hinges on the seamless integration and secure handling of data, ensuring the Data Factory can provide high-quality services to its operators and customers.

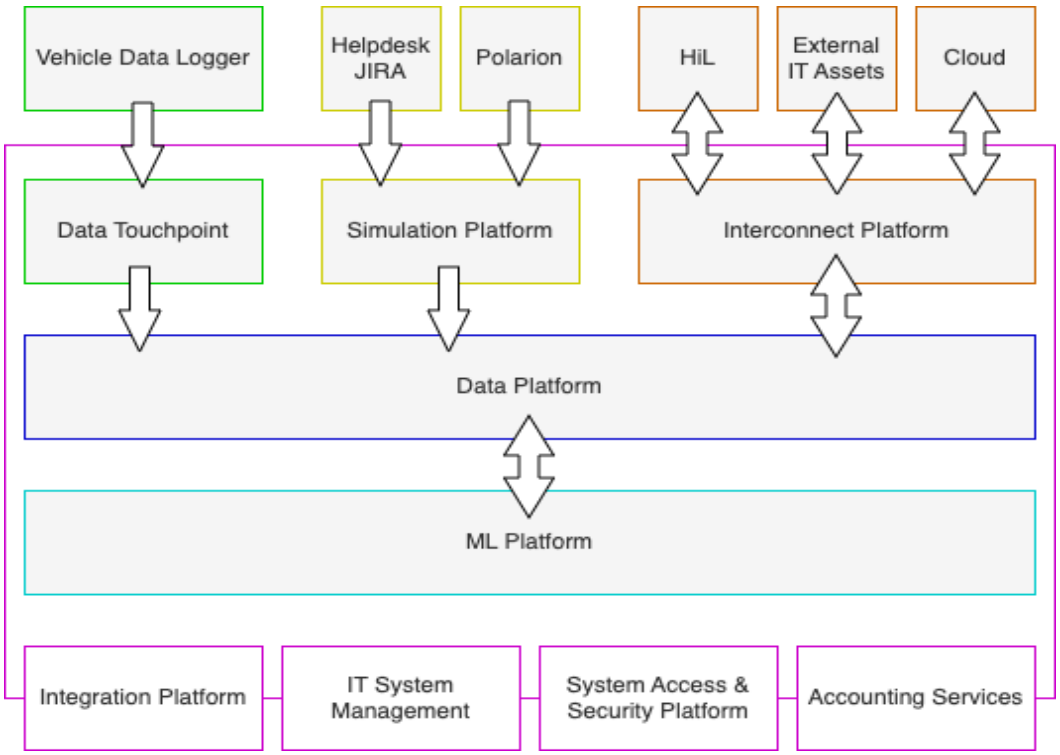


Figure 28: High Level Block Diagram

2.7.1 Dataflow Diagram

The aim of Figure 29 is to provide an overview on the data flows between the main functions of the Data Factory. It anticipates some details, which are explained in more detail in the following sections.

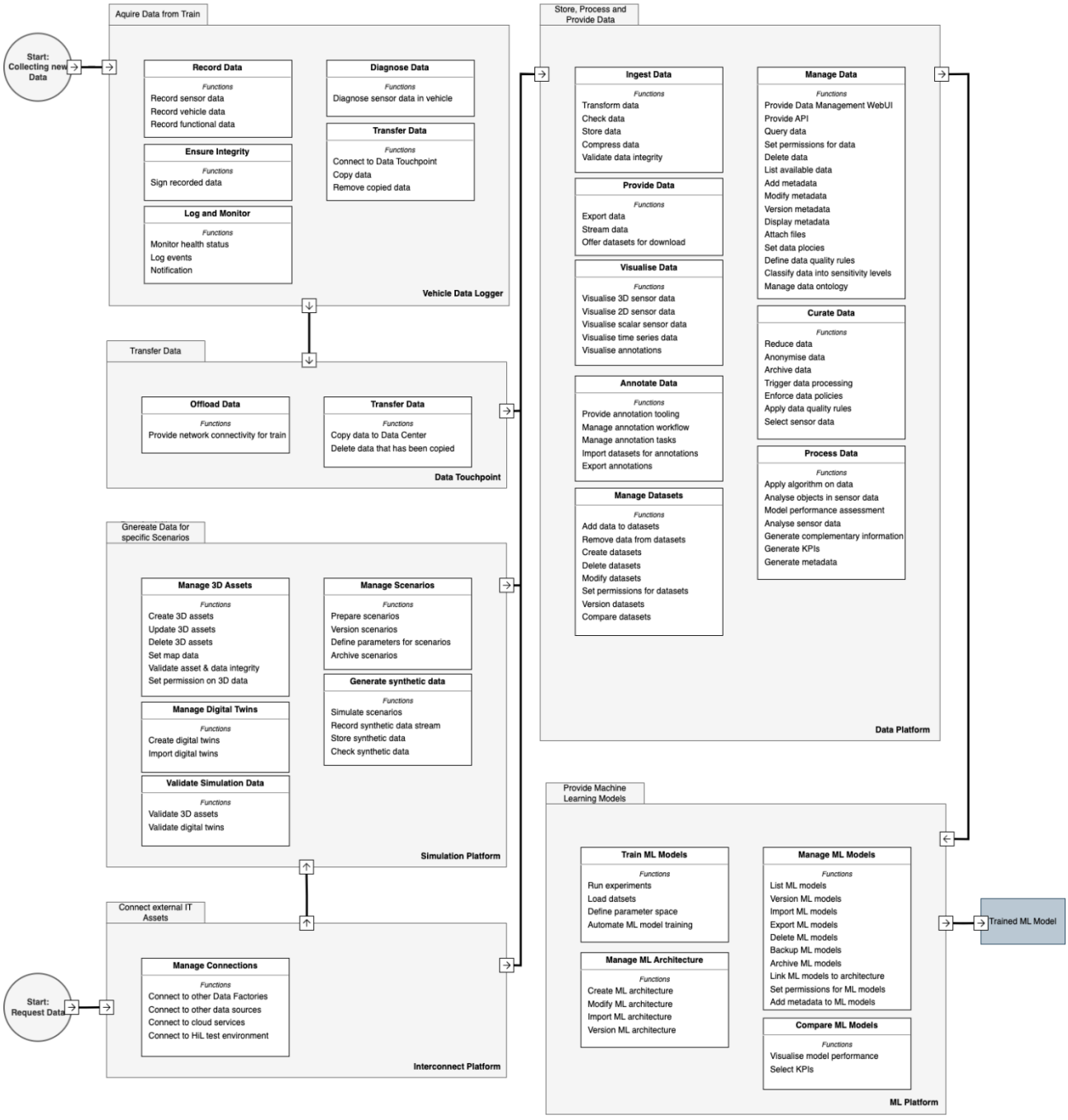


Figure 29: Data Flow Diagram

2.7.1.1 Data Flow Explanation

- Initiation and Data Collection:
  - Start Collecting New Data: The cycle kicks off with the collection of new data from the train (vehicle), which is essential for the subsequent processes.
- Vehicle Data Management:
  - Vehicle Data Logger: Acquire Data from Train: The vehicle data logger's function is to acquire data from the train's onboard systems. This data encompasses various operational metrics and sensor outputs.

- Data Transfer and Simulation:
    - Data Touchpoint: Transfer Data: At this point, the acquired data is offloaded from the train (vehicle) and transferred into the Data Centre. The data touchpoint acts as a gateway, facilitating the data's movement from the train to the system's internal processes.
    - Simulation Platform: Generate Data for Specific Scenarios: Another way to feed data into the data factory is via the simulation platform. Using this platform it is possible to generate synthetic data by executing non-regular situations.
    - Interconnect Platform: Request Data: Using the Interconnect platform, it is possible to bring additional data into the Data Factory, but this platform also enables the exchange (import/export) of data by connecting other IT systems and data sources.
- Data Consolidation and Processing:
    - Data Platform: Store, Process, and Provide Data: As a central hub, the data platform receives input from the actual train data, the simulated data and the externally connected data sources. The data is persisted here. The data is also prepared in such a way that the required data quality and information is available for the ML models to be trained.
- Machine Learning Integration:
    - ML Platform: Provide Machine Learning Models: The ML platform comes into play by using the prepared data to make it available for machine learning. Functions are also provided within the platform to re-train and improve existing ML models. This phase is crucial for determining the quality and performance of the ML model.
- Output**:**
    - Trained ML Model: As a result of the ML platform, a specific algorithm or a trained ML model leaves this platform, which has been trained and developed for a specific application.

2.7.1.2 Data Flow Summary

- The entire process starts with data acquisition from the train, ensuring a fresh and relevant dataset.
- The Vehicle Data management is conducted through a vehicle data logger that channels the data to a touchpoint.
- The simulation platform enhances the dataset with synthetic data, while the interconnect platform introduces external IT assets into the mix.
- The combined streams of real, synthetic, and external data will be flow at the data platform.
- The data is comprehensively prepared and processed on the data platform in order to make it available for machine learning.
- The ML platform is where ML Models and algorithms are provided and trained and re-trained and making use of the processed data.
- The output of the Data Platform is a trained machine learning model that can be tested and applied to real-world scenarios.

2.7.1.3 Key Process Interactions

- Real and synthetic data integration supports robust model training.
- The integration of external IT assets ensures a more comprehensive range of data that can be used for machine learning.
- The central processing node, the data platform, is crucial for data management and data storage, as well as for data integrity and data quality.

2.7.2 Data Platform

Training machine learning models demands a substantial amount of data. To select the most suitable data for a given application, efficient data management is indispensable. The Data Platform delivers the essential functionalities for this purpose within the Data Factory. These encompass ingesting, processing, managing, curating, visualising, and annotating data as well as providing data to other subsystems.

Upon the arrival of new data at the Data Factory, the data ingestion process verifies file integrity and content, transforming the data into suitable formats for storage. Subsequently, data processing pipelines are triggered to extract and generate metadata. This metadata enriches the unstructured sensor data, making it more manageable and searchable.

Users can explore stored data using a data catalogue and employ queries to find relevant samples for their applications. These samples can then be organised into datasets for machine learning and software validation. The Data Platform also governs data policies and rules that are enforced by the data curation engine, such as access restrictions, data anonymisation, and data deletion.

Furthermore, it facilitates both manual and automated labelling of sensor data and provides visualizations alongside annotations. Serving as the central repository for data, the Data Platform ensures seamless access to its data for other subsystems.

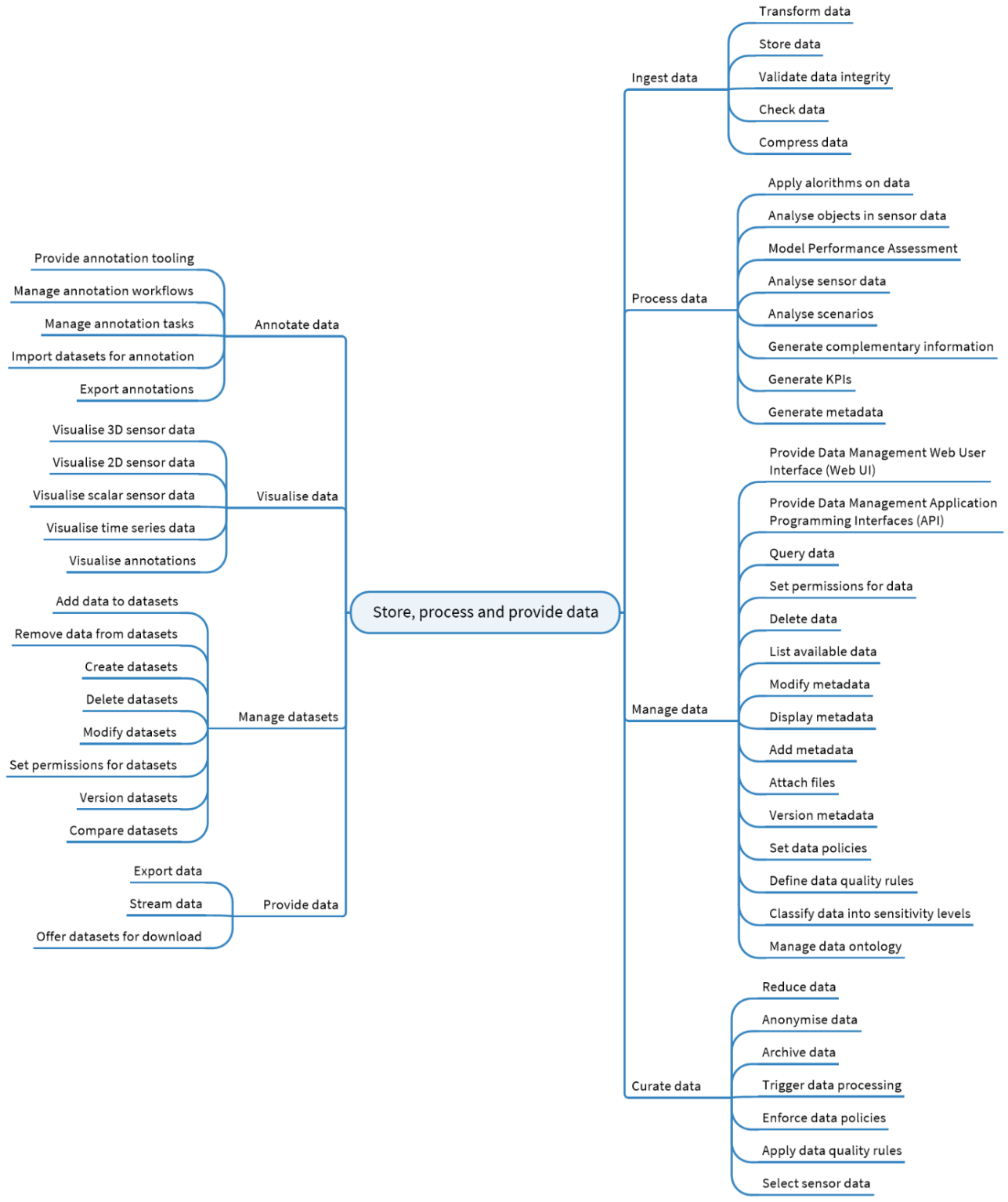


Figure 30: Functional tree Data Platform

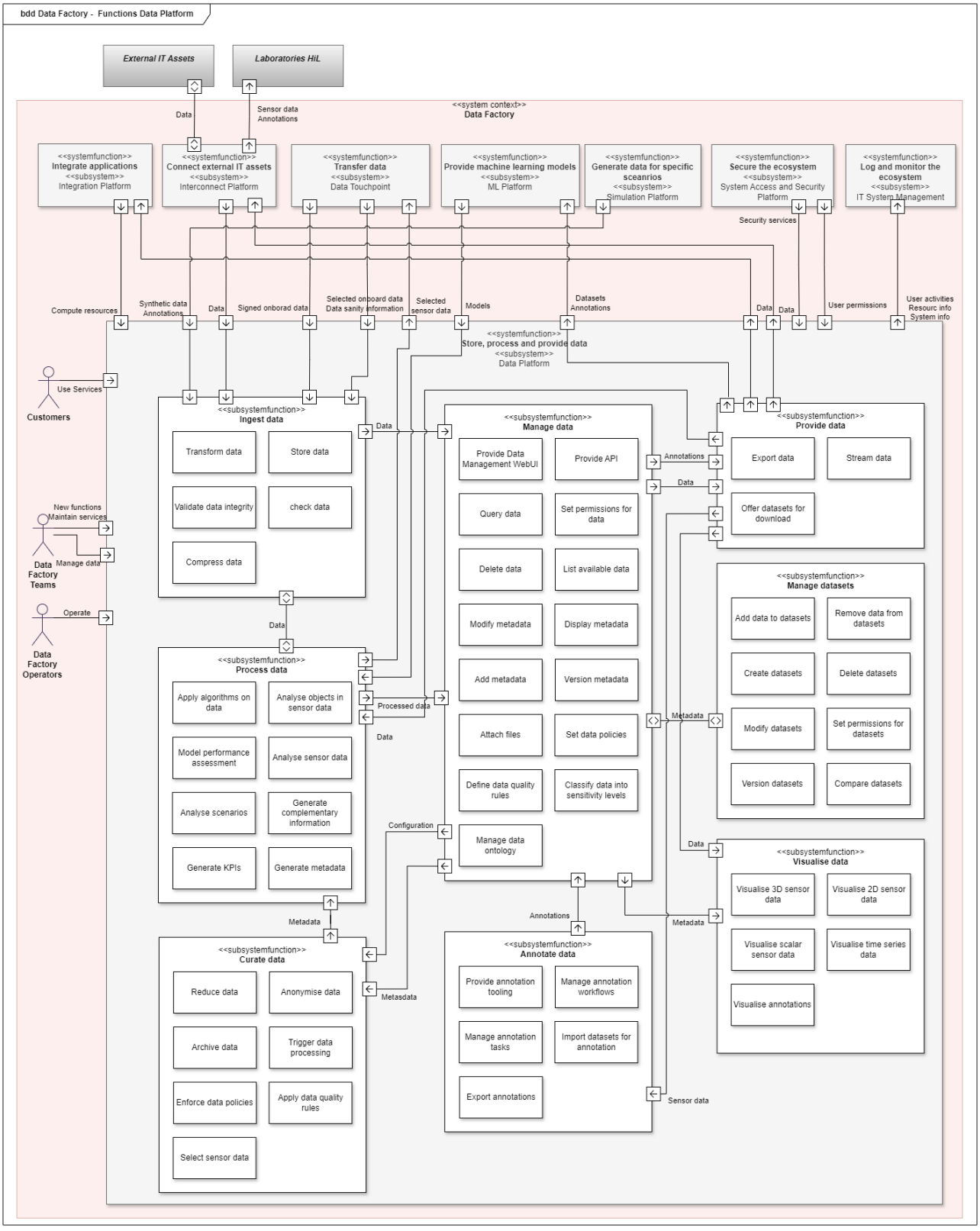


Figure 31: Context diagram Data Platform

The Data Platform is the subsystem within the Data Factory that is responsible for data storage, management, and processing. This includes the ingestion of incoming data, the curation and processing of the stored data. It provides access to the data for other subsystems within the Data Factory.

- Function: Ingest Data

Onboard data streams from trains and data from other data sources are transformed and stored.

  - Input:
    - synthetic data, annotations from “Subsystem: Generate data for specific scenarios”
    - data from “Subsystem: Connect external IT assets”
    - signed onboard data, selected onboard data, data sanity information from “Subsystem: Transfer data”
- Output:
  - data to “Function: Manage data”
- Requirements:
  - The Data Pipelines shall transform data.
  - The Data Pipelines shall store data.
  - The Data Pipelines shall validate data integrity.
  - The Data Pipelines shall check data.
  - The Data Pipelines shall compress data.
- Function: Manage Data

Handling user and system requests for data administration, metadata handling, and policy enforcement to get well-managed datasets with accurate metadata, controlled access, and adherence to data policies and quality rules.

  - Input:
    - data from “Function: Ingest data”
    - metadata from “Function: Process data”
    - metadata from “Function: Manage datasets”
    - annotations from “Function: Annotations”
  - Output:
    - data, annotations to “Function: Provide data”
    - metadata to “Function: Manage datasets”
    - data, metadata to “Function: Visualise data”
    - metadata, config to “Function: Curate data”
  - Requirements:
    - The Data Platform shall manage data.
    - The Data Management shall provide its functionalities via Web User Interface.
    - The Data Management shall provide its functionalities via Application Programming Interfaces (API).

        - The Data Management shall enable the querying of data.
        - The Data Management shall set permissions for data.
        - The Data Management shall enable the deletion of data.
        - The Data Management shall list available data.
        - The Data Management shall enable the modification of metadata.
        - The Data Management shall display metadata.
        - The Data Management shall add metadata.
        - The Data Management shall enable the file attachment.
        - The Data Management shall version metadata.
        - The Data Management shall enable the setting of data policies.
        - The Data Management shall enable the definition of data quality rules.
        - The Data Management shall enable the setting of classifications into predefined sensitivity levels.
        - The Data Management shall manage the data ontology.

- Function: Process Data

Ingested data is analysed with applied algorithms, object detections, classifications, and performance assessments.

    - Input:
        - data from “Function: Ingest data”
        - models from “Subsystem: Provide machine learning models”
        - metadata from “Function: Manage data”
        - data from “Function: provide data”
        - metadata from “Function: Curate data”
    - Output:
        - selected sensor data to “Subsystem: Transfer data”
        - metadata to “Function: Manage data”
        - data to “Function: Ingest data”
    - Requirements:
        - The Data Platform shall process data.
        - The Data Pipelines shall apply algorithms on data.
        - The Data Pipelines shall analyse the objects in the sensor data.
        - The Data Pipelines shall detect objects in camera data.
        - The Data Pipelines shall classify objects in camera data.
        - The Data Pipelines shall provide its object analysis results for both RGB and IR camera input data.

- The Data Pipelines shall detect objects in lidar data.
- The Data Pipelines shall classify objects in lidar data.
- The Data Pipelines shall detect objects in RADAR data.
- The Data Pipelines shall classify objects in RADAR data.
- The Data Pipelines shall track objects in sensor data.
- The Data Pipelines shall determine model performance values to assess sensor data.
- The Data Pipelines shall determine uncertainty scores for object detection results in camera data.
- The Data Pipelines shall determine uncertainty scores for object classification results in camera data.
- The Data Pipelines shall determine uncertainty scores for object detection results in lidar data.
- The Data Pipelines shall determine uncertainty scores for object classification results in lidar data.
- The Data Pipelines shall determine uncertainty scores for object detection results in RADAR data.
- The Data Pipelines shall determine uncertainty scores for object classification results in RADAR data.
- The Data Pipelines shall determine anomalous situations in camera data.
- The Data Pipelines shall provide an anomaly score for sensor data.
- The Data Pipelines shall determine anomalous situations in lidar data.
- The Data Pipelines shall determine anomalous situations in RADAR data.
- The Data Pipelines shall determine frequency and type of anomalous events in sensor data.
- The Data Pipelines shall determine statistics of scenario distribution for sensor data.
- The Data Pipelines shall generate complementary information for existing data.
- The Data Pipelines shall support correction of erroneous measurements using complementary ex-post-analyses.
- The Data pipelines shall generate complementary ex-post-localization information from sensor data to correct/complement erroneous localization data in the original data.
- The Data Pipelines shall generate 3D maps from camera data.
- The Data Pipelines shall generate 3D maps from lidar data.
- The Data Pipelines shall generate 2D maps from RADAR data.
- The Data Pipelines shall analyse scenario coverage for sensor data.

        - The Data Pipelines shall generate KPIs on data.
        - The Data Pipelines shall generate metadata.

- Function: Curate Data

Data is refined, anonymized, and archived to adhere to data policies and quality rules

    - Input:
        - metadata, configuration from “Function: Manage data”
    - Output:
        - metadata to “Function: Process data”
    - Requirements:
        - The Data Platform shall curate data.
        - The Data Management shall reduce data.
        - The Data Management shall mask and anonymise data.
        - The Data Management shall archive data.
        - The Data Management shall trigger data processing.
        - The Data Management shall enforce data policies.
        - The Data Management shall apply the defined data quality rules.
        - The Data Pipelines shall assign a priority score for storing new data samples.
        - The Data Pipelines shall decide on storing new sensor data upon the priority score.
        - The Data Pipelines shall assign a priority score for the annotation priority to each data.
        - The Data Pipelines shall decide on annotating new sensor data upon the priority score.

- Function: Provide Data

Data is made available through various channels as data export, streaming or download.

    - Input:
        - data, annotations from “Function: Manage data”
        - datasets from “Function: Manage datasets”
    - Output:
        - data to “Subsystem: Connect external IT assets”
        - datasets, annotations to “Subsystem: Provide machine learning models”
        - sensor data to “Function: Annotate data”
        - data to “Function: Visualize data”
        - data to “Function: Process data”
        - data to “Subsystem: Integrate applications”

- Requirements:
  - The Data Platform shall provide data.
  - The Data Management shall enable to export data.
  - The Data Management shall stream data.
  - The Data Management shall offer datasets for download.

- Function: Manage Datasets

Datasets are updated, versioned, and compared for consistency and integrity.

- Input:
  - metadata from “Function: Manage data”
- Output:
  - datasets to “Function: Provide data”
- Requirements:
  - The Data Platform shall manage datasets.
  - The Data Management shall enable to add data to datasets.
  - The Data Management shall enable to remove data from datasets.
  - The Data Management shall enable to create datasets.
  - The Data Management shall enable to delete datasets.
  - The Data Management shall enable to modify datasets.
  - The Data Management shall enable to set permissions for datasets.
  - The Data Management shall enable to version datasets.
  - The Data Management shall enable to compare datasets.

- Function: Visualise Data

Sensor data is visualized, enhancing understanding and insights.

- Input:
  - data from “Function: Provide data”
  - metadata from “Function: Manage data”
- Output:
  - none
- Requirements:
  - The Data Platform shall visualise data.
- The Data Management shall visualise 3D sensor data.
- The Data Management shall visualise 2D sensor data.
- The Data Management shall visualise scalar sensor data.
- The Data Management shall visualise time series data.

- The Data Management shall visualise annotations.

- Function: Annotate Data

Raw and processed data is enriched with additional context and information via annotations.

- Input:
  - sensor data from “Function: Provide data”
- Output:
  - annotations to “Function: Manage data”
- Requirements:
  - The Data Platforms shall annotate data.
  - The Annotation Platform shall provide annotation tooling.
  - The Annotation Platform shall manage annotation workflows.
  - The Annotation Platform shall manage annotation tasks.
  - The Annotation Platform shall import datasets for annotation.
  - The Annotation Platform shall export annotations.

- Roles Involved:
  - Data Factory Operators:
    - Engage with the Data Platform to utilize simulation data, contributing inputs to enhance the generation of datasets or scenarios. They play a pivotal role in operationalizing the data and aligning outputs with operational requirements.
  - Customers**:**
    - As end-users, customers interact with the simulation outputs for validation and practical application. Their feedback is critical in assessing the quality and utility of the scenarios, which in turn informs the requirements for the Simulation Platform.
  - Data Factory Teams:
    - Collaborate across subsystems by contributing domain expertise and analytics, ensuring that the scenarios and simulations developed are relevant and aligned with the overarching objectives and use cases of the Data Factory. They help bridge the gap between theoretical data models and practical, operational needs.

- Interactions:
  - Inputs:
    - From Integration Platform:

      Compute Resources: Computational capabilities provided to support data operations within the Data Platform.
    - From Interconnect Platform via External IT Assets:

Data: Data received from external IT systems for integration and processing.

- From Data Touchpoint:

  Signed Onboard Data: Authenticated data received from the Vehicle Data Logger.

  Selected Onboard Data and Data Sanity Information: Curated data along with information on its logical consistency.

- From ML Platform:

  Models: Machine learning models received for data analysis and enhancement.

- From Simulation Platform:

  Synthetic Data Annotations: Annotated synthetic data used for testing and scenario analysis.

- From System Access and Security Platform:

  Security Services: Security-related services to ensure the protection and integrity of the data.

  User Permissions: Access rights and permissions granted to users for data interaction.

- Outputs:

  - To Integration Platform:

    Data: Processed or unprocessed data sent for further integration.

  - To Interconnect Platform:

    Data: Data made available to external IT systems or for further system-wide use.

    Sensor Data Annotations: Annotation details for sensor data to be utilized by External Laboratories HIL.

  - To Data Touchpoint:

    Selected Sensor Data: Data specifically filtered and sent back for verification or further use.

  - To ML Platform:

    Dataset Annotations: Annotations of the datasets used by machine learning models for improved learning and insights.

  - To IT System Management:

    User Activities, Resource Info, and System Info: Logs and information pertaining to the interaction with data resources and system performance.

- Roles Involved:

  - Customers:

- Use Services: Engage with services provided by the Data Platform for their needs.

- Data Factory Teams:
  - New Functions: Develop and introduce new functionalities to enhance data services.
  - Maintain Services: Ensure ongoing maintenance and optimization of services.
  - Managed Data: Oversee data management processes within the Data Platform.
- Data Factory Operators:
  - Operate: Handle the day-to-day operation and ensure the smooth running of the Data Platform's functions.

This structure ensures that the Data Platform operates efficiently as the central hub for data management, supporting the Data Factory's needs for data integration, processing, and security. The clear delineation of inputs and outputs, along with the defined roles, underscores the Data Platform's importance in maintaining a robust data ecosystem.

2.7.3 Data Touchpoint

The Data Touchpoint subsystem is a pivotal component within the Data Factory, primarily focused on the transfer of data. This subsystem facilitates the offloading of data from trains, which encompasses providing necessary network connectivity at predefined locations, ensuring that data is consistently and efficiently transferred to the designated data centers. The Data Touchpoint serves as a crucial interface, managing the flow of information and bridging the gap between the data's origin on the trains and its subsequent storage and processing. Through its integrated sub-subsystems, such as the Automated Offload and DC Transmitter, the Data Touchpoint guarantees that data not only reaches its destination securely but also that any data redundancy is managed effectively. The Automated Offload is tasked with establishing robust network connections, while the DC Transmitter handles the duplication of data to the Data Center and the meticulous removal of data from the Data Touchpoint post-transfer, ensuring the integrity and optimization of data storage within the Data Factory's ecosystem.

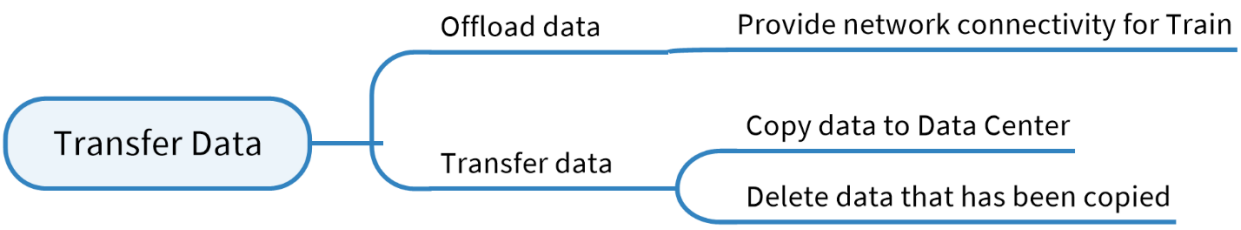


Figure 32: Functional Tree Data Touchpoint

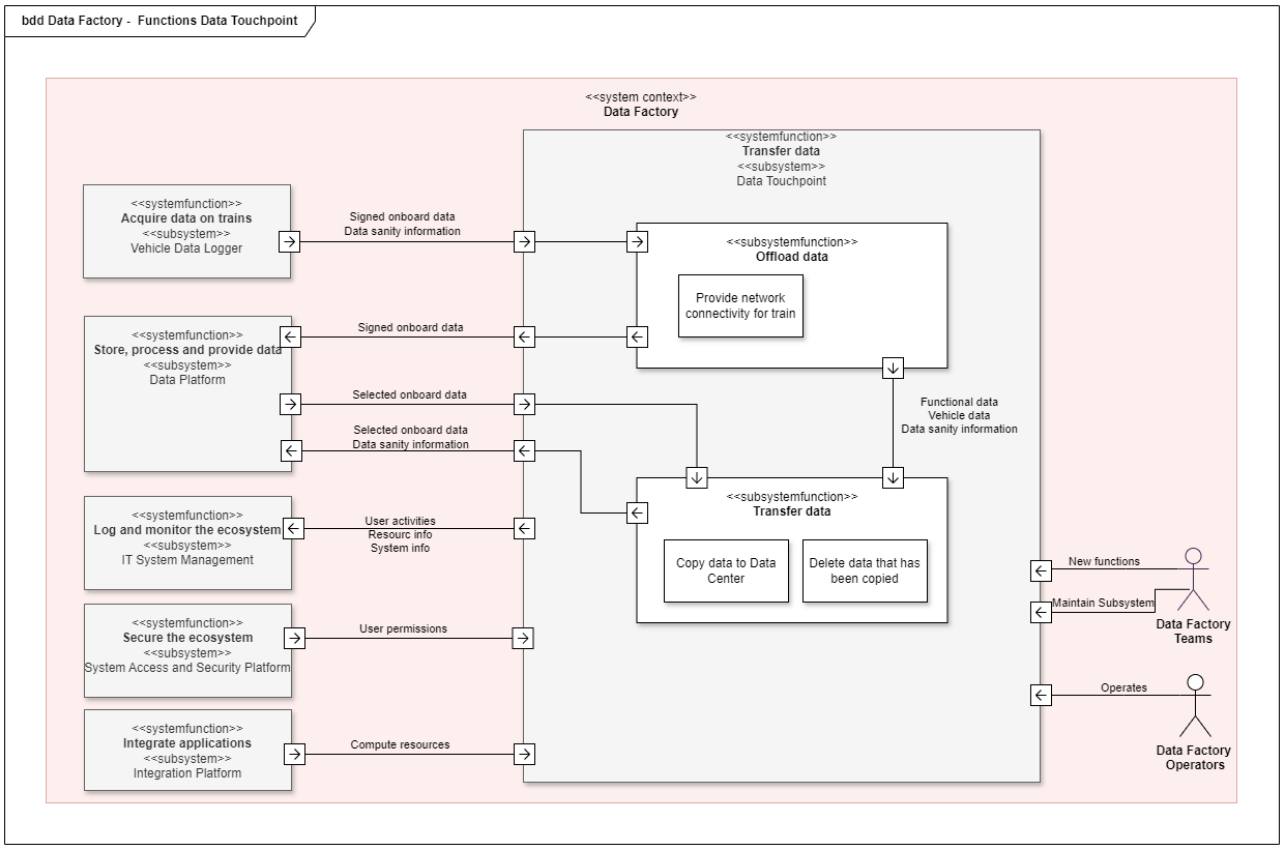


Figure 33: Context diagram Data Touchpoint

The Data Touchpoint Subsystem facilitates data exchange within the Data Factory, acting as an intermediary between data sources like trains and destinations like the Data Centre. Here's a breakdown of its core functionalities with corresponding inputs, outputs, and requirements:

- Function: Offload Data
  - Input:
    - Signed onboard data, Data sanity information
  - Output:
    - Data ready for transfer to the Data Platform
  - Requirements:
    - The Data Touchpoint shall offload data from Train.
- Function: Transfer Data
  - Input:
    - Offloaded data from the Offload Data function
  - Output:
    - Data is transferred to the Data Platform
    - Data that has been copied is deleted from the Data Touchpoint
  - Requirements:
    - The Data Touchpoint shall transfer data to the Data Center.

- Interactions:
    - Inputs:
        - From Vehicle Data Logger:

            Signed Onboard Data: This data is collected from the Vehicle Data Logger, representing onboard data that has been signed to confirm its authenticity.

            Data Sanity Information: Includes diagnostic information on the validity and logical consistency of the onboard data.

        - From Data Platform:
- Selected Onboard Data: Data selected by the Data Platform for specific purposes such as further analysis or model training.
        - From IT System Management:

            User Activities: Information on user interactions with the Data Touchpoint subsystem for monitoring and audit purposes.

            Resource Info: Data regarding the usage of resources by the Data Touchpoint subsystem, such as computational and storage resources.

            System Info: General information about the system status of the Data Touchpoint subsystem.

        - From System Access and Security Platform:
- User Permissions: Permissions and access rights provided by the security platform for users interacting with the Data Touchpoint subsystem.
        - From Integration Platform:

            Compute Resources: Computing resources allocated by the Integration Platform to the Data Touchpoint subsystem for data processing tasks.

    - Outputs:
        - To Data Platform:

            Signed Onboard Data: After verification and signing for authenticity, this data is sent back to the Data Platform.

            Selected Onboard Data: Data selected by the Data Platform, which is then provided to the Data Touchpoint for specific purposes.

            Data Sanity Information: Diagnostic information from the Data Touchpoint that assesses the validity and consistency of the data, shared with the Data Platform for further analysis.

        - To IT System Management:

            User Activities: Logging of user activities within the Data Touchpoint subsystem is shared with IT System Management for monitoring and maintenance.

            Resource Info: Information on resource utilization by the Data Touchpoint subsystem is shared for managing the IT infrastructure.

System Info: Updates on the Data Touchpoint system status, including any significant events or anomalies, are reported for IT system oversight.

- Roles Involved:
  - Data Factory Operators:
    - Manage the operation of the Data Touchpoint, ensuring the data offloading, copying to the Data Centre, and deletion processes are performed correctly and securely.
  - Data Factory Teams:
    - Utilize the data provided by the Data Touchpoint for developing new functionalities and ensuring alignment with operational requirements.

In this subsystem, the Data Touchpoint acts as a pivotal node for data transmission within the Data Factory, interfacing with the Vehicle Data Logger to receive offloaded data, ensuring its secure transfer to the Data Platform, and managing the deletion of data post-transfer to maintain data integrity and prevent redundancy within the ecosystem.

2.7.4 Integration Platform

The Integration Platform within the Data Factory is engineered as a sophisticated nexus for seamless application integration, provision of interactive computation, and robust automation services. It represents a key architectural component that enables the Data Factory to interlink its internal subsystems with external IT assets, thereby facilitating a streamlined and connected operational environment.

At its core, the Integration Platform's capabilities are two-fold. Firstly, the Interactive Computation Environment is designed to establish, manage, and decommission interactive computation environments. This allows for dynamic scalability and adaptability in processing needs. The environment enables connections to be forged to these computation spaces, allowing users to engage with the system interactively. It also ensures that data can be provided, processed, and managed effectively, reflecting a commitment to fostering a collaborative and flexible interactive compute ecosystem.

Secondly, the provision of automation by the Integration Platform underlines its critical role in enhancing efficiency and responsiveness within the Data Factory. The Integration Manager is a pivotal element in this process, tasked with initiating, monitoring, and updating processes that are fundamental to the Data Factory's operations. It informs stakeholders with timely status updates, ensuring transparency and informed decision-making. Furthermore, the Integration Manager orchestrates the scheduling of tasks and jobs, enhancing workflow and productivity. Its capability to provide containerization showcases the Integration Platform's advanced approach to deployment, enabling portable and consistent software environments that are essential for modern DevOps practices.

The Integration Platform acts as a central orchestrator, synchronizing the Data Factory's complex matrix of subsystems and external interfaces. It supports interactive computational tasks, facilitates real-time connectivity, automates process flows, and oversees task scheduling and containerization, thereby ensuring that the Data Factory operates as a cohesive, efficient, and agile entity.

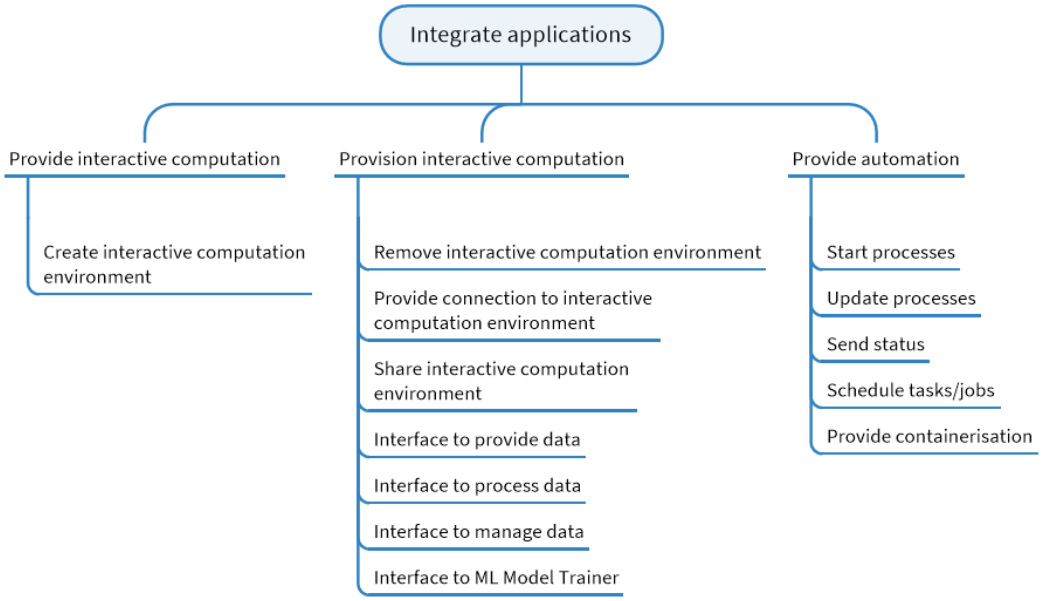


Figure 34: Functional tree Integration Platform

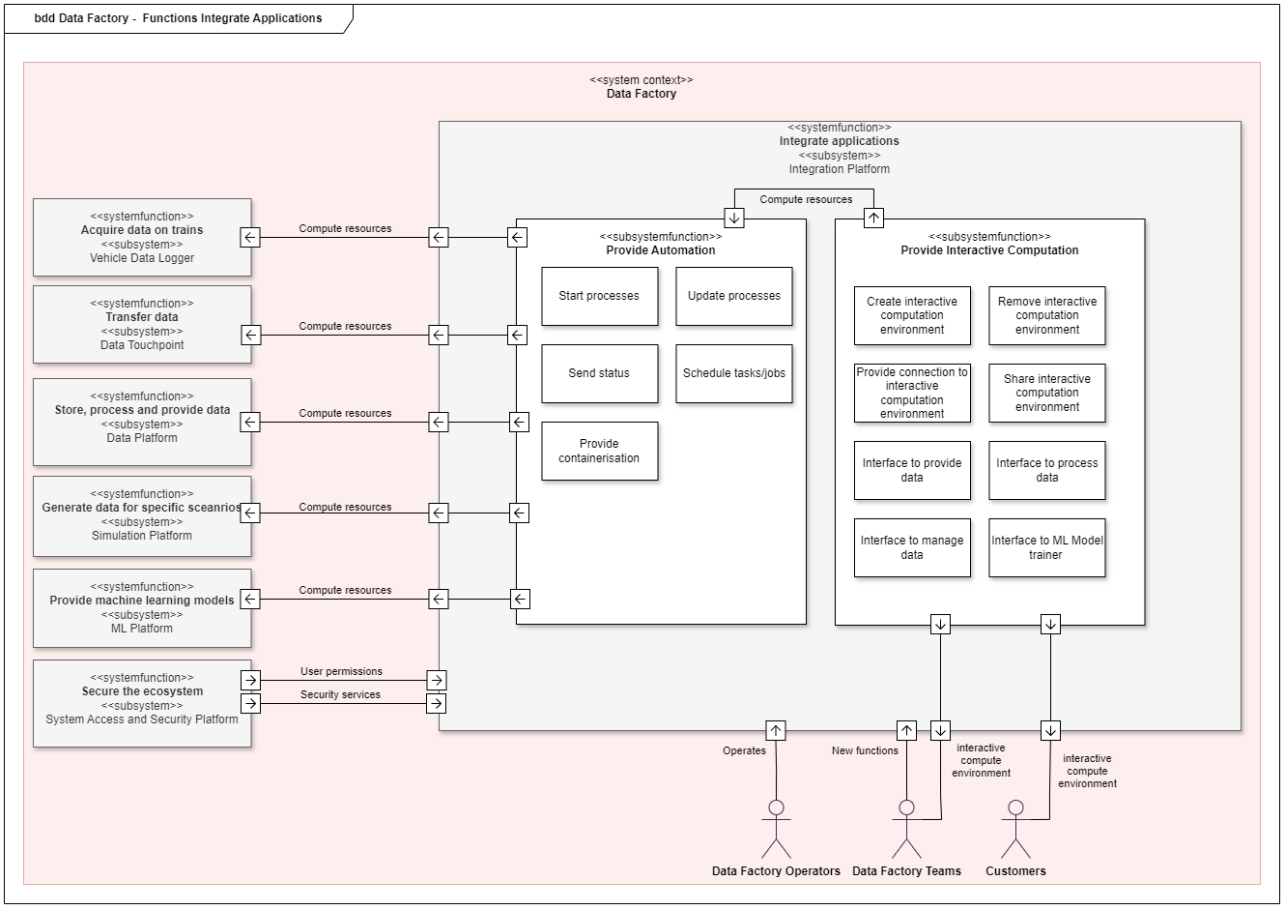


Figure 35: Context diagram Integration Platform

The Integration Platform is a pivotal subsystem within the Data Factory, facilitating seamless interoperability among various subsystems and external entities. It is central to enabling dynamic and efficient interactive computation and automation within the broader ecosystem.

- Function: Provide Interactive Computation

- Input:
  - Requests for the creation, removal, and connection of interactive computation environments.
- Output:
  - Created, updated, and connected interactive computation environments for user engagement.
- Requirements:
  - The Integration Platform shall create and remove interactive computation environments as needed.
  - The Platform shall manage the connections to these environments and facilitate their sharing among users.
  - It will provide interfaces to Data Factory's "provide data" and "process data" functions, enhancing usability.

- Function: Provide Automation
  - Input:
    - Operational commands for starting, updating, and informing processes, scheduling tasks, and enabling containerization.
  - Output:
    - Executed and managed operational processes, updates to stakeholders, and containerized environments.
  - Requirements:
    - The Integration Manager within the Platform shall initiate, and update processes as required.
    - The Manager will inform stakeholders of status updates and schedule necessary tasks and jobs.
    - It shall support containerization to enhance the scalability and portability of applications.
- Interactions:
  - Outputs:
    - To Vehicle Data Logger:

      Compute Resources: Provides computational power and services.
    - To Data Touchpoint:

      Compute Resources: Supplies the necessary computational capabilities for data management.
    - To Data Platform:

      Compute Resources: Supports data storage, processing, and provisioning with computational resources.
    - To Simulation Platform:

Compute Resources: Facilitates the generation of data for specific scenarios with necessary computing power.

- To ML Platform:

  Compute Resources: Provides resources to aid in the creation and training of machine learning models.

- Inputs:
  - From System Access & Security Platform:

    User Permissions: Manages the access rights of users to the platform.

    Security Services: Implements security measures for safeguarding platform operations.

- Roles Involved:
  - Data Factory Operators:
    - Operates: Responsible for the daily operations of the Integration Platform.
  - Data Factory Teams:
    - New Functions: Integrates new capabilities and enhancements into the Integration Platform.
    - Interactive Compute Environment: Engages with the platform's services to perform interactive computational tasks.
  - Customers:
    - Interactive Compute Environment: Engages with the platform's services to perform interactive computational tasks.

The Integration Platform serves as the Data Factory's central hub for application integration and management of compute resources. It ensures smooth interoperability between the factory's subsystems and provides a scalable environment to support the varying computational demands of the Data Factory ecosystem. The platform's design allows for flexibility and scalability, accommodating new functions introduced by the Data Factory Teams and ensuring that the Data Factory Operators can maintain efficient operations. Through its robust interaction with the System Access & Security Platform, the Integration Platform maintains high standards of security and access control, upholding the Data Factory's commitment to secure and resilient operations.

2.7.5 Interconnect Platform

The Interconnect Platform within the Data Factory serves as a nexus for integrating diverse external IT assets. It is a pivotal subsystem focused on managing connections to ensure seamless communication and interoperability between various data sources, systems, and environments. Its role is critical in establishing robust data pathways that connect the Data Factory to other data factories, enhancing collaborative efforts and data exchange.

Additionally, the Interconnect Platform oversees the linkage to external data sources, enabling the aggregation and synthesis of data from disparate origins. This extends the Data Factory’s capabilities, fostering a rich data ecosystem that supports advanced analytics and machine learning model development.

Connection to cloud services is also within the purview of the Interconnect Platform, facilitating access to scalable computing resources, storage, and advanced cloud-based analytics tools. This connectivity ensures that the Data Factory can leverage cloud efficiencies and innovations for data processing and management tasks.

Moreover, the Interconnect Platform's ability to provide connection to Hardware-in-the-Loop (HiL) test environments underscores its importance in validating simulations and models in real-time, offering an interface for testing against a hybrid of real and virtual components.

Through the Connect Manager, the platform guarantees robust connections to other data factories, data sources, cloud services, and HiL test environments, reflecting the Data Factory's commitment to interoperability and its role as a central hub in a larger network of data-driven enterprises.

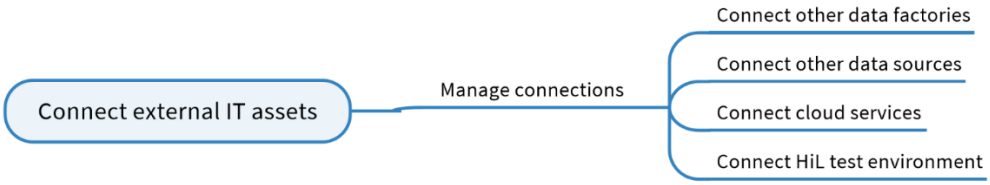


Figure 36: Functional tree Interconnect Platform

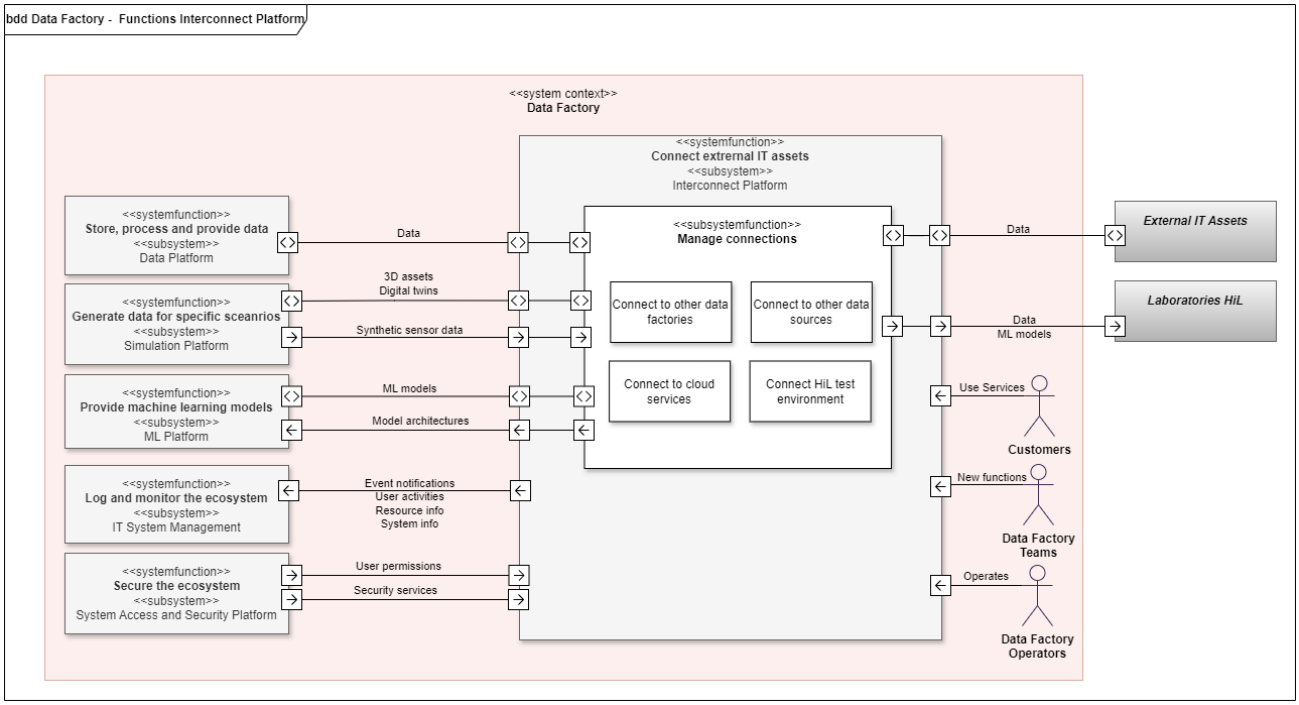


Figure 37: Context diagram Interconnect Platform

The Interconnect Platform is a vital subsystem within the Data Factory, serving as a conduit for seamless data exchange between the Data Factory and various external and internal data sources, including external IT assets and HiL (Hardware in the Loop) laboratories.

- Function: Manage Connections
    - Input:
        - Requests for data exchange from various sources such as other data factories, cloud services, external IT assets, and HIL environments.

- Output:
  - Established data connections that enable the flow of sensor data, synthetic sensor data, annotations, 3D assets, and digital twins.
- Requirements:
  - The Connect Manager shall provide connection other data factories
  - The Connect Manager shall provide connection other data sources
  - The Connect Manager shall provide connection cloud services
  - The Connect Manager shall provide connection to the HiL test environment

- Interactions:
  - Bidirectional Connections:
    - To/From Data Platform:

      Data: Exchange of data between platforms for integrated processing.
    - To/From Simulation Platform:

      3D Assets & Digital Twins: Sharing of assets and digital representations for simulation purposes.
    - To/From ML Platform:

      ML Models: Exchange of machine learning models to enhance learning and predictive capabilities.
    - To/From External IT Assets:

      Data: Sharing of data for collaboration and integration.
  - Inputs:
    - From Simulation Platform:

      Synthetic Sensor Data: Data generated during simulations for analysis and system refinement.
    - From System Access and Security Platform:

      User Permissions: Authorization credentials for user access control.

      Security Services: Protections for data and system integrity.
  - Outputs:
    - To ML Platform:

      Model Architectures: Structure and frameworks of ML models provided for development and refinement.
    - To IT System Management:

      Event Notifications, Resource Info, System Info: Notifications and logs pertaining to system events and resources.
    - To Laboratories HIL:

Data, ML Models: Outputs for testing and validation in laboratory environments.

- Roles Involved:
  - Customers:
    - Use Services: Engage with services offered by the Interconnect Platform for various requirements.
  - Data Factory Teams:
    - New Functions: Develop and integrate new functionalities to advance the Data Factory's capabilities.
  - Data Factory Operators:
    - Operate: Manage the daily operations, ensuring the Interconnect Platform is functioning efficiently and effectively.

This structure solidifies the Interconnect Platform's role as the connective tissue of the Data Factory, facilitating seamless integration across various subsystems and external assets, bolstering the ecosystem's overall connectivity and functionality.

2.7.6 IT System Management

IT System Management is the comprehensive subsystem within the Data Factory responsible for overseeing and tracking all interactions with IT infrastructure. It logs user logins and logouts, service access, application start and end times, resource access, and data deletion. These logs provide a detailed record of user activities, ensuring transparency and accountability in the use of IT resources.

Beyond user interactions, IT System Management also offers alerting capabilities, managing alerts and thresholds and sending notifications. This enables proactive responses to system events, maintaining system integrity.

The subsystem is equipped to display information vital for operational health, such as resource status, application and hardware events, resource usage, and any alerts. This visual representation of the system’s status is key for quick diagnostics and decision-making.

On a resource level, IT System Management keeps track of all resource events, including hardware status and the usage of CPUs, GPUs, memory, networks, and disk space. It also logs file transfers and triggers, which is essential for understanding system performance and planning resource allocation.

Moreover, application activities are closely monitored, with the subsystem logging application status, which helps in maintaining application performance and uptime.

By centralizing monitoring and combining the monitoring of application status and system thresholds, IT System Management provides a unified view of the ecosystem. This is crucial for identifying patterns, predicting potential issues, and coordinating maintenance and updates across the Data Factory’s IT landscape.

In summary, IT System Management's role is to log, monitor, and provide alerting capabilities to ensure the smooth operation of the Data Factory. Through its detailed logs, the subsystem ensures that any action or event within the IT ecosystem is accounted for, which is fundamental to securing and optimizing the Data Factory’s operations.

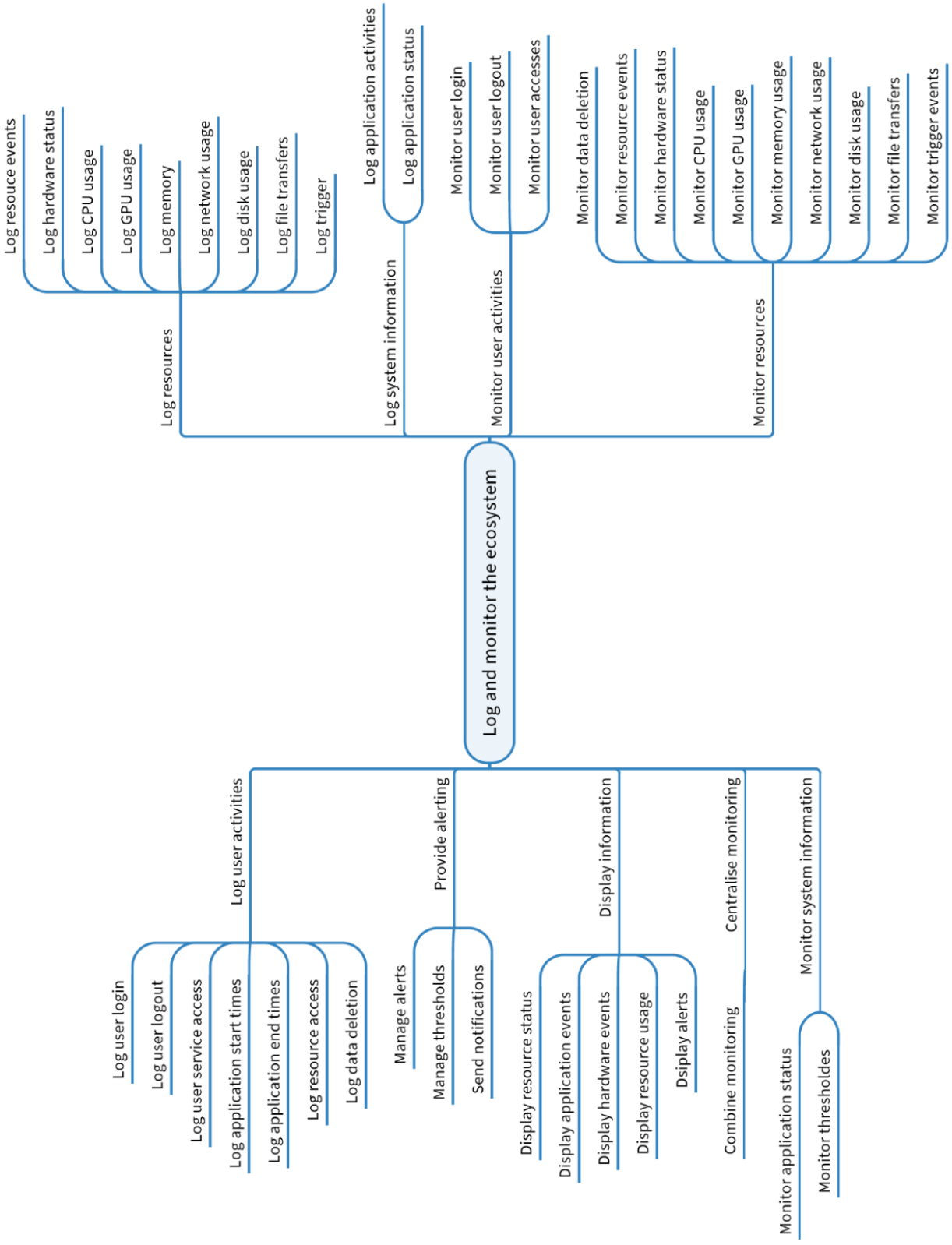


Figure 38: Functional tree IT System Management

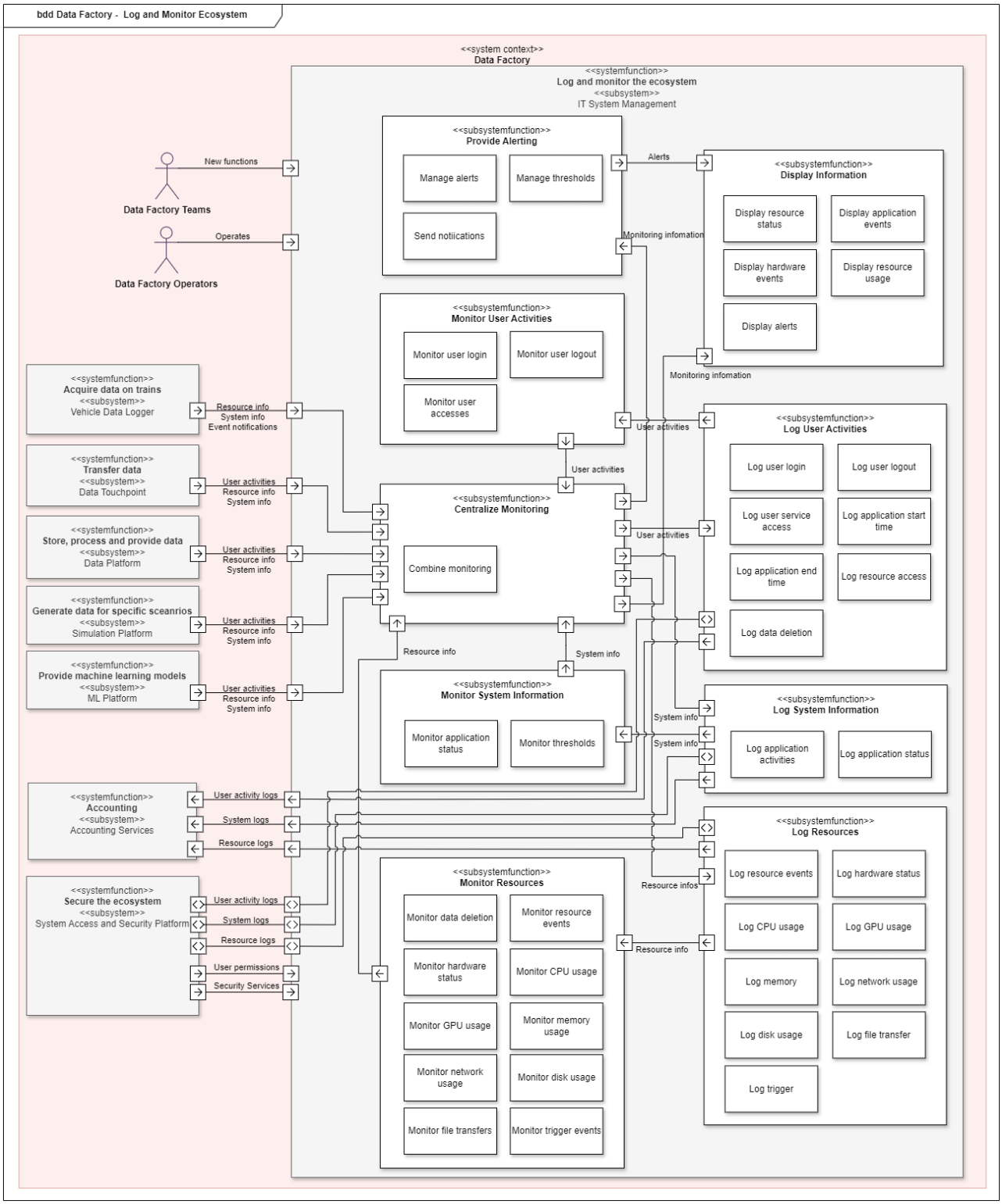


Figure 39: Context diagram IT System Management

The IT System Management subsystem is an integral component of the Data Factory, tasked with the overarching responsibility of monitoring and logging various user and system activities, resource utilization, and providing alerting mechanisms. This ensures the operational health and security of the Data Factory ecosystem.

- Function: Log User Activities
  - Input:
    - User interactions with the Data Factory.
  - Output:
    - Recorded logs of login, logout, service access, and data manipulation activities.
  - Requirements:
    - The Logging Manager shall log all user activities, including login, logout, and access to services.
    - The system shall record the times of user activities and the deletion of data by users.
- Function: Log Resources
  - Input:
    - Resource utilization data.
  - Output:
    - Logs detailing the usage and condition of hardware resources such as CPU, GPU, memory, network, and disk usage.
  - Requirements:
    - The Logging Manager shall capture detailed resource events and statuses.
- Function: Log System Information
  - Input:
    - Application activities and system status data.
  - Output:
    - System logs with detailed application activities and conditions.
  - Requirements:
    - The Logging Manager shall keep a comprehensive log of all system-level information.
- Function: Monitor User Activities
  - Input:
    - Real-time data of user interactions.
  - Output:
    - Alerts and reports on user activity.

- Requirements:
  - The Monitor Manager shall track user logins, logouts, and access to ensure compliance and security.

- Function: Monitor Resources
  - Input:
    - Continuous data feed of resource usage.
  - Output:
    - Real-time alerts and status updates on resource consumption.
  - Requirements:
    - The Monitor Manager shall continuously assess the status and usage of all critical resources.
- Function: Monitor System Information
  - Input:
    - Ongoing system performance data.
  - Output:
    - Monitoring insights for application status and system health.
  - Requirements:
    - The Monitor Manager shall ensure all applications are running optimally and maintain system health checks.
- Function: Centralize Monitoring
  - Input:
    - Collected monitoring data from various subsystems.
  - Output:
    - A unified dashboard of monitoring information.
  - Requirements:
    - The Monitor Manager shall integrate various streams of monitoring data into a central view.
- Function: Display Information
  - Input:
    - Logged and monitored data.
  - Output:
    - Visual displays for resources, events, and alerts.
  - Requirements:
    - The Display Manager shall present all critical information in an accessible format.

- Function: Provide Alerting
  - Input:
    - Monitoring data indicating thresholds are reached or anomalies detected.
  - Output:
    - Alert notifications and management of alert thresholds.
  - Requirements:
    - The Alert Manager shall generate, manage, and distribute alerts based on predefined criteria.
- Interactions:
  - Bidirectional:
    - From/to System Access and Security Platform:

      User Activity Logs, System Logs, Resource Logs: This is a two-way exchange of logs detailing user activities, system events, and resource utilization between IT System Management and the System Access and Security Platform for enhanced security management and oversight.
  - Inputs:
    - From Vehicle Data Logger:

      Resource Info, System Info, Event Notifications: Data concerning system resource utilization, system status, and significant system events received for management and logging.
    - From Data Touchpoint:

      User Activities, Resource Info, System Info: User interaction data, resource allocation metrics, and system status updates are received for monitoring and response.
    - From Data Platform:

      User Activities, Resource Info, System Info: Information on user engagement with the data platform, resource usage, and system health is collected.
    - From Simulation Platform:

      User Activities, Resource Info, System Info: Details on user activities related to simulations, resource, and system information are gathered for analysis and support.
    - From ML Platform:

      User Activities, Resource Info, System Info: Insights into user interactions with machine learning models and associated resource and system information are received.
    - From System Access and Security Platform:

User Permissions: Access rights and permissions data are received to manage user interactions with IT systems securely.

Security Services: Information regarding security protocols and services is received to reinforce the data protection measures.

- Outputs:
  - To Accounting Services:

    User Activity Logs, System Logs, Resource Logs: Logs that detail user actions, system events, and resource utilization are sent for accounting and auditing purposes.

- Roles Involved:
  - Data Factory Teams:
    - New Functions: Develop and integrate new functionalities to enhance the IT System Management subsystem's capabilities.
  - Data Factory Operators:
    - Operates: Oversee the daily operations of the IT System Management subsystem, ensuring smooth and secure processing of system-related activities.

IT System Management acts as a central hub within the Data Factory, handling a wealth of data from various subsystems. It ensures that the ecosystem operates efficiently by monitoring, analysing, and responding to a wide range of data inputs, from user activities to system health indicators. The subsystem's integration with the System Access and Security Platform is essential for maintaining a secure operational environment, managing user access, and implementing security measures. It is a foundational component of the Data Factory's structure, enabling oversight and governance across all interconnected systems.

2.7.7 ML Platform

The ML Platform is a dedicated subsystem within the Data Factory whose primary role is to support the lifecycle of machine learning models. It provides a unified environment for the development, management, and operationalisation of machine learning capabilities within the Data Factory.

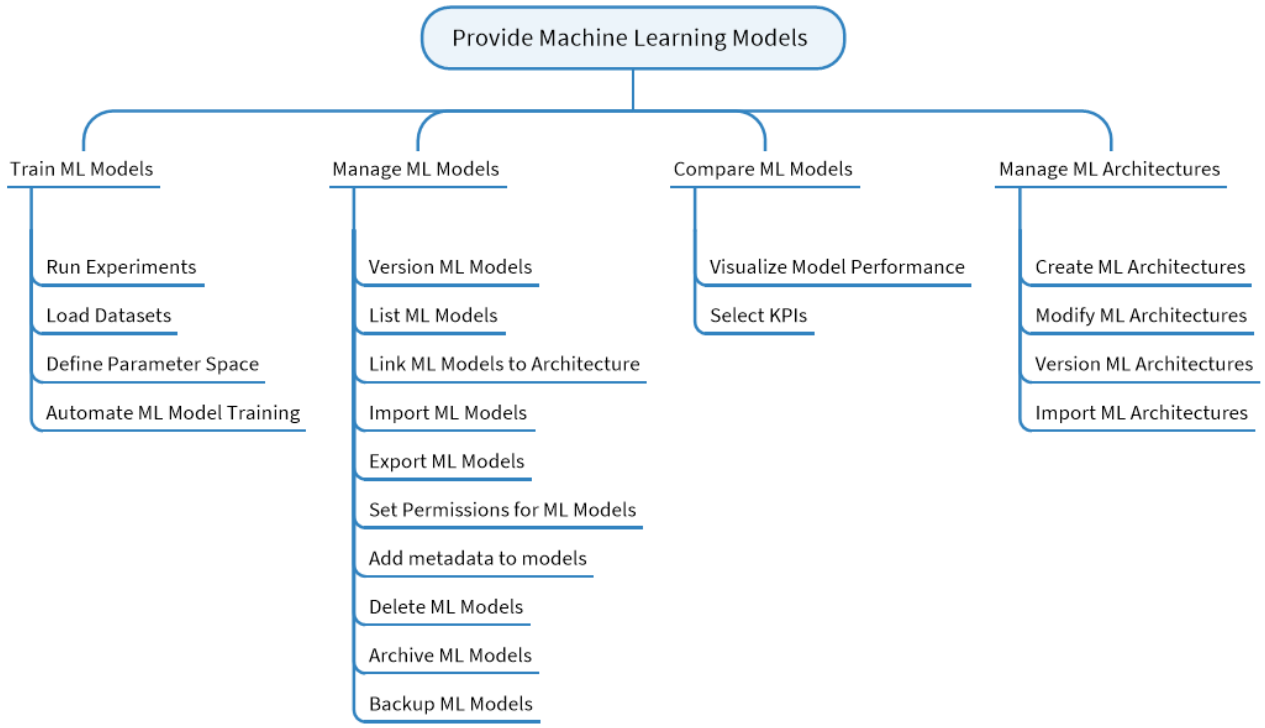


Figure 40: Functional tree ML Platform

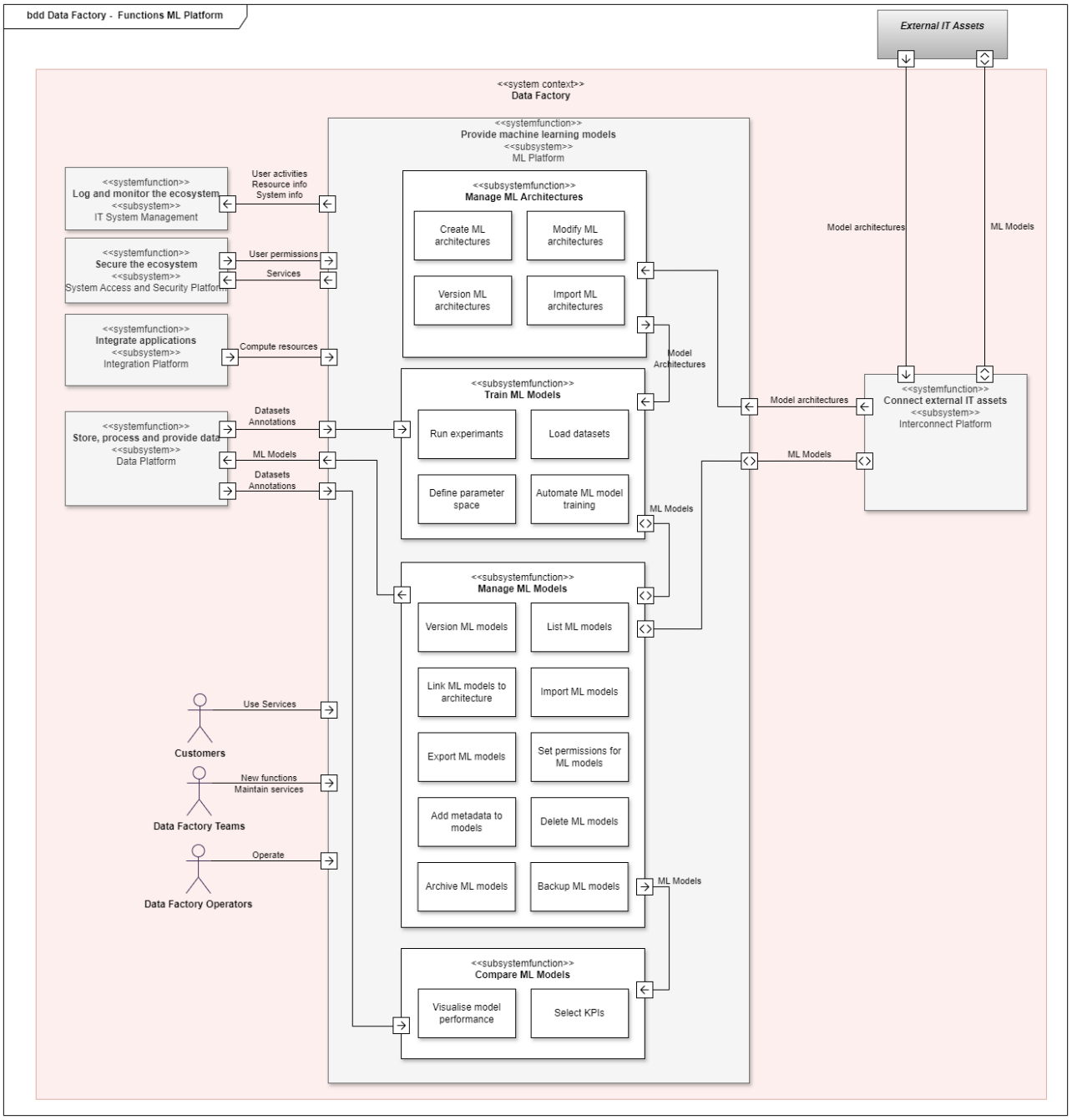


Figure 41: Context diagram ML Platform

The ML Platform is an integral component of the Data Factory, designed to create, manage, and utilise machine learning models. It provides the infrastructure necessary to support the lifecycle of machine learning models, from training and versioning to archiving and deployment.

- Function: Train ML Models
  - Input:
    - Datasets from Data Platform, ML model specifications.
  - Output:
    - Trained ML models ready for evaluation and deployment.
  - Requirements:

        - The ML Model Trainer shall run experiments.
        - The ML Model Trainer shall load datasets.
        - The ML Model Trainer shall define the parameter space.
        - The ML Model Trainer shall automate machine learning model training.
- Function: Manage ML Models
    - Input:
        - Trained models, metadata, model parameters.
    - Output:
        - Managed models stored, versioned, and prepared for use.
    - Requirements:
        - The ML Model Manager shall version machine learning models.
        - The ML Model Manager shall list machine learning models.
        - The ML Model Manager shall link machine learning models to architecture.
        - The ML Model Manager shall import machine learning models.
        - The ML Model Manager shall export machine learning models.
        - The ML Model Manager shall set permissions for machine learning models.
        - The ML Model Manager shall add metadata to models.
        - The ML Model Manager shall delete machine learning models.
        - The ML Model Manager shall archive machine learning models.
        - The ML Model Manager shall backup machine learning models.
- Function: Manage ML Architectures
    - Input:
        - Model architectures, update requests, versioning commands.
    - Output:
        - Current and historical versions of ML architectures.
    - Requirements:
        - The ML Architecture Manager shall create machine learning architectures.
        - The ML Architecture Manager shall modify machine learning architectures.
        - The ML Architecture Manager shall version machine learning architectures.
        - The ML Architecture Manager shall import machine learning architectures.
- Function: Compare ML Models
    - Input:
        - Performance data from different ML models.
    - Output:

        - Comparative analysis results, visualisations, and KPI selections.
    - Requirements:
        - The ML Model Comparer shall visualise machine learning model performance.
        - The ML Model Comparer shall select KPIs.

- Interactions:
    - Bidirectional Connections:
        - With External IT Assets via Interconnect Platform:

          ML Models: Two-way exchange of ML models between external assets and the ML Platform for continuous enhancement and collaboration.
    - Inputs:
        - From External IT Assets via Interconnect Platform:

          Model Architectures: Structural designs for ML models that are brought into the ML Platform.
        - From System Access and Security Platform:

          User Permissions: Authorization details that enable secure access to the ML Platform.

          Services: Various support services that underpin ML operations.
        - From Integration Platform:

          Compute Resources: Necessary computing power allocated for ML tasks.
        - From Data Platform:

          Datasets and Annotations: Information and metadata used for ML model training and validation.
    - Outputs:
        - To IT System Management:

          User Activities, Resource Info, System Info: Logs and updates regarding the utilization of the ML Platform by users.
        - To System Access and Security Platform:

          User Information: Feedback and data concerning user interactions with the ML Platform.
        - To Data Platform:

          ML Models: Developed and trained ML models ready for deployment or further refinement.

          Datasets and Annotations: Data that has been processed or augmented by ML algorithms.

- Roles Involved:
    - Customers:

        - Use Services: Engage with and benefit from the ML services provided.
    - Data Factory Teams:
        - New Functions & Maintain Services: Continuously develop and maintain ML services to improve and expand the platform's capabilities.
    - Data Factory Operators:
        - Operate: Oversee the day-to-day operations, ensuring the ML Platform's functionality and service delivery.

This delineation of interactions and roles solidifies the ML Platform as an essential component in the Data Factory's ecosystem, tasked with transforming data into actionable insights and enabling advanced predictive modeling through a secure, collaborative, and well-maintained environment.

2.7.8 Simulation Platform

Central to the Simulation Platform is the Asset Manager, which ensures that 3D assets are not only created, updated, and deleted as necessary but also that their data and integrity are rigorously validated, and permissions strictly controlled. This meticulous oversight enables accurate representation and consistent quality in the simulation environment.

In parallel, the Scenario Sampler subsystem is engaged in the meticulous preparation, versioning, archiving, and parameter setting for scenarios. This process is vital for the development of diverse and realistic simulation scenarios, which are essential for testing and refining Data Factory operations.

Moreover, the Generator is the driving force behind synthetic data creation. It is responsible for simulating scenarios and recording the resulting synthetic data streams. It also conducts checks and stores this synthetic data, playing a crucial role in expanding the breadth and depth of data available for simulations.

To ensure the reliability and accuracy of the simulations, the Validator is tasked with the validation of both digital twins and 3D assets. This ensures that the simulated environments and objects closely mirror their real-world counterparts.

Additionally, the Digital Twins Manager oversees the creation and importation of digital twins into the Simulation Platform. This function is integral to maintaining an updated and accurate digital representation of physical assets within the simulation.

The Simulation Platform’s integration with the wider Data Factory environment is pivotal, providing critical data that feeds into the operational decision-making process, driving innovation, and enhancing the overall functionality of the Data Factory.

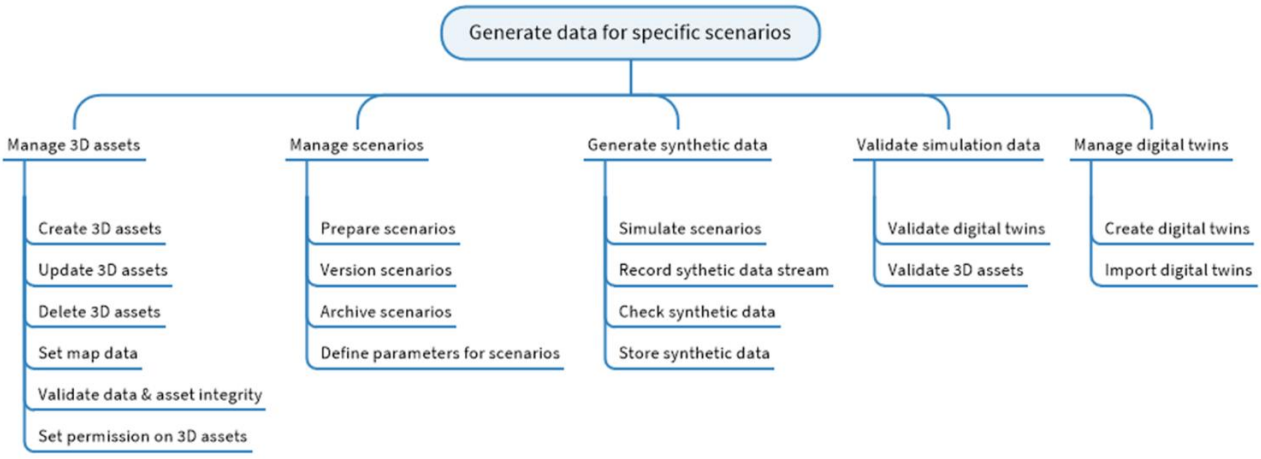


Figure 42: Functional tree Simulation Platform

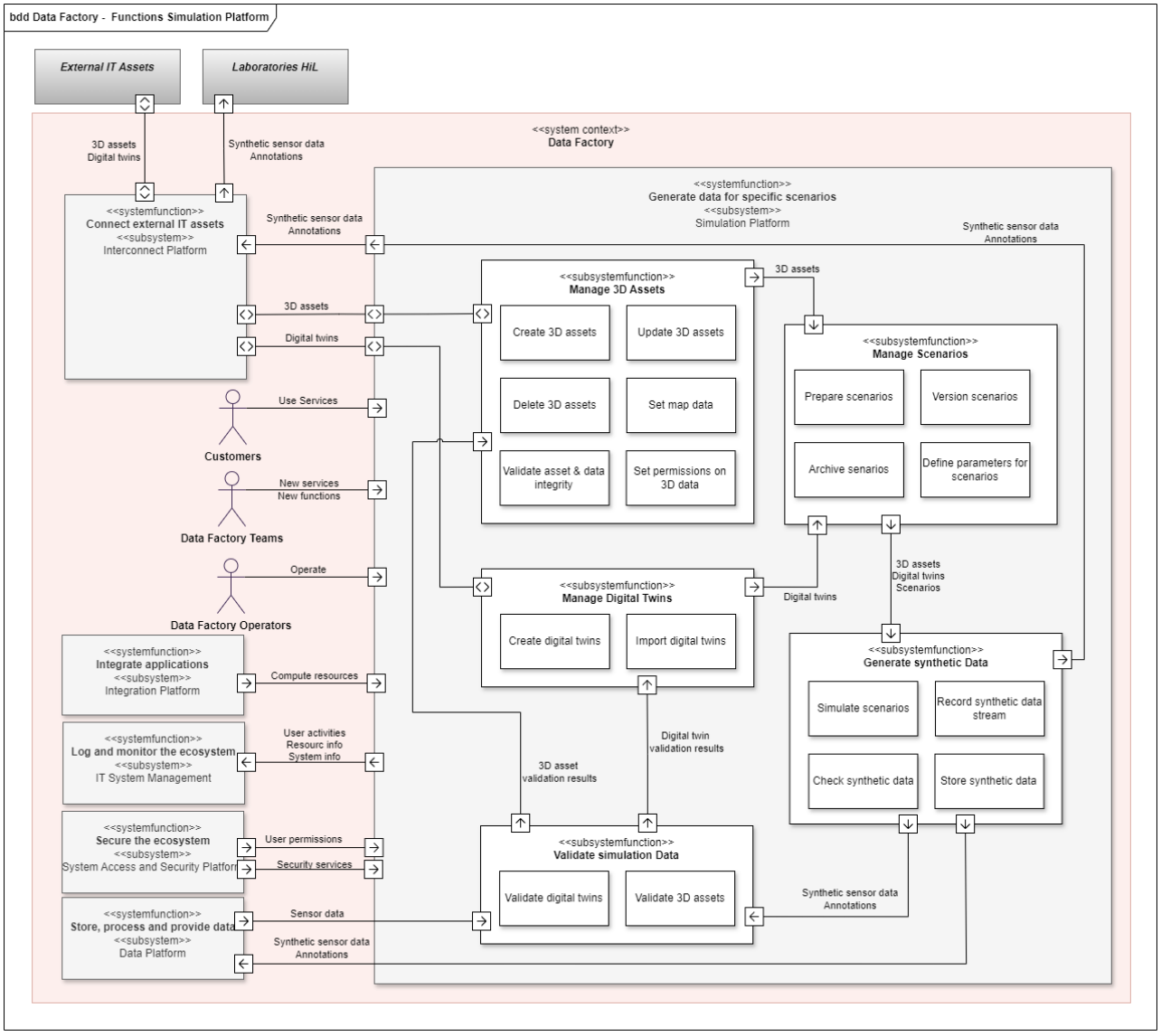


Figure 43: Context diagram Simulation Platform

The Simulation Platform is a central component of the Data Factory, designed to support the creation and management of virtual models and scenarios for testing and simulation. Its capabilities enable

stakeholders to generate, manipulate, and validate synthetic and digital twin data, enhancing the predictive modelling and decision-making processes within the Data Factory.

- Function: Manage 3D Assets
  - Input:
    - Digital assets data, user requests for creation, update, and deletion
  - Output:
    - Newly created 3D assets, updated 3D assets, deletion confirmations
  - Requirements:
    - The Asset Manager shall create, update, and delete 3D assets.
    - The Asset Manager shall set map data.
    - The Asset Manager shall ensure validation & integrity of 3D assets.
    - The Asset Manager shall restrict access to 3D assets.
- Function: Manage Scenarios
  - Input:
    - User requests for scenario preparation, versioning, archiving, and parameter definition
  - Output:
    - Prepared, versioned, archived scenarios, and scenarios with defined parameters
  - Requirements:
    - The Scenario Sampler shall prepare, version, archive scenarios, and define parameters for scenarios.
- Function: Generate Synthetic Data
  - Input:
    - Simulation parameters and scenario data
  - Output:
    - Simulated synthetic sensor data streams, synthetic data records
  - Requirements:
    - The Generator shall simulate scenarios and record synthetic data streams.
    - The Generator shall check synthetic data.
- Function: Validate Simulation Data
  - Input:
    - Digital twin data, synthetic sensor data streams
  - Output:
    - Validation results for digital twins and synthetic data

    - Requirements:
        - The Validator shall validate digital twins and synthetic data.
- Function: Manage Digital Twins
    - Input:
        - Requests for creation and importation of digital twins
    - Output:
        - Newly created digital twins, imported digital twins
    - Requirements:
        - The Digital Twins Manager shall create and import digital twins.
- Roles Involved:
    - Customers:
        - Utilize the services provided by the Simulation Platform.
        - Provide feedback on new services and functions that are developed based on simulation outcomes.
    - Data Factory Teams:
        - Operate the Simulation Platform, utilizing its capabilities for developing and testing.
        - Collaborate on new services and functions, ensuring they align with customer needs and project goals.
    - Data Factory Operators:
        - Facilitate the day-to-day operations of the Simulation Platform.
        - Ensure that the platform is running efficiently and is available for use by the Data Factory Teams and customers.
- Interactions:
    - Bidirectional:
        - Between Simulation Platform and External IT Assets via Interconnect Platform:

          3D Assets and Digital Twins: Exchanged for simulation and external enhancement.
    - Outputs:
        - To Laboratories HIL via Interconnect Platform:

          Sensor Data and Annotations: Output from simulations for laboratory analysis and interpretation.
        - To Data Platform:

          Synthetic Sensor Data and Annotations: Outputs of the simulated data for further analysis, storage, or feedback into the system.

- Inputs:
  - From Integration Platform:

    Compute Resources: Essential computational resources provided to the Simulation Platform for simulating complex scenarios.
  - From IT System Management:

    User Activities, Resource Info, System Info: Information and logs related to user interactions and resource utilization by the Simulation Platform.

In this configuration, the Simulation Platform acts as a central hub for the creation and processing of simulations, extensively interacting with the Interconnect Platform. It receives essential data for generating simulation scenarios and utilizes the resources of the Integration Platform for the interactive computing environment. The Simulation Platform ensures that user activities, resource, and system information are relayed to IT System Management, and it secures necessary permissions from the System Access and Security Subsystem while also transmitting user information back to the system.

2.7.9 System Access and Security Platform

The System Access and Security Platform (SASP) is integral to the Data Factory, ensuring secured access and robust identity verification. It's vital for preserving data integrity and confidentiality. SASP authenticates users, controls account access, and upholds secure communication standards to prevent unauthorized network interactions. Its proactive security measures safeguard system integrity, reacting quickly to any audit failures or security threats, and maintaining stringent data confidentiality through encryption. SASP also plays a critical role in ensuring operational resilience by overseeing the system's functionality against disruptions and attacks, with mechanisms in place for system recovery and maintaining continuous service availability.

Overall, SASP is the Data Factory's shield against digital threats, managing a broad array of security services and ensuring compliance with regulatory standards. It collaborates with other subsystems for holistic security management and evolves with the Data Factory's growth, embodying the commitment to secure, reliable operations.

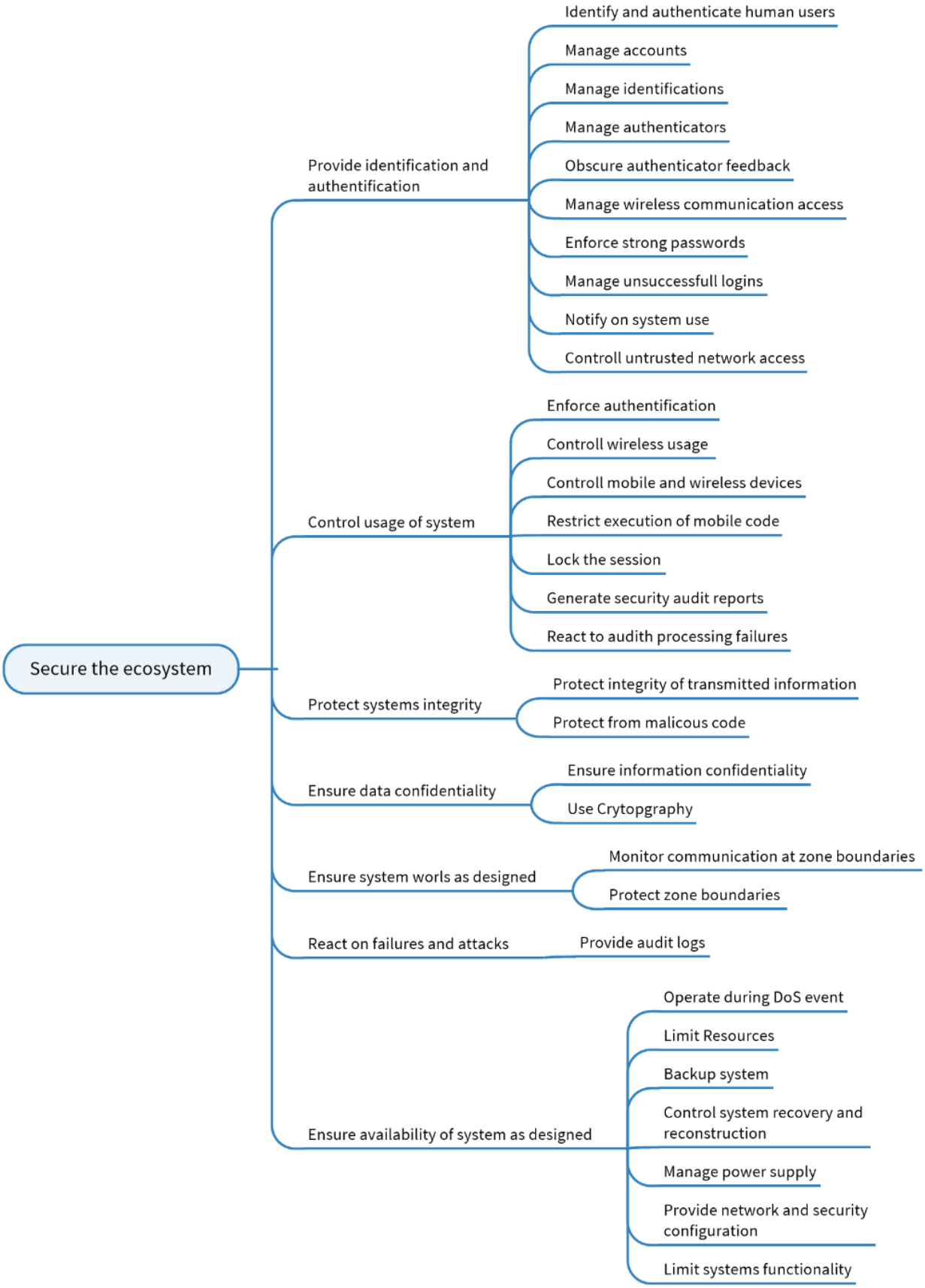


Figure 44: Functional tree System Access & Security Platform

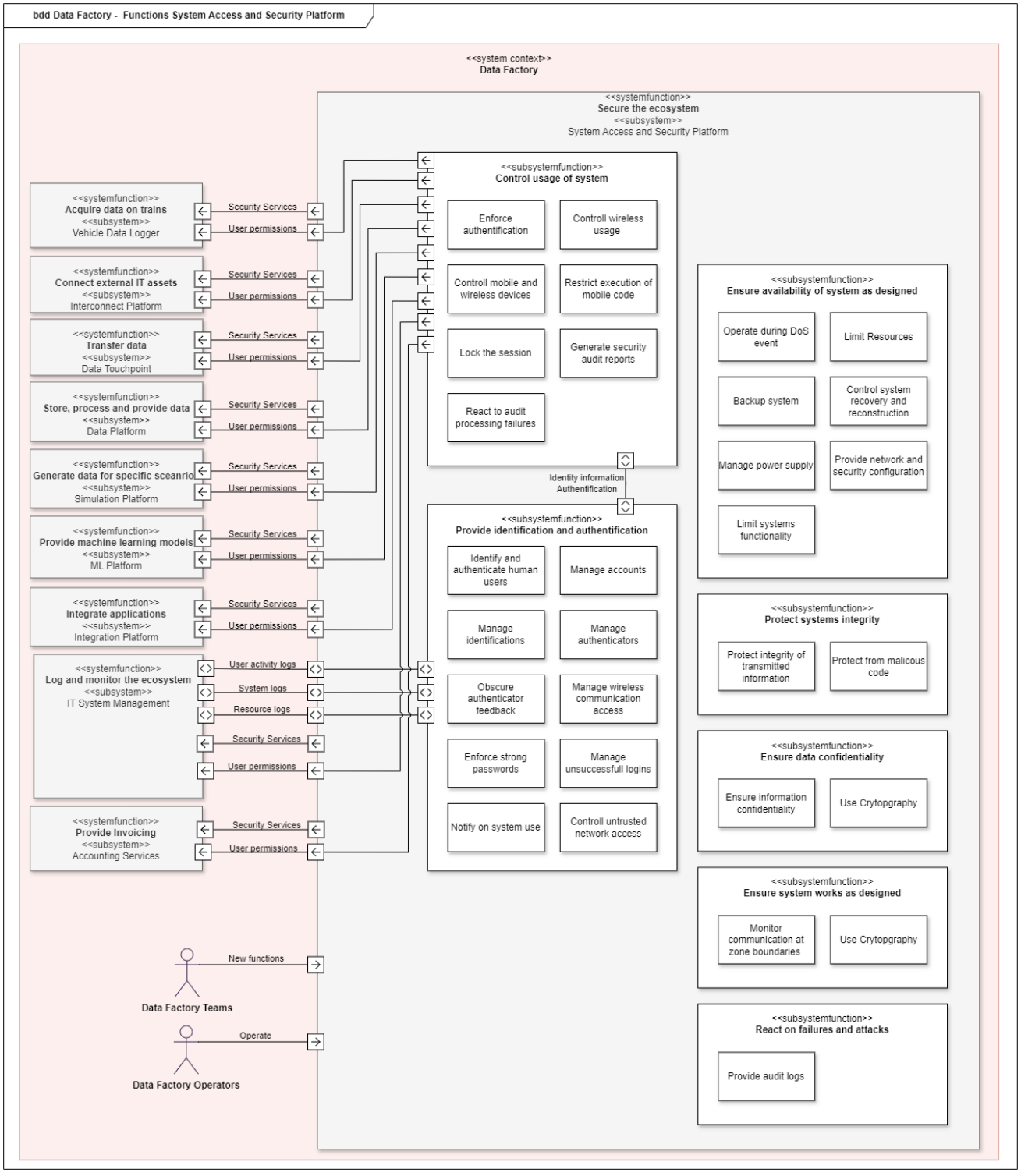


Figure 45: Context diagram System Access & Security Platform

This subsystem ensures the secure and regulated access to the Data Factory's systems and data, providing a suite of security measures and controls to protect against unauthorized access and to manage user permissions.

- Function: Provide Identification and Authentication
  - Input:
    - User credentials and access requests.
  - Output:
    - Authentication status and user identity confirmation.
  - Requirements:
    - The System Access & Security Platform shall identify and authenticate human, software processes and devices to allow them to access th ecosystem or IT-assets.
    - The System Access & Security Platform shall identify and authenticate human, software processes and devices to allow them to access th ecosystem or IT-assets.
    - The System Access & Security Platform shall identify and authenticate human, software processes and devices to allow them to access th ecosystem or IT-assets.
    - The System Access & Security Platform shall identify and authenticate human, software processes and devices to allow them to access th ecosystem or IT-assets.
    - The System Access & Security Platform shall identify and authenticate human, software processes and devices to allow them to access th ecosystem or IT-assets.
    - The System Access & Security Platform shall identify and authenticate human, software processes and devices to allow them to access th ecosystem or IT-assets.
    - The System Access & Security Platform shall identify and authenticate human, software processes and devices to allow them to access th ecosystem or IT-assets.
    - The System Access & Security Platform shall identify and authenticate human, software processes and devices to allow them to access th ecosystem or IT-assets.
    - The System Access & Security Platform shall identify and authenticate human, software processes and devices to allow them to access th ecosystem or IT-assets.
    - The System Access & Security Platform shall identify and authenticate human, software processes and devices to allow them to access th ecosystem or IT-assets.
    - The System Access & Security Platform shall identify and authenticate human, software processes and devices to allow them to access th ecosystem or IT-assets.Function: Control Usage of System
  - Input:
    - Access and operational requests from users and systems.
  - Output:
    - Permissions granted or denied, system usage monitored..
- Function: Protect Systems Integrity
  - Input:
    - System operations and data flow.

- Output:
  - Protected system against data tampering and code exploitation.
- Requirements:
  - The System Access & Security Platform shall enforce that only authorized users can read data.
  - The System Access & Security Platform shall enforce that only authorized users can read data.Function: Ensure Data Confidentiality
- Input:
  - Requests to access or transfer data.
- Output:
  - Data access provided with confidentiality maintained.

- Function: Ensure System Works as Designed
  - Input:
    - System performance and status data.
  - Output:
    - Maintained system functionality, performance monitoring.
  - Requirements:
    - The System Access & Security Platform shall ensure that the system solely works as designed.
    - The System Access & Security Platform shall ensure that the system solely works as designed.Function: React on Failures and Attacks
  - Input:
    - System alerts and error logs.
  - Output:
    - Responses to system threats and failure events.
- Function: Ensure Availability of System as Designed
  - Input:
    - System status and health checks.
  - Output:
    - Continuous system functionality, even under adverse conditions.
  - Requirements:
    - The System Access & Security Platform shall ensure the availability of the system as designed.
    - The System Access & Security Platform shall ensure the availability of the system as designed.
    - The System Access & Security Platform shall ensure the availability of the system as designed.

- The System Access & Security Platform shall ensure the availability of the system as designed.
- The System Access & Security Platform shall ensure the availability of the system as designed.
- The System Access & Security Platform shall ensure the availability of the system as designed.
- The System Access & Security Platform shall ensure the availability of the system as designed.Interactions:

- Bidirectional Connectors:
  - With IT System Management:

    User Activity Logs, System Logs, Resource Logs: Exchanges detailed logs for a comprehensive understanding and management of security across the system.

- Outputs:
  - To Vehicle Data Logger:

    Security Services, User Permissions: Provides essential security protocols and access permissions to ensure secure data logging operations.

  - To Interconnect Platform:

    Security Services, User Permissions: Delivers security and access control services for safe and secure interconnection with external IT assets.

  - To Data Touchpoint:

    Security Services, User Permissions: Ensures secure data transfer and access control for touchpoint operations.

  - To Data Platform:

    Security Services, User Permissions: Facilitates the secure storage, processing, and provisioning of data.

  - To Simulation Platform:

    Security Services, User Permissions: Provides security measures and user access management for simulation data handling.

  - To ML Platform:

    Security Services, User Permissions: Secures machine learning operations by managing access and protecting integrity.

  - To Integration Platform:

    Security Services, User Permissions: Ensures secure application integration and user access management.

  - To Accounting Services:

    Security Services, User Permissions: Provides security and access permissions for secure billing and invoicing processes.

        - To IT System Management:

            Security Services, User Permissions: Delivers security protocols and user permissions for IT system oversight.

- Roles Involved:
    - Data Factory Teams:
        - New Functions: Develop and introduce new security functions, enhancing the platform's capabilities.
    - Data Factory Operators:
        - Operate: Manage the day-to-day operations, ensuring that security measures are actively maintained and effectively implemented.
        -

The System Access and Security Platform is a keystone within the Data Factory, integral to the security and access control framework. It is the guardian of the ecosystem, establishing rigorous protocols and managing permissions to safeguard against unauthorized access and potential vulnerabilities. This platform is responsible for securing the Data Factory's digital environment, from data logging to user interactions with various subsystems, ensuring that every transaction and operation is conducted within a secure and controlled setting. It maintains a continuous state of vigilance, ready to respond to security incidents while providing essential data to the IT System Management for proactive security governance. The role of the Data Factory Teams and Operators is crucial in sustaining this security posture, keeping the Data Factory's mission of safe, secure, and uninterrupted operations.

2.7.10 Vehicle Data Logger

The Vehicle Data Logger is specialized in recording large data streams of onboard data. Depending on the acquisition frequency, particularly high-resolution cameras generate a data rate of over one hundred megabytes per second.

In addition to camera data, the data streams from lidars, radars, localisation sensors, diagnostic functions and the vehicle are recorded simultaneously.

The Vehicle Data Logger is rail certified and supports data transfer to the track-side via manual hard disk exchange, as well as a wireless connection to the Data Touchpoint (see section 2.7.3).

The functionalities of the Vehicle Data Logger include storing and transferring data, ensuring data integrity, logging and monitoring the subsystem and data diagnostics functions to ensure high data quality.

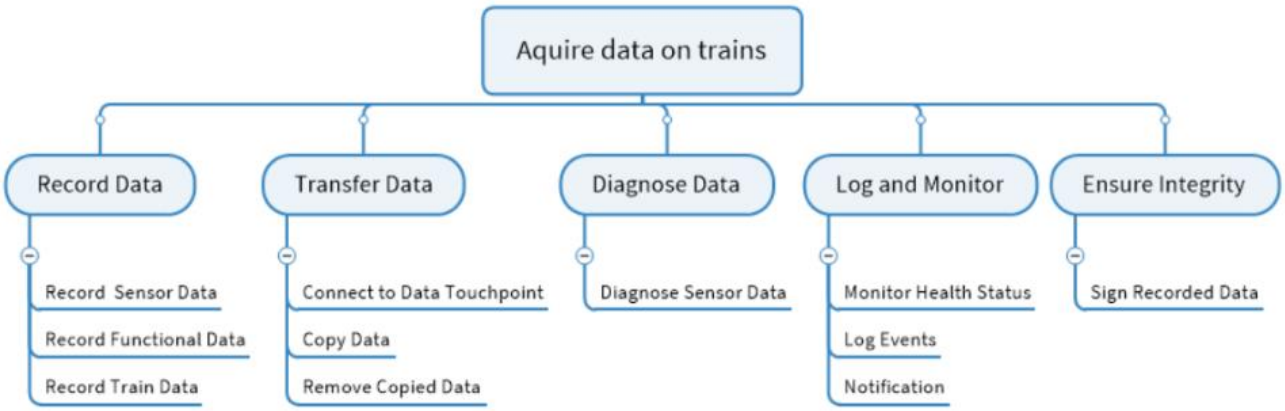


Figure 46: Functional tree Vehicle Data Logger

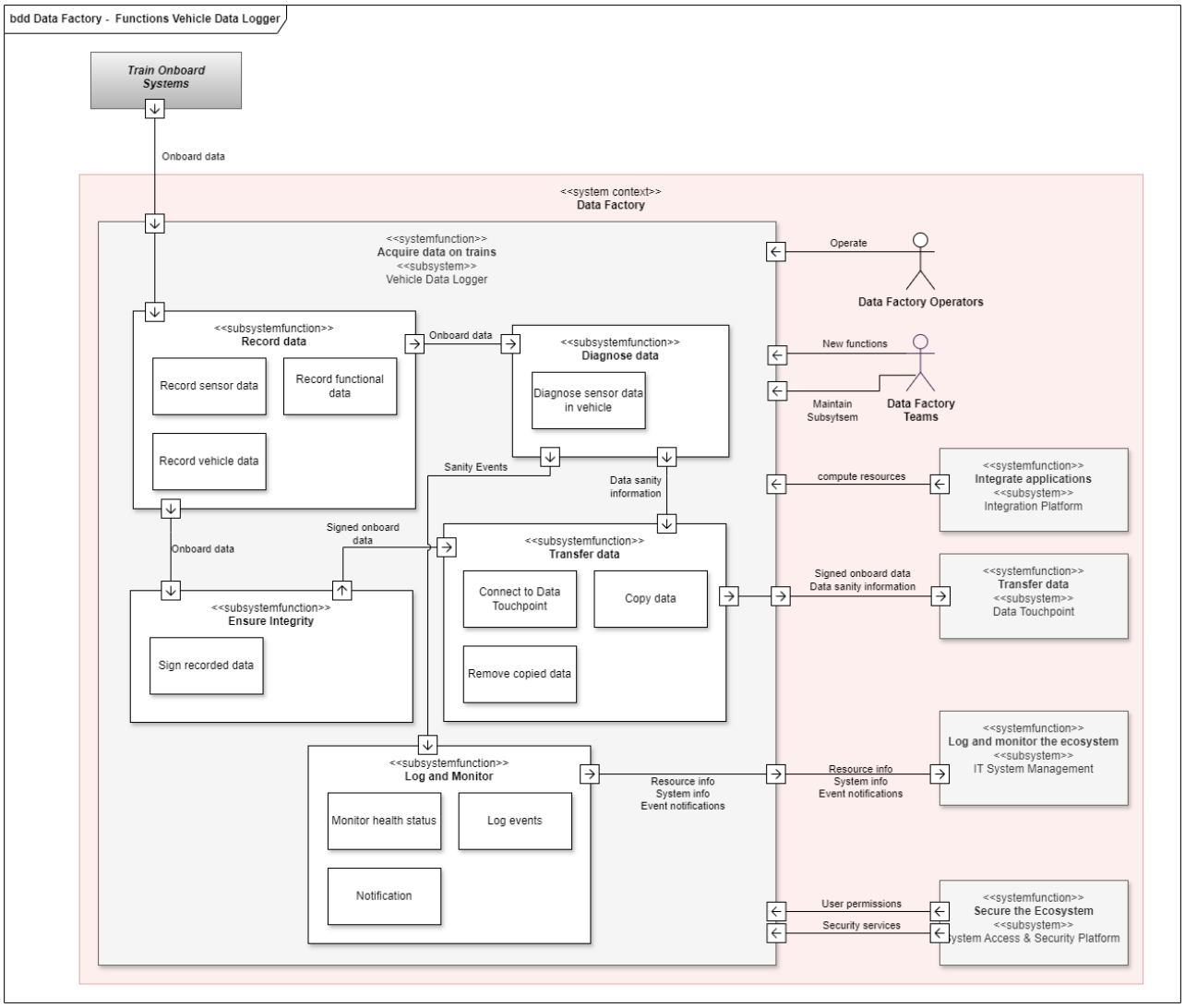


Figure 47: Context diagram Vehicle Data Logger

The Vehicle Data Logger subsystem, which operates under the system function "Acquire Data on Trains," acts as the data acquisition unit within the Data Factory. Here's a breakdown of its core functionalities with their corresponding inputs, outputs, and requirements:

- Function: Record Data
  - Input:
    - Sensor data, Functional data, Vehicle data from Train Onboard Systems
  - Output:
    - Onboard data directed towards the "Diagnose Data" function.
  - Requirements:
    - The Recorder shall record data.
    - The Recorder shall record functional data.
    - The Recorder shall record vehicle data.
- Function: Diagnose Data

    - Input:
        - Sensor data from the "Record Data" function
    - Output:
        - Data sanity information which goes to the "Transfer Data" function
    - Requirements:
        - The Diagnostics shall perform sanity checks on the data in the vehicle.
        - The Diagnostics shall perform integrity checks on the data in the vehicle.
- Function: Ensure Data Integrity
    - Input:
        - Sensor data from the "Record Data" function
    - Output:
        - Signed sensor data that is transferred to the "Transfer Data" function
    - Requirements:
        - The Integrity Guardian shall sign recorded data.
- Function: Transfer Data
    - Input:
        - Signed sensor data, Data sanity information, Functional data, and Vehicle data from the "Ensure Data Integrity" and "Diagnose Data" functions
    - Output:
        - Data transferred to the Data Touchpoint Subsystem includes Copy data and remove copied data operations
    - Requirements:
        - The Transmitter shall connect to data touchpoints.
        - The Transmitter shall copy data.
        - The Transmitter shall remove copied data.
- Function: Log and Monitor
    - Input:
        - Monitor health status and Log events
    - Output:
        - Notifications that are likely communicated to the IT System Management and possibly other Data Factory Teams
    - Requirements:
        - The Monitoring collector shall monitor the device health status.
        - The Monitoring collector shall log events.
        - The Monitoring collector shall give notifications if specific events occur.

- Interaction:
  - Inputs:
    - From Train Onboard Systems:

      Onboard Data: This data includes various types of information collected directly from the trains, such as operational, diagnostic, and sensor data.

    - From Integration Platform:

      Compute Resources: These are computational resources allocated for processing the onboard data, possibly including CPU time, memory, and storage.

    - From the System Access and Security Platform:

      User Permissions: This would include the access rights and permissions for users to interact with the data collected by the Vehicle Data Logger, ensuring that only authorized personnel can access or modify the onboard data.

      Security Services: The Vehicle Data Logger receives security-related services such as authentication protocols, encryption methods, and data protection mechanisms to secure the onboard data from unauthorized access and ensure data integrity.

  - Outputs:
    - To Data Touchpoint Subsystem:

      Signed Onboard Data: This is the onboard data that has been verified and signed off for authenticity and integrity.

      Data Sanity Information: This includes information on the validity and logical consistency of the onboard data.

    - To IT System Management:

      Resource Info: Information on the utilization of resources by the Vehicle Data Logger subsystem.

      System Info: General information about the system status of the Vehicle Data Logger.

      Event Notifications: Alerts and notifications generated by the Vehicle Data Logger subsystem concerning significant events or anomalies.

  - Roles Involved:
    - Data Factory Operators:

      They interact with the Vehicle Data Logger as part of the system function "Acquire Data on Trains" to operate and manage the data acquisition process, making sure that the data collected from Train Onboard Systems is properly logged and transmitted to the Data Touchpoint. They play a significant role in the practical application of the data and alignment with operational requirements.

    - Data Factory Teams:

These groups are involved in the broader aspects of the Data Factory's operations. They may not directly engage with the "Acquire Data on Trains" function but are likely to utilize the data outputs for analysis, development of new functionalities within the Data Factory, and ensuring that the data collected aligns with the overarching objectives and use cases of the Data Factory. Their role serves as a bridge between the theoretical data models and practical, operational needs, influencing how data is acquired, processed, and applied.

In this subsystem, the Vehicle Data Logger acts as a comprehensive data collection and processing unit, interfacing directly with the Train Onboard Systems to capture, secure, and transfer train data within the Data Factory ecosystem.

2.7.11 Accounting Services

The Accounting Services subsystem within the Data Factory is a crucial component that ensures the financial transactions related to the usage of services are tracked, recorded, and billed accurately. It encompasses the end-to-end process of capturing consumption data, generating detailed invoices, and providing transparent billing information to customers. This subsystem is integral to maintaining financial integrity, supporting the Data Factory's revenue streams, and delivering clear communication regarding service costs to customers. Through meticulous data collection and processing, the Accounting Services subsystem stands as a cornerstone of the Data Factory's commercial operations, exemplifying the commitment to accountability and precision in financial management.

Figure 48: Functional tree Accounting Services

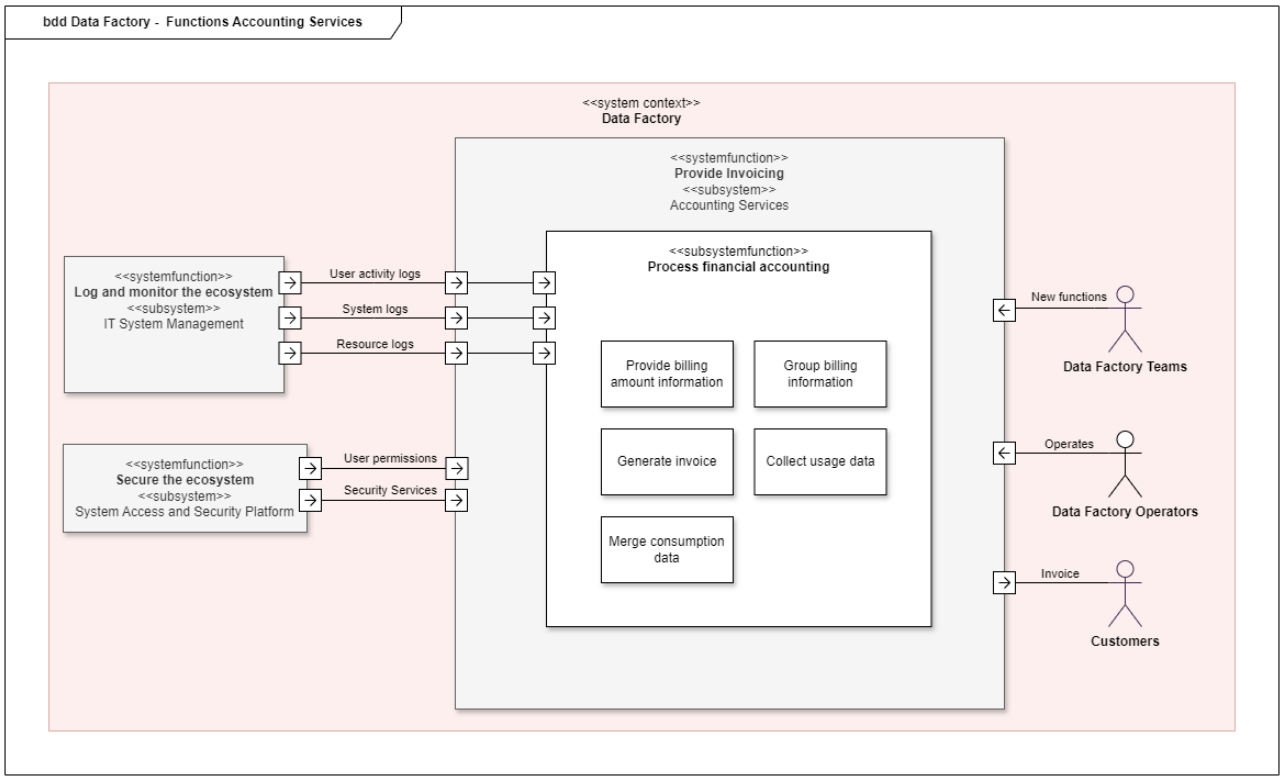


Figure 49: Context diagram Accounting Services

The Accounting Services subsystem within the Data Factory is integral for managing financial transactions and resource usage data. It is designed to handle billing processes, from providing information on billing amounts to consolidating data for invoicing. This subsystem plays a pivotal role in ensuring transparent and accurate billing for Data Factory services.

- Function: Provide financial accounting
  - Input:
    - Requests for billing information.
  - Output:
    - Billing amount information provided to requesters.
  - Requirements:
    - The Accounting Service shall cary out the enttire financial processing.
- Interactions:
  - Inputs
    - from IT System Management:

      User Activity Logs: These logs provide a record of user actions within the Data Factory, which can be used for billing based on service usage.

      System Logs: These logs record system events that may affect billing, such as system downtime or performance issues.

      Resource Logs: These logs detail the use of resources within the Data Factory, which could influence cost calculations.

- from System Access and Security Platform:

  User Permissions: Permissions data is used to determine access levels and potentially the scope of billing for different users or groups.

  User Information: Identifiable information about users that can be used to tailor and dispatch invoices.

- Roles Involved:
  - Data Factory Teams:
    - They introduce new functions to the Data Factory, potentially affecting the billing services by adding new billable features or services.
  - Data Factory Operators:
    - They operate the Data Factory and, as part of their role, may use data from the Accounting Services for operational reporting and monitoring.
  - Customers:
    - They receive invoices generated by the Accounting Services, which reflect their usage and costs associated with the Data Factory services.
    - In essence, the Accounting Services subsystem functions as the economic heartbeat of the Data Factory, interfacing with various roles and systems to ensure that all financial transactions related to the Data Factory's service offerings are accounted for accurately and transparently.

3 CONCLUSIONS

Conclusion: The completion of Deliverable 7.1 within work package 7 represents a milestone for the R2DATO project and a major step forward towards an advanced, digitized and automated European rail system. The efforts documented here reflect the development of the Data Factory - a cornerstone of the project architecture designed to lead the intelligent development of rail infrastructure.

Business Requirements and Goals Achievement: Regarding section 2.1, the Data Factory was set up with the needs and requirements of the stakeholders in mind. Its ability to handle data at all stages - collection, ingestion, processing and use for machine learning - demonstrates that the business objectives of the project are being met. These fundamental elements have been set up physically and toolchain-wise to ensure that the Data Factory is ready for further advances in rail operations technology.

Scenario Analysis: The evaluation of the scenarios in section 2.2, the assumptions and the project structure based on InfraGO's experience played an important role in determining the course of the project. However, due to the complexity and associated regulatory aspects, the use of the models in vehicles and the certification and approval processes posed a challenge that could not be fully overcome within this task. This results in areas for future developments and regulatory considerations.

Interconnected Work Packages and Projects: As outlined in section 2.3, the synergies with interrelated work packages and predecessor projects such as X2Rail4 [16], Tauro Shift2Rail [3] and in particular CEF 2 RailDataFactory [4] and GaiaX CartenaX [21] were significant. The continuity provided by the same thematic focus and staff number provided valuable opportunities to shape the outcome by leveraging previous findings and proven expertise.

Data Requirement Specification Compliance: In accordance with section 2.5, the project has effectively met the data requirement specifications and created a comprehensive data and sensor model. This achievement forms the basis for future data exchange, data sharing and the provision of an open dataset as envisaged in D7.6.

System Evaluation: The system and its subsystems described in sections 2.6 and 2.7 have been assessed to effectively meet the needs of stakeholders and comply with security requirements. The various subsystems were divided into different functional units, each of which contributes to the smooth operation of the Data Factory.

Integration and Dataflow: The analysis of the integration platforms and the data flow diagrams in sections 2.7.1 and 2.7.4 revealed a harmonious integration of components that together contribute to achieving the project objectives. The data flow diagrams in particular offer a differentiated view of the relationships between the subsystems and open up new perspectives on their dependencies.

Subsystems Within the Data Factory Architecture: The creation and mapping of the architecture of the Data Factory, as seen in the subsystems such as Data Platform, Data Touchpoint and others, has enabled a structured and efficient development path. This made it possible to capture the goals already achieved and a clear vision for the subsequent project phases.

Subsystem Summaries Each subsystem has been evaluated for its contribution to the Data Factory:

- The **Vehicle Data Logger** has proven effective in acquiring train data, a foundational step in the data lifecycle.

- **Data Touchpoint** efficiently manages data transfers, a critical function ensuring continuous data integrity.
- The **Integration and Interconnect Platforms** successfully ensure interoperability, enhancing the Data Factory's capabilities through strategic connections with external IT assets and laboratories for Hardware-in-the-Loop (HiL) simulations.
- **IT System Management** and the **ML Platform** manage the data flow and machine learning models, directly impacting the analytical functions of the Data Factory.
- The **Simulation Platform** provides invaluable synthetic data, enriching the pool of information for model training and evaluation.
- The **System Access and Security Platform** upholds stringent security protocols, essential for the protection and integrity of the entire ecosystem.
- **Accounting Services** ensures transparency in resource utilization and billing, highlighting the commercial viability of the Data Factory.

Machine Learning and Simulation Platforms: The effectiveness of the ML Platform and Simulation Platform in Sections 2.7.7 and 2.7.8 has been affirmed. They form a vital suite of functionalities that have contributed substantially to the project's objectives and are essential for the creation of AI-supported GoA4 ATO functionalities.

Security and Access: The System Access and Security Platform, explored in Section 2.7.9, has comprehensively addressed and implemented the required security measures. The holistic security assessment based on established standards provides an essential guideline for IT and data security within the Data Factory.

Lessons Learned and Future Recommendations: A key finding from the comprehensive analysis is that the design and construction of such a complex system requires continuous, iterative development. For future activities, it is recommended that the project focus on these iterative processes, which are crucial for managing the complexity of such a dynamic and multi-layered system.

Concluding Remarks: Deliverable 7.1 not only summarizes current achievements, but also sets the stage for future innovation within the R2DATO project and the wider Europe's Rail initiatives. It symbolizes the collective commitment, technological capabilities and strategic insights that will lead us into an era of improved efficiency and potential for GoA4 automated train operations. The foundation laid with this report will pave the way for further progress and move the project towards the ultimate goal of a digitized, connected and autonomous European rail network.

4 APPENDIX

4.1.1 General (all classes)

4.1.1.1 Unique Identifiers

DAE-507 - Unique identifier for each annotation

A unique identifier (ID) shall be assigned to each annotation (i.e. annotation geometry).

DAE-506 - Unique identifier for each object

Each annotation geometry shall have a unique identifier (ID) for the objects. The unique identifier (ID) is used to distinguish different real-world objects. Annotations that belong to the same real-world object shall have the same unique object identifier (ID) (in the different sensors and over time). Hint: This identifier is often also referred to as temporal or tracking ID.

DAE-712 - Assign annotations to object

Each annotation (i.e. geometry) shall be assigned to an object. All annotations assigned to an object shall represent the same physical instance in different sensor modalities and across frames.

DAE-162 - UUID for all unique identifiers

All unique identifiers shall follow the Universally Unique Identifier (UUID) standard in version 4.

DAE-713 - Implementation example

Having an annotation_id and an object_id for each annotation geometry satisfies the requirements DAE-507, DAE-506 and DAE-712.

4.1.1.2 Attributes

General requirements for attributes

DAE-741 - Attribute types

Each attribute shall be of a fixed type based on the attribute types described in the child requirements.

DAE-742 - Attribute type Boolean

Attributes of the type Boolean shall have the values true and false.

DAE-744 - Attribute type Single-Select

Attributes of the type Single-Select shall allow selection of a single value out of a list of pre-defined strings. These strings are defined for each attribute individually.

DAE-745 - Attribute type Multi-Select

Attributes of the type Multi-Select shall allow selection of multiple values out of a list of pre-defined strings. These strings are defined for each attribute individually.

DAE-884 - Attribute type Reference

Attributes of type Reference shall reference any existing object by allowing to select or enter the UUID of the referenced object.

DAE-751 - Attribute Scopes

Each attribute has a scope and shall meet the consistency requirements of its scope. An overview is the scopes and consistencies is provided in the following matrix. The scopes are defined as child requirements.

Attribute Consistency Matrix	**Frame-Specific**	**Cross-Frame**
Sensor-Specific	Scope Annotation Describe an annotation type (geometry) and can therefore change across sensors and frames. Examples: - Occlusion - Truncation	-
Cross-Sensor	Scope Frame Describe the state of an object and are therefore constant across sensors but can change with time. Examples: - Pose - Carrying - Distracted	Scope Object Describe the physical properties of an object and can therefore change neither across sensors nor across frames. Examples: - Age - Train Type - Railside

DAE-709 - Attribute scope Annotation

Attributes with scope Annotation shall be evaluated and set for each frame and sensor independently.

DAE-710 - Attribute scope Frame

Attributes with scope Frame shall be evaluated and set according to the most suitable sensor modality (i.e. the sensor in which the value of the attribute can be determined most accurately). They have to be consistent across all sensor modalities within a frame but shall be evaluated for each multi-modal frame individually.

DAE-711 - Attribute scope Object

Attributes with scope Object shall be evaluated and set based on the most suitable sensor-modality and frame. They have to be consistent across all sensor modalities and frames for each object within a sequence.

DAE-511 - Use of attribute value Unknown

If an attribute has the option “unknown”, this option shall only be selected when the actual value can not be determined due to a lack of optical information.

4.1.1.3 Annotation Types

DAE-714 - 2D Bounding Box

Annotations with type 2D Bounding Box shall be labelled with a rectangle. The rectangle shall have a centre point (x, y), a width w and a height h. The rectangle shall be axis aligned with the image and shall not have any rotation.

DAE-716 - 2D Polygon

Annotations with type 2D Polygon shall be labelled with a polygon with N anchor points. N depends on the sub-type of the polygon described in the child requirements.

DAE-748 - 4-Point Polygon

Annotations with type 4-Point Polygon shall be labelled with a 2D Polygon with 4 anchor points.

DAE-749 - Outline Polygon

Annotations with type Outline Polygon shall be labelled with a 2D Polygon with 5 to 20 anchor points. The polygon should describe the basic outline of the object, but do not have to meet the 3px precision requirements at all parts of the object. Small details can be neglected.

DAE-717 - 2D Polyline

Annotations with type 2D Polyline shall be labelled with a line with multiple anchor points. The number of anchor points shall be selected in order to follow the shape of the object precisely. Especially curves shall be supported with enough anchor points.

DAE-718 - 2D Rotated Bounding Box

Annotations with type 2D Rotated Bounding Box shall be labelled with a rectangle that can be rotated around its centre. The rectangle shall have a centre point (x, y), a width w, a height h and a rotation angle alpha in radians. The rotation shall be defined as a right-handed rotation.

DAE-715 - 3D Bounding Box

Annotations with type 3D Bounding Box shall be labelled with a cuboid in 3D Euclidean space defined by position (x, y, z), rotation (qa, qb, qc, qd) and size (sx, sy, sz). The position and size shall be defined as 3-vectors and the rotation shall be defined as 4-vector quaternion.

Attribute	Unit	Description
x	m	Specifies the x-coordinate of the 3D position of the centre of the cuboid.
y	m	Specifies the y-coordinate of the 3D position of the centre of the cuboid.
z	m	Specifies the z-coordinate of the 3D position of the centre of the cuboid.
qa		Specify the quaternion in non-unit form (x, y, z, and w) as in the SciPy convention.
qb		Specify the quaternion in non-unit form (x, y, z, and w) as in the SciPy convention.
qc		Specify the quaternion in non-unit form (x, y, z, and w) as in the SciPy convention.
qd		Specify the quaternion in non-unit form (x, y, z, and w) as in the SciPy convention.
sx	m	Specifies the x-dimension of the cuboid.
sy	m	Specifies the y-dimension of the cuboid.
sz	m	Specifies the z-dimension of the cuboid.

DAE-750 - Export 3D Semantic Segmentation for 3D Bounding Boxes

Each 3D Bounding Box shall additionally be exported as 3D Semantic Segmentation by adding all points within the 3D Bounding Box to the 3D Semantic Segmentation.

DAE-719 - 3D Semantic Segmentation

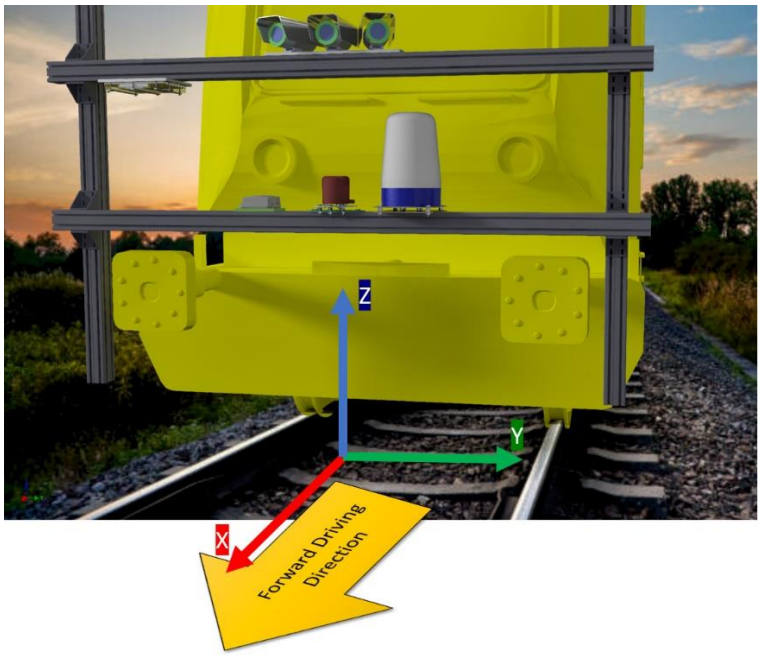
Annotations with type 3D Semantic Segmentation shall be labelled by marking all points that belong to the object. The identifier of the 3d point within the point cloud shall be used to add the point to the segmentation.

4.1.1.4 Annotation Rules

This category contains annotation rules that apply to each class.

DAE-539 - Use reference coordinate system

The following coordinate system shall be used to define the position of the annotation types.



DAE-885 - Annotations match the physical object

Annotations shall represent the physical (real) object in terms of position and size. The annotation types shall be amodal, i.e. occluded parts of objects shall be included within the annotation geometry.

DAE-888 - Annotate entire object

Annotations that enclose objects such as Bounding Boxes and Polygons shall enclose all points or pixels of the physical object.

DAE-887 - Annotate tightly

Annotations that enclose objects shall surround the corresponding objects tightly, i.e. the smallest geometry is searched for that includes the physical object entirely.

DAE-886 - Estimate real size of occluded objects

The real size of annotations that enclose objects such as Bounding Boxes and Polygons shall be estimated when the object is occluded by other objects.

DAE-889 - Annotate within image boundaries

Only parts of the objects that are within the image, i.e. in the field of view of the camera, shall be labelled. Annotations shall not exceed image borders and shall not estimate the size of objects truncated by the image borders.

DAE-890 - Apply general and class-specific annotation rules

General and class-specific annotation rules shall be applied for annotation.

DAE-891 - Class-specific rules replace global rules

Class-specific annotation rules shall replace general annotation rules when regulating the same subject matter. In case of contradiction class-specific annotation rules shall overrule general annotation rules.

DAE-892 - Annotate objects in camera images

Objects shall be labelled in visual and infrared camera images.

DAE-894 - Annotation objects with a size of at least 25 px

Objects with a size of 25 pixels or more in height or width shall be labelled. Smaller objects don not used to be labelled.

DAE-155 - Annotate all visible objects

All objects shall be labelled in the cameras, regardless of whether they are visible in another sensor.

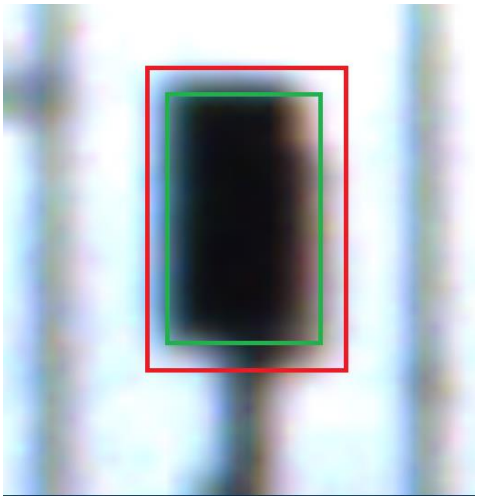
DAE-898 - Annotation precision better than 3 px

Annotations shall have precision of at least 3 px, i.e. they shall not differ by more than 3 px from the representation of the physical object unless specified differently by the annotation type.

DAE-542 - Blurred edges belong to object

If the edges of an object are blurred, the blurred area shall be part of the object and be enclosed by the annotation.

Example: In the following image the red 2D Bounding Box shall be used for labelling the signal.



DAE-893 - Annotate objects in LiDAR point clouds

Objects shall be labelled in the {*}LiDAR point cloud{*}.

Objects shall be labelled in the LiDAR point cloud.

DAE-895 - Annotation objects consisting of at least 3 points

Objects that consist of 3 or more points shall be labelled. Objects with less than do not need to be labelled.

DAE-476 - Annotate all visible objects

All objects shall be labelled in the LiDAR, regardless of whether they are visible in another sensor.

DAE-899 - Annotation precision better than 10 cm

Annotations shall have a precision of at least 10 cm, i.e. they shall not differ by more than 10 cm from the representation of the physical object unless specified differently by the annotation type.

DAE-902 - Rotate around z-axis to match front face

Annotations shall be rotated around the z-axis to match the front face of an object.

DAE-901 - Avoid rotation around x- and y-axis

Rotations around x- and y-axis shall be avoided. Only if the precision requirements can not be meet without an rotation around the x- and y-axis it may be allowed.

DAE-896 - Annotation objects in radar images

Objects shall be labelled in the radar images.

DAE-897 - Use projection from LiDAR for labelling

The annotations for objects within the radar images can be projected from the LiDAR point clouds.

DAE-900 - Annotation precision better than 3 px

Annotations shall have precision of at least 3 px, i.e. they shall not differ by more than 3 px from the representation of the physical object unless specified differently by the annotation type.

DAE-903 - Assign sensor-depended attributes with scope Annotation

The attributes with scope Annotation defined in the child requirements shall be assigned to annotations of all classes.

DAE-904 - Annotation Attribute: Occlusion - for all camera and LiDAR annotations

The attribute Occlusion of type Single-Select shall be set for all annotations within camera (visual + infrared) images and LiDAR point clouds. The occlusion shall define the relative size of the portion of the object that is hidden by other objects. Parts of the object that are truncated by image borders shall not be considered as occluded. The following options shall be available for Single-Select

Options	Description
0% - 25%	Less than 25% of the object are occluded.
25% - 50%	Between 25% and 50% of the object are occluded.
50% - 75%	Between 50% and 75% of the object are occluded.
75% - 100%	Between 75% and 100% of the object are occluded.
100%	The whole object is occluded, and the position of the object is known due to interpolation or projection.

The lower bound is included, the upper bound is excluded in the occlusion interval.

DAE-1262 - Annotation Attribute: isTruncated - for all camera annotations

The attribute isTruncated of type Boolean shall be set to true for objects in a camera images if more than 20% of the object are outside the image (i.e. outside the bounding box).

4.1.2 Classes

These are all required classes for annotation with their respective requirements.

4.1.2.1 Person

DAE-908 - Consider humans and human-like dummies

The class Person shall include all humans and human-like dummies.

DAE-905 - Camera: 2D Bounding Box

Persons shall be labelled with a 2D Bounding Box in the visual and infrared camera images. Examples:

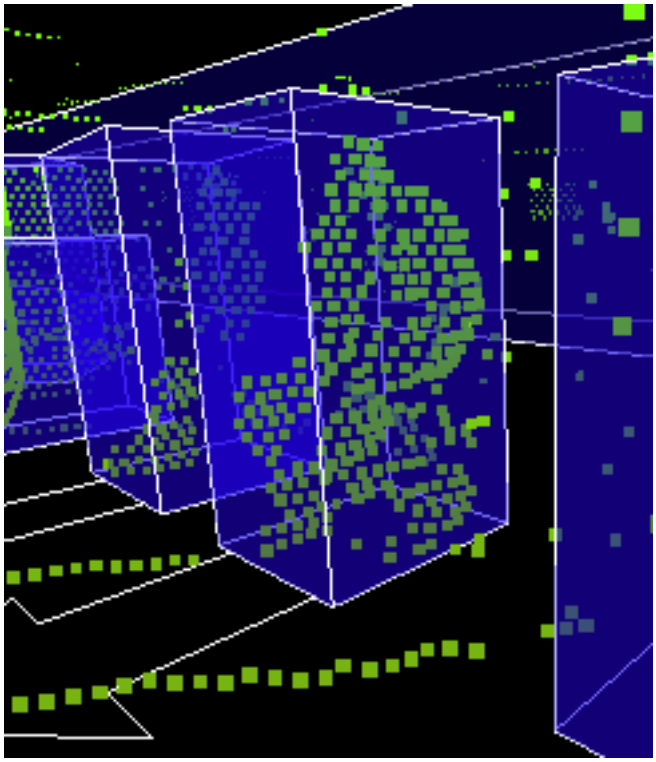
Example for correct annotation of persons in visual cameras

Example for correct annotation of persons in Infrared cameras

DAE-906 - LiDAR: 3D Bounding Box

Persons shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

Example



Example for Person sitting in a 3D Bounding Box

DAE-907 - 3D Bounding Box faces body orientation

The 3D Bounding Box shall face the body orientation of the person, i.e. the rotation around the z-axis shall be set accordingly.

DAE-919 - Radar: 2D Rotated Bounding Box

Persons shall be labelled with a 2D Rotated Bounding Box in the bird's-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-909 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality. The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space). The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame. The scope of each attribute is defined within the attribute's requirement.

DAE-912 - Object attribute: isDummy

The attribute isDummy of type Boolean with the scope Object shall be set for all annotations of class Person. It shall be set to true if the object is not a human, but a human-like dummy.

DAE-913 - Object attribute: age

The attribute age of type Single-Select and scope Object shall be set for all annotations of class persons. The age shall determine whether a person is an adult or a child. The size of a person can be used to estimate the age of a person in case it is not recognisable visually. In this case persons smaller than 1.60 m are considered children.

Option	Description
adult	A person older than 14 years.
child	A person younger than 14 years.

DAE-911 - Frame attribute: function

The attribute function of type Single-Select and scope Frame shall be set for all annotations of class Person. The function shall define the role of the person in the railway context and can be of the following values.

Option	Description
passenger	Passengers are the customers and are using the trains to travel. Persons without any other function are passengers.
worker	Workers are persons working in the railway environment by doing maintenance or cleaning tasks. They are always wearing warning vests or warning cloth, often in orange or yellow.
security	Security staff war a high visibility and protective vest with the letters "Security", "DB Sicherheit" or similar on the back.
staff	Staff is responsible for the rail operations, e.g. a train driver.
uniformed	Persons of public authorities with safety or security tasks, e.g. a police officer, a firefighter, a paramedic or a soldier that wear their uniform.
other	Persons that are clearly no passengers but do not fit in any of the other categories.
unknown	The function of the person cannot be determined, e.g. because the person is only visible in the LiDAR point clouds, but not in camera data.

DAE-914 - Frame attribute: aid

The attribute aid of type Single-Select and scope Frame shall be set for all annotations of class Person. The attribute shall describe the kind of aid used by a person or be None if no walking aid is used.

Option	Description
none	The person is not using any walking aid.
whiteCane	The person is holding or using a white cane.

walkingAid	The person is holding or using a walking aid, such as crutches, walking sticks and rollators.
other	A walking aid is used that is not described by the type of aids above.

DAE-915 - Frame attribute: isDistracted

The attribute isDistracted of type Boolean and scope Frame shall be set for all annotations of class Person. It shall be True when a person is distracted with respect to its environment, e.g., by reading a newspaper or looking at a mobile phone.

DAE-916 - Frame attribute: carrying

The attribute carrying of type Multi-Select and scope Frame shall be set for all annotations of class Person. The attribute shall indicate all items that are carried by the person according to the following options and have no object class of their own in the requirements.

Option	Description
none	The person isn't carrying any objects. This option can't be combined with any of the other options.
backpack	The person is carrying a backpack.
hat	The person is having a headgear
other	The person is carrying any other significant object that does not match any of the other objects.

Explanation: Carrying another person, e.g. a baby, belongs to the attribute connectedTo.

DAE-917 - Frame attribute: pose

The attribute pose of type Single-Select and scope Frame shall be set for all annotations of class Person. The attribute shall describe the current pose of the person with the following options.

Option	Description
upright	The person is in an upright position such as standing or walking.
sitting	The person sitting on an object or on the ground.
lying	The person is lying on an object or on the ground.

DAE-918 - Frame attribute: connectedTo

The attribute connectedTo of type Reference and scope Frame shall be set for all annotations of class Person. The attribute shall reference an object of type Wheelchair,

Bicycle or Motorcycle, when the person is using this object or when the person is carrying an object of type Person (typically a child). The requirements of the referenced classes can contain more information about the definition of the object being used by a person.

4.1.2.2 Personal Item

DAE-1263 - Exclude personal item in crowds

Personal item shall not be labelled in a crowd

DAE-1264 - Camera: 2D Bounding Box

Personal items shall be labelled with a 2D Bounding Box in the visual and infrared camera images.

DAE-1275 - LiDAR: 3D Bounding Box

Personal items shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1308 - 3D Bounding Box faces front of the personal item

The 3D Bounding Box shall face the front of the personal item.

DAE-1337 - Radar: 2D Rotated Bounding Box

Personal items shall be labelled with a 2D Rotated Bounding Box in the bird's-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-1265 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality. The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space). The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame. The scope of each attribute is defined within the attribute's requirement.

DAE-1273 - Frame attribute: connectedTo

The attribute connectedTo of type Reference and scope Frame shall be set for the ObjectIDs of objects belonging to one of the mentioned classes. The attribute shall describe whether the personal item is connected to a person.

DAE-1274 - Object attribute: type

- The attribute type of type Multi-Select and scope Object shall be set for all annotations of class Personal item. The Type shall determine what kind of

personal items are being carried. See below for attribute values: Suitcase Umbrella (unfolded) Others

DAE-1276 - Frame attribute: state

The attribute state of type Single-Select and scope Frame shall be set for all annotations of class personal item. The attribute shall determine if someone is pulling, carrying or whether the object is placed to a next person. See below for attribute values:

Selection	Description
pulled	when an object is pulled (or pushed) by a person
carried	when an object is carried in the arms of a person
alone	when an object is standing without a connection to a person.

4.1.2.3 Pram

This class includes all class of baby carriage such as stroller, buggy and similar.

DAE-1302 - Camera: 2D Bounding Box

Prams shall be labelled with a 2D Bounding Box in the visual and infrared camera images.

DAE-1303 - LiDAR: 3D Bounding Box

Prams shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1309 - 3D Bounding Box faces front of the pram

The 3D Bounding Box shall face the front of the pram.

DAE-1325 - Radar: 2D Rotated Bounding Box

Prams shall be labelled with a 2D Rotated Bounding Box in the bird's-eye view radar images. Projecting the annotations from the LiDAR space is recommended

DAE-1304 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality. The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space). The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame. The scope of each attribute is defined within the attribute's requirement.

DAE-1306 - Object attribute: type

- The attribute type of type Single-Select and scope Object shall be set for all annotations of class Pram. The Type shall determine what kind of baby carriage is defined. See below for attribute values: Stroller Buggy Others

DAE-1307 - Frame attribute: state

The attribute state of type Single-Select and scope Frame shall be set for all annotations of class Pram. The attribute shall determine if someone is pushing or whether the pram is steady, i.e., is standing alone. See below for attribute values:

Selection	Description
carried	when the pram is carried by one or more persons, i.e. at staircases
pushed	when the pram is pushed by a person
steady	when the pram is standing steady

DAE-1305 - Frame Attribute: connectedTo

The attribute connectedTo of type Reference and scope Frame shall be set for the pObjectIDs of objects belonging to one of the mentioned classes. The attribute shall describe whether the Pram is connected to a person.

4.1.2.4 Crowd

A crowd is a group of people that show the same behaviour and that are hard to distinguish from each other, e.g. when the group is far away from the sensor. For example, the same behaviour can be the same walking direction or a close communication with each other.

DAE-548 - Do not consider less than 5 persons as Crowd

Groups of persons having a size of less than 5 overlapping persons shall not be considered as a Crowd.

DAE-1314 - Consider as crowd if 6 or more overlapping persons are visible

A crowd shall consist of people that partially occlude each other (many of those to over 50%). At least 6 persons should overlap each other.

DAE-955 - Camera: 4-Point Polygon

Crowds shall be labelled with a 4-Point Polygon in the visual and infrared camera images. Examples

Annotation example for the object class “Crowd”size: < 25, occlusion: 50-75 %

DAE-963 - LiDAR: 3D Bounding Box

Crowds shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1119 - 3D Bounding Box faces crowd orientation

The 3D Bounding Box shall face the orientation of the crowd, i.e. the body orientation of the majority of persons part of the crowd.

DAE-960 - Radar: 2D Rotated Bounding Box

Crowds shall be labelled with a 2D Rotated Bounding Box in the bird's-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-964 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality. The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space). The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame. The scope of each attribute is defined within the attribute's requirement.

DAE-965 - Frame attribute: size

The attribute size of type Single-Select and scope Frame shall be set for all annotations of class Crowd. The attribute shall describe approximately how many people are members of the crowd. See below for attribute values:

- <25 (default)

- 25-50
- 50-75
- <>75

4.1.2.5 Bicycle

DAE-975 - Consider all types of cycles as bicycles

The class Bicycle shall include all types of bikes, e.g. recumbent bicycles or unicycles.

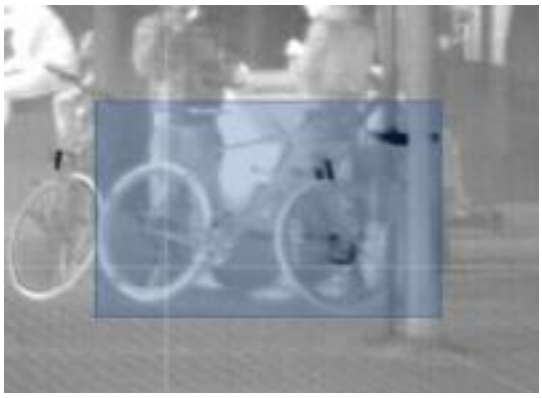
DAE-1117 - Consider parked and moving bicycles

Bicycles shall be labelled in any state, i.e. parked and when in interaction with a person.

DAE-968 - Camera: 2D Bounding Box

Bikes shall be labelled with a 2D Bounding Box in the visual and infrared camera images. Examples

Example for correct annotation of a bicycle in visual cameras



Example for correct annotation of a bicycle in Infrared cameras

DAE-969 - LiDAR: 3D Bounding Box

Bikes shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1118 - 3D Bounding Box faces front of the bicycle

The 3D Bounding Box shall face the front of the bicycle, i.e. heading of the bicycle;

DAE-970 - Radar: 2D Rotated Bounding Box

Bikes shall be labelled with a 2D Rotated Bounding Box in the birds-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-971 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality. The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space). The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame. The scope of each attribute is defined within the attribute's requirement.

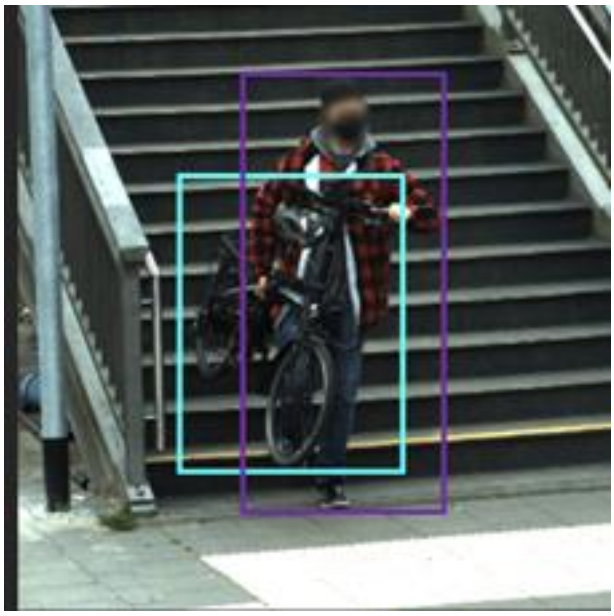
DAE-972 - Frame attribute: state

The attribute state of type Single-Select and scope Frame shall be set for all annotations of class Bicycle. The attribute shall determine if someone is pushing, riding, or carrying the bicycle.

See below for attribute values:

- pushed (default)
- ridden
- carried
- motionless

Example



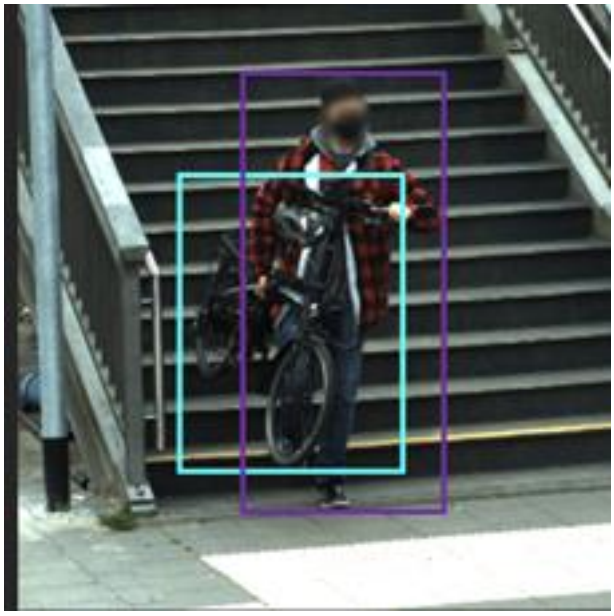
Annotation example for the object class “Bicycle” state: carried. connectedTo: connect the bicycle to the person

Annotation example for the object class “Bicycle” state: pushed. connectedTo: connect the bicycle to the person

DAE-973 - Frame attribute: connectedTo

The attribute connectedTo of type Reference and scope Frame shall be set for all annotations of class Bicycle. The attribute shall determine the person pushing, riding, or carrying the bike.

Examples



Annotation example for the object class “Bicycle” state: carried. ConnectedTo: connect the bicycle to the person

Annotation example for the object class “Bicycle” state: pushed. connectedTo: connect the bicycle to the person

4.1.2.6 Group of Bicycles

DAE-977 - A group of bicycles shall be labelled by an overlapping of more than 50 percent

Multiple bicycles shall be labelled as group of bicycles if there are more than 5 bicycles that overlap by more than 50 percent.

DAE-978 - Camera: 4-Point-Polygon

A group of bicycles shall be labelled with a 4-Point Polygon in the visual and infrared camera images.

Examples

Example for correct annotation of group of bicycles in visual cameras with an occlusion of 50-75 %.

DAE-979 - LiDAR: 3D Bounding Box

A group of bicycles shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1120 - 3D Bounding Box faces front of the group of bicycles

The 3D Bounding Box shall face the front of the bicycle group, i.e direction of the majority of bicycles;

DAE-980 - Radar: 2D Rotated Bounding Box

Group of bicycles shall be labelled with a 2D Rotated Bounding Box in the birds-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-981 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality. The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space). The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame. The scope of each attribute is defined within the attribute's requirement.

Created:	2022-07-28 10:31:07	Reporter:	Betül Sögütlü	Updated:	2022-08-15 14:11:12

DAE-1240 - Frame attribute: size

The attribute size of type Single-Select and scope Frame shall be set for all annotations of class of Group of Bicycles. The attribute shall describe approximately how many bicycles are members of the group.

See below for attribute values:

- <5
- 5-10
- >10

DAE-1241 - Frame attribute: state

The attribute state of type Single-Select and scope Frame shall be set for all annotations of class of Group of Bicycles. The attribute shall describe in what condition and circumstance the bikes within the group are. All bicycles within a Group of Bicycles need to have the same state.

See below for attribute values:

- pushed
- ridden
- carried
- motionless

4.1.2.7 Train

All types of trains and rail vehicles fall into this object class.

DAE-1227 - Consider coupled trains as single train object

If two trains are coupled in such a way that both are controlled by one traction unit, then these coupled trains are treated as one object.

Example

Edge case of connected railcars. In this special case, two regional railcars are connected to each other because of rush hour to have more space for passengers. They are still labelled as one train with one polygon because the second railcar is inactive and is treated as an extension of the first railcar, which is driving.

DAE-1226 - Include coupling device into annotation

The coupling device is always shall be marked as a part of the train.

DAE-1225 - Exclude pantograph

The pantograph of the train shall be excluded from the label of the train.

The pantograph of the train shall be excluded from the label of the train.

DAE-1224 - Consider traction unit as a whole train

If the train traction unit has also space for passengers (combined passenger wagons and traction units), the whole train shall be labelled as “Train”. h2.

If the train traction unit has also space for passengers (combined passenger wagons and traction units), the whole train shall be labelled as “Train”.

DAE-1242 - Consider traction unit without passenger wagon as a locomotive

If the train is a traction unit without an integrated passenger wagon, the traction unit shall be labelled as a “locomotive” and the passenger or freight units belong to the object class “Wagons”.

Traction unit without an integrated passenger wagon is annotated without waggon as locomotive.

DAE-984 - Camera: Two Outline Polygon (train front, whole train)

Trains shall be annotated with two Outline Polygons. The first polygon describes the outline of the whole train. The second polygon describes the outline of the train front.

Example:

Annotation example for the object class “Train” with two separate polygons for the front and the whole body of the train.

DAE-985 - Camera: Outline Polygon (train front)

The train front shall be annotated with an Outline Polygon, with 4 to 20 anchor points. The polygon shall enclose the whole front of the train facing the sensor.

Example

Correct annotation of labelling a train front.

DAE-1326 - Camera: Outline Polygon (whole train)

The whole train shall be annotated with an Outline Polygon.

DAE-997 - Annotation attribute: isFront

The attribute isFront of type Boolean shall be set for all annotations in images of class train. It shall be True for annotations marking the train front (i.e. camera Outline Polygon). The front is always the side facing our cameras, independent of the driving direction. The rear is always the side not facing our cameras.

DAE-986 - LiDAR: 3D Bounding Box

Trains shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1121 - 3D Bounding Box faces train front

The 3D Bounding Box shall face the front of the train.

DAE-1229 - Do not estimate size of trains

In the LiDAR, the real size of a train shall not be estimated. This is an exception to the general annotation rules.

DAE-1228 - Use multiple linked 3D Bounding Boxes if necessary

In the LiDAR, the train shall be labelled by several linked 3D Bounding Boxes, if necessary. This might be the case for example when the train drives through a bending.

DAE-987 - Radar: 2D Rotated Bounding Box

Trains shall be labelled with a 2D Rotated Bounding Box in the birds-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-988 - Sensor-independent attributes (Scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality.

The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space).

The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame.

The scope of each attribute is defined within the attribute's requirement.

DAE-996 - Frame attribute: connectedTo

The attribute connectedTo of type Reference and scope Frame shall be set for the ObjectIDs of objects belonging to one of the mentioned classes. The attribute shall describe whether train is connected to a wagon or to another train object. If several 3D Bounding Boxes are used to describe one train, the Bounding Box transitions shall be between two passenger wagons and the Bounding Boxes shall be linked together via this attribute.

DAE-990 - Object attribute: type

The attribute type of type Single-Select and scope Object shall be set for all annotations of class Trains. The Type shall determine whether a train is either a locomotive (independent traction unit) or a combined traction unit, where passenger wagons at the front and back contain a driver’s cab and the traction unit is integrated into the passenger wagons.

See below for attribute values:

Selection	Description
locomotive	A locomotive is the traction unit of a train, which isn’t supposed to carry passengers. It is usually connected to Wagons that have capacity to transport passengers or goo
intercity	Intercity trains are ICEs and ICs in Germany, where ICEs are always combined traction units and ICs are available either as a combined traction unit or as a combination of locomotive and

	wagons. Intercity trains are usually coloured white with a red or green line in Germany.
regional (default)	Regional trains of the Deutsche Bahn are usually coloured red and either are a combined traction unit or a combination of locomotive and wagons. They are also available in different colours when operated by any other EVUs (railway companies).
commuter	Commuter trains are called S-Bahn or U-Bahn in Germany and are used to transport people within a city. In the dataset S-Bahn trains of Berlin and Hamburg are visible.
construction	Construction vehicles usually coloured yellow and often look different than the other trains as their purpose is to do construction or maintenance work.
other	Other means that an assignment is not possible.

4.1.2.8 Wagons

All types of railway wagons or pulled objects in rail transport belong to this object class.

DAE-1231 - Wagons are often coupled with a train locomotive objects

Wagons can be coupled with locomotives. Then the locomotive shall be labelled individually and the “connectedTo” attribute shall be set to connect the locomotive with the wagons.

DAE-1230 - Consider multiple coupled wagons as one wagons object

Wagons that are coupled shall be considered as one object and labelled as such.

DAE-1000 - Camera: Outline Polygon

Wagons shall be marked as type Outline Polygon, to be labelled with a 2D Polygon with 4 to 20 anchor points. The polygon should describe the basic outline of the wagon.

Examples

Example for correct annotation of a wagon in visual cameras This wagon is not connectedTo a locomotive

DAE-1001 - LiDAR: 3D Bounding Box

Wagons shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1232 - 3D Bounding Box faces wagon front

The 3D Bounding Box shall face the front side of the wagon or opposite direction towards us.

DAE-1002 - Radar: 2D Rotated Bounding Box

Wagons shall be labelled with a Rotated 2D Bounding Box in the birds-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-1003 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality. The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space). The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame. The scope of each attribute is defined within the attribute's requirement.

DAE-1005 - Object attribute: type

The attribute type of type Single-Select and scope Object shall be set for all annotations of class Wagons. The Type shall determine the differentiation among the railway wagons. See below for attribute values:

Selection	Description
intercity	Intercity wagons are part of an IC, usually coloured white with a red line in Germany. For more details see the class “Train”.
regional	If regional wagons are from Deutsche Bahn, they are usually coloured red. If they are from another EVU, they may have any over colour (see Figure 2).
freight (default)	Freight wagons are used to transport goods (see Figure 1).
construction	Construction wagons are part of a construction train and are often coloured yellow. They often differ in their look compared to all other wagons.
other	Other means that an assignment is not possible.

Examples

Annotation example for the object class “Wagons” type: regional. hint: The locomotive and the wagons shall be labelled separately.

Annotation example for the object class “Wagons” type: freight. hint: The locomotive and the wagons shall be labelled separately.

DAE-1004 - Frame attribute: connectedTo

The attribute connectedTo of type Multi-Select and scope Frame shall be set for the ObjectIDs of objects belonging to one of the mentioned class. The attribute shall describe whether wagons are connected to a train with the attribute “locomotive” for setting it correspondingly.

If several 3D Bounding Boxes are used to fit the wagons, the Bounding Box transitions shall be between two wagons and the Bounding Boxes shall be linked together via this attribute.

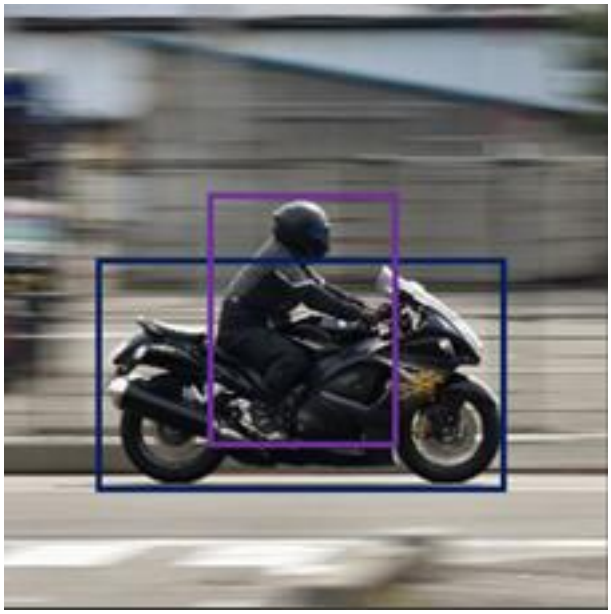
4.1.2.9 Motorcycle

DAE-1009 - Consider all types of motorcycles, mopeds and their electric variants

All types of motorized motorcycles, mopeds and their electric variants belong to this object class.

DAE-1010 - Camera: 2D Bounding Box

Motorcycles shall be labelled with a 2D Bounding Box in the visual and infrared camera images. Example



Annotation example for the object class “Motorcycle”

DAE-1011 - LiDAR: 3D Bounding Box

Motorcycles shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1233 - 3D Bounding Box faces motorcycle front

The 3D Bounding Box shall face the front of the Motorcycle.

DAE-1012 - Radar: 2D Rotated Bounding Box

Motorcycle shall be labelled with a 2D Rotated Bounding Box in the birds-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-1013 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality.

The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space).

The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame.

The scope of each attribute is defined within the attribute's requirement.

DAE-1017 - Frame attribute: connectedTo

The attribute connectedTo of type Multi-Select and scope Frame shall be set for the ObjectIDs of objects belonging to one of the mentioned class. The attribute shall describe

whether the motorcycle is connected to a person or several persons, this attribute is set correspondingly.

4.1.2.10 Road Vehicle

DAE-1018 - Consider all types of road vehicles (except motorcycles)

All types of road vehicles shall be labelled as "Road Vehicles". Including for example trailer. Road Vehicles include combustion and electric engines as well as no engine such as a trailer.

DAE-1234 - Do not label persons within road vehicles

Persons inside a road vehicle shall be ignored in the labelling process.

DAE-1019 - Camera: 2D Bounding Box

Road Vehicles shall be labelled with a 2D Bounding Box in the visual and infrared camera images.

DAE-1020 - LiDAR: 3D Bounding Box

Road Vehicles shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1235 - 3D Bounding Box faces front of the road vehicle

The 3D Bounding Box shall face the front of the road vehicle.

DAE-1021 - Radar: 2D Rotated Bounding Box

Road Vehicles shall be labelled with a 2D Rotated Bounding Box in the birds-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-1022 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality. The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space). The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame. The scope of each attribute is defined within the attribute's requirement.

DAE-1026 - Frame attribute: connectedTo

The attribute connectedTo of type Multi-Select and scope Frame shall be set for all annotations of class Road Vehicles. The attribute shall determine whether a road vehicle

is connected with another road vehicle, e.g. a trailer is connected to a truck, this attribute is set correspondingly.

Example

Annotation example for the object class “Road Vehicle” type: truck connectedTo: trailer

DAE-1025 - Object attribute: type

The attribute type of type Single-Select and scope Object shall be set for all annotations of class Trains. The Type shall determine the base of the classification is the common understanding of road vehicles. The following guidelines can help to categorize vehicles.

Selection	Description
car (default)	A car has room for two to five persons.
van	A van has room for six to 10 persons or larger objects.
truck	A truck is meant to pull larger trailers
bus	A bus has room for more than 10 persons.
constructionVehicle	construction vehicle has a special building or reparation function.
trailer	a non-automotive vehicle designed to be hauled by road
other	other is the category for road vehicles which do not fit in one of the other categories like squads.

Examples

Two road vehicle: truck and trailer which are connected to each other.

Road vehicle of type: constructionVehicle.

Road vehicle of type: car.

Road vehicle of type: van.

4.1.2.11 Animal

DAE-1027 - Consider only animals bigger than a cat

In this class, all animals which are bigger than a cat shall be labelled.

DAE-1313 - Labelling dogs regardless of the size

Dogs shall be labelled in all sizes without being excluded.

DAE-1028 - Camera: 2D Bounding Box

Animals shall be labelled with a 2D Bounding Box in the visual and infrared camera images.

DAE-1029 - LiDAR: 3D Bounding Box

Animals shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1236 - 3D Bounding Box faces body orientation

The 3D Bounding Box shall face the body orientation of an animal, i.e. the face front.

DAE-1030 - Radar: 2D Rotated Bounding Box

Animals shall be labelled with a 2D Rotated Bounding Box in the birds-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-1031 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality.

The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space).

The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame.

The scope of each attribute is defined within the attribute's requirement.

DAE-1033 - Object attribute: size

The attribute size of type Single-Select and scope Object shall be set for all annotations of class animals. The size shall determine whether the potential risk to the train in the event of a collision with the respective animal may occur.

small animals pose a small risk, medium animals a medium risk and large animals a high risk (see Figure 1).

Long and flat animals such as snakes are not dangerous for the train despite their length, as they are flat, which means that the train could easily roll over them and are classified as “small”. To get an idea of this attribute, the following examples are given:

Size selection	Reference
small	Cat
	Badger
	Racoon
medium	Sheep

	Dog
	Stork
	Swan
	Goat
	Lynx
	Fox
	Wolf
large	Deer
	Cow
	Horse
	Deer
	Pig
	Boar

Examples:

Correct annotation suited to a medium-sized animal, i.e. Dog in Bounding Box

DAE-1035 - Frame attribute: pose

The attribute pose of type Single-Select and scope Frame shall be set for all annotations of class Animal. The attribute shall determine the pose of the animal.

Attribute value:

- upright (default)
- sitting

- lying
- other

DAE-1329 - Object attribute: species

The attribute species of type Single-Select and scope Object shall be set for all annotations of class animals. The size shall determine the type of animal which is detected

Attribute values:

- dog (default)
- deer
- fox
- rabbit
- wildBoar
- cow
- bird
- otherAnimal

4.1.2.12 Group of Animals

DAE-1278 - Do not consider less than 5 animals as Group of animals

Group of animals having a size of less than 5 overlapping animals shall not be considered as a group of animals.

DAE-1036 - A group of animals shall be labelled by an overlapping of more than 50 percent

This class is used for a group of animals that occlude each other to a certain extent, i.e. overlapping scenario by more than 50 percentage. And have similar behavioural patterns, e.g., a herd of cows grazing together.

DAE-1037 - Camera: 2D Bounding Box

A group of Animals shall be labelled with a 2D Bounding Box in the visual and infrared camera images.

Example

Annotation example for the object class "Group of Animals" occlusion 0-25%

DAE-1038 - LiDAR: 3D Bounding Box

The group of animals shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1237 - 3D Bounding Box faces body orientation

The 3D Bounding Box shall face the body orientation or the direction where the majority of animals in the group of animals is directing to.

DAE-1039 - Radar: 2D Rotated Bounding Box

A group of animals shall be labelled with a 2D Rotated Bounding Box in the birds-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-1243 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality. The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space). The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame. The scope of each attribute is defined within the attribute's requirement.

DAE-1246 - Frame attribute: pose

The attribute pose of type Single-Select and scope Frame shall be set for all annotations of class Animal. The attribute shall determine the pose of the animal. Attribute value:

- upright (default)
- sitting
- lying

- other

DAE-1245 - Object attribute: species

The attribute species of type Single-Select and scope Object shall be set for all annotations of class group of animals. The size shall determine the type of animal which is detected.

Attribute values:

- dog (default)
- deer
- fox
- rabbit
- wildBoar
- cow
- bird
- otherAnimal

DAE-1244 - Object attribute: size

The attribute size of type Single-Select and scope Object shall be set for all annotations of class animals. The size shall determine whether the potential risk to the train in the event of a collision with the respective animal may occur;

small animals pose a small risk, medium animals a medium risk and large animals a high risk (see Figure 1 and Figure 2).

Long and flat animals such as snakes are not dangerous for the train despite their length, as they are flat, which means that the train could easily roll over them and are classified as “small”. To get an idea of this attribute, the following examples are given:

Selection	Description
Dog	(small-medium)
Deer	(Large)
Fox	(small)
Rabbit	(small)
Wild Boar	(medium)
Bird	(small-medium)
other Animals	(small-large)

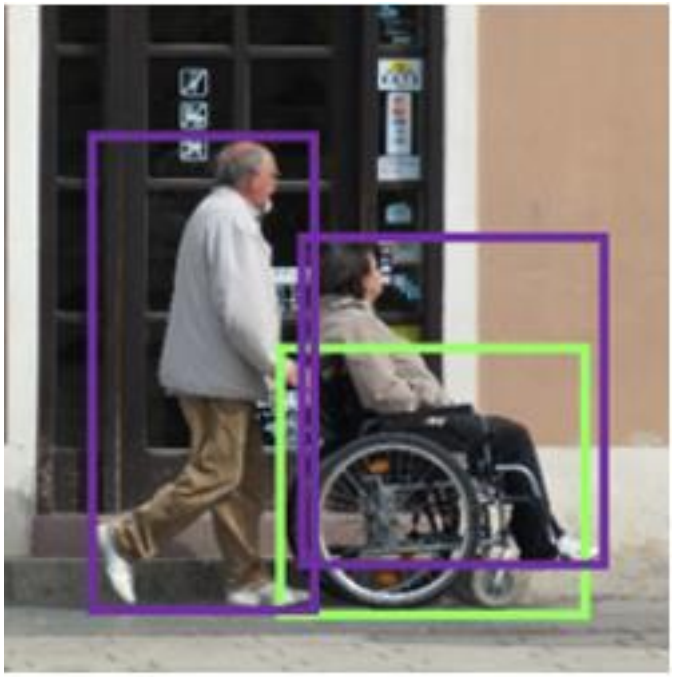
4.1.2.13 Wheelchair

All wheelchairs and their electric variants belong to this object class.

DAE-1041 - Camera: 2D Bounding Box

Wheelchairs shall be labelled with a 2D Bounding Box in the visual and infrared camera images.

Example



Annotation example for the object class “Wheelchair”

DAE-1042 - LiDAR: 3D Bounding Box

A wheelchair shall be labelled with a *3D Bounding Box* in the *LiDAR* point cloud.

A wheelchair shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1238 - 3D Bounding Box faces front of the wheelchair

The 3D Bounding Box shall face the front of the wheelchair, i.e. The front is the direction the wheelchair moves when the wheels are rotated and the seat direction.

DAE-1043 - Radar: 2D Rotated Bounding Box

The wheelchair shall be labelled with a 2D Rotated Bounding Box in the birds-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-1044 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality.

The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space).

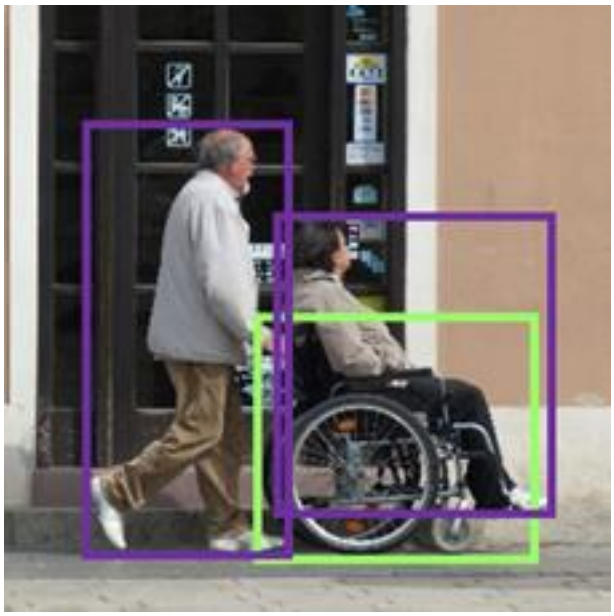
The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame.

The scope of each attribute is defined within the attribute's requirement.

DAE-1045 - Frame Attribute: connectedTo

The attribute connectedTo of type Multi-Select and scope Frame shall be set for all annotations of class wheelchair. The attribute shall determine whether a person can sit in the wheelchair, or a person can push the wheelchair.

Example



connectedTo: connect the wheelchair with the person sitting in and the person pushing the wheelchair

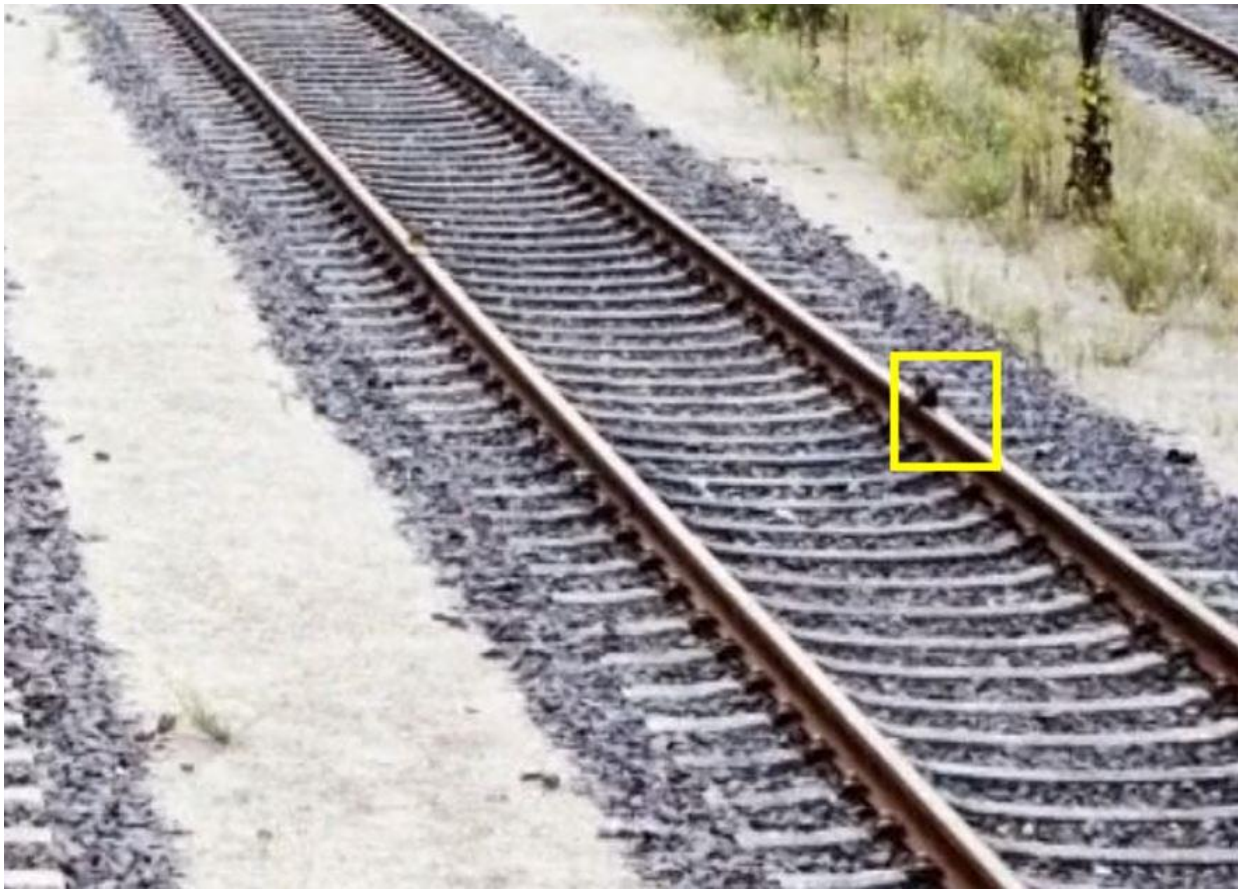
4.1.2.14 Drag Shoe

A drag shoe is used to hold the wheels in place during the train stands and to secure trains during maintenance.

DAE-1123 - Camera: 2D Bounding Box

Drag Shoes shall be labelled with a 2D Bounding Box in the visual and infrared camera images.

Example:



Correct annotation of a drag shoe

DAE-1124 - LiDAR: 3D Bounding Box

A Drag shoe shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1239 - 3D Bounding Box faces front of the Drag Shoe

The 3D Bounding Box shall face the front of the Drag Shoe, i.e. the view to the support bracket.

DAE-1126 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality. The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space). The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame.

The scope of each attribute is defined within the attribute's requirement.

DAE-1128 - Frame attribute: onTrack

onTrack shall be of type Reference and contain the object id of the track on which the drag shoe is on. In case that track is not labelled, or the drag shoe is next to the tracks, this field should remain empty.

DAE-1127 - Frame attribute: railSide

Determines the rail side, the drag shoe is located on. If the drag shoe is not on a rail, then it shall be labelled as “none”.

single select, scope frame:

Attribute values:

leftRail (default)

rightRail

none

4.1.2.15 Track

DAE-1315 - Each track consists of two rails

Each track consists of two rails which shall be considered for labelling and be assigned to the same object.

DAE-1134 - Label tracks as long as visible

Tracks shall be labelled as long as they are visible in the images or contain points in the point clouds.

DAE-1330 - Annotate rails without interruption

Rails that are covered by a pole, sign, or objects shall be labelled without interruption when the route can be estimated.

DAE-1133 - Split track annotations when route is not identifiable due to large occlusion

If very large parts of a track are covered and the track is visible before and after the occlusion, the track annotation shall be split only if the track course is not identifiable at the covered part. All annotations that label parts of the track shall belong to the same track object, i.e. have the same object id.

DAE-1132 - Consider tracks underneath a train

Tracks underneath a train shall also be labelled if they are visible or the track course can be estimated.

DAE-1319 - Track course is not affected by switch positions

The course of a track shall be the straightest possible extension and shall not be affected by switch positions.

DAE-1320 - Label only selected tracks

Only the selected tracks listed in the sub requirements shall be labelled.

DAE-1130 - Label Ego-Track

The ego-track, i.e. the track this train is driving on, shall be labelled.

DAE-1318 - Change of Ego-Track

The Ego-Track shall change to another track as soon as the train has taken a switch. The LiDAR can be used as reference as it shows the track directly in front of the train.

DAE-1131 - Label two left and right neighbouring tracks

The two left and two right neighbouring tracks of the Ego-Track shall be labelled if they exist and are visible.

DAE-1321 - Do not stop labelling tracks that are labelled in any other frame

Tracks that are labelled because they are either the Ego-Track or one of the two neighbouring tracks on either side in any of the frames shall be labelled in the whole sequence, even though they do not meet the criteria of being ego- or first/second neighbouring track anymore.

DAE-1135 - Camera: 2D Polyline

The outer edge of both rails of a Track shall be labelled with a 2D Polyline each in the visual and infrared camera images.

Labelled outer edges. On the left and right rail.

DAE-1150 - Annotate the outer edge of rail head

The 2D Polyline shall follow the outer edge of the rail head.

DAE-1247 - Support polyline with enough anchor points on curved tracks

The number of anchor points shall be selected in order to follow the shape of the object precisely. Especially on curved tracks the 2D Polyline shall follow the outer edge of the rails tightly.

DAE-1129 - Consider tracks with a single visible rail as tracks

Tracks that have only one visible rail shall also be labelled as track. The occlusion for the track must be set accordingly, i.e. >50%.

DAE-1249 - Annotation attribute: railSide

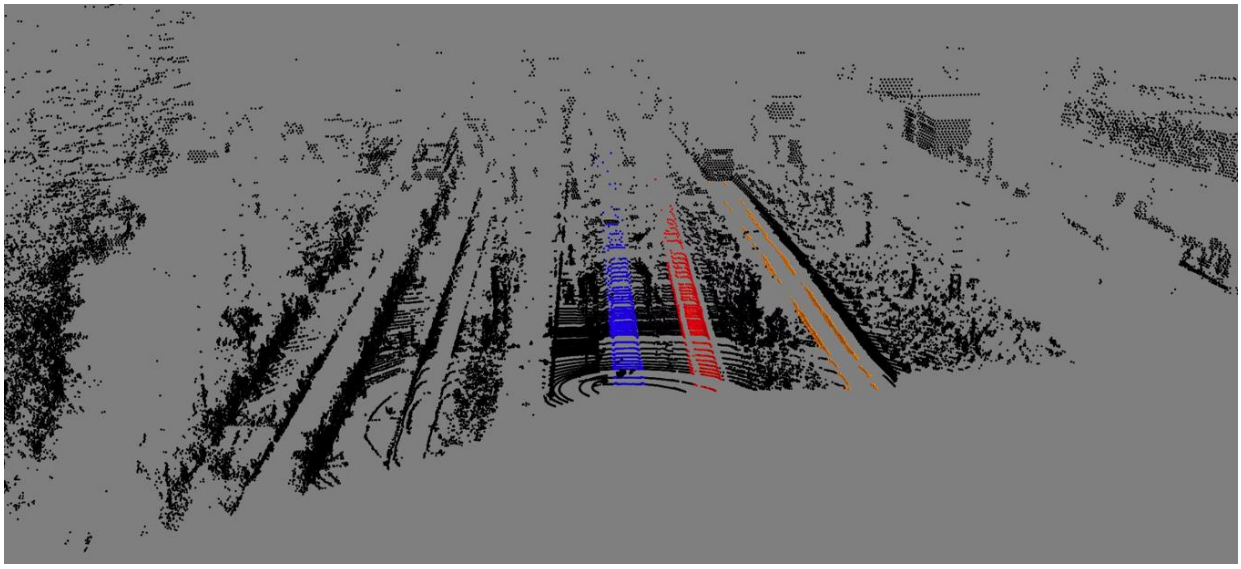
The attribute railSide of type Single-Select with the scope Annotation shall be set for all annotations of class Track. It shall be set according to the side of the rail on the respective track bed. Attribute values:

- leftRail
- rightRail

DAE-1136 - LiDAR: 3D Semantic Segmentation

Tracks shall be labelled with a 3D Semantic Segmentation in the LiDAR point cloud.

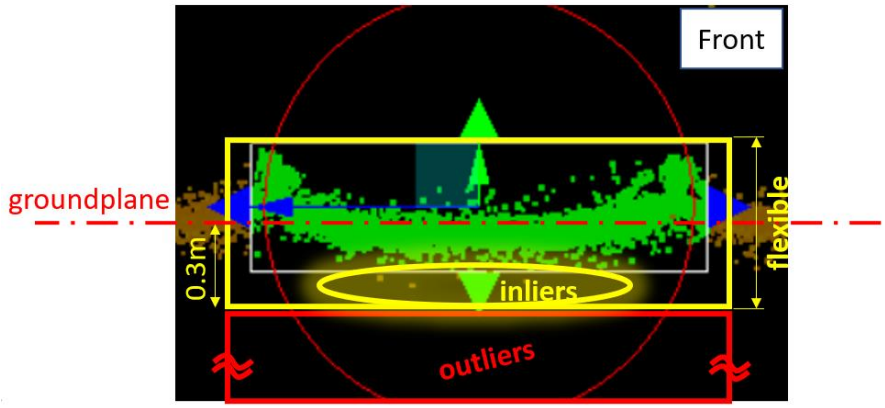
Example



3D Semantic Segmentation of 3 rails (green, red, orange)

DAE-1316 - Segment track bed between outer fasteners

In the LiDAR point cloud, the track bed shall be segmented between the outer fastener (screws) that connect both rails to the sleeper.



Example for area to be labelled in the 3D point cloud. The yellow Box shows the area to be considered, Viewed from the front.

DAE-1317 - Segment only points belonging to the track

The track shall be segmented underneath objects (i.e. trains). Nevertheless, only the points belonging to the track and not to the object shall be labelled.

DAE-1248 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality.

The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space).

The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame.

The scope of each attribute is defined within the attribute's requirement.

DAE-1250 - Frame attribute: isEgoTrack

The attribute isEgoTrack of type Boolean with the scope Frame shall be set for all annotations of class track. It shall be set to true if the test train is driving on the track during the sequence. The other tracks on the left-hand side and right-hand side shall be selected as: false

4.1.2.16 Transition

DAE-1322 - Transitions connect two continuous tracks

Transitions are short tracks that connect two continuous (usually parallel) tracks and are used to switch between those tracks.

DAE-1323 - Consider Track requirements for labelling Transitions

Annotation rules of the class Track shall be applied when labelling Transitions.

DAE-1142 - Ignore transitions that originate or end in unlabelled tracks

Transitions that originate or end in unlabelled tracks shall not be labelled.

DAE-1143 - Labelling transitions until connection to regular trail

The transitions shall be labelled until they connect with a regular rail.

The transitions shall be labelled until they connect with a regular rail.

DAE-1144 - Camera: 2D Polyline

Transitions with type 2D Polyline shall be labelled with a line with multiple anchor points. The number of anchor points shall be selected in order to follow the shape of the object precisely. Especially, curves shall be supported with enough anchor points.

DAE-1251 - Start and end transitions on switch

The rails of the transition shall be labelled from switch to switch. The polylines shall start and end at the beginning or end of the switch blades. The annotations on both rail sides shall start and end on the same horizontal level. Example:

Annotation example for Transitions

Annotation example for Transitions

DAE-1148 - Annotation attribute: railSide

The attribute railSide of type Single-Select with the scope Annotation shall be set for all annotations of class Transition. It shall be set according to the side of the rail on the respective track bed. See below for attributes:

- leftRail

- rightRail

DAE-1145 - LiDAR: 3D Semantic Segmentation

Transitions shall be labelled with a 3D Semantic Segmentation in the LiDAR point cloud.

DAE-1149 - Segment track bed of transition between outer fasteners

In the LiDAR point cloud the track bed of the transition shall be segmented between the outer fastener (screws) that connect both rails to the sleeper, c.f. track requirements.

DAE-1146 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality. The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space). The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame.

The scope of each attribute is defined within the attribute's requirement.

DAE-1147 - Object attribute: startTrack

The attribute startTrack of type Reference and scope Object shall be set the object id of the track from which the Transition originates. The transition always originates on the end closer to the test train.

DAE-1324 - Object attribute: endTrack

The attribute endTrack of type Reference and scope Object shall be set the object id of the track in which the Transition ends. The transition always ends on the end farer away from the test train.

4.1.2.17 Switch

DAE-1153 - Annotate switches on labelled tracks

Switches shall only be labelled if they are on the tracks which are labelled.

Switches shall only be labelled if they are on the tracks which are labelled.

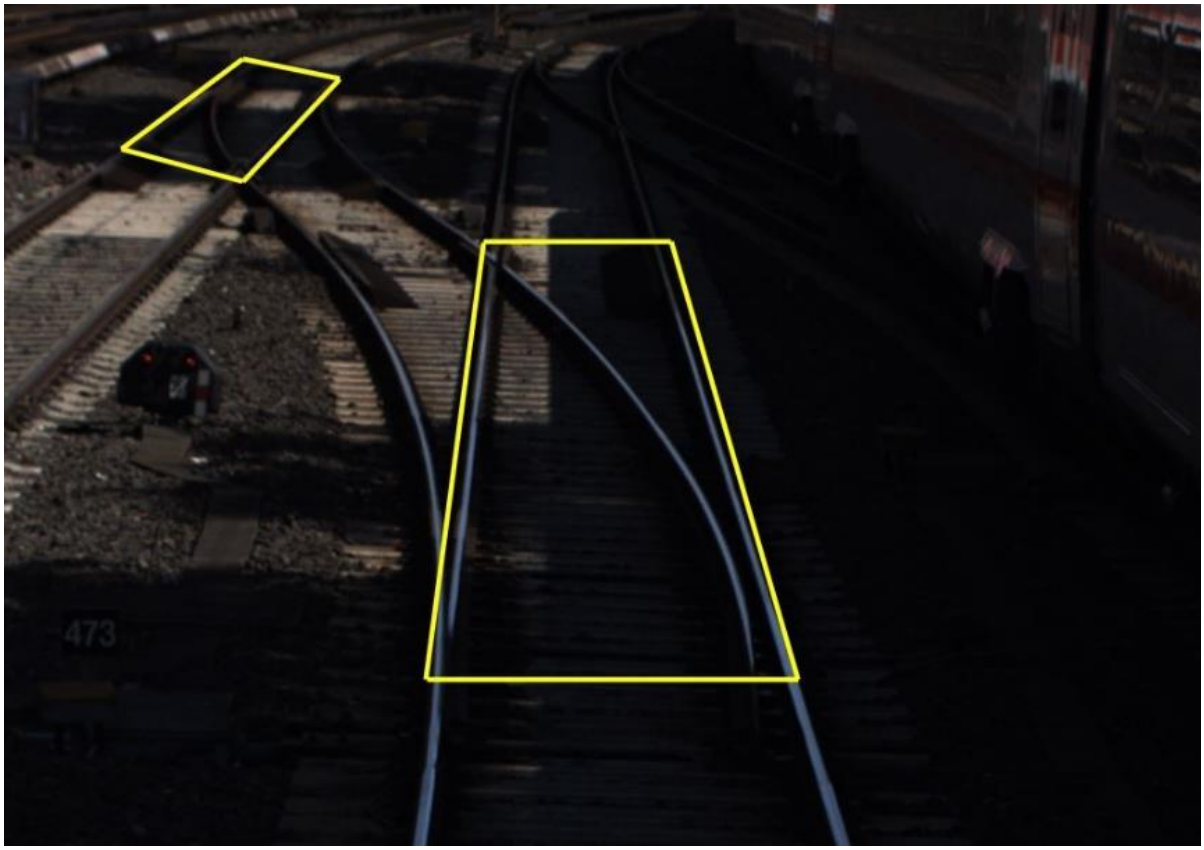
DAE-1155 - Camera: 4-Point Polygon

Switches shall be labelled with a 4-Point Polygon in the visual and infrared camera images.

Switches shall be labelled with a 4-Point Polygon in the visual and infrared camera images.

DAE-1157 - Consider enclosing switch from the tip of the switch toes to the switch frog

In the cameras, the 2D Bounding Box shall enclose the switch from the tip of the switch toes to the switch frog.



Correct annotation of switches

DAE-1156 - Consider involved rails in the surrounding

In the cameras, the 2D Bounding Box shall surround the involved rails

DAE-1158 - LiDAR: 3D Bounding Box

Switches shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

Switches shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1159 - 3D Bounding Box faces front of the switch

The 3D Bounding Box shall face front of the switch, meaning the side where the train drives into the switch.

DAE-1255 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality. The attributes with scope

Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space).

The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame.

The scope of each attribute is defined within the attribute's requirement.

DAE-1256 - Frame attribute: state

The attribute state of type Single-Select with the scope Frame shall be set for all annotations of class Switch. It determines if the switch blade rests against the left rail, right rail or none of the rails. The state is always determined from the view of the train.

Attribute values:

- leftRail
- rightRail
- noRail
- unknown

DAE-1257 - Object attribute: onTrack

The attribute onTrack of type Reference with the scope Object shall be set for all annotations of class Switch. It shall be set to the object id of the track on which the switch placed.

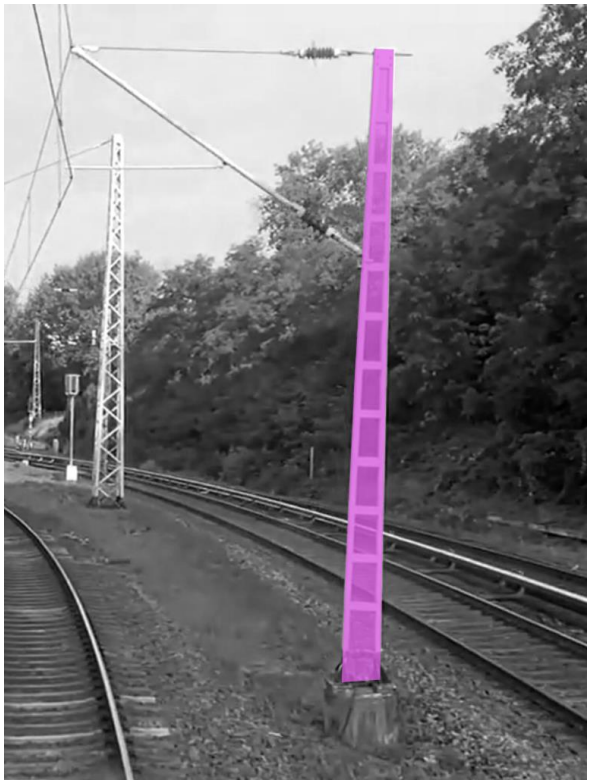
4.1.2.18 Catenary Pole

Catenary Poles hold the overhead lines that are used by railway systems to supply the locomotives with traction current.

DAE-1258 - Exclude catenary pole foundation from the labelling

Foundation of catenary pole shall be excluded from annotation geometry.

Example

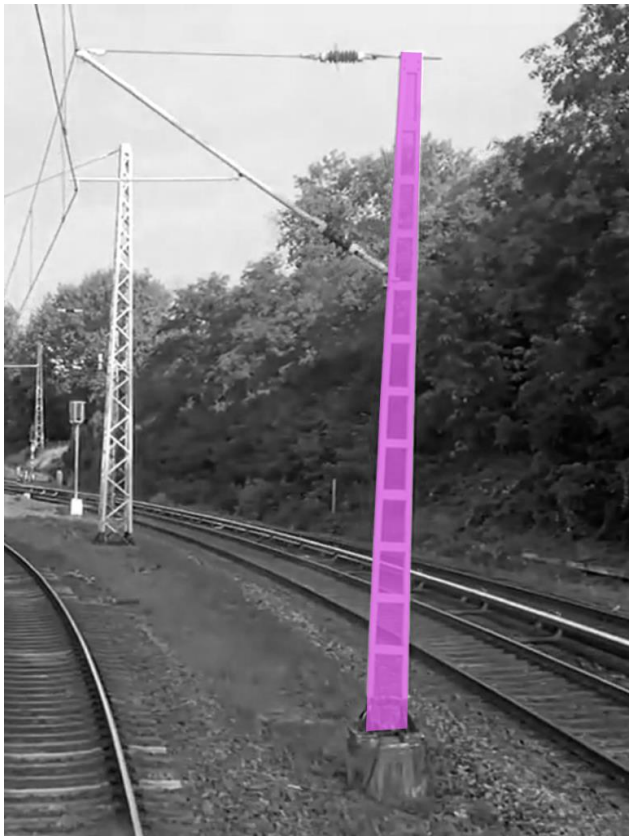


Correct annotation of a structured catenary pole (see the excluded foundation)

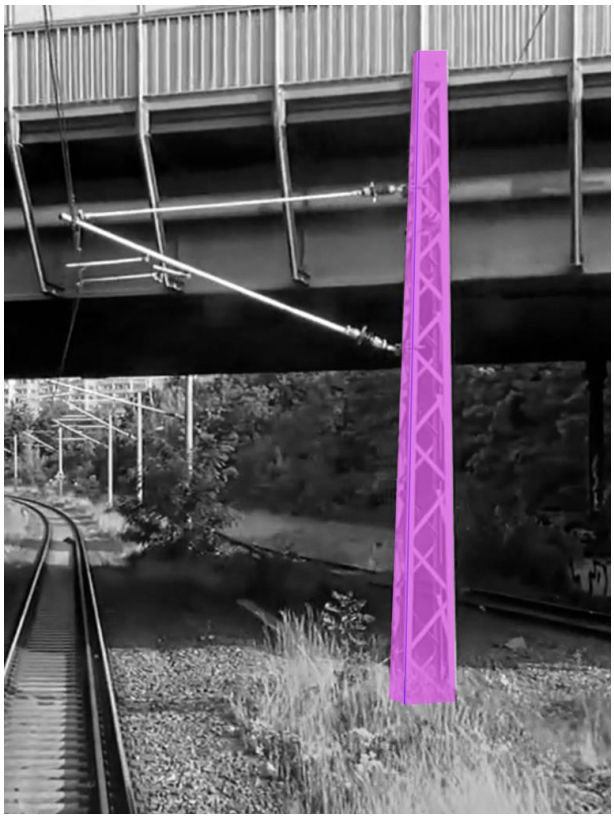
DAE-1163 - Camera: Outline Polygon

Catenary Poles shall be marked as type Outline Polygon, they are to be labelled with a 2D Polygon with 4 to 20 anchor points. The polygon should describe the basic outline of the catenary pole.

Example:



Correct annotation of a structured catenary pole (see the excluded foundation)



Correct annotation of a structured catenary pole (labelling ends on the blade of grass, i.e., foundation excluded)

DAE-1164 - LiDAR: 3D Bounding Box

Catenary Poles shall be labelled with a *3D Bounding Box* in the *LiDAR* point cloud.

Catenary Poles shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1165 - 3D Bounding Box faces front of the catenary pole

The 3D Bounding Box shall face the arm of the catenary pole. If the catenary pole has two arms, it should point to the arm closer to the train.

DAE-1171 - Estimation of size

In the LiDAR, for estimating the true size of a catenary pole, the biggest catenary pole within the sequence shall be chosen as reference.

DAE-1166 - Radar: 2D Rotated Bounding Box

Catenary poles shall be labelled with a 2D* *Rotated Bounding Box in the bird's-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

Catenary poles shall be labelled with a 2D Rotated Bounding Box in the bird's-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-1167 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality. The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space). The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be *consistent across all annotations within a multi-modal frame. The scope of each attribute is defined within the attribute's requirement.

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality.

The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space).

The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame.

The scope of each attribute is defined within the attribute's requirement.

DAE-1169 - Object attribute: type

The attribute type of type Single-Select and scope Object shall be set for all annotations of class catenary pole. The type shall determine whether the pole is structured or of solid material.

See below for attribute values:

- solid (default)
- structured

Examples

Annotation example for the object class “Catenary pole” type: solid

Annotation example for the object class “Catenary pole” type: structured

DAE-1260 - Frame attribute: isBottomVisible

The attribute isBottomVisible of type Boolean with the scope Frame shall be set for all annotations of class catenary poles. It shall be set to true if the bottom is visible.

4.1.2.19 Signal Pole

DAE-1259 - Excluding the signal pole foundation from the labelling

Foundation of signal pole shall be excluded from annotation geometry.

Examples:

Annotation of a signal pole (view as a whole)

The same picture with a close up of an excluded foundation

DAE-1173 - Separate the signal from the signal pole

The signal itself shall not extend the signal pole label. Signals on top of the top of the signal pole are to be excluded, and signals which are not situated on the top shall count as occlusions. Example Fig. 1: display of a signal pole annotation with an occlusion of <25%. |

The signal itself shall not extend the signal pole label. Signals on top of the top of the signal pole are to be excluded, and signals which are not situated on the top shall count as occlusions.

Example:

Display of a signal pole annotation with an occlusion of <25%.

DAE-1174 - Camera: Outline Polygon

Signal Poles shall be marked as type Outline Polygon, they are to be labelled with a 2D Polygon with 4 to 20 anchor points. The polygon should describe the basic outline of the signal pole.

DAE-1175 - LiDAR: 3D Bounding Box

Signal poles shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1176 - 3D Bounding Box faces front of the signal pole

The 3D Bounding Box shall face the visual pointing direction of the signal pole.

DAE-1177 - Radar: 2D Rotated Bounding Box

Signal poles shall be labelled with a 2D Rotated Bounding Box in the bird's-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-1178 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality.

The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space).

The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame.

The scope of each attribute is defined within the attribute's requirement.

DAE-1180 - Object attribute: type

The attribute type of type Single-Select and scope Object shall be set for all annotations of class Signal pole. The Type shall determine, if the signal pole is structured or of solid material.

See below for attribute values:

- solid
- structured (default)

Examples

Annotation example for the class "Signal Pole" type: structured

Annotation example for the class "Signal Pole" type: solid

DAE-1179 - Frame attribute: connectedTo

The attribute connectedTo of type Multi-Select and scope Frame shall be set for all annotations of class signal pole. The attribute shall indicate which analog or light signals are connected to the signal pole. ObjectIDs of objects belonging to one of the mentioned classes.

DAE-1261 - Frame attribute: isBottomVisible

The attribute isBottomVisible of type Boolean with the scope Frame shall be set for all annotations of class signal poles. It shall be set to true if the bottom of the signal pole is visible.

4.1.2.20 Signal

Signals regulate traffic in rail systems and are either shape or light signals.

DAE-1339 - Signals regulate traffic in rail systems

Signals regulate traffic in rail systems and are either shape or light signals.

DAE-1183 - Positioning of the Signals

Signals are mounted on top of a signal pole or signal bridge, or they are placed on the signal pole or signal bridge.

DAE-1186 - Annotate different signals on one signal pole separately

Different signals on the same signal pole shall be labelled separately.

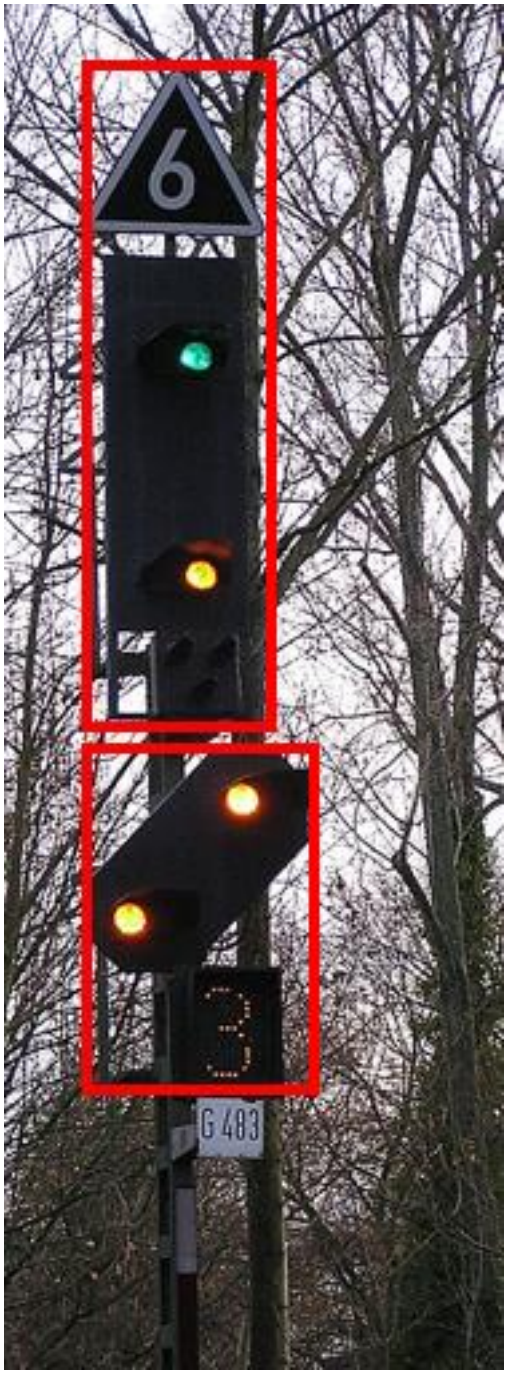
Example:

Two light signals with additional attachments are annotated separately

DAE-1343 - Include attachments to a signal in the bounding box

Light signals often have further attachments (i.e. signalling the speed) below or above the signal. These shall be integrated in the 2D bounding box and be displayed via an attribute.

Examples:



Two signals with attachments. The upper signal has two attachments at the up- and downside. The lower signal has one attachment at the downside.

Light signal with attachments at the up- and downside.

DAE-1188 - Black background part of the signal

The black background of the signal is part of the annotation box.

DAE-1185 - Annotate signals from the front and the back

Signals visible from the front face and from the back shall be labelled.

DAE-1189 - Camera: 2D Bounding Box

Signals shall be labelled with a 2D Bounding Box in the visual and infrared camera images.

Example:

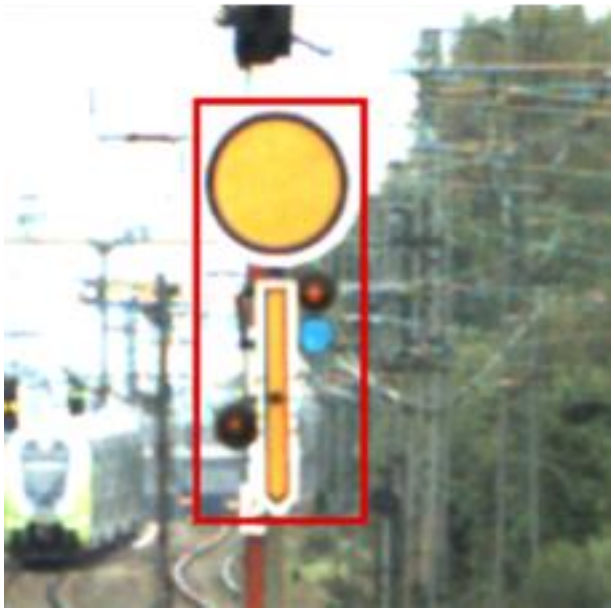
Light signal without attachments.

Light signal without attachments.

DAE-1187 - Light signals used to enhance recognizability of shape signals

Light signals that are used to enhance the recognizability of shape signals at night shall be included within the 2D Bounding Box.

Example



Correct annotation of included enhancing light signals

Correct annotation of included enhancing light signals

DAE-1190 - LiDAR: 3D Bounding Box

Signals shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1191 - 3D Bounding Box faces front of the signal

The 3D Bounding Box shall face the viewing direction of the signal.

DAE-1192 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality.

The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space).

The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame.

The scope of each attribute is defined within the attribute's requirement.

DAE-1193 - Object attribute: type

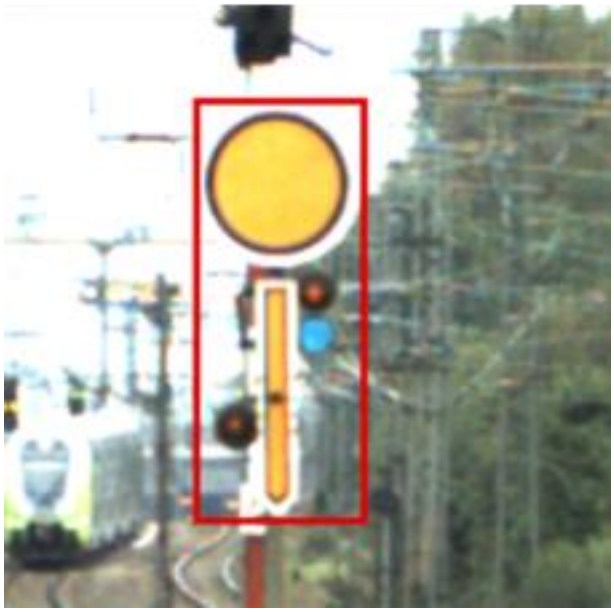
The attribute type of type Single-Select and scope Object shall be set for all annotations of class Signal pole. The Type shall describe the type of the signal. Hence, shape or light signal. Shape signals always have a plate form at the signal.

The shapeVorsignal always consists of a round yellow table and a yellow arm. Both can have several positions. The shapeHauptsignal always has one or more red and white arms.

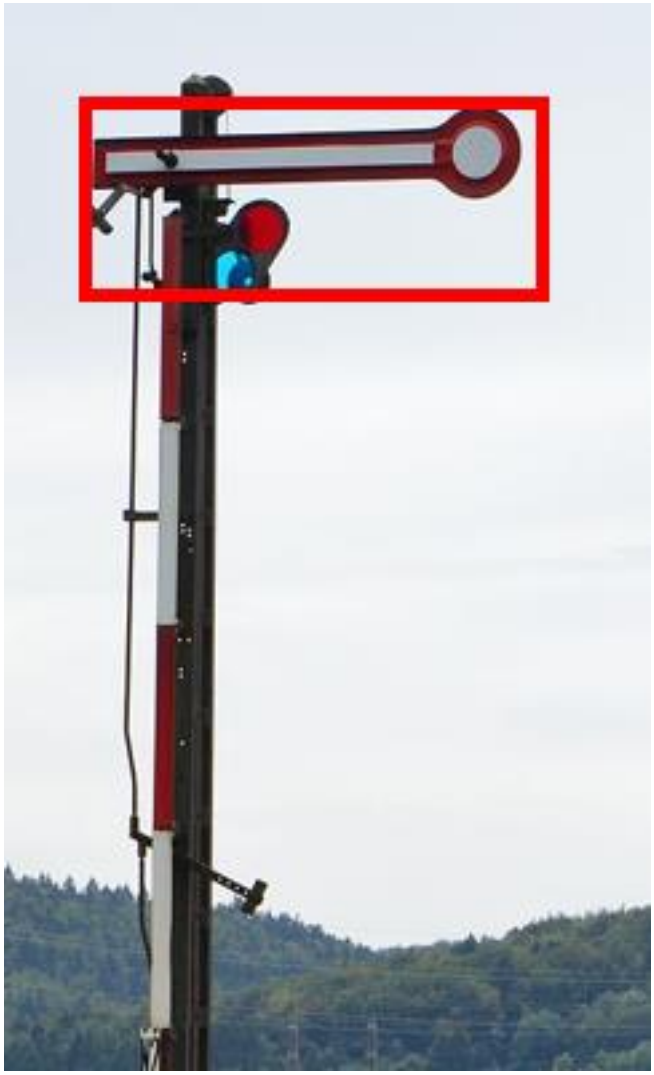
Attribute values:

- light (default)
- shapeVorsignal
- shapeHauptsignal

Examples:



shapeVorsignal



shapeHauptsignal

DAE-1194 - Frame attribute: signalFace

The attribute signalFace of type Single-Select and scope Frame shall be set for all annotations of class signals. The attribute shall describe whether the signal can be seen from the front or from behind. “Unknown” shall be used when a decision is not possible. Tracking-IDs of objects belonging to one of the mentioned classes.

Attribute values:

- front (default)
- back
- unknown

DAE-1195 - Frame attribute: connectedTo

The attribute connectedTo of type Multi-Select and scope Frame shall be set for all annotations of class signal. The attribute shall determine to which pole or signal bridge the signal is connected to with the ObjectIDs of objects belonging to one of the mentioned classes.

DAE-1344 - Object attribute: signalAttachments

The attribute signalAttachment of type Multi-Select and scope Object shall be set for all annotations of class Signal.

Light signals often have attachments added to the main signal. These attachments can be placed on the left, right, lower or upper side of the signal.

Attribute values:

- upside
- downside
- leftside
- rightside
- none (default)
- unknown

4.1.2.21 Signal Bridge

Signal bridges are horizontal structures that protrude from the signal pole over the rail system and to which the signals for the relevant section are attached.

DAE-1196 - Camera: 4-Point-Polygon

Signal bridges shall be labelled with a 4-Point Polygon in the visual and infrared camera images.

Example

correct annotation of a signal bridge

Another example of correct annotation of a signal bridge

DAE-1197 - LiDAR: 3D Bounding Box

Signal bridges shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1198 - 3D Bounding Box faces the visual direction of the signals mounted on

The 3D Bounding Box shall face the visual direction of the signals mounted on the bridge.

DAE-1199 - Radar: Rotated 2D Bounding Box

Signal bridges shall be labelled with a 2D Rotated Bounding Box in the bird's-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-1200 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality.

The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space).

The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame.

The scope of each attribute is defined within the attribute's requirement.

DAE-1203 - Frame attribute: connectedTo

The attribute connectedTo of type Multi-Select and scope Frame shall be set for all annotations of class signal bridge. The attribute shall contain the ObjectIDs of the signals and poles the signal bridge is connected to.

DAE-1201 - Object attribute: type

The attribute Type of type Single-Select and scope Object shall be set for all annotations of class Signal pole. The Type shall determine if the signal bridge is structured or of solid material.

Attribute values:

- solid
- structured (default)

4.1.2.22 Buffer Stop

A buffer stop marks the end of a track and is therefore a sign to halt for a train.

DAE-1204 - Sign of the buffer stop inside of the annotation

Several buffer stops may have a sign on them or on top of them. The sign of the buffer stop shall be inside the label.

DAE-1336 - Label all buffer stops

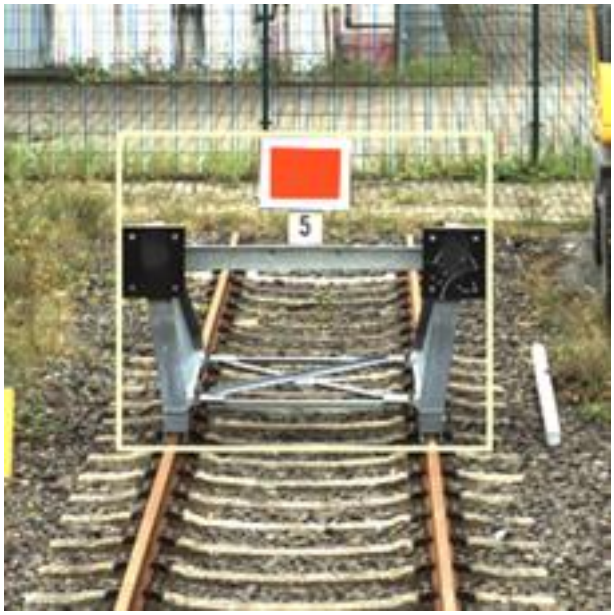
All buffer stops shall be labelled. This means, buffer stops that are not on a labelled track shall be labelled as well.

DAE-1205 - Cameras: 2D Bounding Box

Buffer stops shall be labelled with a 2D Bounding Box in the visual and infrared camera images.

Example

Correct annotation of a buffer stop on the neighbouring track



Correct annotation of a buffer stop on the ego track

DAE-1206 - LiDAR: 3D Bounding Box

Buffer stop shall be labelled with a 3D Bounding Box in the LiDAR point cloud.

DAE-1207 - 3D Bounding Box faces front of the buffer stop

The 3D Bounding Box shall face the front of the buffer stop, i.e. in direction of the track.

DAE-1208 - Radar: 2D Rotated Bounding Box

Buffer stops shall be labelled with a 2D Rotated Bounding Box in the birds-eye view radar images. Projecting the annotations from the LiDAR space is recommended.

DAE-1209 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality.

The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space).

The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame.

The scope of each attribute is defined within the attribute's requirement.

DAE-1210 - Frame Attribute: connectedTo

The attribute connectedTo of type Single-Select and scope Frame shall be set for all annotations of class buffer stop. The attribute shall define whether the buffer stop is outside the ObjectID range, then it should be labelled as “other”. If the buffer stop is not on a track, then it shall be labelled as “none”.

See below for attribute values:

- ObjectID
- other
- none

4.1.2.23 Flame

This class describes flames captured by the cameras.

DAE-1212 - Including the flame itself only

The annotation shall only include the flame itself, not the source

Example:

Correct annotation of an open flame

DAE-1213 - Camera: 2D Bounding Box

Flames shall be labelled with a 2D Bounding Box in the visual and infrared camera images.

DAE-1214 - Sensor-independent attributes (scopes Object and Frame)

The requirements defined as children are independent of the sensor modalities. They have to be assigned with the same attribute value for each sensor modality.

The attributes with scope Object shall be assigned to all annotations of a real-world object and shall be consistent across all annotations (time and space).

The attributes with scope Frame shall be assigned to all annotations of a real-world object and shall be consistent across all annotations within a multi-modal frame.

The scope of each attribute is defined within the attribute's requirement.

DAE-1215 - Frame attribute: size

The attribute size of type Single-Select and scope Frame shall be set for all annotations of class fire. The attribute shall define the size of the fire.

Fires with a height smaller than 4 m are considered as small and fires with a height larger than 4 m and a surface area of 4 square meters are considered as big.

Attribute values:

- big

- small (default)

Example:

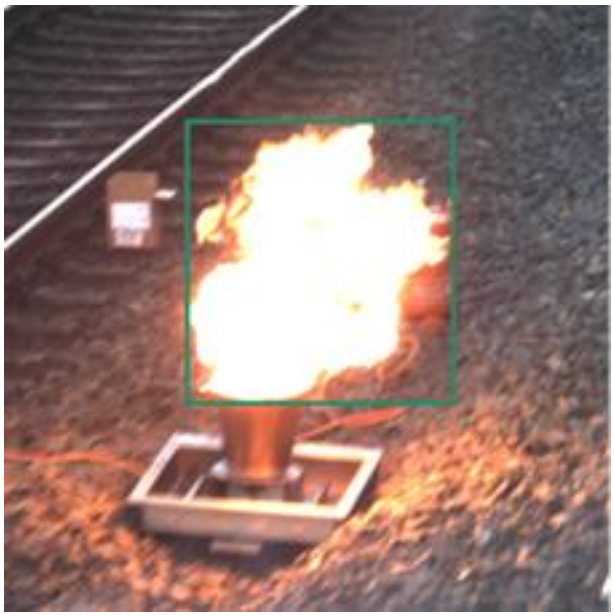


Illustration of a fire of height up to 4 meters

4.1.2.24 Smoke

This class describes smoke seen by the camera sensors.

DAE-1216 - Annotate areas where structure behind the smoke is hardly- or non-visible

Smoke shall be labelled if the structure behind the smoke is hardly- or non-visible

Smoke shall be labelled if the structure behind the smoke is hardly- or non-visible

DAE-1217 - Camera: Outline Polygon

Smoke shall be labelled with an Outline Polygon.

DAE-1219 - Consider bound boxing the smoke only and not the source

The Bounding Box shall only include the smoke itself, not the source.

Examples:

Annotation of smoke with a 2D point polygon is set

Annotation of smoke with a 2D point setted polygon

DAE-1220 - Sensor-independent attributes (scopes Object and Frame)

The attributes with scope Frame shall be assigned to all annotations and consistent across all annotations of all smokes within a multi-modal frame.

DAE-1221 - Frame attribute: size

The attribute size of type Single-Select and scope Frame shall be set for all annotations of class smoke. The attribute shall define the size of the smoke. Smoke clouds with a height smaller than 4 m are considered as small. Smoke clouds with a height bigger than 4 m and a surface area of 4 square meters are considered as big.

Attribute values:

- big (default)
- small

DAE-1222 - Frame attribute: colour

The attribute colour of type Single-Select and scope Frame shall be set for all annotations of class smoke. The attribute shall determine the colour of the smoke.

Attribute values:

- white
- grey (default)
- black
- other

REFERENCES

[1] R. Tilly, P. Neumaier, K. Schwalbe, P. Klasek, R. Tagiew, P. Denzler, T. Klockau and M. Köppel, “Offener Multisensordatensatz für die Entwicklung der Umfeldwahrnehmung beim vollautomatischen Fahren,” ETR, vol. 4, 2023

[2] R. Tilly, P. Neumaier, K. Schwalbe, P. Klasek, R. Tagiew, P. Denzler, T. Klockau, Martin Boekhoff and M. Köppel, https://data.fid-move.de/dataset/osdar23, 2023.

[3] "Technologies for Autonomous Rail Operation | TAURO | Project | Fact sheet | H2020 | CORDIS | European Commission.", https://cordis.europa.eu/project/id/101014984/reporting, 2023

[4] “Innovative concepts and technologies for a Pan-European Railway Data Factory network for future digital and automated rail operation (21-EU-DIG-RailDataFactory).”, LINK, 2023

[5] “CEF2 RailDataFactory D 1, “Data Factory Concept, Use Cases and Requirements”, LINK, 2023

[6] “CEF2 RailDataFactory D 2.1 – “Technical specifications and available solutions for building blocks, components, Cloud / hybrid-Cloud and Edge-Orchestration & Operational concept”, LINK, 2023

[7] “CEF2 RailDataFactory D 2.2 – “Technical specifications and available solutions for Identity Access Management (IAM), Data Management and Transfer and Cyber-Security”, LINK, 2023

[8] “CEF2 RailDataFactory D 2.3 – “High-speed pan-European Railway Data Factory Backbone Network”, LINK, 2023

[9] “CEF2 RailDataFactory D 3.1 – “Report of bottlenecks data application in rolling stock”, LINK, 2023

[10] “CEF2 RailDataFactory D 3.2 – “Business case whether open data infrastructure would be attractive for European rail”, LINK, 2023

[11] “CEF2 RailDataFactory D 3.3 – “Description of cybersecurity vulnerabilities, threat scenario’s and usable standards to mitigate associated risks”, LINK, 2023

[12] “CEF2 D 3.4 CEF2 RailDataFactory D 3.4 – “Legal and regulatory assessment catalogue", LINK, 2023

[13] “CEF2 RailDataFactory, D 4.1 – “Deployment activities description for a pan-European Railway Data Factory”, LINK, 2023

[14] “CEF2 RailDataFactory, D 4.2 – “Pan-European Railway Data Factory deployment planning and strategy proposal”, LINK, 2023

[15] “CEF2 RailDataFactory, D 5.2 – “Final Study Results”, LINK, 2023

[16] "Advanced signalling and automation system - Completion of activities for enhanced automation systems, train integrity, traffic management evolution and smart object controllers | X2Rail-4 | Project | Results | H2020 | CORDIS | European Commission.", https://cordis.europa.eu/project/id/881806/results, 2023.

[17] "Information technology — Security techniques — Information security management systems — Requirements | ISO/IEC 27001." International Organization for Standardization. https://www.iso.org/standard/54534.html, 2024.

[18] "Information technology — Security techniques — Code of practice for information security controls | ISO/IEC 27002." International Organization for Standardization. https://www.iso.org/standard/54533.html, 2024.

[19] "Industrial communication networks - Network and system security - Part 2-4: Security for industrial automation and control systems | IEC 62443-2-4." International Electrotechnical Commission. https://webstore.iec.ch/publication/6199, 2024.

[20] P. Neumaier, “Data Factory - Data Production" for the training of AI software”, LINK, 2022

[21] CartenaX Consortium. (2023). CartenaX: Empowering the Automotive Industry through Secure Data Exchange within Gaia-X. Retrieved from https://www.gaia-x.eu/cartenax