D8.1 Specification of Industrial validation Use Cases

Deliverable Id:	**D8.1**
Deliverable Name:	**Specification of Industrial validation Use Cases**
Status:	**Final**
Dissemination Level:	**Public**
Due date of deliverable:	**2022/08/31**
Actual submission date:	**2022/09/16**
Work Package:	**WP8 "Case Studies, Specification, Benchmarking & Justification File"**
Organization name of lead contractor for this deliverable:	MODIS Consultant
Authors:	Pietro Abbatangelo, MODIS Luca Visconti, MODIS Jon Mikel Olmos Serna, CAF Markus Postl, Virtual Vehicle Florian Thaler, Virtual Vehicle Bernhard Winkler, Virtual Vehicle Artur Kaufmann, BEEA Stefano Delucchi, AITEK Giacomo Valente, UNIVAQ Luigi Pomante, UNIVAQ Fernando Eizaguirre, IKERLAN Iñaki Paz, LKS
Reviewers:	Martin Matschnig, Siemens Tania Di Mascio, UNIVAQ

Abstract

D8.1 "Specification of Industrial Validation Use Cases" aims at providing a detailed description of each Use Case in terms of implementation, objectives, evaluation of results, and expectations from FRACTAL.

Implementation plan and justification plan are defined for each Use Case and they will guide third year project activities related to Use Cases from UC5 to UC8.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 877056

Co-funded by the Horizon 2020 Programme of the European Union under grant agreement No 877056.

Contents

1 History

Version	Date	Modification reason	Modified by
0.1	26/08/2022	First complete draft version	All contributors
0.2	05/09/2022	Complete version for internal review	All contributors
1.0	16/09/2022	Submission	All contributors

Table 1 - Document history

2 Summary

This document is the output of Tasks T8.1 and T8.2 and will guide the tasks T8.3 and T8.4. Four of the FRACTAL Use Cases take part in WP8 (Case Studies, Benchmarking and Quality) for the industrial validation of FRACTAL developments:

- UC5 Increasing the safety of an autonomous train through AI techniques;
- UC6 Elaborate data collected using heterogeneous technologies;
- UC7 Autonomous robot for implementing safe movements;
- UC8 Improve the performance of autonomous warehouse shuttles for moving goods in a warehouse.

This document declares the objectives of these Use Cases, how they are going to be implemented in FRACTAL and finally how the Use Cases are going to evaluate results. In this sense, the document contains 4 main sections (one per UC) where each of the sections presents the following structure:

- A small introduction to the UC;
- How the system would be implemented without FRACTAL;
- The objectives of the UC within FRACTAL;
- An exploration of the state of the art in the UC field;
- The main contributions expected from FRACTAL;
- How the results are going to be evaluated (UC KPI definition);
- How the Use Case will be implemented using FRACTAL;
- And the justification plan: how defined KPIs will be evaluated in the context of the UC to evaluate FRACTAL.

3 Introduction

The goal of the FRACTAL project is to create a cognitive edge node, called FRACTAL node, enabling a FRACTAL Edge that can be qualified to work in different domains under industry standards. The FRACTAL node will be the basic building block of intelligent and scalable Internet of Things (ranging from Low-Energy Computing to High-Performance Computing Edge Nodes).

The strategic objective to implement and prioritize the different requirements of a FRACTAL node are presented in Table 3 of the Deliverable D2.1 and shown below in Table 2. The objectives of this project are related to the main technological pillars, representing all the characteristics and specifications that need to be integrated into the hardware and software of the node.

Obj. #	Objective	Relates to
O1	Design and Implement an Open-Safe-Reliable Platform to Build Cognitive Edge Nodes of Variable Complexity	Pillar 1 WP3
O2	Guarantee extra-functional properties (dependability, security, timeliness, and energy-efficiency) of FRACTAL nodes and systems built using FRACTAL nodes (i.e., FRACTAL systems).	Pillar 2 WP4
O3	Evaluate and validate the analytics approach by means of AI to help the identification of the largest set of working conditions still preserving safe and secure operational behaviors	Pillar 3 WP5
O4	To integrate FRACTAL communication and remote management features into FRACTAL nodes	Pillar 4 WP6

Table 2 - FRACTAL Objective

Even if belonging to different application fields, the Use Cases share some needs that are closely related to the objectives of FRACTAL. These general needs are common to all the Use Cases, but some of them are more crucial in some use cases.

Table 4 of the DoA, shown below, synthesizes the main needs associated with each use case, a brief description of them, and their relationship with FRACTAL pillars with different priorities (H: High, M: Medium, L: Low) for each pillar.

Use case	Name	Lead Partner	Technological Pillars: Open Safe	Low Pow. Security	Cognition Safe	Fractal Mutable
VAL-UC5	Increasing the safety of an autonomous train through AI techniques	CAF	H	H	H	M
VAL-UC6	Elaborate data collected using heterogeneous technologies	AITEK/ UNIVAQ	H	M	H	M
VAL-UC7	Autonomous robot for implementing safe movements	VIF	H	H	H	M
VAL-UC8	Improve the performance of autonomous warehouse shuttles for moving goods in a warehouse	BEE	H	H	M	H

Table 3 - Use Case, brief description, leading partners, and their relationship with FRACTAL pillars

This deliverable is the output of the FRACTAL Task T8.1, related to "Case Study Coordination", and of FRACTAL Task T8.2, related to "Case Study and Benchmark Specification"

D8.1 belongs to WP8 whose goal is to demonstrate how the FRACTAL building blocks, technologies and methodologies are applied to industrial applications with well-identified performance, security and safety requirements defined in WP2.

In particular, the Deliverable D8.1, entitled "Specification of Industrial validation Use Cases", aims to provide a detailed description of each Use Case by defining the following.

The functioning of the system before FRACTAL, intended as the system is managed without the implementation of this Use Case in FRACTAL.

The objectives of the use case, the desired functioning of the system after the implementation of this FRACTAL Use Case, then how the system is managed with FRACTAL.

The architecture of the use case, based on the "Big Picture" defined in WP2. The architecture related to the "Big Picture" is composed of main components and their interaction within FRACTAL. There are three main aspects: FRACTAL services in the cloud, Software components of FRACTAL on edge node and finally hardware platform used in FRACTAL. Within the Deliverable D2.3, entitled "Platform Specification (b)" is possible to identify every Hardware and Software component that is present within the architecture of each Use Case.

The state of the art in the field of the use case, for instance projects, technologies and approaches that are currently used in the scenario of the specific use case, to identify existing systems and their performance. The scope of this investigation is to define actual reference in each Use Case field in terms of performance to compare with FRACTAL UC solutions. Comparison will be defined in the Deliverable D8.2, entitled "System Requirement", in terms of benchmarks.

The main contributions expected from FRACTAL, in particular, how the Use Case implements the main pillars of the project.

The implementation plan by defining the implementation stages (and which components will be used at each stage) with the tasks, a brief description, the relationship between tasks, the objectives, and the KPIs. In each implementation phase, the KPIs are continuously evaluated. The main KPIs objectives are to evaluate the progress in each implementation phase and evaluate the progress compared to Use Case objectives.

Justification plan with justification methods like demonstrations, tests, simulations, calculations, etc. The justification plan can be considered as a test plan designed to be executed during the implementation phases. Justification plan execution will guide the D8.3 deliverable "Evaluation Result" that will collect all results.

4 VAL-UC5 Increasing the safety of an autonomous train through AI techniques

The railway industry is evolving towards autonomous vehicles presenting a roadmap for porting certain technologies with proven efficiency in other industries such as the automotive industry. Increasing the autonomy of a train implies deploying a wide variety of technologies based on artificial intelligence which has critical hardware computing capacity requirements. This focus differs from the traditional hardware used in railway industry that gives critical importance to robustness, liability, predictability, and other considerations needed to develop a safety platform. For this reason, conventional railway hardware does not satisfy the requirements for deploying AI based functionalities and HW accelerated platforms come into the scene.

Autonomy relies on implementing automatic operations that were previously performed by a human driver. Further analysis of all the human-based operation leads to a large list of processes of several types (security checks, environment perception, incident prevention, driving …) and some of them require AI techniques to work, concretely the functions related to environment perception. The UC5 is based on this environment perception functions, from their development until their deployment on an embedded platform.

Environment perception means extracting all information related to train operation from the real environment and converting it to data understandable by the rest of the systems. The train operation environment contains several types of information, in the UC5 scope the relevant information is related to the train driving operation (traffic signals) and to the human-train interaction (human presence in dangerous surroundings during train movement).

From all the functions presented in autonomous train operation UC5 will cover the following functions:

- **Accurate Stop:** Correct train stopping location based on landmark references
- **Safe Passenger Transfer:** Verify that there is no human presence on the station platform near the train at gate closing and departure.

4.1 Before FRACTAL

In a conventional environment, accurate stop and safe passenger transfer functions are covered by the train driver. The driver has complete control of train driving manipulation so that the stopping location is directly corrected by the driver. The departure and the door control are also performed by the driver. In order to provide those functions in an autonomous way and fulfill real-time performance requirements, several algorithms present in the state-of-the-art are applied.

The first approach for covering the UC5 functions is based on detection algorithms working together with distance estimation techniques. On the one hand, an accurate stop requires detection algorithms to detect the landmarks defined for the stopping location using two frontal cameras as input and stereovision distance estimation techniques to calculate the remaining distance until the landmarks. On the other hand, safe passenger transfer requires detection algorithms to detect passengers near the train using the rear mirror camera as input. The algorithms selected for the use case are following.

- Accurate Stop: Transfer Learning on Yolov3 and Yolov4 608x608 for landmark detection and OpenCV's stereo SGBM (Semi Global Block Matching) for distance estimation.
- Safe Passenger Transfer: Transfer Learning on Yolov3 and Yolov4 608x608 for person detection in train surroundings.

First Validation setup is achieved in a laboratory through the implementation of custom SW using these algorithms integrated on an X86 machine with dedicated GPU acceleration and using recorded videos as input. The training of YOLO models is performed using a custom recorded and labeled dataset gathered in the Case Study environment (Line 3 from Euskotren Metro Bilbao). The results show that selected algorithms and gathered data are suitable for the environment characteristics related to ambient lighting and object characteristics.

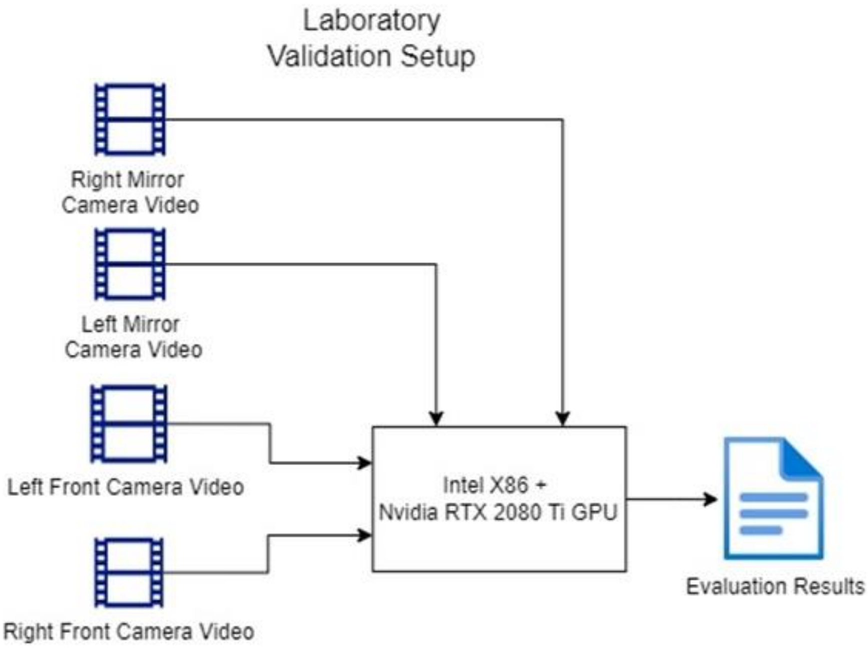


Figure 1 - Laboratory validation setup over X86 and GPU platform

The laboratory validation setup, shown in Figure 1, proves the efficiency of the selected algorithms in an early stage for the use case but it is not suitable for train integration. The next step is based on embedding those algorithms into a commercial platform suitable for the train environment. The selected platform is the Jetson family from Nvidia which provides the embedded GPU requirements for inference on real-time operation. The results show that the selected models and algorithms can provide a real-time cycle (100 ms) on this platform. The deployment setup and its validation setup based on Nvidia Jetson Xavier are described in Figure 2.

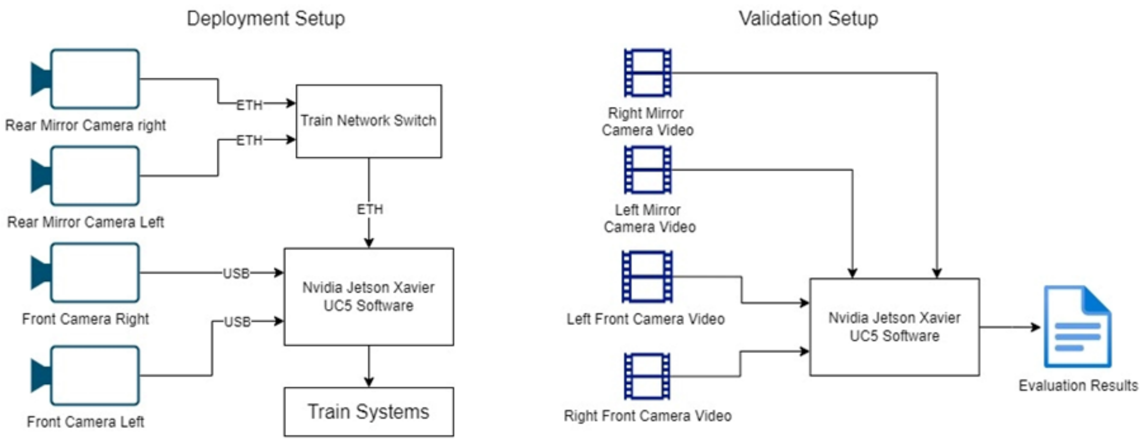


Figure 2 - a) Deployment setup based on Jetson Xavier platform. b) Validation setup based on Jetson Xavier platform

The setup is based on an Nvidia Jetson Platform obtaining input from two frontal cameras and one rear-mirror camera. Operation software is deployed based on Yolo model GPU accelerated inference and SGBM CPU based algorithm.

This functional setup is dependent on specific hardware that does not provide any safety considerations, avoiding the capacity for eventually hosting safety-critical functions. PER system (PERception systems) functions definition is currently in progress and there are no safety functions defined yet but foreseeing the safety needs for any function, the exploration for high-performance safety-critical hardware alternatives is needed. At this point, the FRACTAL project presents a suitable candidate for embedding the PER system with the two specific functions, safe passenger transfer and accurate stop, in scope.

4.2 Use Case Objectives

4.2.1 Specific FRACTAL Technical Objectives

Based on the starting point consisting of a commercial embedded platform deploying Landmark detection and passenger detection, the technical objectives for this use case aim to:

- To Integrate the safety-critical high-performance computing platform within a railway control system;
- To Test and evaluate of CV&AI-enhanced autonomous train operation processes over safety-critical high-performance computing platform with actual in-the-field data and operating in the real railway vehicle environment. The use case will perform CV&AI based:
  - Automatic platform detection;
  - Accurate automatic stop at door equipped platforms, aligning the vehicle and platform for correct passenger transfer;
  - Detection of the passengers who are getting in/out the train (in platform area) avoiding any door closing operation before all train's doors are free of crossing-passengers.

4.2.2 General Objectives

The general objectives regarding the autonomous vehicle roadmap are:

- Give autonomy and decision-making capabilities to vehicles so they can observe and interpret the environment in an independent manner, complementing the information already received from railroad signaling modes;
- Reduce installation and maintenance costs by lowering both complexity and price with new optical sensors and increasing the installation's lifecycle;
- Increase flexibility in different railway operations that are attached to delimited areas and delimited time slots depending on the type of railroad and its configuration;

- Enhance variable calculation and operations both in precision and speed with new optical sensors information;
- Increase railway systems safety;
- Increase railway exploitation capacity and flexibility by CV&AI based more precise measurements (optical metrics, object detection/identification).

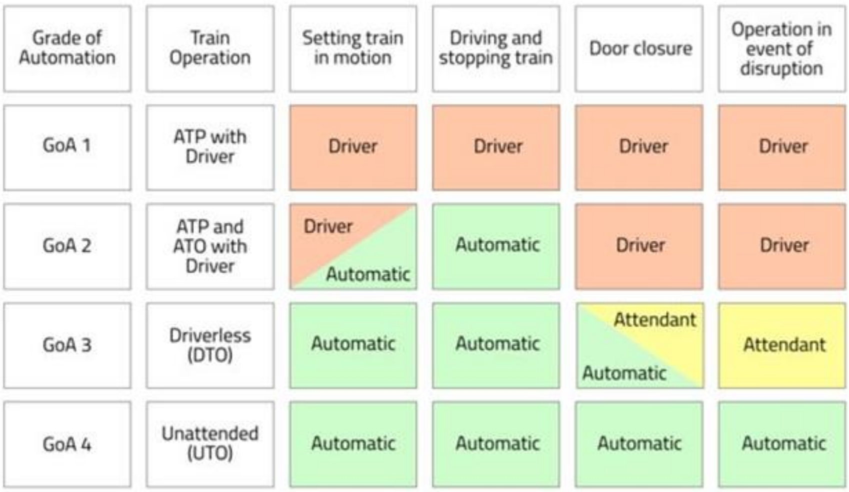
4.3 State of the art

4.3.1 Technological Context in Railway Industry

The technical evolution roadmap in the railway industry is regulated either by growing customer needs and by European Consortiums such as Shift2Rail [1], Tauro [2] and their successor EURail [3]. Customer needs are mainly based on improving the efficiency in transport lines following the criteria:

- Increase Transport density: Increase the train density on a transport line (headway) by improving signaling systems higher speeds and lower train to train distance;
- Reduce Energy consumption: Reduce energy consumption of vehicles by regulating driving profiles;
- Increase Transport Flexibility: Dynamically rearrange transport lines to fit variable schedules and cover passenger demand;
- Increase Safety: Introduce new systems for driving assistance that complement driver's reaction;
- Increase Ride Comfort: Introduce new systems to generate smooth driving profiles that lead to higher comfort during travel;
- Reduce Maintenance Costs: Introduce new techniques to reduce long term maintenance costs.

Aligned to industry needs, European Consortiums elaborate the agreements and standards that make progress towards the transport efficiency goal. For increasing the vehicles autonomy, Shift2Rail presents the roadmap that can be seen in Figure 3.



Grade of Automation	Train Operation	Setting train in motion	Driving and stopping train	Door closure	Operation in event of disruption
GoA 1	ATP with Driver	Driver	Driver	Driver	Driver
GoA 2	ATP and ATO with Driver	Driver / Automatic	Automatic	Driver	Driver
GoA 3	Driverless (DTO)	Automatic	Automatic	Attendant / Automatic	Attendant
GoA 4	Unattended (UTO)	Automatic	Automatic	Automatic	Automatic

Figure 3 - Grade of Automation Levels Presented in S2R

The Grade of Automation levels (GoA) introduce the required steps towards the autonomous train (GoA4).

GoA1 provides ATP system, which stands for "Automatic Train Protection". The ATP system supervises the operation and prevents overspeed situations or end-of-authority trespassing. This is a safety-critical system and reacts to dangerous situations over driver's actions

GoA2 level introduces ATO System. The ATO system, which stands for "Automatic Train Operation" handles the traction and brake commands required to drive the train between stations. The ATO system is always triggered by the driver and cannot override the driver's actions. It works also under the supervision of GoA1 systems.

GoA3 and GoA4 levels introduce the DTO concept, which stands for "Driverless Train Operation". The new systems required for this level of autonomous operations are currently under definition. One of those former systems is the PER system. This system, which is responsible for detecting environment vehicle surroundings information, presents the context for UC5 requiring the application of heavy computational costed AI techniques.

The state of the art in the railway industry is located between levels GoA2 and GoA3 settling down the level GoA2 with the introduction of the ATO system specification and planning the step to GoA3 through several European Consortiums like Shift2Rail, Tauro, and EURail.

4.3.2 State of the Art for Relevant AI Techniques

The functions in scope of UC5 require both (1) *object detection* techniques and (2) *distance estimation* techniques which represent separated technical challenges.

In recent years, deep learning based object detection applications are emerging in the railway domain for tasks such as signal/objects detection and distance estimation [4] [5] [6] [7]. Efficient and robust embedding of these models into embedded hardware, such as NVIDIA Jetson or FPGA boards, as in FRACTAL, is a challenge.

In 2012, the use of GPUs and a CNN (Convolutional Neural Network) called AlexNet [8] changed the AI paradigm in the object visual recognition area, winning the ImageNet 2012 challenge. Most of the image processing architectures in deep learning are built from CNNs, as these have shown the ability to learn features from the imagery. These feature extractors, which take images as input and outputs feature maps of the corresponding input image are named as *Backbone networks*. Different backbone architectures have been designed in CNN based object detection approaches. These architectures are divided in different categories; some architectures are focused on accelerating the inference speed while others are focused on increasing the achieved accuracy [9]. Some tasks like real-time video processing require not only high processing speed but high accuracy, which require well-designed backbone networks to overcome the existing trade-off between speed and accuracy.

Deep Learning based object detection models have been generally tested on the ImageNet database challenge (ILSVRC) and since 2012, every year the accuracy error has been decreasing. In 2013 ZFNet [10] improved 0.5% to AlexNet. In 2014

GoogleNet (InceptionV1) [11] and VGGNet [12] appeared. Eventually in 2015, ResNet [13] beat human-level performance on this dataset (with 3.57% error). Some very popular models followed such as DetNet [14] , R-CNN [15] and Fast R-CNN [16].

Most of these models are based in a two-stage method in which a first network proposes regions of interest in the image and a second network detect objects in those regions. These models may be very accurate but, due to their two stage nature, add an overhead to the inference process.

In 2015 the first one-stage detector, called YOLO [17] model (You Only Look Once) appears, and since then, it has become the most popular one-stage detector. Yolo is specially designed for real-time object detection. In next years, other one-stage detectors, as for example SSD [18] and RetinaNet [19] appeared but Yolo remains one of the most successful models.

From 2016 to the present day, new more accurate and faster versions of YOLO appeared, such as YoloV2 [20], YoloV3 [21] and YoloV4 [22]. Both YoloV3 and YoloV4 achieves a state-of-art combination of speed and accuracy and, hence, have been selected for UC5 for the detection of train stop signals in the platform and the detection of persons or obstacles around the train doors.

Regarding distance estimation, nowadays, some of the technologies that estimate the train position is based on wheel odometry and radars: a beacon-based system in the track and encoders and radars installed onboard to estimate train odometry data. The inaccuracy of radar and encoder sensors estimation is corrected when the onboard controlling system receives track beacon distance information. However, at a stopping point in a station, the driver's eyes and experience are still the key factors to align the train correctly with the platform area and to remove the final localization error. These systems have a high installation cost (as a lot of beacons must be placed in the rail infrastructure), high maintenance cost, and the deployment is slowed down.

In recent years, Deep Learning models approaches to estimate the distance from RGB cameras have been proposed. Some of the most interesting approaches can be summarized as:

- *Position Estimation by Visual Odometry techniques*: In this case, the aim is to estimate the train's geographical position by analysingn real time the images acquired by a camera in front of the train. The algorithm compares consecutive images to estimate the speed of the train and the turn (left, right) thus, computing the accumulated position of the train. Some of the most popular algorithm includes DeepVO [23] and state-of-art ORB-SLAM2 [24] . However, these algorithms are better suited for general geographical positioning but not for accurate positioning in a train stop.

- *Distance Estimation*: In this case, the aim is to estimate the distance to a given signal in the track, for example, the train stop signal in the platforms. The techniques in this group may use one single monocular camera or stereo-cameras:

    - Algorithms with *monocular cameras*: A Deep Learning model is trained to estimate the distance to a Signal/Obstacle detected with Yolo. DisNet [25] is one of the most well-known monocular algorithms. Unfortunately, the algorithm suffers when the object/obstacle to be detected is not always of the same size and is relatively inaccurate even in well-known size objects. This is due to the fact the monocular vision faces problems estimating the scale of objects of varying size (i.e., an adult or a child in the track, for example, both are persons, but of different sizes, so the algorithm can conclude that the child is an adult who is further away).

    - Algorithms with *stereo-cameras*: In this type of algorithms a stereo-camera, asfor example, a double camera in the front left and front right of the train is used. Both cameras detect, for example, the train signal stop signal in the platform using Yolo algorithm. Both detections are fed to a stereo-matching algorithm that can infer the visual angle difference of the signals. With the angle difference a simple triangulation can be used to infer the distance to the signal. These algorithms are especially suited for accurate estimation of the distance to relative near objects (where the angle difference is relevant) as for example the detection of the train stop signal in the platform when the train is entering the station platform.

In the case of UC5, the aim is to estimate the distance to a relative near train stopping point signal in the platform, therefore, the last technique type has been selected.

4.3.3 Deep Learning Hardware, Power Computing and functional safety

AI solutions to be used in UC5, as for example YoloV3/V4, area heavily based in CNN Neural Networks. Convolution operations pose hard processing speed power requirements on hardware and, in addition to this, are a challenge to achieve functional safety.

In general terms, in the computing area, there are several hardware solutions available [26]￼the classical CPU, the GPUs, and the FPGAs. CNN networks require computing parallelization to achieve high inference speed, therefore, GPUs and FPGAs are the options to consider. According to [27], each option has its own advantages and disadvantages and can be summarized as follows:

- **Computing Power**: According to Xilinx an Ultrascale+TM XCVU13P FPGA reaches 38.3 INT8 TOP/s, and has almost the same computing power as an NVIDIA Tesla P40 that reaches 40 INT8 TOP/s. Regarding on-chip memory (something crucial in deep learning) FPGAs can have a high amount of cache memory which reduces memory bottlenecks and allow high bandwidth. FPGAs produce low latency and deterministic latency, ensuring that the deep learning model provides a stable response time, something critical in many real-time applications as for example object recognition in real-time video. Therefore, both options, GPUs and FPGAs can be considered for UC5.

- **Flexibility and Power Efficiency**: GPUs are designed to run arbitrary code; therefore, the data flow in the GPU is defined by software and has to accommodate to the complex memory hierarchy and fixed cores of the GPU. If the task is vastly parallel and suits well to the GPU structure, the execution can be very efficient, however, this is not always the case. FPGAs can deliver more flexible architectures, adapted to the exact problem structure (i.e., the structure of a given neural network, which is parallel by nature), and thus may achieve maximum task parallelization and power efficiency when FPGAs are programmed as systolic arrays. For specific problems, the ability to reconfigure the FPGA to the exact parallel nature of the task may be a big win. In the case the processor unit has to execute completely changing nature tasks, the CPU/GPU classic schema will be more appropriate.

- **Functional Safety**: GPUs are originally designed for graphic high-performance tasks where safety is not a concern;then, to meet functional safety a costly and time-consuming redesign for GPU vendors would be needed (although NVIDIA Jetson AGX Xavier has given a huge step in this way [28] paving the way to IEC-61508 and IEC 26262 certification), but in general terms, achieving functional safety via GPU may be very complex/impossible. However, FPGAs have been designed in a way to meet functional safety requirements of sectors such as avionics, defense, industrial automation, etc. For example, Xilinx Zynq®-7000 and Ultrascale+TM MPSoC devices are designed to support safety-critical applications such as Autonomous Driving. In the case of UC5, applications associated with doors operation in a train do have functional safety requirements so it seems that FPGA may be the way to go.

4.3.4 Object Detection Metrics (KPIs)

To compare the results of object recognition networks, such as **YOLO**, **Fast_CNN**, and **DetNet**, the research community has defined precise numeric metrics [29] that allow evaluating and comparing the different networks.

In the computer vision research area, and more precisely in object recognition, the most popular metrics are:

- **Average Precision** (AP)
- **Mean Average Precision** (mAP)
- **Precision Threshold** curve
- **Recall Threshold** curve
- **Precision Recall** curves (PR Curves)

Before explaining these metrics, some key concepts need to be understood:

- *True Positive (TP)* – Object correctly identified by the model.
- *False Positive (FP)* – Object detection that is not correct.
- *False Negative (FN)* – Object that should have been detected but it is not.
- *True Negative (TN)* – Regions of the image in which, correctly, the model does not detect any objects.

4.3.4.1 Intersection over Union

This metric, abbreviated as **IoU**, defines the percentage of overlap between the object ground truth and the predicted area of the object. Figure 4 explains the concept.

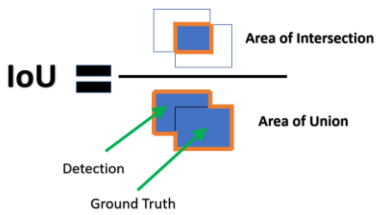


Figure 4 - Intersection over Union

IoU ranges from 0 (complete fail) to 1.0 (perfect match). Usually, a **threshold** is defined to determine if the detection is a True Positive (i.e., IoU > 0.75), a False Positive (i.e., 0 < IoU < 0.25) or a False Negative.

4.3.4.2 Precision and Recall

Precision defines how exact is the model when detecting only relevant objects. Mathematically (Figure 5), is the ratio of True Positives over the Total Detections made by the model (True Positives + False Positives). As an example, in a sequence of several frames of a video, if the model identifies 250 persons but only 220 of them are true positives, the precision is P = 220/250= 0.88 (88%).

$$P = \frac{TP}{TP + FP}$$

Figure 5 - Precision calculation

Recall, on the other hand, defines the ability of the model to detect all the right objects in the scenes. Mathematically (Figure 6), is the ratio between the True Positives and all ground truths (True Positive + False Negatives). As an example, in the sequence of frames mentioned before, if there are really 230 persons but only 220 of them are detected, the recall is R = 220/230 = 0.95 (95%).

$$R = \frac{TP}{TP + FN}$$

Figure 6 - Recall calculation

A good model must have a high Precision and a high Recall. These two metrics are the key KPIs to define the rest of the metrics, as explained in the following sections:

4.3.4.3 PR Curves

The Precision curve (Figure 7) and Recall curve (Figure 8) are plots of Precision and Recall values given by a model at different IoU thresholds. The figures below show these curves in which the recall and precision of the model can be seen against the threshold curve.

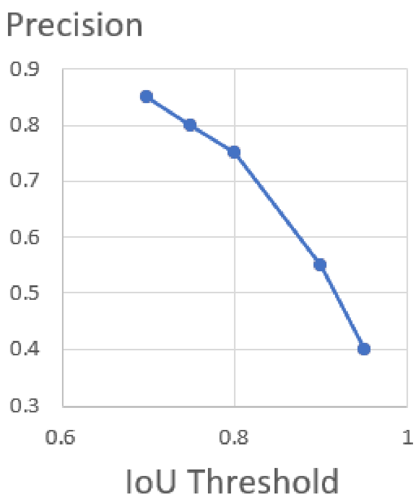


Figure 7 - Precision curve

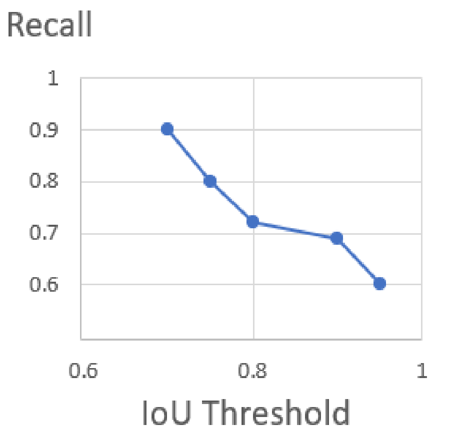


Figure 8 - Recall curve

Finally, a third and very important curve can be defined, the **PR Curve** (Figure 9), in which, for a given threshold, the X axis shows the Precision, and the Y axis shows the Recall. This curve shows how the Precision and Recall are interrelated and that usually maximizing one will decrease the other, so a trade-off has to be found because a high number of false positives results into to low precision, and a high number of false negatives results into low recall.

Ideally, both metrics should be high, but in practice it is better to optimize one of them, depending on the case, or at least, decide what is an acceptable trade-off point.

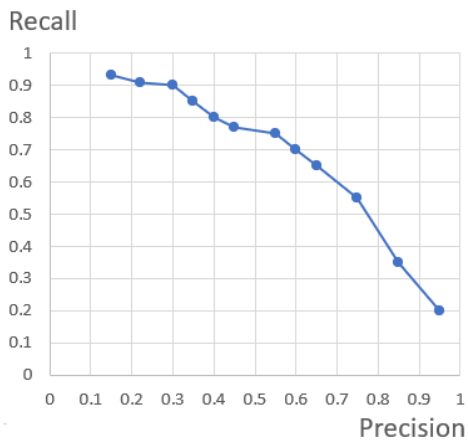


Figure 9 - PR Curve

4.3.4.4 Average Precision (AP) and Mean Average Precision (mAP)

Finally, there are two metrics that are used to summarizes the others:

- **Average Precision** (**AP**) - for a given model and threshold: It is defined as the total area under the PR Curve. AP is a value between 0 and 1, where 1 means a perfect model. As an example, Figure 10 shows the AP_{50} calculation, it is the area under the **PRCurve** for a model evaluated with a threshold equal to 50.

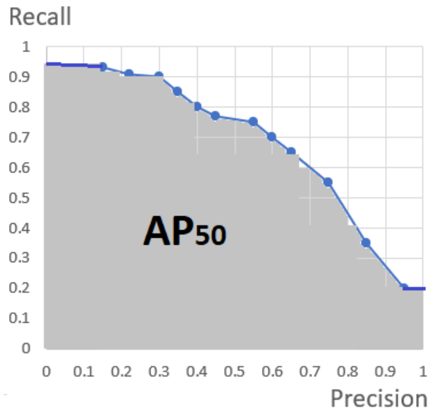


Figure 10 - Average Precision (AP) curve

- **Mean Average Precision** (**mAP**) – finally, the *mean average precision* is the mean of all **APi** values for all classes, and for a given threshold of course. As an example, if the model has five classes and we compute the AP_{75}-i for each class *i=1…5*, the mean average precision is calculated as shown in Figure 11.

$$mAP_{75} = \frac{1}{5}\sum_{i=1}^{5} APi_{75}$$

Figure 11 - Mean Average Precision (mAP)

4.4 Main contributions expected from FRACTAL

The FRACTAL project provides an opportunity to explore safety-critical and high-performance alternatives to commercial GPU based non-safety platforms aligned to the main FRACTAL pillars.

The main focus for the UC5 stands for executing heavy computational load algorithms on a platform providing safety considerations. The Use Case presents real-time requirements caused by the need for updated information for the correct operation of the train. This need also influences latency considerations in order to avoid presenting obsolete and therefore incorrect information to the systems in charge of train operation. For those reasons, relying on the cloud, where heavy computing machines can be emplaced avoiding railway-specific requirements for AI, inference purposes is not an option and inference in the edge requirement is enforced. The FRACTAL platform is expected to provide AI inference on the edge mechanisms to execute selected Yolo models inference at a real-time frequency defined as 10 frames processed per second. Amongst the platform variants presented, Xilinx Versal-based FRACTAL platform is selected due to its accelerated nodes capacity and memory capacity.

As the UC5 proposal and requirements are focused on single-train PER system, other pillars of the FRACTAL project are outside of the UC5 scope but present an opportunity to cover the transversal needs of the railway industry. Service orchestration and fractality concept provide a way to centralize train fleet management for UC5 scope software. These cloud/edge additional services allow an extension of UC5 setup with an additional functionality: centralized SW and AI Model management. The extension of the Use Case implies introducing new needs for other FRACTAL characteristics, such as security for handling train(edge) to control center(cloud) connection, cloud services for SW and AI model updates and edge services for automatic software updates from the cloud.

Related to FRACTAL project objectives, UC5 provides an application environment for each FRACTAL Pillar (Table 2).

The main UC5 development is contained in Pillar 1 and WP3 scope. UC5 provides a strong requirement for edge real-time inference which leads to high-performance cognitive edge node requirements.

The extended UC5 establishes a connection with the cloud and, therefore, provides an application environment for Pillar 2. Pillars 3 and 4 are also present for the remote AI model and software management services and the automatic update application.

4.5 Evaluation of the implementation results

This section defines the KPIs defined for UC5 implementation. These KPIs are classified into three groups:

- KPIs for Implementation Plan Task;
- KPIs for FRACTAL Objectives related to FRACTAL Pillars;
- KPIs for UC Features.

For each KPI, an *Identifier*, a *Description* and the type of result *Value* is defined. The *Test* to be performed for the KPI will be defined later in the Justification Plan, therefore is marked as TBD (To Be Defined).

Next sub sections describe in some detail the three groups of KPIs.

4.5.1 KPI for Implementation Plan Tasks

This section defines the ***KPIs defined for the Implementation Tasks***. Figure 12 shows the complete list of KPIs defined for the Implementation Tasks of UC5.

KPI UC for Implementation Plan

KPI ID	Description	Value	Test	Comment
UC5_KPI_IP_01	All subtask success	True/False	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_02	Inference time	< 100 ms	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_03	Build OpenCV on for Target (Versal ARM64) success	True/False	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_04	Accuracy % with respect to X86 platform	> 95%	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_05	Build CAF Demonstration Software on Target success	True/False	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_06	Build Safe Passenger Transfer application success	True/False	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_07	Build Accurate Stop application success	True/False	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_08	All subtask success	True/False	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_09	Hours of video recorded	> 80	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_10	Image Database size	> 40000	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_11	Model accuracy over test database	> 75%	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_12	All subtask success	True/False	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_13	Inference time	< 100 ms	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_14	Working under secure connection success	True/False	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_15	Model and docker image cloud hosting success	True/False	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_16	Cloud repositories version handling success	True/False	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_17	Metrics obtained for defined model both in X86 and Versal	True/False	TBD	*Defined for Implementation Tasks*
UC5_KPI_IP_Req_01	Edge Node Platform Ruggedized	True/False	TBD	*Comming from UC Requirements*
UC5_KPI_IP_Req_02	Edge Node Inference Time	< 100 ms	TBD	*Comming from UC Requirements*
UC5_KPI_IP_Req_03	Edge OpenCV Support	True/False	TBD	*Comming from UC Requirements*
UC5_KPI_IP_Req_04	Edge ONNX Support	True/False	TBD	*Comming from UC Requirements*
UC5_KPI_IP_Req_05	HW Accelerator Compatible wiht TensorFlow	True/False	TBD	*Comming from UC Requirements*
UC5_KPI_IP_Req_06	Edge Node with al least 4 cores	True/False	TBD	*Comming from UC Requirements*
UC5_KPI_IP_Req_07	Edge Node with mutlithreading	True/False	TBD	*Comming from UC Requirements*
UC5_KPI_IP_Req_08	Edge Node at least 60 GFLOPS	True/False	TBD	*Comming from UC Requirements*
UC5_KPI_IP_Req_09	Edge Node at least 16GB DDR RAM	True/False	TBD	*Comming from UC Requirements*
UC5_KPI_IP_Req_10	Edge Node with HW Acccelerator	True/False	TBD	*Comming from UC Requirements*
UC5_KPI_IP_Req_11	Edge Node Multi-Interfaces and their Linux Drivers	True/False	TBD	*Comming from UC Requirements*
UC5_KPI_IP_Req_12	Edge Node with Linux OS	True/False	TBD	*Comming from UC Requirements*
UC5_KPI_IP_Req_13	Platform Release with C++ compiler/crosscompiler	True/False	TBD	*Comming from UC Requirements*

Figure 12 - KPIs for UC5 Implementation Plan Tasks

The KPIs are divided into two subgroups:

- ***KPIs specifically defined for each Task*** – These KPIs have been defined to check the success of the task. *When possible, they are defined as a numerical criterion* (i.e., inference time < 100 ms), otherwise they are defined as a True/False indicating that the task finished successfully.

- ***KPIs related to Tasks, allowing checking the Requirements* defined by the UC *in the general Excel defined in WP1 FRACTAL - Requirements_KPIs_Components.xlsx*** (see Tab Requirements) - These KPIs have been defined taking into accounts the general requirements posted by the Use Case. These KPIs are defined as a True/False value indicating that *the task finished* and allows checking whether *the requirements is met.*

4.5.2 KPI for FRACTAL Objectives related to FRACTAL Pillars

KPIs defined to measure ***how the Implementation Tasks contribute to demonstrate the FRACTAL Objectives*** (Related to Pillars and found in the FRACTAL proposal, Section 1.1.2.). Figure 13 shows the complete list of KPIs defined for this purpose.

KPI for Fractal Objective (an related Pillar)				Helps to demonstrate the following Fractal Specific Objective		
KPI ID	Description	Value	Test	ID Obj	Description	Relates to Pillar
UC5_KPI_FO_00	Fractal Technology helps improving State-of-Art in Railways Sector	True/False	TBD			
UC5_KPI_FO_01	Real-time Inference Time and Accurate high perfomance Cognitive AI based node implemented and running.	< 100ms	TBD	O1	Design and Implement an Open-Safe-Reliable Platform to Build Cognitive Edge Nodes of Variable Complexity	Pillar 1 (WP3) - Open-Safe-Realiable and low power node architecture.
UC5_KPI_FO_02	Edge Node application with Secure connection to the Cloud implemented and running.	True/False	TBD	O2	Guarantee extra-functional properties (dependability, security, timeliness and energy-efficiency) of FRACTAL nodes and systems built using FRACTAL nodes (i.e., FRACTAL systems).	Pillar 2 (WP4) - Low power, safety, security and high-preformance trade-off.
UC5_KPI_FO_03	Edge Node Software and Model update Max Time, guaranteeing data is not corrupted, implemented and running.	< 1 min	TBD	O3	Evaluate and validate the analytics approach by means of AI to help the identification of the largest set of working conditions still preserving safe and secure operational behaviors	Pillar 3 (WP5) - Cognitive & Autonomous Node.
UC5_KPI_FO_04	Software and Model version handling on cloud implemented and running.	True/False	TBD	O4	To integrate fractal communication and remote management features into FRACTAL nodes	Pillar 4 (WP6) - Mutable and fractal communications.

Figure 13 - KPIs for UC5 Implementation Plan to measure the contribution to FRACTAL Objectives

4.5.3 KPI for UC Features

KPIs defined to measure ***how the Implementation Tasks contribute to demonstrate the UC Features*** (defined in the Tab *FRACTAL Features* in the general Excel defined in *WP1 FRACTAL - Requirements_KPIs_Components.xlsx*). Figure 14 shows the complete list of KPIs defined for this purpose.

KPI for UC Feature				Helps to demonstrate the following UC Feature	
KPI ID	Description	Value	Test	ID Feat	Description
UC5_KPI_FT_01	Edge Node has USB-C port	True/False	TBD	F1_CAF	ADAPTABILITY - EXTENSIBILITY - PORT CONNECTION - USB C
UC5_KPI_FT_02	Edge Node has Ethernet connector RJ45	True/False	TBD	F2_CAF	ADAPTABILITY - EXTENSIBILITY - PORT CONNECTION - ETH RJ45
UC5_KPI_FT_03	Edge Node Vitis-AI allows importing and executing ONNX models	True/False	TBD	F3_CAF	ADAPTABILITY - AI - SW - INFERENCE - MODEL - ONNX
UC5_KPI_FT_04	Edge Node Vitis-AI allows importing and executing Yolo V3/V4	True/False	TBD	F4_CAF	ADAPTABILITY - AI - SW - INFERENCE - ALGORITHMS - YOLO V3/V4
UC5_KPI_FT_05	Edge Node inference time allows real-time processing of frames	< 100ms	TBD	F5_CAF	ADAPTABILITY - AI - SW - INFERENCE - REALTIME
UC5_KPI_FT_06	Build & System Integration	True/False	TBD	F6_CAF	ADAPTABILITY - EXTENSIBILITY - PORT CONNECTION - BUILD - SYSTEM INTEGRATION
UC5_KPI_FT_07	AI Inference Accuracy on Model	> 75%	TBD	F7_CAF	ADAPTABILITY - AI - SW - INFERENCE - MODEL - ACCURACY / VALIDATION
UC5_KPI_FT_08	Edge Node has the ability to track locattion	True/False	TBD	F8_CAF	ADAPTABILITY - AI - SW - INFERENCE - LOCATION - NODE
UC5_KPI_FT_09	Edge Node has OpenCV	True/False	TBD	F9_CAF	ADAPTABILITY - AI - SW - LIBRARY - OPENCV
UC5_KPI_FT_10	Cloud Data Set Version Control	True/False	TBD	F10_CAF	ADAPTABILITY - AI - DATA ORCHESTRATION - DATA SET - VERSION CONTROL
UC5_KPI_FT_11	Edge Node frame processing rate > 10fps	>10fps	TBD	F11_CAF	REALIABILITY - RESPONSE TIME - FRAME RATE
UC5_KPI_FT_12	Edge Node allows video processing	True/False	TBD	F12_CAF	CONTEXT_AWARENESS - SENSORS - VIDEO
UC5_KPI_FT_13	Safety Regulation ISO 26262 Automotion	True/False	TBD	F13_CAF	SAFETY - REGULATION - ISO 26262 - CAR - VARIATION OF 61508
UC5_KPI_FT_14	Safety Regulation ISO 61508 Generic	True/False	TBD	F14_CAF	SAFETY - REGULATION - ISO 61508 - Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems
UC5_KPI_FT_15	Safety Regulation CENELEC EN50126/8/9: Railway Industry	True/False	TBD	F15_CAF	SAFETY - REGULATION - CENELEC EN50126/8/9: Railway Industry
UC5_KPI_FT_16	Edge Node in Low Power has ONNX models	True/False	TBD	F16_CAF	LOW POWER - AI - LIBRARY - MODELS - ONNX
UC5_KPI_FT_17	Edge Node allows secure storage of data	True/False	TBD	F17_CAF	SECURITY - SECURE STORAGE
UC5_KPI_FT_18	Edge Node allows Authentication / Authorization	True/False	TBD	F18_CAF	SECURITY - AUTHENTICATION - AUTHORIZATION
UC5_KPI_FT_19	Fractality communication via Ethernet	True/False	TBD	F19_CAF	FRACTALITY - COMMUNICATION / CONNECTIVITY - TECHNOLOGIES - ethernet
UC5_KPI_FT_20	Edge Node is implemented on Versal	True/False	TBD	F20_CAF	OTHER: NON-FUNCTIONAL - PLATFROM - VERSAL
UC5_KPI_FT_21	Edge Node executes LINUX Ooperating	True/False	TBD	F21_CAF	OTHER: NON-FUNCTIONAL - OS - LINUX

Figure 14 - KPIs for UC5 Implementation Plan to measure the contribution to FRACTAL Features

4.6 Implementation plan

4.6.1 Architecture

4.6.1.1 FRACTAL Big Picture

The Use Case integrates several FRACTAL components, both from the edge and the cloud. Those components can be seen in Figure 15,within the context of FRACTAL Big Picture representation.

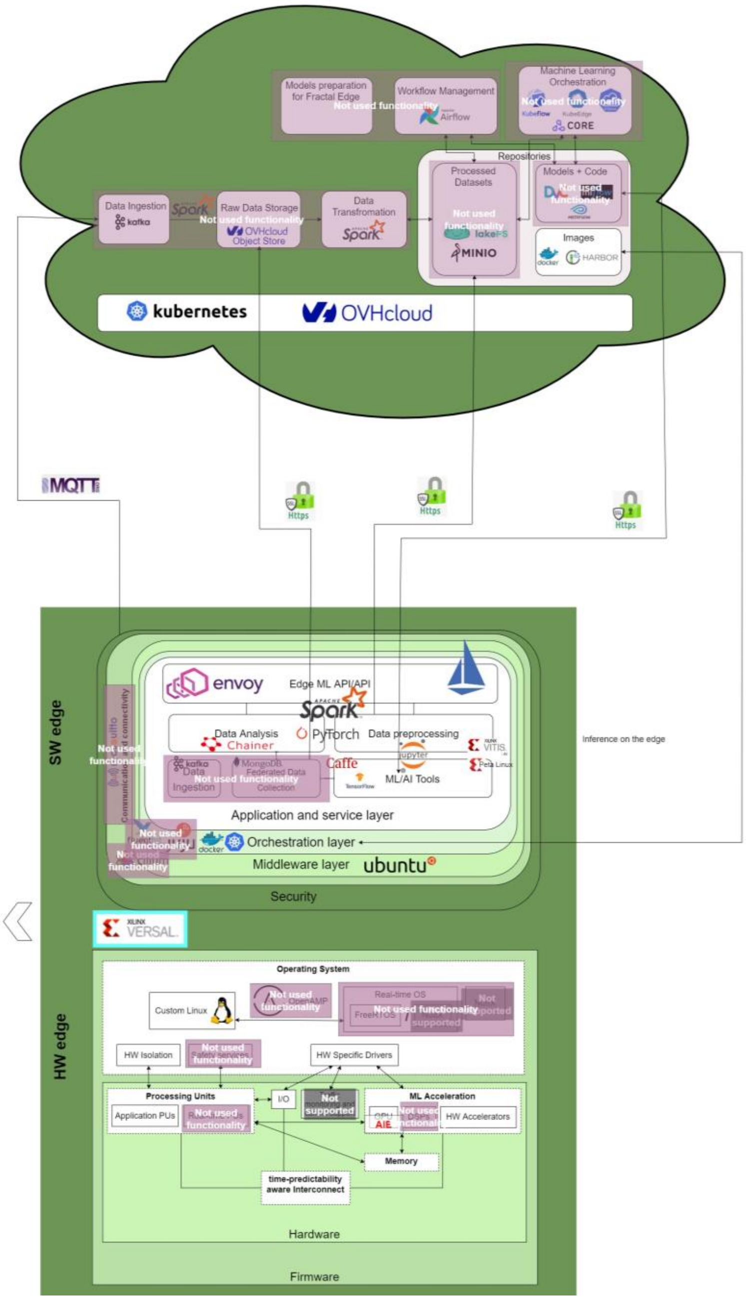


Figure 15 - FRACTAL Big Picture Instantiation for UC5.

The UC5 Applications are mainly contained on the FRACTAL edge node. Safe passenger transfer application and automatic accurate stop applications are wrapped into docker containers that can be orchestrated using Kubernetes and automatically updated from the cloud Harbor Repository. Figure 16 shows a more specific components relationship with specific FRACTAL components involved

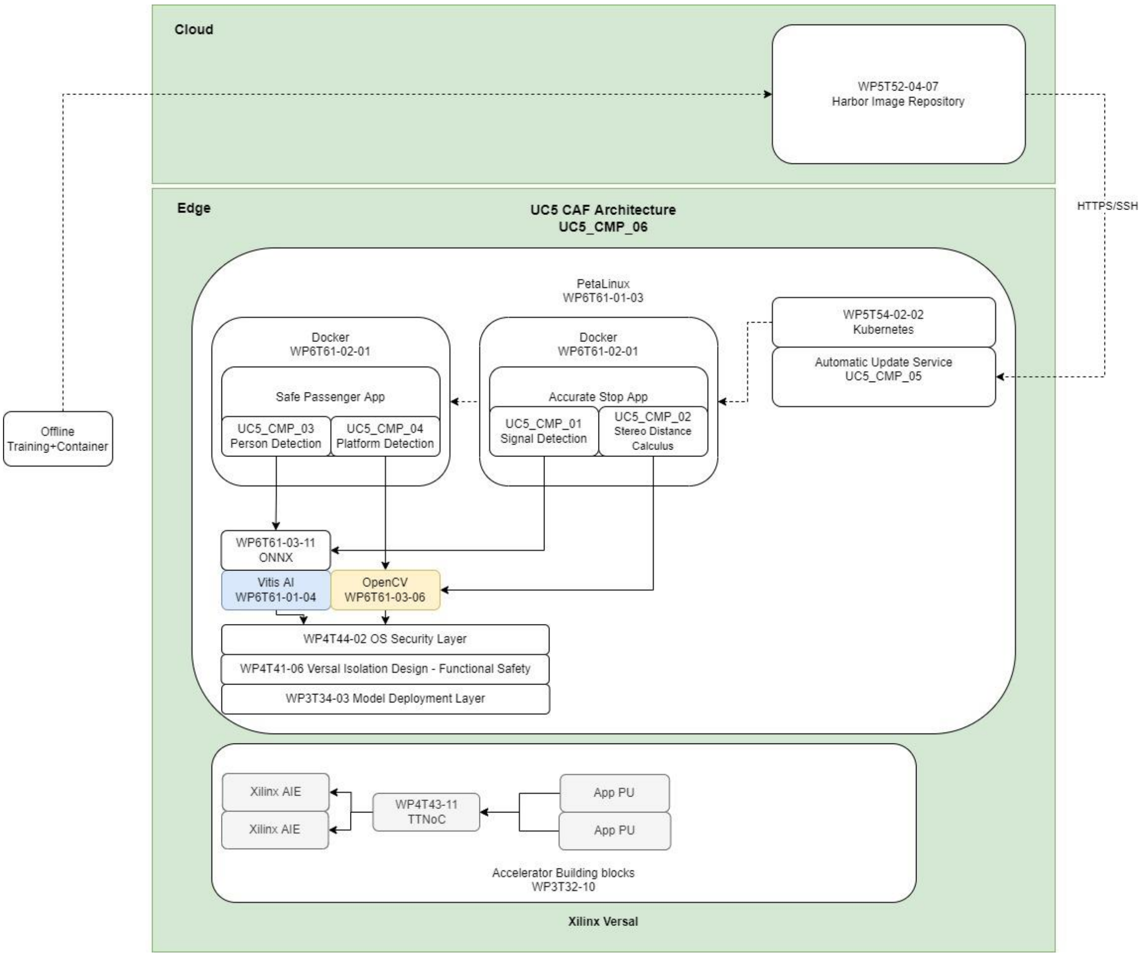


Figure 16 - UC5 Architecture and components

4.6.2 Tasks

4.6.2.1 Chronogram

Figure 17 shows the implementation plan tasks and chronogram for UC5. It is basically divided into four main tasks with subtasks. Sections to follow describe the tasks in some detail.

Implementation Plan

Task ID	Description	M19	M20	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30	M31	M32	M33	M34	M35	M36
UC5_T1	**Basic Target Environment Setup (Versal Edge node)**							■	■	■	■	■	■	■					
UC5_T1_1	Test CAF ONNX models over Vitis AI runtime on Target								■	■	■								
UC5_T1_2	Build OpenCV on for Target (Versal ARM64)								■	■	■								
UC5_T1_3	Test OpenCV Stereo Algorithms on Target									■	■	■							
UC5_T1_4	Build CAF Demonstration Software on Target										■	■	■	■					
UC5_T1_4_1	Build Safe Passenger Transfer application										■	■	■						
UC5_T1_4_2	Build Accurate Stop application											■	■	■					
UC5_T2	**Benchmark Preparation**	■	■	■	■	■	■	■	■	■	■	■	■	■					
UC5_T2_1	Database recording on real environment	■	■	■	■	■	■	■	■	■	■	■	■	■					
UC5_T2_2	Database processing, labelling and splitting into train and validation	■	■	■	■	■	■	■	■	■	■	■	■	■					
UC5_T2_3	Model training	■	■	■	■	■	■	■	■	■	■	■	■	■					
UC5_T3	**Extended Target Environment Setup and automatic update service**													■	■	■			
UC5_T3_1	Test docker hosted application integration with Vitis AI Runtime													■	■	■			
UC5_T3_2	Test edge automatic update services and connection with cloud													■	■	■			
UC5_T3_3	Test cloud repositories version handling													■	■	■			
UC5_T4	**System Evaluation**							■	■	■	■	■	■	■	■	■	■	■	■
UC5_T4_1	Metrics Calculation							■	■	■	■	■	■	■	■	■	■	■	■

Figure 17 - UC Implementation Plan Chronogram

4.6.2.2 Task: UC5_T1 - Basic Target Environment Setup (Versal Edge node)

The sub tasks under this task are devoted to implement the prime basic scenario of CAF UC5 that involves only the Edge Versal-based node of FRACTAL.

4.6.2.2.1 Sub Task: UC5_T1_1 - Test CAF ONNX models over Vitis AI runtime on Target

This task consists of checking whether Xilinx Vitis AI, once installed in the Versal platform, can import and execute successfully the ONNX neural network models. This task allows evaluation of multiple KPI, as for example, inference time, and many more.

4.6.2.2.2 Sub Task: UC5_T1_2 - Build OpenCV on for Target (Versal ARM64)

This task consists of cross-compiling the OpenCV framework and installing it into de the Versal ARM64 platform. The success of OpenCV can be evaluated by executing the *OpenCV standard test battery*.

4.6.2.2.3 Sub Task: UC5_T1_3 - Test OpenCV Stereo Algorithms on Target

This task consists of checking the OpenCV stereo matching algorithms on the Versal Platform. The Accurate Stop application calculates the distance to the stop signals by using one of the stereo matching algorithms provided by OpenCV, so, this task will check if this algorithm executes with an accuracy similar to x86 platforms.

4.6.2.2.4 Sub Task: UC5_T1_4 - Build CAF Demonstration Software on Target

This task consists of integrating into one application both the (a) *Safe Passenger Transfer Application* and the (b) *Accurate Stop Application*. After integration, KPIs are defined to check if inference time is met (it might happen that inference time holds for a single application but not for both applications running together), to check if accuracy holds, etc.

4.6.2.2.5 Sub Task: UC5_T1_4_1 - Build Safe Passenger Transfer application

This task consists of building (cross-compiling) and executing the *Safe Passenger Transfer Application* into the Versal Platform and checking if it reaches the required FPS (Frames Per Second) and an accuracy similar to x86 systems.

4.6.2.2.6 Sub Task: UC5_T1_4_2 - Build Accurate Stop application

This task consists of building (cross-compiling) and executing the *Accurate Stop Application* into the Versal Platform and checking if it reaches the required FPS and an accuracy similar to x86 systems.

4.6.2.3 Task: UC5_T2 - Benchmark Preparation

The sub-tasks under this task are devoted to preparing the data needed to benchmark the FRACTAL-based implementation of UC5 and compare it against the x86 solution.

4.6.2.3.1 Sub Task: UC5_T2_1 - Database recording on real environment

This task consists of recording a video in a real environment. On the one side, both left-right cameras on the front of the train will record the train entering the station and stopping at the platform stop signal. In addition to this, backward cameras will record the passenger area near the train doors. More than 80 hours of video are expected to be recorded.

4.6.2.3.2 Sub Task: UC5_T2_2 - Database processing

This task consists of processing the video captured in the previous task, extracting the frames, labeling the frames, and separating them into the three typical sets for training neural networks: *training set, validation set*, and test set These three groups will be created for both applications (Accurate Stop and Safe Passenger Transfer). More than 40.000 frames are expected to be labeled and separated into different sets.

4.6.2.3.3 Sub Task: UC5_T2_3 - Model training

This task consists of training both neural networks (Accurate Stop and Safe Passenger Transfer) in x86 environment (outside FRACTAL environment) and exporting them into ONNX format to be used later in FRACTAL Edge. KPIs are defined to measure the accuracy obtained during training.

4.6.2.4 Task: UC5_T3 - Extended Target Environment Setup and automatic update service

The sub-tasks under this task are devoted to implement the extended scenario of CAF UC5 that involves both the Edge Versal-based node and Cloud node of FRACTAL.

4.6.2.4.1 Sub Task: UC5_T3_1 - Test docker hosted application integration with Vitis AI Runtime

This task aims to evaluate the integration between docker containerized application and its integration with Vitis AI runtime libraries which execute the accelerated inference on Versal FRACTAL Node. It also aims to measure the impact on libraries' performance when called from a docker environment to preserve the time cycle requirement implementing the automatic update UC extension.

4.6.2.4.2 Sub Task: UC5_T3_2 - Test edge automatic update services and connection with cloud

This task is defined as a node-cloud connection test. Its objective is to guarantee the following points:

- Node-Cloud Connection is established.
- Connection is done under secure protocol.
- Scheduled task on Versal Node can pull SW and AI models from the cloud repositories without data corruption and preserving data confidentiality by applying cyphering.
- In case of detected failure on node SW, SW and Model pull operation is done within a defined amount of time to reduce the failure lifetime updating to a most recent version.

4.6.2.4.3 Sub Task: UC5_T3_3 - Test cloud repositories version handling

This task has the cloud SW and Model handling as targets. It is defined for testing the properties of the Cloud data handling mechanisms to verify that SW and Model describing information integrity is preserved as well as transferred to the node during the update operation. The relevant information for SW is the name of the application/container, the version in format X.Y.Z, and the release date. For the model, the desired information is the name, the release date, and the dataset used to train it as an optional parameter.

4.6.2.5 Task: UC5_T4 - System Evaluation

The only sub task under this task is devoted to compare FRACTAL solution against the state-of-art x86 solution.

4.6.2.5.1 Sub Task: UC5_T4_1 - Metrics Calculation

This task consists of calculating the metrics results for both applications (Accurate Stop and Safe Passenger Transfer) in the FRACTAL Edge node and comparing them with metrics in the x86 solution. This final task will help to clarify how much FRACTAL solution contributes to improving state-of-art solutions in the railways sector. The test of the KPI associated with this task will consist of several criteria, not only accuracy and FPS, but also criteria such as safety, security, updating from the cloud, model management, etc.

4.6.3 Components

This section summarizes the components involved in the Implementation Plan. All the components listed here have been extracted from Tab ***Components*** in the general Excel defined in WP1 ***FRACTAL - Requirements_KPIs_Components.xlsx.***

Components are basically divided into two groups:

- ***Components produced by the UC*** *resulting from executing the Implementation Plan*.
- ***Common FRACTAL Components*** (from WP3, WP4, WP5, WP6) that are needed to execute the Implementation Plan.

Following two sub sections list these components.

4.6.3.1 Components produced by the Implementation Plan

These components, Figure 18, *are* ***produced by executing the tasks*** of the Implementation Plan.

UC Components		
KPI ID	**Name**	**Description**
UC5_CMP_01	Accurate Stop/ Signal Detection	Neural Network yolov3/yolov4 deploy for signalling detection
UC5_CMP_02	Accurate Stop/ Stereo Distance Calculus	OpenCV's stereoSGBM algorithm for stereo distance calculus
UC5_CMP_03	Safe Passenger Transfer/ Person Detection	Neural Network yolov3/yolov4 deploy for person detection near the train
UC5_CMP_04	Safe Passenger Transfer/ Platform Detection	Neural Network yolov3/yolov4 deploy for platform detection
UC5_CMP_05	Automatic Models Update Service	Neural networks models Automatic Update service
UC5_CMP_06	Integrated Demostration Software on Target	Demostration Software Integrating Accurate Stop and Safe Passenger

Figure 18 - Components produced by the execution of UC5 Implementation Plan

4.6.3.2 FRACTAL components needed to execute the Implementation Plan

These components, Figure 19, *are* ***Common FRACTAL Components*** (from WP3, WP4, WP5, WP6) that are ***needed to execute the Implementation Plan***.

FRACTAL Components needed by the UC		
KPI ID	**Name**	**Description**
WP3T32-10	VERSAL accelerator building-blocks	Development of building-blocks for accelerators for VERSAL
WP3T34-03	Versal Model deployment layer	Model deployment on the Versal APU + DPU control from model repository images
WP4T41-06	Versal Isolation Design - Functional Safety	Enhance the common Versal platform to strictly separate functional accesses, services from underlying HW access
WP4T43-11	Time-Triggered Extension Layer for VERSAL NoC	Time Triggered extension layer is an extension layer developed for VERSAL NoC that allow the VERSAL NoC to transfer messages using Time triggered traffic.
WP4T44-02	OS Security Layer	Implementation of security countermeasures in a transversal security layer
WP5T52-06-01	Model preparation for Fractal Edge (Versal Xilinx Vitis AI)	Workflows to compile models for Versal with Xilinx Vitis AI, add containerized toolchain to the cloud
WP5T54-02-02	Kubernetes	Open-Source orchestrator for cluster management and container orchestration.
WP5T52-04-07	Harbor Image repository	Container Registry for Docker Images
WP6T61-01-01	Operating system - Ubuntu	Linux for ARM64 & RISC_V64
WP6T61-01-03	Petalinux	Tools necessary to customize, build and deploy Embedded Linux solutions on Xilinx processing systems. Tailored to accelerate design productivity, the solution works with the Xilinx hardware design tools to ease the development of Linux systems for Versal
WP6T61-01-04	Vitis AI	The Vitis™ AI development environment is Xilinx's development platform for AI inference on Xilinx hardware platforms
WP6T61-03-11	ONNX	cross-platform inference and training machine-learning accelerator
WP6T61-03-06	OpenCV	open-source computer vision and machine learning software library
WP6T61-02-01	Docker	platform as a service product that uses OS-level virtualization to deliver software in packages called containers.
WP6T61-15	Standard C++ Library	Library for C++

Figure 19 - Common FRACTAL components from WP3, WP4, WP5, WP6 needed to execute UC5 Implementation Plan

4.6.4 Traceability relationships of Tasks-Components-KPIs

Finally, this section ***links together*** *tasks*, *components* and *KPIs*. For each Task, the following traceability-relationships are given:

- ***Components***
    - IN Components – Input components needed by the task.
    - OUT Components – Output components produced by the task.
- ***KPIs for UC Implementation Plan***
- ***KPIs for FRACTAL Objectives & Features***

The following subsections detail this information for each task.

4.6.4.1 UC_T1 - Basic Target Environment Setup (Versal Edge node) Tasks

4.6.4.1.1 Task: UC5_T1_1 - Test CAF ONNX models over Vitis AI runtime on Target

Figure 20 shows traceability relationships for Task UC5_T1_1:

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
		UC5_KPI_IP_02	Inference time	< 100 ms			
		UC5_KPI_IP_Req_12	Edge Node with Linux OS	True/False			
		UC5_KPI_IP_Req_11	Edge Node Multi-Interfaces and their Linux Drivers	True/False			
WP3T32-10		UC5_KPI_IP_Req_06	Edge Node with al least 4 cores	True/False	UC5_KPI_FT_01	Edge Node has USB-C port	True/False
WP6T61-01-01		UC5_KPI_IP_Req_07	Edge Node with mutlithreading	True/False	UC5_KPI_FT_02	Edge Node has Ethernet connector RJ45	True/False
WP6T61-01-03		UC5_KPI_IP_Req_09	Edge Node at least 16GB DDR RAM	True/False	UC5_KPI_FT_03	Edge Node Vitis-AI allows importing and executing ONNX models	True/False
WP6T61-01-04		UC5_KPI_IP_Req_08	Edge Node at least 60 GFLOPS	True/False	UC5_KPI_FT_04	Edge Node Vitis-AI allows importing and executing Yolo V3/V4	True/False
WP6T61-03-11		UC5_KPI_IP_Req_13	Platform Release with C++ compiler/crosscompiler	True/False	UC5_KPI_FT_05	Edge Node inference time allows real-time processing of frames	< 100ms
		UC5_KPI_IP_Req_04	Edge ONNX Support	True/False	UC5_KPI_FT_07	AI Inference Accuracy on Model	> 75 %
		UC5_KPI_IP_Req_10	Edge Node with HW Acccelerator	True/False			
		UC5_KPI_IP_Req_05	HW Accelerator Compatible wiht TensorFlow	True/False			
		UC5_KPI_IP_Req_02	Edge Node Inference Time	< 100 ms			

Figure 20 - Task UC5_T1_1 traceability relationship between task, components and KPIs

4.6.4.1.2 Task: UC5_T1_2 - Build OpenCV on for Target (Versal ARM64)

Figure 21 shows traceability relationships for Task UC5_T1_2:

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
WP3T32-10 WP6T61-01-01 WP6T61-01-03 WP6T61-03-06 WP6T61-15		UC5_KPI_IP_03 UC5_KPI_IP_Req_11	Build OpenCV on for Target (Versal ARM64) success Edge OpenCV Support	True/False True/False	UC5_KPI_FT_09 UC5_KPI_FT_12	Edge Node has OpenCV Edge Node allows video processing	True/False True/False

Figure 21 - Task UC5_T1_2 traceability relationship between task, components and KPIs

4.6.4.1.3 Task: UC5_T1_3 - Test OpenCV Stereo Algorithms on Target

Figure 22 shows traceability relationships for Task UC5_T1_3:

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
WP3T32-10 WP6T61-01-01 WP6T61-01-03 WP6T61-03-06		UC5_KPI_IP_04 UC5_KPI_IP_Req_06	Accuracy % with respect to X86 platform Edge OpenCV Support	> 95% True/False			

Figure 22 - Task UC5_T1_3 traceability relationship between task, components and KPIs

4.6.4.1.4 Task: UC5_T1_4 - Build CAF Demonstration Software on Target

Figure 23 shows traceability relationships for Task UC5_T1_4:

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
WP3T32-10 WP4T41-06 WP4T43-11 WP6T61-01-01 WP6T61-01-03 WP6T61-01-04 WP6T61-03-11 WP6T61-03-06 WP6T61-15 UC5_CMP_01 UC5_CMP_02 UC5_CMP_03 UC5_CMP_04	UC5_CMP_06	UC5_KPI_IP_05 UC5_KPI_IP_02 UC5_KPI_IP_04 UC5_KPI_IP_Req_08	Build CAF Demonstration Software on Target success Inference time Accuracy % with respect to X86 platform Edge Node Platform Ruggedized	True/False < 100 ms >= 95% True/False	UC5_KPI_FO_01 UC5_KPI_FT_06 UC5_KPI_FT_17 UC5_KPI_FT_18 UC5_KPI_FT_20 UC5_KPI_FT_21 UC5_KPI_FT_19 UC5_KPI_FT_13 UC5_KPI_FT_14 UC5_KPI_FT_15	Real-time Inference Time and Accurate high perfomance Cognitive AI based node implemented and running. Build & System Integration Edge Node allows secure storage of data Edge Node allows Authentication / Authorization Edge Node is implemnted on Versal Fractality communication via Ethernet Edge Node executes LINUX Ooperating System Safety Regulation ISO 26262 Automotion Safety Regulation ISO 61508 Generic Safety Regulation CENELEC EN50126/8/9: Railway Industry	< 100ms True/False True/False True/False True/False True/False True/False True/False True/False True/False

Figure 23 - Task UC5_T1_4 traceability relationship between task, components and KPIs

4.6.4.1.5 Task: UC5_T1_4_1 - Build Safe Passenger Transfer application

Figure 24 - shows traceability relationships for Task UC5_T1_4_1:

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
WP3T32-10 WP6T61-01-01 WP6T61-01-03 WP6T61-01-04 WP6T61-03-11 WP6T61-03-06 WP6T61-15	UC5_CMP_03 UC5_CMP_04	UC5_KPI_IP_06 UC5_KPI_IP_02 UC5_KPI_IP_04	Build Safe Passenger Transfer application success In Inference time Accuracy % with respect to X86 platform	True/False < 100 ms >= 95%			

Figure 24 - Task UC5_T1_4_1 traceability relationship between task, components and KPIs

4.6.4.1.6 Task: UC5_T1_4_2 - Build Accurate Stop application

Figure 25 shows traceability relationships for Task UC5_T1_4_2:

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
WP3T32-10 WP6T61-01-01 WP6T61-01-03 WP6T61-01-04 WP6T61-03-11 WP6T61-03-06 WP6T61-15	UC5_CMP_01, UC5_CMP_02	UC5_KPI_IP_07 UC5_KPI_IP_02 UC5_KPI_IP_04	Build Accurate Stop application success Inference time Accuracy % with respect to X86 platform	True/False < 100 ms >= 95%	UC5_KPI_FT_08	Edge Node has the ability to track locattion	True/False

Figure 25 - Task UC5_T1_4_2 traceability relationship between task, components and KPIs

4.6.4.2 UC_T2 - Benchmark Preparation Tasks

4.6.4.2.1 Task: UC5_T2_1 - Database recording on real environment

Figure 26 shows traceability relationships for Task UC5_T2_1:

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
Outside Fractal		UC5_KPI_IP_09	Hours of video recorded	> 80			

Figure 26 - Task UC5_T2_1 traceability relationship between task, components and KPIs

4.6.4.2.2 Task: UC5_T2_2 - Database processing

Figure 27 shows traceability relationships for Task UC5_T2_2:

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
Outside Fractal		UC5_KPI_IP_10	Image Database size	>40000			

Figure 27 - Task UC5_T2_2 traceability relationship between task, components and KPIs

4.6.4.2.3 Task: UC5_T2_3 - Model training

Figure 28 shows traceability relationships for Task UC5_T2_3:

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
Outside Fractal		UC5_KPI_IP_11	Model accuracy over test database	> 75%			

Figure 28 - Task UC5_T2_3 traceability relationship between task, components and KPIs

4.6.4.3 UC_T3 - Extended Target Environment Setup and automatic update service Tasks

4.6.4.3.1 Task: UC5_T3_1 - Test docker hosted application integration with Vitis AI Runtime

Figure 29 shows traceability relationships for Task UC5_T3_1:

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
WP3T32-10 WP3T34-03 WP5T52-06-01 WP5T52-04-07 UC5_CMP_06		UC5_KPI_IP_13	Inference time	< 100 ms			

Figure 29 - Task UC5_T3_1 traceability relationship between task, components and KPIs

4.6.4.3.2 Task: UC5_T3_2 - Test edge automatic update services and connection with cloud

Figure 30 shows traceability relationships for Task UC5_T3_2:

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
WP3T32-10 WP4T44-02 WP5T52-06-01 WP5T52-04-07 UC5_CMP_06		UC5_KPI_IP_14	Working under secure connection	True/False	UC5_KPI_FO_02 UC5_KPI_FO_03	Edge Node application with Secure connection to the Cloud implemented and running. Edge Node Software and Model update time<1 min (i.e. example), guaranteeing data is not corrupted, implemented and running.	True/False < 1 min

Figure 30 - Task UC5_T3_2 traceability relationship between task, components and KPIs

4.6.4.3.3 Task: UC5_T3_3 - Test cloud repositories version handling

Figure 31 shows traceability relationships for Task UC5_T3_3:

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
WP5T52-06-01 WP5T52-04-07 WP5T54-02-02		UC5_KPI_IP_15 UC5_KPI_IP_16	Model and docker image cloud hosting success Cloud repositories version handling success	True/False True/False	UC5_KPI_FO_04 UC5_KPI_FT_10	Software and Model version handling on cloud implemented and running. Cloud Data Set Version Control	True/False True/False

Figure 31 - Task UC5_T3_3 traceability relationship between task, components and KPIs

4.6.4.4 UC_T4 - System Evaluation Tasks

4.6.4.4.1 Task: UC5_T4_1 - Metrics Calculation

Figure 32 shows traceability relationships for Task UC5_T4_1:

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
WP3T32-10 WP6T61-01-01 WP6T61-01-03 WP6T61-01-04 WP6T61-03-11 WP6T61-03-06 WP6T61-15 UC5_CMP_01 UC5_CMP_02 UC5_CMP_03 UC5_CMP_04		UC5_KPI_IP_17	Metrics obtained for defined model both in X86 and Versal	True/False	UC5_KPI_FO_00	Fractal Technology helps improving State-of-Art in Railways Sector	True/False

Figure 32 - Task UC5_T4_1 traceability relationship between task, components and KPIs

4.7 Justification plan

This section defines the justification methods (like demonstrations, tests, simulations, calculations, etc.) for KPIs evaluation, Use Case Requirements validation and Components validation.

4.7.1 KPI evaluation method

4.7.1.1 KPI for Implementation Plan

4.7.1.1.1 UC5_KPI_IP_01

- ***Description***: All subtask success
- ***Result type***: True/False
- ***Evaluation method***: The KPI is True when all subtasks under the task succeed.

4.7.1.1.2 UC5_KPI_IP_02

- ***Description***: Inference time
- ***Result type***: < 100 ms
- ***Evaluation method***: The KPI is True when the inference time of both applications (Accurate Stop and Safe Passenger), executed separately and considering preprocessing + network inference + postprocessing, is less than or equal to 100ms, it is, the applications can process 10 FPS. In this case, front RGB cameras capture frames of 1280x960 pixels, and rear pointing RGB cameras may reach up to 1920x1080 pixels. The test consists of passing the video (mp4 format) to the application and measuring the fps achieved.

4.7.1.1.3 UC5_KPI_IP_03

- ***Description***: Build OpenCV on for Target (Versal ARM64) success
- ***Result type***: True/False
- ***Evaluation method***: OpenCV must be built in Versal Platform. OpenCV must be downloaded in Linux x86 and cross-compiled for ARM64 and installed in Versal. After installation OpenCV provides and standard benchmark of tests that checks that OpenCV is working properly. The test consists of passing this benchmark.

4.7.1.1.4 UC5_KPI_IP_04

- ***Description***: Accuracy % with respect to X86 platform
- ***Result type***: > 95%
- ***Evaluation method***: This KPI applies to OpenCV Stereo Vision on Target Platform, in this case Versal. This test passes if OpenCV Stereo Vision benchmarks executed in Versal achieve > 95% of the Accuracy obtained by these benchmark algorithms in Windows.

4.7.1.1.5 UC5_KPI_IP_05

- ***Description***: Build CAF Demonstration Software on Target success
- ***Result type***: True/False
- ***Evaluation method***: CAF demonstrator on Versal integrates both the *Accurate Stop Application* and *Safe Passenger Transfer Application* into only

one application. This KPI checks that both applications are integrated and run together in the final platform and keep satisfying KPIS UC5_KPI_IP_02 (inference time) and UC5_KPI_IP_04 (accuracy % with respect to the X86 platform). This prevents from integration problems, when applications executed alone in the target reach the corresponding KPIs, but when executed together, do not reach the KPIs.

4.7.1.1.6 UC5_KPI_IP_06

- **Description**: Build Safe Passenger Transfer application success
- **Result type**: True/False
- **Evaluation method**: This KPI checks if the Safe Passenger application compiles and executes successfully in Versal platform. Other KPIs check performance.

4.7.1.1.7 UC5_KPI_IP_07

- **Description**: Build Accurate Stop application success
- **Result type**: True/False
- **Evaluation method**: This KPI checks if the Accurate Stop application compiles and executes successfully in Versal platform. Other KPIs check performance.

4.7.1.1.8 UC5_KPI_IP_08

- **Description**: All subtask success
- **Result type**: True/False
- **Evaluation method**: The KPI is True when all subtasks under the task succeed.

4.7.1.1.9 UC5_KPI_IP_09

- **Description**: Hours of video recorded
- **Result type**: > 80
- **Evaluation method**: This test is passed if more than 80 hours of video have been recorded, 75% hours from the front cameras of the train and 25% hours from the camera pointing to the rear of the train, capturing train doors.

4.7.1.1.10 UC5_KPI_IP_10

- **Description**: Image Database size
- **Result type**: > 4000
- **Evaluation method**: This test is passed if 4000 images are extracted from the front camera videos (only for stop signal and platform detection) and are divided into the following groups: (a) 2250 images are labeled (box around stop signal and platform) for the training set, (b) 750 images are labeled (box around stop signal and platform) for the Development Set, and the rest, (c) 1000 images, are also labeled and selected for the Test Set.

4.7.1.1.11 UC5_KPI_IP_11

- **Description**: Model accuracy over test database

- **Result type**: > 75%
- **Evaluation method**: The stop signal and platform detection models are trained offline. This test passes if model accuracy over the Test Set after training is > 75%.

4.7.1.1.12 UC5_KPI_IP_12

- **Description**: All subtask success
- **Result type**: True/False
- **Evaluation method**: The KPI is True when all subtasks under the task succeed.

4.7.1.1.13 UC5_KPI_IP_13

- **Description**: Inference time (of containerized application)
- **Result type**: < 100 ms
- **Evaluation method**: This test passes if the docker hosted CAF Demonstration Software (integrated with Vitis AI Runtime) on Target Versal executes in the Edge with an inference time less than 100 ms. Inference time includes preprocessing + network inference + postprocessing.

4.7.1.1.14 UC5_KPI_IP_14

- **Description**: Working under secure connection success
- **Result type**: True/False
- **Evaluation method**: This test passes if the edge models automatic update services and connection with cloud works under a secure connection. The test consists of checking the following aspects:
  - Is communication encrypted?
  - Is communication authenticated? Minimum one factor authentication.
  - Are authentication credentials exposed to the network? Protected credentials within the cloud and edge.

4.7.1.1.15 UC5_KPI_IP_15

- **Description**: Model and docker image cloud hosting success
- **Result type**: True/False
- **Evaluation method**: The test passes if the cloud can host the docker images containing the models and the applications of Accurate Stop and Safe Passenger transfer.

4.7.1.1.16 UC5_KPI_IP_16

- **Description**: Cloud repositories version handling success
- **Result type**: True/False
- **Evaluation method**: The test passes if the cloud can host and handle different versions of the docker images containing the models and the applications of Accurate Stop and Safe Passenger transfer.

4.7.1.1.17 UC5_KPI_IP_17

- **Description**: Metrics obtained for defined model both in X86 and Versal

- **Result type**: True/False
- **Evaluation method**: The test passes if for both applications, Accurate Stop and Safe Passenger Transfer, the following metrics will be obtained for both the x86 platform and the Versal platform (see section 4.3.4 for a detailed description):
    - Average Precision (AP)
    - Mean Average Precision (mAP)
    - Precision Threshold curve
    - Recall Threshold curve
    - Precision Recall curves (PR Curves).

4.7.1.2 KPI for FRACTAL Objectives

4.7.1.2.1 UC5_KPI_FO_00

- **Description**: FRACTAL Technology helps improving State-of-Art in Railways Sector.
- **Result type**: True/False
- **Evaluation method**: After collecting all metrics obtained for the implementation plan (UC5_KPI_IP_**) this test will decide if FRACTAL technology helps CAF improve the current state-of-art in the Railways Sector. CAF has already implemented some of the functions (Accurate Stop and Safe Passenger Transfer) with NVIDIA Jetson AGX Xavier technology. Therefore, this KPI will be necessarily a complex decision process by CAF, comparing results from FRACTAL with results from NVIDIA Jetson AGX Xavier. This test will compare performance in the Edge, but however, undoubtedly, FRACTAL cloud services will definitively be an improvement for CAF.

4.7.1.2.2 UC5_KPI_FO_01

- **Description**: Real-time Inference Time and Accurate high performance Cognitive AI based node implemented and running.
- **Result type**: < 100 ms
- **Helps to demonstrate FRACTAL Objective:** O1 - Design and Implement an Open-Safe-Reliable Platform to Build Cognitive Edge Nodes of Variable Complexity.
- **Relates to FRACTAL Pillar:** Pillar 1 (WP3) - Open-Safe-Reliable and low power node architecture.
- **Evaluation method**: FRACTAL Edge node, Versal, and CAF applications running on the target, contribute to Pillar 1 as they represent an open-safe-reliable platform with an AI-based node. This test will pass if the combined inference time of both applications in the Edge is less than 100ms.

4.7.1.2.3 UC5_KPI_FO_02

- **Description**: Edge Node application with Secure connection to the Cloud implemented and running.
- **Result type**: True/False

- **Helps to demonstrate FRACTAL Objective:** O2 - Guarantee extra-functional properties (dependability, security, timeliness, and energy-efficiency) of FRACTAL nodes and systems built using FRACTAL nodes (i.e., FRACTAL systems).
- **Relates to FRACTAL Pillar:** Pillar 2 (WP4) - Low power, safety, security and high-performance trade-off.
- **Evaluation method**: This test consists of checking if FRACTAL Edge node, Versal, and CAF applications running on the target, provide at least one the extra-functional properties, in the case of CAF, security. The test (similar to UC5_KPI_IP_14) consists of checking the following aspects:
  - Is communication encrypted?
  - Is communication authenticated? Minimum one factor authentication.
  - Are authentication credentials exposed to the network? Protected credentials within the cloud and edge.

4.7.1.2.4 UC5_KPI_FO_03

- **Description**: Edge Node Software and Model update Max Time, guaranteeing data is not corrupted, implemented and running.
- **Result type**: < 1 min
- **Helps to demonstrate FRACTAL Objective:** O3 - Evaluate and validate the analytics approach by means of AI to help the identification of the largest set of working conditions still preserving safe and secure operational behaviors.
- **Relates to FRACTAL Pillar:** Pillar 3 (WP5) - Cognitive & Autonomous Node.
- **Evaluation method**: This test passes if the Edge Node software (applications + neural networks models) can be updated in less than 1 minute guaranteeing is not corrupted. The test will be carried out by triggering and update, measuring the time, checking that the update is not corrupted and checking if the update is up and running.

4.7.1.2.5 UC5_KPI_FO_04

- **Description**: Software and Model version handling on cloud implemented and running.
- **Result type**: True/False
- **Helps to demonstrate FRACTAL Objective:** O4 - To integrate FRACTAL communication and remote management features into FRACTAL nodes.
- **Relates to FRACTAL Pillar:** Pillar 4 (WP6) - Mutable and FRACTAL communications.
- **Evaluation method**: Introduce several versions for SW and Model on Cloud repository whit different version numbers and check that they remain differentiable.

4.7.1.3 KPI for FRACTAL Features

4.7.1.3.1 UC5_KPI_FT_01

- **Description**: Edge Node has USB-C port
- **Result type**: True/False

- **Helps to demonstrate UC Feature**: F1_CAF - ADAPTABILITY - EXTENSIBILITY - PORT CONNECTION - USB C
- **Evaluation method**: This test passes if the Edge node has at least one usable USB-C port from which data can be read (models and test data, in this case, mp4 files).

4.7.1.3.2 UC5_KPI_FT_02

- **Description**: Edge Node has Ethernet connector RJ45
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F2_CAF - ADAPTABILITY - EXTENSIBILITY - PORT CONNECTION - ETH RJ45
- **Evaluation method**: This test passes if the Edge node provides and Ethernet connector from which the target can be connected. In this case a SSH connection will be tried for testing.

4.7.1.3.3 UC5_KPI_FT_03

- **Description**: Edge Node Vitis-AI allows importing and executing ONNX models
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F3_CAF - ADAPTABILITY - AI - SW - INFERENCE - MODEL - ONNX
- **Evaluation method**: This test passes if both, the Accurate Stop application and Safe Passenger Transfer application, can load at start-up time the neural network models provided in ONNX format.

4.7.1.3.4 UC5_KPI_FT_04

- **Description**: Edge Node Vitis-AI allows importing and executing Yolo V3/V4
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F4_CAF - ADAPTABILITY - AI - SW - INFERENCE - ALGORITHMS - YOLO V3/V4
- **Evaluation method**: This test passes if both, the Accurate Stop application and Safe Passenger Transfer application, can load at start-up time the neural network models provided in ONNX format, can be executed and allow making inference with the models producing outputs (Stop Signal identification in case of Accurate Stop) and (Person/obstacles identification in case of Safe Passenger Transfer).

4.7.1.3.5 UC5_KPI_FT_05

- **Description**: Edge Node inference time allows real-time processing of frames
- **Result type**: < 100ms
- **Helps to demonstrate UC Feature**: F5_CAF - ADAPTABILITY - AI - SW - INFERENCE - REALTIME
- **Evaluation method**: This test passes if CAF Demonstrator (Accurate Stop + Safe Passenger Transfer) reaches 10 fps.

4.7.1.3.6 UC5_KPI_FT_06

- **Description**: Build & System Integration
- **Result type**: True/False

- **Helps to demonstrate UC Feature**: F6_CAF - ADAPTABILITY - EXTENSIBILITY - PORT CONNECTION - BUILD - SYSTEM INTEGRATION
- **Evaluation method**: Docker container can successfully access the HW acceleration.

4.7.1.3.7 UC5_KPI_FT_07

- **Description**: AI Inference Accuracy on Model
- **Result type**: > 75%
- **Helps to demonstrate UC Feature**: F7_CAF - ADAPTABILITY - AI - SW - INFERENCE - MODEL - ACCURACY / VALIDATION
- **Evaluation method**: mAP calculated over validation dataset giving more than 75% on the X86 platform. Test pass if accuracy on FRACTAL is also more than 75% and equal (+-1%) to X86 evaluation.

4.7.1.3.8 UC5_KPI_FT_08

- **Description**: Edge Node can track location
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F8_CAF - ADAPTABILITY - AI - SW - INFERENCE - LOCATION - NODE
- **Evaluation method**: This test passes if the train can track location. In the case of UC5, the test passes if the Accurate Stop application can estimate the distance with respect to the platform stop signal (it is a relative position of the train with respect to the stop point). However, stop precision cannot be measured, as this would imply a real connection of the FRACTAL node to the ATO control of the train, and taking stop precision measurements of the whole system integrated, which is out of the scope of the project.

4.7.1.3.9 UC5_KPI_FT_09

- **Description**: Edge Node has OpenCV
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F9_CAF - ADAPTABILITY - AI - SW - LIBRARY - OPENCV
- **Evaluation method**: This test passes if OpenCV is successfully installed in the node and passes OpenCV standard benchmarks.

4.7.1.3.10 UC5_KPI_FT_10

- **Description**: Cloud Data Set Version Control
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F10_CAF - ADAPTABILITY - AI - DATA ORCHESTRATION - DATA SET - VERSION CONTROL
- **Evaluation method**: This test passes if the Cloud service has a Data Set Version control service.

4.7.1.3.11 UC5_KPI_FT_11

- **Description**: Edge Node frame processing rate > 10fps
- **Result type**: >10fps
- **Helps to demonstrate UC Feature**: F11_CAF - REALIABILITY - RESPONSE TIME - FRAME RATE

- **Evaluation method**: This test passes if the node can execute CAF Demonstrator with a frame rate of 10fps.

4.7.1.3.12 UC5_KPI_FT_12

- **Description**: Edge Node allows video processing
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F12_CAF - CONTEXT_AWARENESS - SENSORS - VIDEO
- **Evaluation method**: This test passes if the node can be fed with direct video input.

4.7.1.3.13 UC5_KPI_FT_13

- **Description**: Safety Regulation ISO 26262 Automotion
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F13_CAF - SAFETY - REGULATION - ISO 26262 - CAR - VARIATION OF 61508
- **Evaluation method**: FRACTAL Developer WP provide documentation.

4.7.1.3.14 UC5_KPI_FT_14

- **Description**: Safety Regulation ISO 61508 Generic
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F14_CAF - SAFETY - REGULATION - ISO 61508 - Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems
- **Evaluation method**: FRACTAL Developer WP provide documentation.

4.7.1.3.15 UC5_KPI_FT_15

- **Description**: Safety Regulation CENELEC EN50126/8/9: Railway Industry
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F15_CAF - SAFETY - REGULATION - CENELEC EN50126/8/9: Railway Industry
- **Evaluation method**: FRACTAL Developer WP provide documentation.

4.7.1.3.16 UC5_KPI_FT_16

- **Description**: Edge Node in Low Power has ONNX models
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F16_CAF - LOW POWER - AI - LIBRARY - MODELS - ONNX
- **Evaluation method**: This test passes if the Edge Node can execute processes that load ONNX models, as for example Accurate Stop application and Safe Passenger Transfer application.

4.7.1.3.17 UC5_KPI_FT_17

- **Description**: Edge Node allows secure storage of data
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F17_CAF - SECURITY - SECURE STORAGE

- **Evaluation method**: This test passes if data stored in the Edge Node has secured storage mechanisms, as for example, encryption, access authentication, etc.

4.7.1.3.18 UC5_KPI_FT_18

- **Description**: Edge Node allows Authentication / Authorization
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F18_CAF - SECURITY - AUTHENTICATION - AUTHORIZATION
- **Evaluation method**: This test passes if the Edge Node has authenticated and authorized access.

4.7.1.3.19 UC5_KPI_FT_19

- **Description**: Fractality communication via Ethernet
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F19_CAF - FRACTALITY - COMMUNICATION / CONNECTIVITY - TECHNOLOGIES - ethernet
- **Evaluation method**: This test passes if the Edge Node has a usable ethernet port.

4.7.1.3.20 UC5_KPI_FT_20

- **Description**: Edge Node is implemented on Versal
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F20_CAF - OTHER: NON-FUNCTIONAL - PLATFROM - VERSAL
- **Evaluation method**: This test passes if the Versal Edge Node is available in FRACTAL.

4.7.1.3.21 UC5_KPI_FT_21

- **Description**: Edge Node executes LINUX Operating System
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F21_CAF - OTHER: NON-FUNCTIONAL - OS - LINUX
- **Evaluation method**: This test passes if the Versal Edge Node has Linux installed.

4.7.2 Use Case Requirement Validation methods

Use case requirements validation methods are defined under the KPI defined for Use Case Requirements.

4.7.2.1.1 UC5_KPI_IP_Req_01

- **Description**: Edge Node Platform Ruggedized
- **Result type**: True/False
- **Evaluation method**: Railway regulation compliance (EN50155, EN50125, EN45545, EN50121, UNE EN 61373)

4.7.2.1.2 UC5_KPI_IP_Req_02

- **Description**: Edge Node Inference Time
- **Result type**: < 100 ms
- **Evaluation method**: This test passes if the Edge Node Versal allows for an inference time of less than 100ms for both applications, the Accurate Stop and Safe Passenger Transfer, working simultaneously. The test will carry out by feeding two videos, one of the train entering the station, and the other of people around the train doors. Both applications should reach 10 FPS.

4.7.2.1.3 UC5_KPI_IP_Req_03

- **Description**: Edge OpenCV Support
- **Result type**: True/False
- **Evaluation method**: This test passes if the Edge Node Versal has OpenCV installed and running, and OpenCV passes the standard OpenCV benchmarks.

4.7.2.1.4 UC5_KPI_IP_Req_04

- **Description**: Edge ONNX Support
- **Result type**: True/False
- **Evaluation method**: This test passes if Edge node Versal can execute processes that load ONNX models, as for example Accurate Stop application and Safe Passenger Transfer application. The test passes if both processes can load ONNX models.

4.7.2.1.5 UC5_KPI_IP_Req_05

- **Description**: HW Accelerator Compatible with TensorFlow
- **Result type**: True/False
- **Evaluation method**: Test passes if HW accelerator API is compatible with Tensorflow API

4.7.2.1.6 UC5_KPI_IP_Req_06

- **Description**: Edge Node with at least 4 cores
- **Result type**: True/False
- **Evaluation method**: This test passes if the Edge Node has at least 4 cores.

4.7.2.1.7 UC5_KPI_IP_Req_07

- **Description**: Edge Node with multithreading
- **Result type**: True/False
- **Evaluation method**: This test passes if the Edge node allows multiple threads per core.

4.7.2.1.8 UC5_KPI_IP_Req_08

- **Description**: Edge Node at least 60 GFLOPS
- **Result type**: True/False
- **Evaluation method**: Inference speed test, less than 100 ms per inference required on CAF Yolo V4 model.

4.7.2.1.9 UC5_KPI_IP_Req_09

- **Description**: Edge Node at least 16GB DDR RAM

- **Result type**: True/False
- **Evaluation method**: This test passes if the Edge Node has at least 16GB of RAM.

4.7.2.1.10 UC5_KPI_IP_Req_10

- **Description**: Edge Node with HW Accelerator
- **Result type**: True/False
- **Evaluation method**: This test passes if the Edge Node has an HW Accelerator implemented and if EDDL can make use of the acceleration.

4.7.2.1.11 UC5_KPI_IP_Req_11

- **Description**: Edge Node Multi-Interfaces and their Linux Drivers
- **Result type**: True/False
- **Evaluation method**: Speed transmission test for the available required interfaces (Ethernet), NFS with at least 100 Mbps.

4.7.2.1.12 UC5_KPI_IP_Req_12

- **Description**: Edge Node with Linux OS
- **Result type**: True/False
- **Evaluation method**: This test passes if the Edge Node has Linux OS installed and running.

4.7.2.1.13 UC5_KPI_IP_Req_13

- **Description**: Platform Release with C++ compiler/cross compiler
- **Result type**: True/False
- **Evaluation method**: This test passes if the Edge Node is released along with the right cross compilers to cross compile C++ code, possibly including EDDL and OpenCV functions, and generate code for the Edge Node.

4.7.3 Components Validation

Components used by the Use Case can be divided into two groups: *specific* components produced by the Use Case, and general *common* FRACTAL Components used by the Use Case.

The validation of Use Case Specific components is done through the corresponding KPIs. However, FRACTAL Common Components cannot be validated by just one UC, therefore, validation through this Use Case can be considered only as just a partial validation of the component.

4.7.3.1 Case Specific Components

4.7.3.1.1 UC5_CMP_01- Accurate Stop/ Signal Detection

This component basically consists of a Neural Network YOLOv3/YOLOv4 deployed into the Versal platform for stop signal detection. Validation is done by successfully executing the tests of the following KPIs (see the corresponding KPI for details of the test):

- UC5_KPI_IP_07 – Signal Detection application success (app compiles and execute)
- UC5_KPI_IP_02 - Inference time of Signal Detection is < 100 ms
- UC5_KPI_IP_04 - Accuracy % of Signal Detection with respect to X86 platform is >= 95%

4.7.3.1.2 UC5_CMP_02- Accurate Stop/ Stereo Distance Calculus

This component basically consists of an OpenCV's stereo SGBM algorithm for stereo distance calculus deployed into the Versal platform to estimate the distance from the train to the stop signal. The algorithm works by matching signal detections front left/right camera in the front. Validation is done by successfully executing the tests of the following KPIs (see the corresponding KPI for details of the test):

- UC5_KPI_IP_07 – Distance Calculus application success (app compiles and execute)
- UC5_KPI_IP_02 - Inference time of Distance Calculus is < 100 ms
- UC5_KPI_IP_04 - Accuracy % of Distance Calculus with respect to X86 platform is >= 95%

4.7.3.1.3 UC5_CMP_03- Safe Passenger Transfer/ Person Detection

This component basically consists of a Neural Network YOLOv3/YOLOv4 deployed into the Versal platform for person detection near the train doors. Validation is done by successfully executing the tests of the following KPIs (see the corresponding KPI for details of the test):

- UC5_KPI_IP_06 – Person Detection application success (app compiles and execute)
- UC5_KPI_IP_02 - Inference time of Person Detection is < 100 ms
- UC5_KPI_IP_04 - Accuracy % of Person Detection with respect to X86 platform is >= 95%

4.7.3.1.4 UC5_CMP_04- Safe Passenger Transfer/ Platform Detection

This component basically consists of a Neural Network YOLOv3/YOLOv4 deployed into the Versal platform for platform detection. Validation is done by successfully executing the tests of the following KPIs (see the corresponding KPI for details of the test):

- UC5_KPI_IP_06 – Platform Detection application success (app compiles and execute)
- UC5_KPI_IP_02 - Inference time of Platform Detection is < 100 ms
- UC5_KPI_IP_04 - Accuracy % of Platform Detection with respect to X86 platform is >= 95%

4.7.3.1.5 UC5_CMP_05- Automatic Models Update Service

This component is the client part for the Automatic Update service of models in the cloud. Validation is done by successfully executing the tests of the following KPIs (see the corresponding KPI for details of the test):

- UC5_KPI_IP_14 - Working under secure connection.
- UC5_KPI_FO_02 - Edge Node application with Secure connection to the Cloud implemented and running.
- UC5_KPI_FO_03 - Edge Node Software and Model update time < 1 min, guaranteeing data is not corrupted, implemented and running.

4.7.3.1.6 UC5_CMP_06- Integrated Demonstration Software on Target

This component is the Integration of previous components: Accurate Stop, Safe Passenger and Platform detection components, and deployed into the Versal platform. Validation is done by successfully executing the tests of the following KPIs (see the corresponding KPI for details of the test):

- UC5_KPI_IP_05 - Build CAF Demonstration Software on Versal success (app compiles and execute).
- UC5_KPI_IP_02 – Simultaneous Inference time of all integrated applications (Signal Detection, Platform Detection and Person Detection) is less than < 100 ms.
- UC5_KPI_IP_04 - Accuracy % with respect to X86 platform of all integrated applications (Signal Detection, Platform Detection and Person Detection) is >= 95%.
- UC5_KPI_IP_Req_08 - Edge Node Platform Ruggedized.

4.7.3.2 FRACTAL Common Components

4.7.3.2.1 WP3T32-10 - Versal accelerator building-blocks

This component consists of the development of building-blocks for accelerators for Versal. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_01 - Accurate Stop/ Signal Detection
- UC5_CMP_02 - Accurate Stop/ Stereo Distance Calculus
- UC5_CMP_03 - Safe Passenger Transfer/ Person Detection
- UC5_CMP_04 - Safe Passenger Transfer/ Platform Detection

4.7.3.2.2 WP3T34-03 - Versal Model deployment layer

This component consists of the model deployment on the Versal APU + DPU control from model repository images. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_06 - Integrated Demonstration Software on Target

4.7.3.2.3 WP5T52-04-07 - Harbor Image repository

This component consists of the Container Registry for Docker Images. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_06 - Integrated Demonstration Software on Target

4.7.3.2.4 WP4T41-06 - Versal Isolation Design

This component consists of enhancing the common Versal platform to strictly separate functional accesses, services from underlying HW access. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_01 - Accurate Stop/ Signal Detection
- UC5_CMP_02 - Accurate Stop/ Stereo Distance Calculus
- UC5_CMP_03 - Safe Passenger Transfer/ Person Detection
- UC5_CMP_04 - Safe Passenger Transfer/ Platform Detection

4.7.3.2.5 WP4T43-11 - Time-Triggered Extension Layer for Versal NoC

This component consists of a Time-Triggered extension layer that is an extension layer developed for Versal NoC (Network on Chip) that allows the Versal NoC to transfer messages using Time triggered traffic. Partial validation is done by successfully executing the tests of the KPIs of the Use Case specific components that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_01 - Accurate Stop/ Signal Detection
- UC5_CMP_02 - Accurate Stop/ Stereo Distance Calculus
- UC5_CMP_03 - Safe Passenger Transfer/ Person Detection
- UC5_CMP_04 - Safe Passenger Transfer/ Platform Detection

4.7.3.2.6 WP4T44-02 - OS Security Layer

This component consists of an implementation of security countermeasures in a transversal security layer. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_06 - Integrated Demonstration Software on Target

4.7.3.2.7 WP5T52-06-01 - Model preparation for FRACTAL Edge (Versal Xilinx Vitis AI)

This component consists of the workflows to compile models for Versal with Xilinx Vitis AI and add containerized toolchain to the cloud. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_01 - Accurate Stop/ Signal Detection
- UC5_CMP_02 - Accurate Stop/ Stereo Distance Calculus
- UC5_CMP_03 - Safe Passenger Transfer/ Person Detection
- UC5_CMP_04 - Safe Passenger Transfer/ Platform Detection

4.7.3.2.8 WP5T54-02-02 - Kubernetes

This component consists of an open-source orchestrator for cluster management and container orchestration. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_06 - Integrated Demonstration Software on Target

4.7.3.2.9 WP5T52-04-07 - Harbor Image repository

This component consists of a container registry for docker images. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific*

components that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_06 - Integrated Demonstration Software on Target

4.7.3.2.10 WP6T61-01-01 - Operating system - Ubuntu

This component consists of the Linux for ARM64 & RISC_V64. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_01 - Accurate Stop/ Signal Detection
- UC5_CMP_02 - Accurate Stop/ Stereo Distance Calculus
- UC5_CMP_03 - Safe Passenger Transfer/ Person Detection
- UC5_CMP_04 - Safe Passenger Transfer/ Platform Detection

4.7.3.2.11 WP6T61-01-03 - Petalinux

This component consists of the tools necessary to customize, build, and deploy Embedded Linux solutions on Xilinx processing systems. Tailored to accelerate design productivity, the solution works with the Xilinx hardware design tools to ease the development of Linux systems for Versal. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_01 - Accurate Stop/ Signal Detection
- UC5_CMP_02 - Accurate Stop/ Stereo Distance Calculus
- UC5_CMP_03 - Safe Passenger Transfer/ Person Detection
- UC5_CMP_04 - Safe Passenger Transfer/ Platform Detection

4.7.3.2.12 WP6T61-01-04 - Vitis AI

This component consists of the Vitis™ AI development environment is Xilinx's development platform for AI inference on Xilinx hardware platforms. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_01 - Accurate Stop/ Signal Detection
- UC5_CMP_02 - Accurate Stop/ Stereo Distance Calculus
- UC5_CMP_03 - Safe Passenger Transfer/ Person Detection
- UC5_CMP_04 - Safe Passenger Transfer/ Platform Detection

4.7.3.2.13 WP6T61-03-06 - OpenCV

This component consists of the open-source computer vision and machine learning software library OpenCV. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_01 - Accurate Stop/ Signal Detection
- UC5_CMP_02 - Accurate Stop/ Stereo Distance Calculus
- UC5_CMP_03 - Safe Passenger Transfer/ Person Detection
- UC5_CMP_04 - Safe Passenger Transfer/ Platform Detection

4.7.3.2.14 WP6T61-03-11 - ONNX

This component consists of the cross-platform neural network interchange format and the functions necessary to import ONNX format files in Versal. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_01 - Accurate Stop/ Signal Detection
- UC5_CMP_02 - Accurate Stop/ Stereo Distance Calculus
- UC5_CMP_03 - Safe Passenger Transfer/ Person Detection
- UC5_CMP_04 - Safe Passenger Transfer/ Platform Detection

4.7.3.2.15 WP6T61-02-01 - Docker

This component consists of a Platform-as-a-Service product, namely Docker, that uses OS-level virtualization to deliver software in packages called containers. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_06 - Integrated Demonstration Software on Target

4.7.3.2.16 WP6T61-15 - Standard C++ Library

This component consists of the library for C++ in the platform Versal. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC5_CMP_06 - Integrated Demonstration Software on Target

5 VAL-UC6 Elaborate data collected using heterogeneous technologies (intelligent totem)

The UC6 reference scenario is a shopping mall transformed into a sentient space, by embedding processing resources within the set physical environment. This space can be considered as a network of interconnected nodes, each able to collect and process data locally. In more detail, such components are smart cameras and advertising smart totems (equipped with cameras, microphones, and a large touch-screen display) strategically located inside the shopping mall (e.g., in crowded areas).

As described in detail in the next sections, the main pillars of this Use Case are:

- An AI-based intelligent totem, for personalized advertisement and dedicated customer support will be developed adopting the FRACTAL framework;
- In such space, interactive totems will be equipped with heterogeneous sensors, like for example cameras and microphones, in order to collect a huge amount of data;
- Innovative and advanced AI algorithms deployed on the edge are used to process data collected to better understand their surroundings.

5.1 Before FRACTAL

Digital Signage solutions distribute multimedia content and interactive applications via any display format, touch screen, video wall, and digital totem located at highly attended locations, such as stores, shopping malls, rail or air terminals,and museums.

Typically, they include user-friendly scheduling tools and powerful framework control for secure and reliable real-time broadcasting of info and content that can be constantly updated and disseminated according to location and timing.

Moreover, such tools allow the broadcasting management of monitors with live content and playlists consisting of audio / video clips, slides, texts, links to websites, news feeds, RSS, and the creation of layout, channels and schedules.

Components of a traditional Digital Signage platform are listed here and shown in Figure 33:

- Central servers hosting the front-end of the content management applications;
- A central "director" application (CMS) managing and distributing audio, video and textual content;
- A framework of remote IP-connected multimedia players;
- A network of multi-format LCD/Plasma TVs, pro displays, video-walls, interactive totems and touch screens.

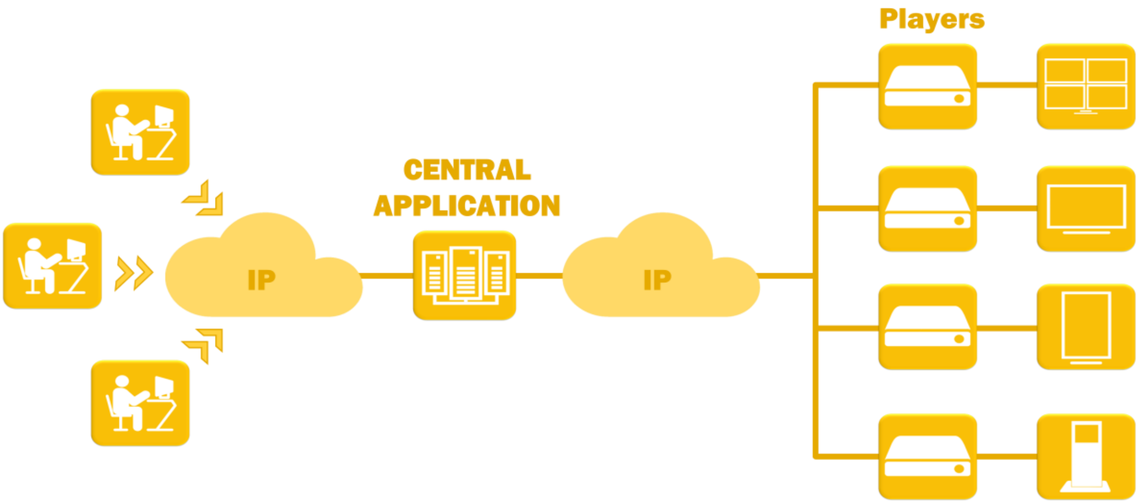


Figure 33 – Digital Signage platform

Unfortunately, content is predefined and its broadcasting/display is statically scheduled. As a matter of fact, totems are usually simple actuators without any sensors or data processing capabilities. The unique interaction possible with their surrounding is by means of interactive displays eventually used by customers or visitors.

5.2 Use Case Objectives

Shopping is one of the most important leisure activities in our life. Despite e-commerce is becoming a fast-growing area of business, shopping mall still remains a relevant reference point above all [30]. Currently, indoor medium-large shopping malls are shopping areas from which traffic is excluded and, sometimes, distributed over several floors. Typically, in these buildings, there are video-surveillance systems and informative totems which provide spatial type information, such as maps and advertising.

Shopping malls are in general crowded and noisy environments, due to the presence of many shops. When looking for a specific shop, it can be difficult to locate and reach, especially when the noise is loud and the space to move is limited by the presence of other people [31].

Other features of those spaces are a good level of lighting. A typical shopping mall is depicted in Figure 34.

Figure 34 – Totems installed in shopping mall

Here the goal is to transform the shopping mall into a **sentient space,** by embedding processing resources within the set physical environment [32] [33] [34]. Hence, this space can be seen **as a network of interconnected nodes**, each with its processing resources, such as smart cameras and advertising smart totems (equipped with cameras, microphones, and a large touch-screen display) located strategically inside the shopping mall (e.g., near entrances, in the hallways), able to provide support to the users uniquely tailored to their needs.

An AI-based intelligent totem, for personalized advertisement and dedicated customer support, will be developed by adopting the FRACTAL framework. The totem will be equipped with heterogeneous sensors (cameras, microphones, etc.) to collect a huge amount of data that can be processed to better understand their surroundings. Advanced AI approaches for data collection and processing will be developed and deployed on the edge.

5.3 State of the art

5.3.1 Interactive totem for retail application

Most relevant solutions for digital interactive communications are listed here below. They are the results of a state-of-the-art analysis done by UC6 leaders.

- ***SmartMedia*** produces and distributes interactive devices like multi-touch monitors, LED panels and Interactive kiosks (indoor and outdoor) for self-payment services [35].
- ***SITA:*** Kate, intelligent robotic kiosk. Based on AI technology and geo-navigation, SITA Lab has developed an intelligent check-in kiosk that will autonomously move to busy or congested areas in the airport as needed, to minimize check-in queues [36].
- ***Intel;*** Interactive kiosk solutions, based on Intel technology are proposed to speed up ordering, purchasing, and other self-service tasks, to simplify everyday problems for all users. Such interactive devices are used also in banks, restaurants, smart cities, transportation providers, hospitals, and stores, in order to automate many self-service tasks [37].
- ***Zebra;*** small size mini- and micro-kiosks for interactive services; typical use is price and inventory check, patient check-in, electronic Hotel Concierge and Maps, E-ticket purchase and pick-up, merchandising and digital-signage, etc [38].
- **Emoji;** EMOJ proposes a set of solutions for online and in-store to really achieve an omnichannel approach to customer experience. Customer segmentation, engagement,and satisfaction are measured in context. Real-time reactions are activated to create unique customer experiences in order to increase sales and optimize conversion rates. They work mainly in Fashion Retail. EMOJ owns an Italian patent about the convolutional networks used for customer recognition and reaction control in real-time. Claims regard:
  - Method to recognize age, gender, emotions and gaze;
  - Method for the elaboration of pictures and video and data transmission;
  - Algorithms for recommendation and content customization;
  - Algorithms for the creation of adaptive and sensible spaces based on customer's profile and emotions.

5.3.2 Object and people detection using video analytics

In UC6 video analysis plays a crucial role, being used for several different tasks. So far, traditional video analysis algorithms have been based on **background estimation** methods which separate the static background from the moving targets, the *foreground*, by comparing the video flow, frame-by-frame. The result of this preprocessing is a set of metadata describing the characteristics of detected targets (position, sizes, direction, speed, etc.). If such characteristics meet a particular predefined rule, which describes a dangerous event like for example a wrong-way vehicle, the system triggers a specific alarm to notify this event. The most important part of this type of algorithm is the rules definition which is basically demanded to video processing experts and SW developers.

Recently, innovative video analytics systems are based on **Artificial Intelligence** approaches and in particular on **Deep learning**. Thanks to its training-based data-driven approach, it allows to minimize errors and increase the detection precision, therefore, system performance. In addition, the advantages offered by deep learning are innumerable: the use of neural networks eliminates the problems associated with sudden changes in the framed scene, caused by changes in weather conditions, lighting or camera movement, situations that generally affect the traditional video analysis system. Unfortunately, these algorithms are quite resource-demanding and with high execution time, which have negative implications for their widespread adoption.

There are several different approaches in the scientific literature, most of them related to object detection and classification, as they are the most important features for each video surveillance applications, including traffic and road monitoring. Recently proposed in [RPN], the **RPN** (Region Proposal Network) is a quite promising technique for object detection, consisting of a convolutional neural network able to detect object bounding boxes and classifying them. Another interesting method, called **SSD** (Single Shot Detection), presented in [SSD], is able to detect objects just using a single deep neural network analyzing images acquired. Finally, **YOLO** (You only look once) [YOLO] is a real-time object detection that applies a single neural network to the whole image, not dividing it in different region as done by the other methods. Performance comparison among these algorithms is not an easy job. Nevertheless, a quite good comparative analysis is reported in [COMP].

5.3.3 Idiom recognition

Idiom Recognition (IR) can be defined as the problem of recognizing the language of the current speaker, given an unknown speech utterance. Literature works related to the IR task can be broadly categorized into the following main approaches: (i) Text-Dependent (TD) and (ii) Text-Independent (TI).

TD solutions are aimed at recognizing the language of the current speaker given the presence of a specific word contained in the speech utterance. The identification is performed on the audio sample, using two different approaches: (i) Speech-to-Text (STT) and (ii) Non Speech-to-Text (NSTT).

TD-STT idiom recognition is based on the transcript of the audio, which must contain a specific word among one or more pre-defined keywords for each available language. STT approaches rely on a typically short list of hot-words for each language: when a hot-word is identified inside the transcript, the system outputs the corresponding language [39].

On the other hand, TD-NSTT solutions are based on machine learning algorithms taking into account the audio features of specific utterances. In particular, keyword matching is not performed on the speech transcript, but it is achieved by comparing the audio sample by employing specific algorithms and models, such as for instance Dynamic Time Warping (DTW) [40], Gaussian Mixture Models (GMM) [41] [42].

For what concerns TI methods, TI-STT idiom recognition is based on the transcript of the audio, which contains generic words belonging to a specific language. STT approaches usually rely on a database storing a large vocabulary of different languages and use it to compare the transcript with the words available for the considered idioms. Matching words and possible analysis on consistency and grammar syntax models allow identifying the correct language. Widely employed methods rely on deep learning using Neural Network (NN) architectures or common code books and Discrete Hidden Markov Models (DHMM) [43] [44].

In the same frameworks, TI-NSTT solutions are based upon language models based on acoustic and phonotactic features related to each language. The acoustic features reflect low-level spectral characteristics, while the phonotactic features represent the phonological constraints that govern a spoken language. Both features have been shown to be effective in spoken language recognition [41] [42]. Examples of commonly employed features are Mel Frequency Cepstral Coefficients (MFCC), Delta-Delta coefficients (DD), and Shifted Delta Cepstral coefficients (SDC) [45]. The speech audio sample is usually fed into a classifier trained with a huge number of samples related to the specific language, possibly uttered in different conditions. The system compares the speech and acoustic features of the audio sample with those derived from the language models and recognizes the idiom that best matches the attributes. Widely employed algorithms in this approach are for instance Support Vector Machines (SVM) [46], or probabilistic models such as Artificial Neural Networks (ANN) and GMM [47].

5.4 Main contributions expected from FRACTAL

The FRACTAL project provides an opportunity to explore alternatives to current commercial smart totems aligned to the main FRACTAL Pillars. The main focus for the UC6 stands for cognitive smart totems based on explainable AI and running on reliable edge platforms able to exploit fractality (especially with respect to communications and load distribution) to provide energy/performance trade-off while guaranteeing soft real-time constraints with respect to user experience. Among the platform variants presented, Xilinx Ultrascale+/Versal-based FRACTAL platform is selected due to its support for inference at the edge.

More in detail, UC6 contributes to the implementation of the FRACTAL Pillars as described below.

Pillar 1: Open-Safe-Reliable and low power Node architecture

UC6 nodes are able to monitor their status so enforcing safety and reliability (e.g., fault and anomaly detection), and low power processing (e.g., energy consumption monitoring to check the energy-efficiency of the inference at the edge).

Pillar 2: Low Power, safety, security, and high-performance trade-off

UC6 nodes are able to monitor their performances so enforcing the power/performance trade-off (e.g., energy efficient acceleration with variable accuracy) in order to satisfy soft real-time requirements related to user experience by means of the inference at the edge.

Pillar 3: Cognitive and Autonomous node

UC6 nodes are able to detect users' features and activities so giving rise to cognitive smart totems based on explainable AI that improves their behavior (performance, power efficiency) against the uncertainty of the environment. UC6 nodes are also autonomous since they integrate all the needed computational power to process data and AI algorithms at the edge.

Pillar 4: FRACTAL mutable Communications

UC6 nodes are able to share the workload among them by also identifying the best communication technology in mutable scenarios, so enabling different performance/power trade-off and enforcing reliability. In such a way, UC6 nodes can collaboratively organize and distribute the work over the FRACTAL in an autonomous manner.

5.5 Evaluation of the implementation results

UC6 KPIs are listed in this section. They are classified as:

- KPIs for Implementation Plan Tasks and for Requirements (defined in WP2)
- KPIs for FRACTAL Objectives related to FRACTAL Pillars
- KPIs for UC Features

For each KPI are reported:

- an *Identifier*,
- a *Description* the type of result *Value*.
- *the Test(s)* to be performed to validate them (defined in section intitled "Justification Plan")

The next subsections are devoted to each group of KPIs.

5.5.1 KPIs for Implementation Plan Tasks

This section defines the ***KPIs defined for the Implementation Plan Tasks*** (more details about the Implementation Plan Tasks are provided in Section 5.6.2). Figure 35 shows the complete list of KPIs defined for the Implementation Plan Tasks of UC6. Figure 36 shows the complete list of KPIs defined for the UC6 requirements defined in WP2.

KPI for UC Implementation Plan				
KPI ID	**Description**	**Value**	**Test**	**Comment**
UC6_KPI_IP_01	Density Estimator Correctness	True/False	KPI is True if the related components are correct	*Defined in section 5.7.1*
UC6_KPI_IP_02	People Detector Correctness	True/False	KPI is True if the related components are correct	*Defined in section 5.7.1*
UC6_KPI_IP_03	Face Detector Correctness	True/False	KPI is True if the related components are correct	*Defined in section 5.7.1*
UC6_KPI_IP_04	Age Estimator Correctness	True/False	KPI is True if the related components are correct	*Defined in section 5.7.1*
UC6_KPI_IP_05	Gender Classifier Correctness	True/False	KPI is True if the related components are correct	*Defined in section 5.7.1*
UC6_KPI_IP_06	Idiom Recognizer Correctness	True/False	KPI is True if the related components are correct	*Defined in section 5.7.1*
UC6_KPI_IP_07	Runtime Manager Correctness	True/False	KPI is True if the related components are correct	*Defined in section 5.7.1*
UC6_KPI_IP_08	Rule-based Recommender Correctness	True/False	KPI is True if the related components are correct	*Defined in section 5.7.1*
UC6_KPI_IP_09	Data Compressor Correctness	True/False	KPI is True if the related components are correct	*Defined in section 5.7.1*
UC6_KPI_IP_10	UC Components Integration Correctness	True/False	KPI is True if the following related KPIs are True: UC6_KPI_FO_00 UC6_KPI_FO_01 UC6_KPI_FO_02 UC6_KPI_FO_03 UC6_KPI_FO_04	*Defined in section 5.7.1*

Figure 35 – KPI for Implementation Plan Tasks

KPI for UC Requirements				
KPI ID	**Description**	**Value**	**Test**	**Comment**
UC6_KPI_IP_Req_01	Cognitiveness Reqs	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_01 UC6_KPI_IP_02 UC6_KPI_IP_03 UC6_KPI_IP_04 UC6_KPI_IP_05	Related to: REQ_UC6_01, REQ_UC6_03, REQ_UC6_04, REQ_UC6_05, REQ_UC6_06, REQ_UC6_07, REQ_UC6_08, REQ_UC6_09, REQ_UC6_16, REQ_UC6_21
UC6_KPI_IP_Req_02	Monitoring&Management Reqs	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_01 UC6_KPI_IP_02 UC6_KPI_IP_03 UC6_KPI_IP_04 UC6_KPI_IP_05 UC6_KPI_IP_06 UC6_KPI_IP_07 UC6_KPI_IP_08 UC6_KPI_IP_09 UC6_KPI_IP_10	Related to: REQ_UC6_02, REQ_UC6_11, REQ_UC6_12, REQ_UC6_13, REQ_UC6_14, REQ_UC6_15, REQ_UC6_17, REQ_UC6_18, REQ_UC6_19, REQ_UC6_20, REQ_UC6_22, REQ_UC6_23, REQ_UC6_24, REQ_UC6_25, REQ_UC6_27, REQ_UC6_28
UC6_KPI_IP_Req_03	User Experience Reqs	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_07 UC6_KPI_IP_08 UC6_KPI_IP_10	Related to: REQ_UC6_10, REQ_UC6_26

Figure 36 – KPI for UC6 requirements

5.5.2 KPIs for FRACTAL Objectives

KPIs defined to measure ***how the Implementation Plan Tasks contribute to demonstrate the FRACTAL Objectives*** (Related to Pillars and found in the FRACTAL proposal, Section 1.1.2.). Figure 37 shows the complete list of KPIs defined for this purpose.

KPI for Fractal Objectives (an related Pillars)				Helps to demonstrate the following Fractal Specific Objective		
KPI ID	**Description**	**Value**	**Test**	**ID Obj**	**Description**	**Relates to Pillar**
UC6_KPI_FO_00	The nodes are able to detect users features	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_03 UC6_KPI_IP_04 UC6_KPI_IP_05 UC6_KPI_IP_06	O3	Evaluate and validate the analytics approach by means of AI to help the identification of the largest set of working conditions still preserving safe and secure operational behaviors	Pillar 3 (WP5) - Cognitive & Autonomous Node.
UC6_KPI_FO_01	The nodes are able to detect users activities	True/False	KPI is True if the following related IP KPIs are True: UC6_KPI_IP_01 UC6_KPI_IP_02	O3	Evaluate and validate the analytics approach by means of AI to help the identification of the largest set of working conditions still preserving safe and secure operational behaviors	Pillar 3 (WP5) - Cognitive & Autonomous Node.
UC6_KPI_FO_02	The nodes are able to monitor their status	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_07	O1	Design and Implement an Open-Safe-Reliable Platform to Build Cognitive Edge Nodes of Variable Complexity	Pillar 1 (WP3) - Open-Safe-Realiable and low power node architecture.
UC6_KPI_FO_03	The nodes are able to monitor their performances	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_07 UC6_KPI_IP_08	O2	Guarantee extra-functional properties (dependability, security, timeliness and energy-efficiency) of FRACTAL nodes and systems built using FRACTAL nodes (i.e., FRACTAL systems).	Pillar 2 (WP4) - Low power, safety, security and high-preformance trade-off.
UC6_KPI_FO_04	The nodes are able to share the workload among them	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_07 UC6_KPI_IP_09	O4	To integrate fractal communication and remote management features into FRACTAL nodes	Pillar 4 (WP6) - Mutable and fractal communications.

Figure 37 – KPI for FRACTAL Objectives

5.5.3 KPIs for UC Features

KPIs defined to measure ***how the Implementation Plan Tasks contribute to demonstrate the UC Features*** (defined in the Tab *FRACTAL Features* in the general Excel defined in *WP1 FRACTAL - Requirements_KPIs_Components.xlsx*). Figure 38 shows the complete list of KPIs defined for this purpose.

KPI for UC Feature				Helps to demonstrate the following UC Feature	
KPI ID	**Description**	**Value**	**Test**	**ID Feature**	**Description**
UC6_KPI_FT_00	The nodes are able to accelerate AI/ML models	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_01, UC6_KPI_IP_04, UC6_KPI_IP_05	UC6_F0	ADAPTABILITY->AI->HW->AI/ML_ACCELERATOR
UC6_KPI_FT_01	The nodes are able to perform inference in real-time	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_01, UC6_KPI_IP_02, UC6_KPI_IP_03, UC6_KPI_IP_04, UC6_KPI_IP_05, UC6_KPI_IP_06, UC6_KPI_IP_08	UC6_F1	ADAPTABILITY->AI->SW->INFERENCE->REAL-TIME
UC6_KPI_FT_02	The nodes are able to import and execute ONNX models	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_01, UC6_KPI_IP_02, UC6_KPI_IP_03, UC6_KPI_IP_04, UC6_KPI_IP_05	UC6_F2	ADAPTABILITY->AI->SW->INFERENCE->MODEL->FORMAT->ONNX
UC6_KPI_FT_03	The nodes are able to import and execute VERSAL models	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_01, UC6_KPI_IP_04, UC6_KPI_IP_05	UC6_F3	ADAPTABILITY->AI->SW->INFERENCE->MODEL->FORMAT->VERSAL
UC6_KPI_FT_04	The nodes are able to perform inference	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_01, UC6_KPI_IP_02,UC6_KPI_IP_03, UC6_KPI_IP_04, UC6_KPI_IP_05, UC6_KPI_IP_06, UC6_KPI_IP_08	UC6_F4	ADAPTABILITY->AI->SW->INFERENCE->MODEL->LOCATION>NODE
UC6_KPI_FT_05	The nodes are able to exploit offline learning/training	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_01, UC6_KPI_IP_02, UC6_KPI_IP_03, UC6_KPI_IP_04, UC6_KPI_IP_05, UC6_KPI_IP_06, UC6_KPI_IP_08	UC6_F5	ADAPTABILITY->AI->SW->LEARNING/TRAINING->LOCATION>OTHER
UC6_KPI_FT_06	The nodes are able to exploit supervised learning/training	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_01, UC6_KPI_IP_02, UC6_KPI_IP_03, UC6_KPI_IP_04, UC6_KPI_IP_05, UC6_KPI_IP_06, UC6_KPI_IP_08	UC6_F6	ADAPTABILITY->AI->SW->LEARNING/TRAINING->PARADIGM->SUPERVISED
UC6_KPI_FT_07	The nodes are able to exploit CNN	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_01, UC6_KPI_IP_02, UC6_KPI_IP_03, UC6_KPI_IP_04, UC6_KPI_IP_05	UC6_F7	ADAPTABILITY->AI->SW->LEARNING/TRAINING->ALGORITHMS->CNN
UC6_KPI_FT_08	The nodes are able to exploit TENSORFLOW/KERAS libraries	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_01, UC6_KPI_IP_02, UC6_KPI_IP_03, UC6_KPI_IP_04, UC6_KPI_IP_05	UC6_F8	ADAPTABILITY->AI->SW->LIBRARY>TENSORFLOW/KERAS
UC6_KPI_FT_09	The nodes are able to perform load balancing	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_07, UC6_KPI_IP_09, UC6_KPI_IP_10	UC6_F9	RELIABILITY->AVALABILITY->LOAD_BALANCING
UC6_KPI_FT_10	The nodes are able to monitor their performances	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_07, UC6_KPI_IP_08	UC6_F10	SAFETY->MONITORING->PERFORMANCES
UC6_KPI_FT_11	The nodes can acquire video streams from a camera	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_01, UC6_KPI_IP_02, UC6_KPI_IP_03, UC6_KPI_IP_04, UC6_KPI_IP_05	UC6_F11	CONTEXT->AWARENESS->SENSORS->CAMERA
UC6_KPI_FT_12	The nodes can acquire audio streams from a microphone	True/False	KPI is True if the following related KPI is True: UC6_KPI_IP_06	UC6_F12	CONTEXT->AWARENESS->SENSORS->MICROPHONE
UC6_KPI_FT_13	The nodes can generate and trasmit alarms	True/False	KPI is True if the following related KPI is True: UC6_KPI_IP_02	UC6_F13	CONTEXT->AWARENESS->ACTIONS->ALARM
UC6_KPI_FT_14	The nodes have Ethernet interface	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_01, UC6_KPI_IP_02, UC6_KPI_IP_07, UC6_KPI_IP_10	UC6_F14	FRACTALITY->COMMUNICATIONS/CONNECTIVITY->TECHNOLOGIES->ETHERNET
UC6_KPI_FT_15	The nodes have WI-FI interface	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_01, UC6_KPI_IP_02, UC6_KPI_IP_07, UC6_KPI_IP_10	UC6_F15	FRACTALITY->COMMUNICATIONS/CONNECTIVITY->TECHNOLOGIES->WI-FI
UC6_KPI_FT_16	The nodes support MQTT communication	True/False	KPI is True if the following related KPIs are True: UC6_KPI_IP_01, UC6_KPI_IP_02, UC6_KPI_IP_07, UC6_KPI_IP_10	UC6_F16	FRACTALITY->COMMUNICATIONS/CONNECTIVITY->DATAPROTOCOLS->MQTT
UC6_KPI_FT_17	The nodes are implemented on Versal	True/False	KPI is True if the following related KPI is True: UC6_KPI_IP_10	UC6_F17	OTHER:NON-FUNCTIONAL->PLATFORM->VERSAL-ARM
UC6_KPI_FT_18	The nodes are implemented on ZYNQ ULTRASCALE+	True/False	KPI is True if the following related KPI is True: UC6_KPI_IP_10	UC6_F18	OTHER:NON-FUNCTIONAL->PLATFORM->ZYNQU_LTRASCALE+
UC6_KPI_FT_19	The nodes execute LINUX OS	True/False	KPI is True if the following related KPI is True: UC6_KPI_IP_10	UC6_F19	OTHER:NON-FUNCTIONAL->PLATFORM->OS->LINUX

Figure 38 - KPIs for UC5 Implementation Plan to measure the contribution to FRACTAL Features

5.6 Implementation Plan

5.6.1 Big Picture in UC6

Figure 39 shows the FRACTAL big picture with respect to the UC6 implementation. In particular, it shows the functionalities that are currently not used (pink boxes) or not supported at all (gray boxes). With respect to the other functionality, especially for those related to the CLOUD part, we are currently evaluating which ones to effectively use. Therefore, we leave all of them enabled for now.

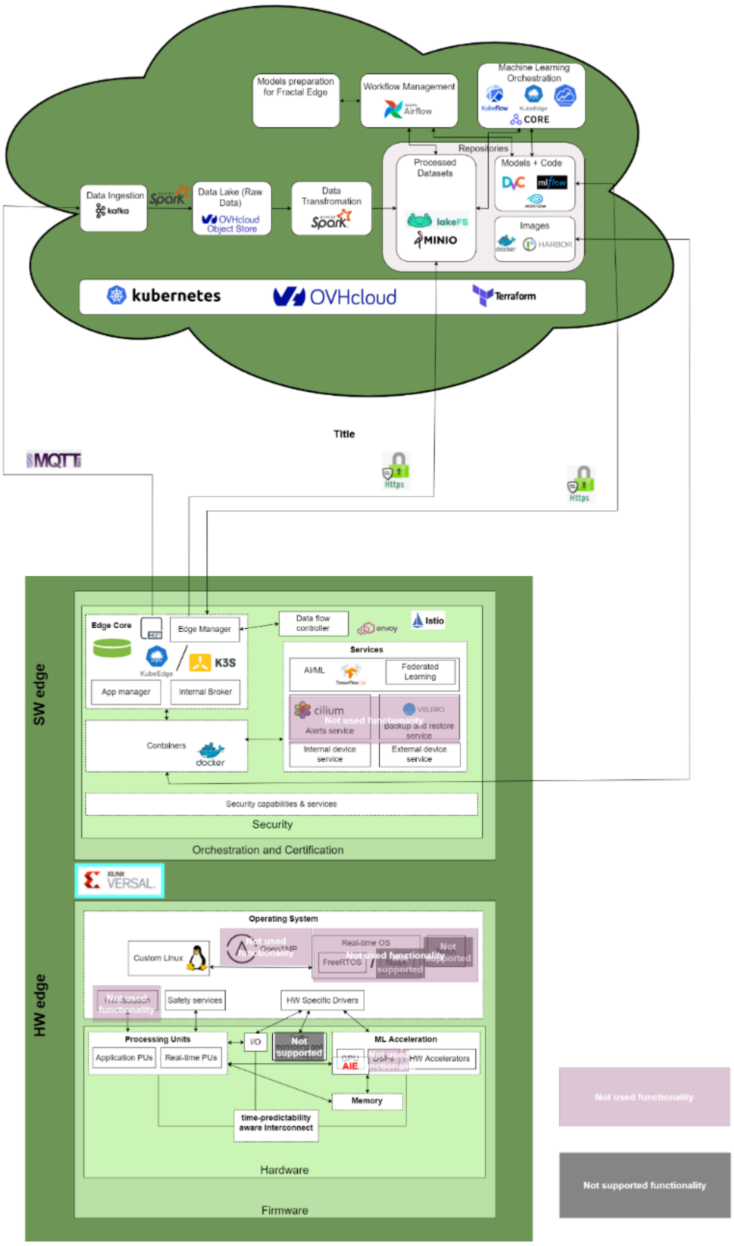


Figure 39 – Big Picture customization for UC6

5.6.2 Tasks

Figure 40 shows the implementation plan tasks and chronogram for UC6. The table contains the following information:

- Task ID, structured as UC<X>_T<N>, in which X represents the use case number, 6 in this case and N is a progressive number to have a unique ID for each task
- Task name
- Task duration

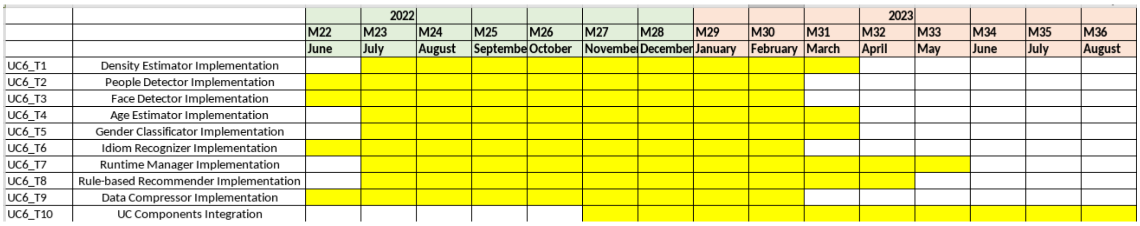


Figure 40 - Chrono program of the Implementation Plan

All Tasks are described in the next subsections.

5.6.2.1 Density Estimator Implementation

This task consists in estimating the rate of people that are on the roof node field of view of the UC6.

The people density is estimated using a CNN-based detector, which has been deployed on a Xilinx ZUS+ FRACTAL Edge. The application is accelerated using Vitis-AI and a porting to the Xilinx Versal platform is planned as well.

The density estimator application provides output to the rest of the system via MQTT (Message Queuing Telemetry Transport). Configuration parameters can be changed at runtime, using MQTT, in order to match the computational load of the system and to exploit fractality.

5.6.2.2 People detector Implementation

This task consists of training a neural network (People Object Detector) on a cloud server (offline training outside the FRACTAL environment) and exporting it into ONNX format to be deployed in FRACTAL Edge (Roof Node) in combination with a component that can generate MQTT alarms based on the neural network outputs.

5.6.2.3 Face Detector Implementation

The aim of the face detector task is to extract the face of a person in front of the smart totem of the UC6.

The Face Detector application runs on top of a Xilinx ZUS+ FRACTAL edge. The application takes advantage of the Vitis-AI acceleration and a porting to the Xilinx Versal platform is planned as well.

The output of the Face Detector is forwarded to the other AI-based tasks using shared memory.

5.6.2.4 Age Estimator Implementation

This task consists of training a CNN for age estimation on a host computer (offline training outside the FRACTAL environment) and cross-compiling it to execute the inference phase on two targets edge-computing platforms. The network is based on Tensorflow and the trained model is processed with Xilinx Vitis AI; the two targets are the Xilinx Zynq Ultrascale+ and the Xilinx Versal. In both cases, the workload is managed by the ARM application cores and it is accelerated on a dedicated processor. In the case of Zynq Ultrascale+, the dedicated processor is the DPU, while in the case of Versal the dedicated processor is the AI Engine. ARM starts the application by configuring the accelerator, setting it to automatically fetch input data from external RAM memory. Then, the DPU/AI Engine executes the CNN and triggers an interrupt to ARM when the inference ends. It is worth noting that input is provided by the Face Detector application, and outputs are sent to the Rule-based Recommender application through shared external memory. The application is executed on the totem node, and it can share the workload with a roof node with available computing resources.

5.6.2.5 Gender Classifier Implementation

This task consists of training a CNN for gender classification on a host computer (offline training outside the FRACTAL environment) and cross-compiling it to execute the inference phase on two targets edge-computing platforms. The network is based on Tensorflow and the trained model is processed with Xilinx Vitis AI; the two targets are the Xilinx Zynq Ultrascale+ and the Xilinx Versal. In both cases, the workload is managed by the ARM application cores and it is accelerated on a dedicated processor. In the case of Zynq Ultrascale+, the dedicated processor is the DPU, while in the case of Versal the dedicated processor is the AI Engine. ARM starts the application by configuring the accelerator, setting it to automatically fetch input data from external RAM memory. Then, the DPU/AI Engine executes a part of the CNN, up to a hard sigmoid layer of the CNN; then, it triggers an interrupt to the ARM core that performs the hard sigmoid and sends back the control to the accelerator. The process repeats four times, and the result is stored in external memory to be available for the rule-based recommender module. The input is provided by the face detector module. The application is executed on the totem node, and it can share the workload with a roof node with available computing resources.

5.6.2.6 Idiom Recognizer Implementation

This task tackles the problem of recognizing the language (or idiom) spoken by a person who is approaching the intelligent totem. Namely, the Idiom Recognition (IR) algorithm works as follows: (i) it converts the spoken audio into a series of words through a Speech-To-Text (STT) engine and successively (ii) it recognizes the uttered language by following a word-by-word dictionary comparison approach. The outcome of the IR task is then propagated by using a MQTT-based protocol.

5.6.2.7 Runtime Manager Implementation

The aim of the task is to develop a component, called Runtime Manager, capable of managing the communication and interaction between several nodes or modules. Regarding communication management, it uses MQTT protocol for totem-roof nodes communication whereas it uses Rest API and Socket API for modules communication. For what concern the interaction management, the component plays the role of master scheduling the tasks based on inputs received.

5.6.2.8 Rule-based Recommender Implementation

The aim of the task is to develop a component that receives the data of the person and suggests the corresponding content to be reproduced to get the attention of the user. The component is developed as a REST server which accepts the data in the form of a JSON dictionary {"age", "gender", "language"}, and responds with the index of the media file that should be reproduced. This index is computed by a rule-based explainable AI algorithm which should be previously trained on a suitable dataset.

5.6.2.9 Data Compression

This component performs data compression and decompression operations using the universal lossless data compression algorithm LZW, with the aim to investigate an energy-aware solution to reduce data transfer for low-power services. The component is developed as a software library written in C++, with a single entry-point for both actions of compression and decompression. As a part of the internal device services, it can be integrated into any component needing compression and decompression features. In the context of UC6, the component can be used in conjunction with the Load Balancing module to improve the efficiency of the managing loads operation.

5.6.2.10 UC Components Integration

This task is related to the integration of all the components developed in the other tasks. It is performed in an iterative way in order to allow incremental verification as soon as the different components are verified individually. Finally, it will allow to validate the whole UC integration.

5.6.3 Components

This section summarizes the components involved in the Implementation Plan. Such components are:

- ***Components produced by the UC*** *resulting from executing the Implementation Plan*.
- ***Common FRACTAL Components*** (from WP3, WP4, WP5, WP6) that are needed to execute the Implementation Plan.

5.6.3.1 Components produced by the Implementation Plan

The Figure 41 contains the following information:

- Components ID, structured as UC_<X>_COMP<N>, in which X represents the use case number, 6 in this case and N is a progressive number to have a unique ID for each component
- Name of the component
- Description of the component

UC Components		
KPI ID	**Name**	**Description**
UC6_CMP_01	DE – Density Estimator	The DE component is a CNN-based system for people density estimation. The DE component takes as input an IP-camera video flow and using the neural accelerators of the ZUS+, outputs the density estimation. The rate of people is sent to the other components using an MQTT-based communications. Configuration parameters can be changed at runtime using MQTT.
UC6_CMP_02	PD – People Detector	SW component that implements People detection algorithms (WP5T56_01), alarm generation module (analyzing metadata generated by people detector) and communication interfaces.
UC6_CMP_03	FD – Face Detector	The FD component allows to extract a face crop of a person in front of the smart totem. The face bounding box is computed using a CNN-based detector, that takes advantage of a Vitis-AI acceleration. The output of the component is an image that contains a face. The output image is forwarded to the other components using shared memory.
UC6_CMP_04	AE – Age Estimator	Age estimator based on a convolutional neural network. The system takes as input the picture of a person and outputs an estimation of the age.
UC6_CMP_05	GC – Gender Classifier	Gender classifier based on a convolutional neural network. The system takes as input the picture of a person and classifies it as female or male.
UC6_CMP_06	IR – Idiom Recognizer	Automatic language recognition based on speech processing. The system outputs the language of the current speaker by processing the pre-recorded audio. If necessary, the IR component can also elaborate "live" audio.
UC6_CMP_07	RM - Runtime Manager	Component that manages node-to-node communications, interaction between components and task scheduling based on input received
UC6_CMP_08	RBR - Rule-based Recommender	The component is implemented as a REST server which receives the data of the people as a JSON dictionary {"age", "gender", "language"} and it responds with the suggestion computed using the already trained rule-based clear-box algorithm.
UC6_CMP_09	DC - Data Compression	Component that perform Data compression and decompression operations through a energy-aware version of the LZW algorithm.

Figure 41 – Component produced by execution of UC6 Implementation Plan

5.6.3.2 Common FRACTAL Components

Common FRACTAL components defined and developed in technical Work Packages (i.e., WP3, 4, 5 and 6) and used in UC6 are listed in the figure below. Figure 42 contains the following information:

- Components ID, structured as WP<X>T<Y>-<N>, in which X and Y represent respectively WP and task in which the component has been defined and N is a progressive number to have a unique ID for each task of each WP;
- Component Name;
- Short description about the components.

FRACTAL Components needed by the UC		
KPI ID	**Name**	**Description**
WP3T32-07	Age and Gender identifier at the edge	CNN development and integration on Zynq Ultrascale+ and VERSAL for UC6
WP3T32-09	Runtime Bandwidth Regulator	Memory bandwidth regulator that can be integrated on FPGA-based accelerator clusters, to improve main memory QoS and interference mitigation. This is a joint innovation with UNIVAQ.
WP3T35-05	Idiom Recognizer	Idiom/Language regonition system based on speech signal registration through Speech-to-Text ML solutions.
WP3T36-02	Load Balancing Module	Software module designed to collect computational loads from nodes and, in case of overload, able to distribute compuational load.
WP4T41-01	Data Compression for Low-Power Services	Data compression technique for low-power devices, to be applied at system level.
WP4T44-06	GDPR Compliance	Data Protection Impact Analysis
WP5T56_01	People detector example	Video Content Analysis algorithms (based on CNN) for people detection inside the monitored area. Focus on UC6: customer detection and estimation of the position with respect to the totem.
WP6T61-03-02	End-to-end machine learning toolkit	end-to-end open-source platform for machine learning (Tensorflow)
WP6T61-03-06	Computer vision toolkit	open-source computer vision and machine learning software library (OpenCV)
WP6T61-03-11	Machine learning training accelerator	cross-platform inference and training machine-learning accelerator (ONNX)
WP6T61-13	Deep learning toolkit	deep learning framework (Caffe)
WP6T62-03	Run time Manager	Management of component interaction, task sheduling and node-to-node communication

Figure 42 - Common FRACTAL components from WP3, WP4, WP5, WP6 needed to execute UC6 Implementation Plan

5.6.4 Traceability relationships of Tasks-Components-KPIs

Finally, this section ***links together*** *tasks*, *components* and *KPIs*. For each task, the following traceability-relationships are given:

- ***Components***
  - IN Components – Input components needed by the task.
  - OUT Components – Output components produced by the task.
- ***KPIs for UC Implementation Plan***
  - *KPI ID*
  - *KPI Description*
  - *Value*
  - *Test*
- ***KPIs for FRACTAL Objectives and Features***
  - *KPI ID*
  - *KPI Description*
  - *Value*
  - *Test*

Following subsections detail this information for each task.

5.6.4.1 *Density Estimator Implementation*

Figure 43 shows traceability relationships for Task UC6_CMP_01:

Components		KPIs for UC Implementation Plan				KPIs for Fractal Objectives & Features			
IN Comp	OUT Comp	KPI ID	KPI Description	Value	Test	KPI ID	KPI Description	Value	Test
WP4T44-06 WP6T61-03-11 WP6T61-12	UC6_CMP_01	UC6_KPI_IP_01	KPI description, value and test are reported in Section 5.7.1			UC6_KPI_FO_01 UC6_KPI_FT_00 UC6_KPI_FT_01 UC6_KPI_FT_02 UC6_KPI_FT_03 UC6_KPI_FT_04 UC6_KPI_FT_05 UC6_KPI_FT_06 UC6_KPI_FT_07 UC6_KPI_FT_08 UC6_KPI_FT_11 UC6_KPI_FT_14 UC6_KPI_FT_15 UC6_KPI_FT_16	KPI description, value and test are reported in Section 5.7.1		

Figure 43 - Task UC6_CMP_01 traceability relationship between task, components and KPIs

5.6.4.2 *People Detector Implementation*

Figure 44 shows traceability relationships for Task UC6_CMP_02:

Components		KPIs for UC Implementation Plan				KPIs for Fractal Objectives & Features			
IN Comp	OUT Comp	KPI ID	KPI Description	Value	Test	KPI ID	KPI Description	Value	Test
WP4T44-06 WP5T56_01 WP6T61-03-11 WP6T61-03-06 WP6T61-03-02	UC6_CMP_02	UC6_KPI_IP_02	KPI description, value and test are reported in Section 5.7.1			UC6_KPI_FO_01 UC6_KPI_FT_01 UC6_KPI_FT_02 UC6_KPI_FT_04 UC6_KPI_FT_05 UC6_KPI_FT_06 UC6_KPI_FT_07 UC6_KPI_FT_08 UC6_KPI_FT_11 UC6_KPI_FT_13 UC6_KPI_FT_14 UC6_KPI_FT_15 UC6_KPI_FT_16	KPI description, value and test are reported in Section 5.7.1		

Figure 44 - Task UC6_CMP_02 traceability relationship between task, components and KPIs

5.6.4.3 *Face detector Implementation*

Figure 45 shows traceability relationships for Task UC6_CMP_03:

Components		KPIs for UC Implementation Plan				KPIs for Fractal Objectives & Features			
IN Comp	OUT Comp	KPI ID	KPI Description	Value	Test	KPI ID	KPI Description	Value	Test
WP4T44-06 WP6T61-03-11 WP6T61-13	UC6_CMP_03	UC6_KPI_IP_03	KPI description, value and test are reported in Section 5.7.1			UC6_KPI_FO_00 UC6_KPI_FT_01 UC6_KPI_FT_02 UC6_KPI_FT_04 UC6_KPI_FT_05 UC6_KPI_FT_06 UC6_KPI_FT_07 UC6_KPI_FT_08 UC6_KPI_FT_11	KPI description, value and test are reported in Section 5.7.1		

Figure 45 - Task UC6_CMP_03 traceability relationship between task, components and KPIs

5.6.4.4 Age Estimator Implementation

Figure 46 shows traceability relationships for Task UC6_CMP_04:

Components		KPIs for UC Implementation Plan				KPIs for Fractal Objectives & Features			
IN Comp	OUT Comp	KPI ID	KPI Description	Value	Test	KPI ID	KPI Description	Value	Test
WP4T44-06 WP5T56_01 WP3T32-07 WP6T61-03-11 WP6T61-03-06 WP6T61-03-02 WP6T61-13	UC6_CMP_04	UC6_KPI_IP_04	KPI description, value and test are reported in Section 5.7.1			UC6_KPI_FO_00 UC6_KPI_FT_00 UC6_KPI_FT_01 UC6_KPI_FT_02 UC6_KPI_FT_03 UC6_KPI_FT_04 UC6_KPI_FT_05 UC6_KPI_FT_06 UC6_KPI_FT_07 UC6_KPI_FT_08 UC6_KPI_FT_11	KPI description, value and test are reported in Section 5.7.1		

Figure 46 - Task UC6_CMP_04 traceability relationship between task, components and KPIs

5.6.4.5 Gender Classifier Implementation

Figure 47 shows traceability relationships for Task UC6_CMP_05:

Components		KPIs for UC Implementation Plan				KPIs for Fractal Objectives & Features			
IN Comp	OUT Comp	KPI ID	KPI Description	Value	Test	KPI ID	KPI Description	Value	Test
WP4T44-06 WP5T56_01 WP3T32-07 WP6T61-03-11 WP6T61-03-06 WP6T61-03-02 WP6T61-13	UC6_CMP_05	UC6_KPI_IP_05	KPI description, value and test are reported in Section 5.7.1			UC6_KPI_FO_00 UC6_KPI_FT_00 UC6_KPI_FT_01 UC6_KPI_FT_02 UC6_KPI_FT_03 UC6_KPI_FT_04 UC6_KPI_FT_05 UC6_KPI_FT_06 UC6_KPI_FT_07 UC6_KPI_FT_08 UC6_KPI_FT_11	KPI description, value and test are reported in Section 5.7.1		

Figure 47 - Task UC6_CMP_05 traceability relationship between task, components and KPIs

5.6.4.6 Idiom Recognizer Implementation

Figure 48 shows traceability relationships for Task UC6_CMP_06:

Components		KPIs for UC Implementation Plan				KPIs for Fractal Objectives & Features			
IN Comp	OUT Comp	KPI ID	KPI Description	Value	Test	KPI ID	KPI Description	Value	Test
WP4T44-06 WP3T35-05	UC6_CMP_06	UC6_KPI_IP_06	KPI description, value and test are reported in Section 5.7.1			UC6_KPI_FO_00 UC6_KPI_FT_01 UC6_KPI_FT_04 UC6_KPI_FT_05 UC6_KPI_FT_06 UC6_KPI_FT_12	KPI description, value and test are reported in Section 5.7.1		

Figure 48 - Task UC6_CMP_06 traceability relationship between task, components and KPIs

5.6.4.7 Runtime Manager Implementation

Figure 49 shows traceability relationships for Task UC6_CMP_07:

Components		KPIs for UC Implementation Plan				KPIs for Fractal Objectives & Features			
IN Comp	OUT Comp	KPI ID	KPI Description	Value	Test	KPI ID	KPI Description	Value	Test
WP3T36-02 WP3T32-09 WP6T62-03 WP4T41-01	UC6_CMP_07	UC6_KPI_IP_07	KPI description, value and test are reported in Section 5.7.1			UC6_KPI_FO_02 UC6_KPI_FO_03 UC6_KPI_FO_04 UC6_KPI_FT_09 UC6_KPI_FT_10 UC6_KPI_FT_14 UC6_KPI_FT_15 UC6_KPI_FT_16	KPI description, value and test are reported in Section 5.7.1		

Figure 49 - Task UC6_CMP_07 traceability relationship between task, components and KPIs

5.6.4.8 Rule-based Recommender Implementation

Figure 50 shows traceability relationships for Task UC6_CMP_08:

Components		KPIs for UC Implementation Plan				KPIs for Fractal Objectives & Features			
IN Comp	OUT Comp	KPI ID	KPI Description	Value	Test	KPI ID	KPI Description	Value	Test
none	UC6_CMP_08	UC6_KPI_IP_08	KPI description, value and test are reported in Section 5.7.1			UC6_KPI_FO_03 UC6_KPI_FT_01 UC6_KPI_FT_04 UC6_KPI_FT_05 UC6_KPI_FT_06	KPI description, value and test are reported in Section 5.7.1		

Figure 50 - Task UC6_CMP_08 traceability relationship between task, components and KPIs

5.6.4.9 Data Compression

Figure 51 shows traceability relationships for Task UC6_CMP_09:

Components		KPIs for UC Implementation Plan				KPIs for Fractal Objectives & Features			
IN Comp	OUT Comp	KPI ID	KPI Description	Value	Test	KPI ID	KPI Description	Value	Test
WP4T41-01	UC6_CMP_09	UC6_KPI_IP_09	KPI description, value and test are reported in Section 5.7.1			UC6_KPI_FO_04 UC6_KPI_FT_09	KPI description, value and test are reported in Section 5.7.1		

Figure 51 - Task UC6_CMP_09 traceability relationship between task, components and KPIs

5.6.4.10 UC Component Integration

Figure 52 shows traceability relationships for Task UC6_CMP_10:

Components		KPIs for UC Implementation Plan				KPIs for Fractal Objectives & Features			
IN Comp	OUT Comp	KPI ID	KPI Description	Value	Test	KPI ID	KPI Description	Value	Test
UC6_CMP_01 UC6_CMP_02 UC6_CMP_03 UC6_CMP_04 UC6_CMP_05 UC6_CMP_06 UC6_CMP_07 UC6_CMP_08 UC&_CMP_09	none	UC6_KPI_IP_10	KPI description, value and test are reported in Section 5.7.1			UC6_KPI_FT_09 UC6_KPI_FT_14 UC6_KPI_FT_15 UC6_KPI_FT_16 UC6_KPI_FT_17 UC6_KPI_FT_18 UC6_KPI_FT_19	KPI description, value and test are reported in Section 5.7.1		

Figure 52 - Task UC6_CMP_10 traceability relationship between task, components and KPIs

5.7 Justification plan

This section describes the justification methods that will be used for:

- KPIs evaluation;
- Use Case requirements validation and components validation.

5.7.1 KPIs evaluation method

This section describes the method that will be used to evaluate:

- KPIs for Implementation Plan;
- KPIs for FRACTAL Objectives.

5.7.1.1 KPIs for Implementation Plan

5.7.1.1.1 UC6_KPI_IP_01

- ***Description***: Density Estimator Correctness
- ***Result type***: True/False
- ***Evaluation method***: KPI is True if the related component (UC6_CMP_01) is correct. The detailed way to assess components correctness is specified in 5.7.3.1.1

5.7.1.1.2 UC6_KPI_IP_02

- ***Description***: People Detector Correctness
- ***Result type***: True/False
- ***Evaluation method***: KPI is True if the related component (UC6_CMP_02) is correct. The detailed way to assess components correctness is specified in 5.7.3.1.2

5.7.1.1.3 UC6_KPI_IP_03

- ***Description***: Face Detector Correctness
- ***Result type***: True/False
- ***Evaluation method***: KPI is True if the related component (UC6_CMP_03) is correct. The detailed way to assess components correctness is specified in 5.7.3.1.3

5.7.1.1.4 UC6_KPI_IP_04

- ***Description***: Age Estimator Correctness
- ***Result type***: True/False
- ***Evaluation method***: KPI is True if the related component (UC6_CMP_04) is correct. The detailed way to assess components correctness is specified in 5.7.3.1.4

5.7.1.1.5 UC6_KPI_IP_05

- ***Description***: Gender Classifier Correctness
- ***Result type***: True/False
- ***Evaluation method***: KPI is True if the related component (UC6_CMP_05) is correct. The detailed way to assess components correctness is specified in 5.7.3.1.5

5.7.1.1.6 UC6_KPI_IP_06

- ***Description***: Idiom Recognizer Correctness
- ***Result type***: True/False
- ***Evaluation method***: KPI is True if the related component (UC6_CMP_06) is correct. The detailed way to assess components correctness is specified in 5.7.3.1.6

5.7.1.1.7 UC7_KPI_IP_07

- ***Description***: Runtime Manager Correctness
- ***Result type***: True/False
- ***Evaluation method***: KPI is True if the related component (UC6_CMP_07) is correct. The detailed way to assess components correctness is specified in 5.7.3.1.7

5.7.1.1.8 UC6_KPI_IP_08

- ***Description***: Rule-based Recommender Correctness
- ***Result type***: True/False
- ***Evaluation method***: KPI is True if the related component (UC6_CMP_08) is correct. The detailed way to assess components correctness is specified in 5.7.3.1.8

5.7.1.1.9 UC6_KPI_IP_09

- ***Description***: Data Compressor Correctness
- ***Result type***: True/False
- ***Evaluation method***: KPI is True if the related component (UC6_CMP_09) is correct. The detailed way to assess components correctness is specified in 5.7.3.1.9

5.7.1.1.10 UC6_KPI_IP_10

Description: UC Components Integration Correctness
Result type: True/False
Evaluation method: KPI is True if the following related KPIs are True:
UC6_KPI_FO_00;
UC6_KPI_FO_01;
UC6_KPI_FO_02;
UC6_KPI_FO_03;
UC6_KPI_FO_04.

5.7.1.2 KPIs for FRACTAL Objectives

5.7.1.2.1 UC6_KPI_FO_00

- **Description**: The nodes are able to detect users' features.
- **Result type**: True/False.
- **Helps to demonstrate FRACTAL Objective:** O3 - Evaluate and validate the analytics approach by means of AI to help the identification of the largest set of working conditions still preserving safe and secure operational behaviors.
- **Relates to FRACTAL Pillar:** Pillar 3 (WP5) - Cognitive & Autonomous Node.
- **Evaluation method:** KPI is True if the following related KPIs are True:

- UC6_KPI_IP_03;
- UC6_KPI_IP_04;
- UC6_KPI_IP_05;
- UC6_KPI_IP_06.

5.7.1.2.2 UC6_KPI_FO_01

- **Description**: The nodes are able to detect users' activities.
- **Result type**: True/False.
- **Helps to demonstrate FRACTAL Objective:** O3 - Evaluate and validate the analytics approach by means of AI to help the identification of the largest set of working conditions still preserving safe and secure operational behaviors.
- **Relates to FRACTAL Pillar:** Pillar 3 (WP5) - Cognitive & Autonomous Node.
- **Evaluation method:** KPI is True if the following related IP KPIs are True:
  - UC6_KPI_IP_01;
  - UC6_KPI_IP_02.

5.7.1.2.3 UC6_KPI_FO_02

- **Description**: The nodes are able to monitor their status.
- **Result type**: True/False.
- **Helps to demonstrate FRACTAL Objective:** O1 - Design and Implement an Open-Safe-Reliable Platform to Build Cognitive Edge Nodes of Variable Complexity.
- **Relates to FRACTAL Pillar:** Pillar 1 (WP3) - Open-Safe-Realiable and low power node architecture.
- **Evaluation method:** KPI is True if the following related KPIs are True:
  - UC6_KPI_IP_07.

5.7.1.2.4 UC6_KPI_FO_03

- **Description**: The nodes are able to monitor their performances.
- **Result type**: True/False.
- **Helps to demonstrate FRACTAL Objective:** O2 - Guarantee extra - functional properties (dependability, security, timeliness and energy-efficiency) of FRACTAL nodes and systems built using FRACTAL nodes (i.e., FRACTAL systems).
- **Relates to FRACTAL Pillar:** Pillar 2 (WP4) - Low power, safety, security and high-performance trade-off.
- **Evaluation method:** KPI is True if the following related KPIs are True:
  - UC6_KPI_IP_07;
  - UC6_KPI_IP_08.

5.7.1.2.5 UC6_KPI_FO_04

- **Description**: The nodes are able to share the workload among them.
- **Result type**: True/False.
- **Helps to demonstrate FRACTAL Objective:** O4 - To integrate FRACTAL communication and remote management features into FRACTAL nodes.
- **Relates to FRACTAL Pillar:** Pillar 4 (WP6) - Mutable and FRACTAL communications.
- **Evaluation method:** KPI is True if the following related KPIs are True:
  - UC6_KPI_IP_07;
  - UC6_KPI_IP_09.

5.7.1.3 KPIs for FRACTAL Features

5.7.1.3.1 UC6_KPI_FT_00

- **Description:** The nodes are able to accelerate AI/ML models.
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** ADAPTABILITY->AI->HW->AI/ML_ACCELERATOR
- **Evaluation method:** KPI is True if the following related KPIs are True:
  - UC6_KPI_IP_01;
  - UC6_KPI_IP_04;
  - UC6_KPI_IP_05.

5.7.1.3.2 UC6_KPI_FT_01

- **Description:** The nodes are able to perform inference in real-time.
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** ADAPTABILITY->AI->SW->INFERENCE->REAL-TIME
- **Evaluation method:** KPI is True if the following related KPIs are True:
  - UC6_KPI_IP_01;
  - UC6_KPI_IP_02;
  - UC6_KPI_IP_03;
  - UC6_KPI_IP_04,;
  - UC6_KPI_IP_05;
  - UC6_KPI_IP_06;
  - UC6_KPI_IP_08.

5.7.1.3.3 UC6_KPI_FT_02

- **Description:** The nodes are able to import and execute ONNX models
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** ADAPTABILITY->AI->HW->AI/ML_ACCELERATOR
- **Evaluation method:** KPI is True if the following related KPIs are True:
  - UC6_KPI_IP_01;
  - UC6_KPI_IP_02;
  - UC6_KPI_IP_03;
  - UC6_KPI_IP_04;
  - UC6_KPI_IP_05.

5.7.1.3.4 UC6_KPI_FT_03

- **Description:** The nodes are able to import and execute Versal models.
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** ADAPTABILITY->AI->SW->INFERENCE->MODEL->FORMAT->VERSAL
- **Evaluation method:** KPI is True if the following related KPIs are True:
  - UC6_KPI_IP_01;
  - UC6_KPI_IP_04;
  - UC6_KPI_IP_05.

5.7.1.3.5 UC6_KPI_FT_04

- **Description:** The nodes are able to perform inference.
- **Result type:** True/False

- **Helps to demonstrate UC Feature:** ADAPTABILITY->AI->SW->INFERENCE->MODEL->LOCATION->NODE
- **Evaluation method**: KPI is True if the following related KPIs are True:
    - UC6_KPI_IP_01;
    - UC6_KPI_IP_02;
    - UC6_KPI_IP_03;
    - UC6_KPI_IP_04;
    - UC6_KPI_IP_05;
    - UC6_KPI_IP_06;
    - UC6_KPI_IP_08.

5.7.1.3.6 UC6_KPI_FT_05

- **Description:** The nodes are able to exploit offline learning/training.
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** ADAPTABILITY->AI->SW->LEARNING/TRAINING->LOCATION->OTHER
- **Evaluation method:** KPI is True if the following related KPIs are True:
    - UC6_KPI_IP_01;
    - UC6_KPI_IP_02;
    - UC6_KPI_IP_03;
    - UC6_KPI_IP_04;
    - UC6_KPI_IP_05;
    - UC6_KPI_IP_06;
    - UC6_KPI_IP_08.

5.7.1.3.7 UC6_KPI_FT_06

- **Description:** The nodes are able to exploit supervised learning/training. **Result type:** True/False
- **Helps to demonstrate UC Feature:** ADAPTABILITY->AI->SW->LEARNING/TRAINING->PARADIGM->SUPERVISED
- **Evaluation method:** KPI is True if the following related KPIs are True:
    - UC6_KPI_IP_01;
    - UC6_KPI_IP_02;
    - UC6_KPI_IP_03;
    - UC6_KPI_IP_04;
    - UC6_KPI_IP_05;
    - UC6_KPI_IP_06;
    - UC6_KPI_IP_08.

5.7.1.3.8 UC6_KPI_FT_07

- **Description:** The nodes are able to exploit CNN.
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** ADAPTABILITY->AI->SW->LEARNING/TRAINING->ALGORITHMS->CNN
- **Evaluation method:** KPI is True if the following related KPIs are True:
    - UC6_KPI_IP_01;
    - UC6_KPI_IP_02;
    - UC6_KPI_IP_03;
    - UC6_KPI_IP_04;
    - UC6_KPI_IP_05.

5.7.1.3.9 UC6_KPI_FT_08

- **Description:** The nodes are able to exploit TensorFlow/Keras libraries.
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** ADAPTABILITY->AI->SW->LIBRARY->TENSORFLOW/KERAS
- **Evaluation method:** KPI is True if the following related KPIs are True:
  - UC6_KPI_IP_01;
  - UC6_KPI_IP_02;
  - UC6_KPI_IP_03;
  - UC6_KPI_IP_04;
  - UC6_KPI_IP_05.

5.7.1.3.10 UC6_KPI_FT_09

- **Description:** The nodes are able to perform load balancing.
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** RELIABILITY->AVALABILITY->LOAD_BALANCING
- **Evaluation method:** KPI is True if the following related KPIs are True:
  - UC6_KPI_IP_06;
  - UC6_KPI_IP_07;
  - UC6_KPI_IP_09;
  - UC6_KPI_IP_10.

5.7.1.3.11 UC6_KPI_FT_10

- **Description:** The nodes are able to monitor their performances**.**
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** SAFETY->MONITORING->PERFORMANCES
- **Evaluation method:** KPI is True if the following related KPIs are True:
  - UC6_KPI_IP_07;
  - UC6_KPI_IP_08.

5.7.1.3.12 UC6_KPI_FT_11

- **Description:** The nodes can acquire video streams from a camera.
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** CONTEXT->AWARENESS->SENSORS->CAMERA
- **Evaluation method:** KPI is True if the following related KPIs are True:
  - UC6_KPI_IP_01;
  - UC6_KPI_IP_02;
  - UC6_KPI_IP_03;
  - UC6_KPI_IP_04.

5.7.1.3.13 UC6_KPI_FT_12

- **Description:** The nodes can acquire audio streams from a microphone.
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** CONTEXT->AWARENESS->SENSORS->MICROPHONE
- **Evaluation method:** KPI is True if the following related KPI is True:
  - UC6_KPI_IP_06.

5.7.1.3.14 UC6_KPI_FT_13

- **Description:** The nodes can generate and transmit alarms.
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** CONTEXT->AWARENESS->ACTIONS->ALARM
- **Evaluation method:** KPI is True if the following related KPI is True:
  - UC6_KPI_IP_02.

5.7.1.3.15 UC6_KPI_FT_14

- **Description:** The nodes have an Ethernet interface.
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** FRACTALITY-> COMMUNICATIONS /CONNECTIVITY->TECHNOLOGIES->ETHERNET
- **Evaluation method:** KPI is True if the following related KPIs are True:
  - UC6_KPI_IP_01;
  - UC6_KPI_IP_02;
  - UC6_KPI_IP_07;
  - UC6_KPI_IP_10.

5.7.1.3.16 UC6_KPI_FT_15

- **Description:** The nodes have a WI-FI interface.
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** FRACTALITY-> COMMUNICATIONS /CONNECTIVITY->TECHNOLOGIES->WI-FI
- **Evaluation method:** KPI is True if the following related KPIs are True:
  - UC6_KPI_IP_01;
  - UC6_KPI_IP_02;
  - UC6_KPI_IP_07;
  - UC6_KPI_IP_10.

5.7.1.3.17 UC6_KPI_FT_16

- **Description:** The nodes support MQTT communication
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** FRACTALITY-> COMMUNICATIONS /CONNECTIVITY -> DATAPROTOCOLS->MQTT
- **Evaluation method:** KPI is True if the following related KPIs are True:
  - UC6_KPI_IP_01,
  - UC6_KPI_IP_02,
  - UC6_KPI_IP_07,
  - UC6_KPI_IP_10.

5.7.1.3.18 UC6_KPI_FT_17

- **Description:** The nodes are implemented on Versal.
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** OTHER:NON-FUNCTIONAL->PLATFORM->VERSAL-ARM
- **Evaluation method:** KPI is True if the following related KPI is True:
  - UC6_KPI_IP_10.

5.7.1.3.19 UC6_KPI_FT_18

- **Description:** The nodes are implemented on Zynq UltraScale+.
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** OTHER:NON-FUNCTIONAL->PLATFORM->ZYNQ_ULTRASCALE+
- **Evaluation method:** KPI is True if the following related KPI is True:
    - UC6_KPI_IP_10.

5.7.1.3.20 UC6_KPI_FT_19

- **Description:** The nodes execute Linux OS.
- **Result type:** True/False
- **Helps to demonstrate UC Feature:** OTHER: NON-FUNCTIONAL->PLATFORM->OS->LINUX
- **Evaluation method:** KPI is True if the following related KPI is True:
    - UC6_KPI_IP_10.

5.7.2 Use Case Requirements Validation methods

Use case requirements validation methods are defined under the KPI defined for Use Case Requirements:

5.7.2.1 KPIs for Use Case Requirements

5.7.2.1.1 UC6_KPI_IP_Req_01

- **Description**: Cognitiveness Requirements
- **Result type**: True/False.
- **Evaluation method**: KPI is True if the following related KPIs are True:
    - UC6_KPI_IP_01;
    - UC6_KPI_IP_02;
    - UC6_KPI_IP_03;
    - UC6_KPI_IP_04;
    - UC6_KPI_IP_05.
- **Comment**: Related to REQ_UC6_01, REQ_UC6_03, REQ_UC6_04, REQ_UC6_05, REQ_UC6_06, REQ_UC6_07, REQ_UC6_08, REQ_UC6_09, REQ_UC6_16, REQ_UC6_21 defined in WP2.

5.7.2.1.2 UC6_KPI_IP_Req_02

- **Description**: Monitoring & Management Requirements
- **Result type**: True/False.
- **Evaluation method**: KPI is True if the following related KPIs are True:
    - UC6_KPI_IP_01;
    - UC6_KPI_IP_02;
    - UC6_KPI_IP_03;
    - UC6_KPI_IP_04;
    - UC6_KPI_IP_05;
    - UC6_KPI_IP_06;
    - UC6_KPI_IP_07;
    - UC6_KPI_IP_08;

- UC6_KPI_IP_09;
- UC6_KPI_IP_10.

- **Comment**: Related to REQ_UC6_02, REQ_UC6_11, REQ_UC6_12, REQ_UC6_13, REQ_UC6_14, REQ_UC6_15, REQ_UC6_17, REQ_UC6_18, REQ_UC6_19, REQ_UC6_20, REQ_UC6_22, REQ_UC6_23, REQ_UC6_24, REQ_UC6_25, REQ_UC6_27, REQ_UC6_28.

5.7.2.1.3 UC6_KPI_IP_Req_03

- **Description**: User Experience Requirements.
- **Result type**: True/False.
- **Evaluation method**: KPI is True if the following related KPIs are True:
  - UC6_KPI_IP_07;
  - UC6_KPI_IP_08;
  - UC6_KPI_IP_10.
- **Comment**: Related to REQ_UC6_10, REQ_UC6_26 defined in WP2.

5.7.3 Components Validation

As already said, components used by each FRACTAL Use Case can be grouped as:

- *Specific* components produced by the Use Case – partially validated by corresponding KPI defined in Section 5.5;;
- General *common* FRACTAL Components defined during the project used by the Use Case – fully validated by corresponding KPI.

5.7.3.1 Case Specific Components

5.7.3.1.1 UC6_CMP_01

Density estimator (DE) is a CNN-based system for people density estimation. The DE component takes as input an IP-camera video flow and using the neural accelerators of the ZUS+, outputs the density estimation. The rate of people is sent to the other components using MQTT-based communications. Configuration parameters can be changed at runtime using MQTT.

Validation is considered done if the following conditions are verified:

1. THROUGHTPUT [multithread, multi-dpu] > 18 img/sec;
2. THROUGHTPUT [single thread, single dpu] > 5 img/sec;
3. ACCURACY > 60%;
4. RESPONSE TIME (Real-Time Streaming Protocol (RTSP) acquisition + pre-processing + inference + post-processing) < 0.3 s.

5.7.3.1.2 UC6_CMP_02

The People Detector (PD) is a SW component that implements People Detection algorithms (WP5T56_01), alarm generation module (analyzing metadata generated by people detector) and communication interfaces.

Validation is considered done if the following conditions are verified:

1. Accuracy [mAP] > 85%;
2. Throughtput> 8 img/sec;
3. Inference Time < 120 ms;
4. Response Time(image acquisition+ inference (3 consecutive detections) + MQTT alarm publishing) < 500ms.

5.7.3.1.3 UC6_CMP_03

The Face Detector (FD) component allows to extract a face crop of a person in front of the smart totem. The face bounding box is computed using a CNN-based detector, that takes advantage of a Vitis-AI acceleration. The output of the component is an image that contains a face. The output image is forwarded to the other components using shared memory.

Validation is considered done if the following conditions are verified:

1. Throughtput[single thread, single dpu] > 20 img/sec;
2. Accuracy [mAP] > 0.8925;
3. Response Time (RTSP acquisition + pre-processing + inference + post-processing) < 0.05 s.

5.7.3.1.4 UC6_CMP_04

Age Estimator (AE) is based on a convolutional neural network. The system takes as input the picture of a person and outputs an estimation of the age.

Validation is considered done if the following conditions are verified:

1. Mean Square Error < 50;
2. Response Time < 500ms.

5.7.3.1.5 UC6_CMP_05

Gender Classifier (GC) is based on a convolutional neural network. The system takes as input the picture of a person and classifies it as female or male.

Validation is considered done if the following conditions are verified:

1. Accuracy > 90%;
2. Response Time < 500ms.

5.7.3.1.6 UC6_CMP_06

Idiom Recognition (IR) is based on speech processing. The system outputs the language of the current speaker by processing the pre-recorded audio. If necessary, the IR component can also elaborate "live" audio.

Validation is considered done if the following conditions are verified:

1. Accuracy > 90%;
2. Response time < 2s.

5.7.3.1.7 UC6_CMP_07

Runtime Manager (RM) is a Component that manages node-to-node communications, interaction between components and task scheduling based on input received.

Validation is considered done if the following conditions are verified:

1. Latency time < 300ms based on worst case scenario.
   Several test case cases will be defined to test the component in different situations. The more complex one will be defined as "worst case scenario".

5.7.3.1.8 UC6_CMP_08

Rule-based Recommender is a component implemented as a REST server which receives the data of the people as a JSON dictionary {"age", "gender", "language"} and it responds with the suggestion computed using the already trained rule-based clear-box algorithm.

Validation is considered done if the following conditions are verified:

1. Accuracy > 70%;
2. Response Time < 100ms.

5.7.3.1.9 UC6_CMP_09

Data Compressor (DC) is a component that performs data compression and decompression operations through an energy-aware version of the LZW algorithm.

Validation is considered done if the following conditions are verified:

1. Saved Space > 10%
2. Accuracy = 100%
3. Response Time < 100ms

5.7.3.2 FRACTAL Common Components

Such components are partially validated thanks to the validation of the case specific components in which they are included.

For example, FRACTAL Common component WP3T32-07 is used by case specific components UC6_CMP_04 and UC6_CMP_05. So WP3T32-07 can be partially validated thanks to UC6_CMP_04 and UC6_CMP_05 validation (done in Sections 5.7.3.1.4 and 5.7.3.1.5 respectively).

For case specific components validation, please consider section 5.7.3.1. For the link between Case Specific components and FRACTAL Common components, please consider section 5.6.4.

6 VAL-UC7 Autonomous robot for implementing safe movements

The "Smart Physical Demonstration and Evaluation Robot" (SPIDER) [48] is an autonomous robot prototype. Within this Use Case, the Cognitive Edge Node developed in FRACTAL will be integrated into the autonomous robot SPIDER and evaluated against its applicability for performing computationally intensive relevant vehicle functions of variable complexity at the edge of the network (near the source of the data) while still being able to guarantee extra-functional properties (dependability, timeliness) for preserving safety- and security operational behaviors.

Figure 53 - Smart Physical Demonstration and Evaluation Robot (SPIDER)

The SPIDER is a prototype robot developed completely by Virtual Vehicle Research. This includes the planning of the chassis, mechanical construction, electronics, and software system. This deep insight into the architecture makes it possible to integrate the FRACTAL components into the vehicle in a targeted manner. The SPIDER robot is used as a Mobile Hardware-in-the-Loop (HIL) platform for testing and verification of sensors and automated driving functions in the automotive sector. The cooperation with ALP. Lab GmbH enables a commercial use of the robot, while Virtual Vehicle uses the SPIDER for research purposes.

An adaptable, open hardware and software architecture enables the testing of software architectures, perception, and control systems. The SPIDER can imitate the system under test. The easily adaptable mounting rod system, visualized in Figure 54, on the chassis allows sensor positions to be quickly and easily adapted to the target system.

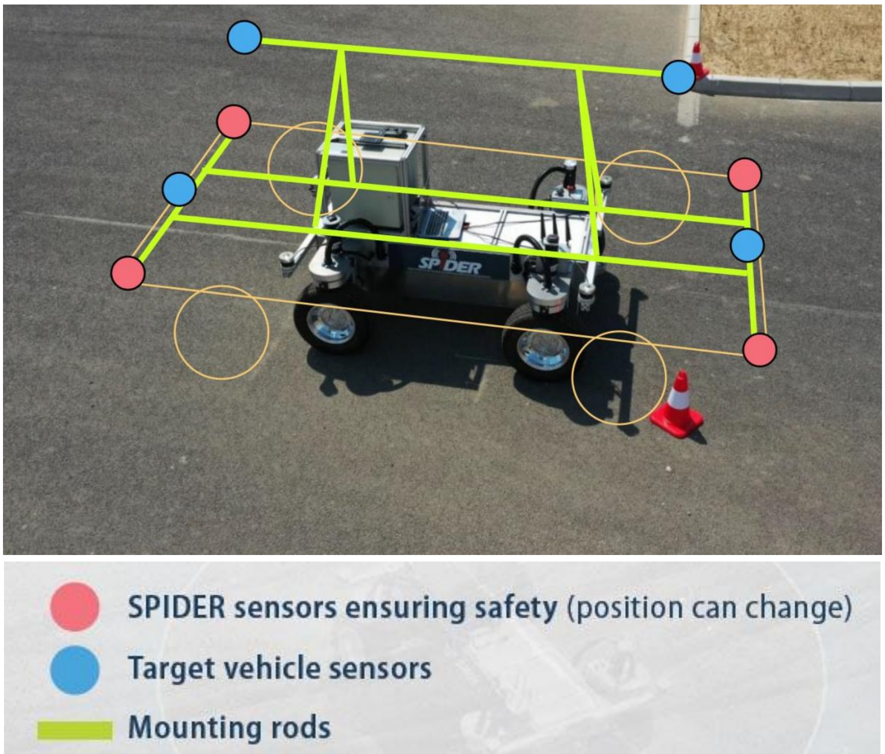


Figure 54 – SPIDER matching sensor positions

Four individually controllable wheels enable almost omni-directional movement. As shown in Figure 55, the SPIDER's software system controls these wheels in a way to precisely mimic the movements of the target vehicle, with a completely different geometry. Integrated safety functions ensure that these tests can be carried out without risk already at the prototype stage of the system under test.

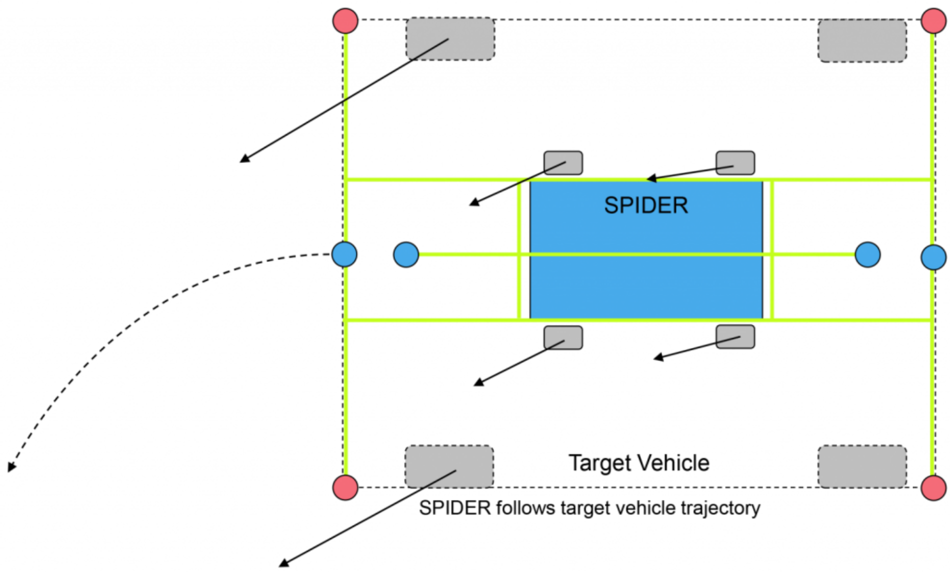


Figure 55 – SPIDER target movement imitation

6.1 Before FRACTAL

This section describes the state of the SPIDER Platform before the integration of the FRACTAL components. Subsections 6.1.1 and 6.1.2 describe the hardware components and the system architecture. In subsections 6.1.3 and 6.1.4, the Collision Avoidance Function (CAF) and the Path Tracking Function (PTF), which are the central elements of the UC7, are examined in more detail.

6.1.1 SPIDER Hardware

The base of the SPIDER is about 380 kg and can be extended easily using mounting rods. Figure 56 shows the SPIDER with an optional sensor box mounted. The vehicle is battery-powered with a top speed of 50 km/h. All four wheels can be controlled individually with a level of freedom of 270 degrees, enabling a pseudo omni-directional movement.

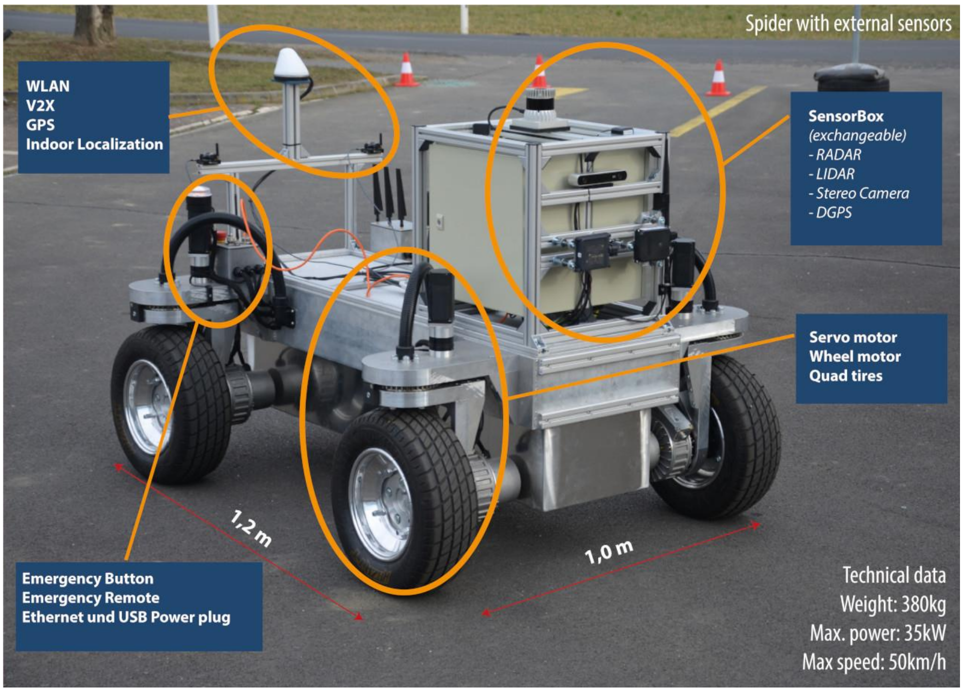


Figure 56 – SPIDER Hardware Overview

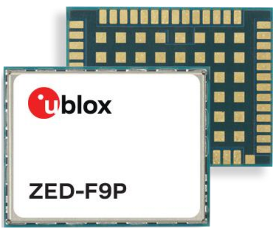
The SPIDER is controlled by two computing platforms, described in Table 4.

Low Level Control Unit (LLCU)	32-Bit TriCore CPU Infineon AURIX	The LLCU is used for controlling and monitoring SPIDER driving hardware, such as the high voltage batteries, motor controllers, and servos.

		
High Level Control Unit (HLCU)	Industrial PC with x64 quad core processor and Nvidia graphics card with CUDA. 	The HCLU is connected via ethernet to the LLCU. It is responsible for all SPIDER functions on application level, like user interaction, path planning tracking, sensor integration and fusion, and localization.

Table 4 - SPIDER computing platforms

Further, the core sensor setup of the platform is described in Table 5.

Inertia Measurement Unit (IMU)	Xsense MTi-630 sensor delivers data on acceleration and velocity for all 6 degrees of freedom with a high frequency up to 100 Hz. 	The velocity and acceleration measurements are fused with odometry information from the motor controllers and position information from the GPS sensors to calculate a precise localization (robot position) with a high frequency of at least 50 Hz.
Differential Global Positioning System (dGPS) system	Two u-blox F9P modules with two separate antennas provide GPS positioning information at cm-level. 	By triangulating the position information from the two antennas the system can calculate an absolute orientation in the world. The GPS position and orientation is the base of the SPIDER localization system and further improved by fusing odometry and IMU data.

Light detection and ranging (Lidar) sensors	Four Ouster OS0 360° lidar sensors with 16 lines provide a pointcloud with a resolution of 1024 points at 10 Hz. 	The pointclouds of the four sensors are filtered, fused, and translated to an occupancy grid. This grid is used for detecting obstacles on the planned path of the robot.

Table 5 - SPIDER sensor setup

6.1.2 SPIDER System Architecture

As described in the last section, the SPIDER has a two-tier system architecture. Figure 57 gives an overview of the functions of the SPIDER system.

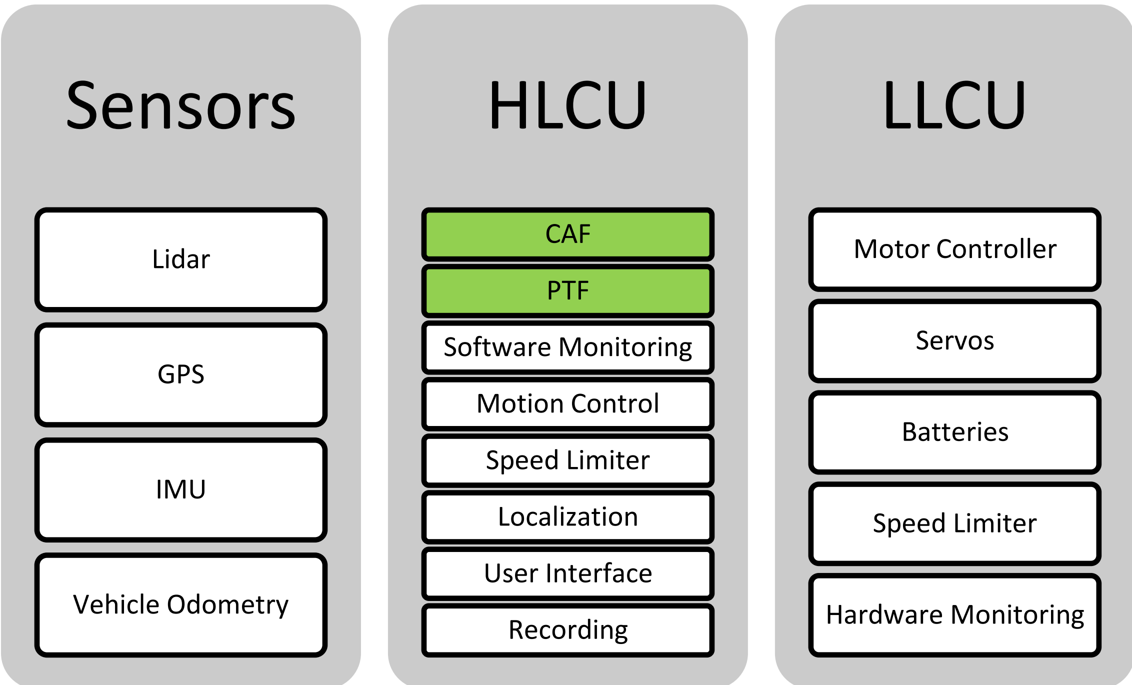


Figure 57 – SPIDER System Architecture

The central node is the HLCU, an industrial PC with powerful computing capacity. The sensors are connected to the HLCU via Ethernet or USB. The HLCU reads the commands from the operator, determines the optimal trajectories, or calculates robot and obstacle positions from the sensor data. A GPU is used for computationally intensive operations. The software stack of the HLCU runs on an Ubuntu Linux distribution and uses the Robot Operating System (ROS) [49] which is available as open source [50]. ROS includes many ready-made libraries and tools in the robotics field and offers a uniform interface to external systems. In the ROS framework at the

HLCU are also running the CAF (section 6.1.3) and PTF (section 6.1.4), which will be mainly adapted for the FRACTAL use-case.

The LLCU provides the connection via CAN interface to the base hardware components like motors and batteries. It uses a safety certified Infineon Aurix microprocessor. The LLCU receives the target values for the motors via an UDP interface from the HLCU, performs safety checks on them and passes those to the corresponding hardware components. Further the LLCU monitors the connection to the HLCU and can initiate a safe stop if the connection is broken, or in case of a hardware failure.

6.1.3 Collision Avoidance Function

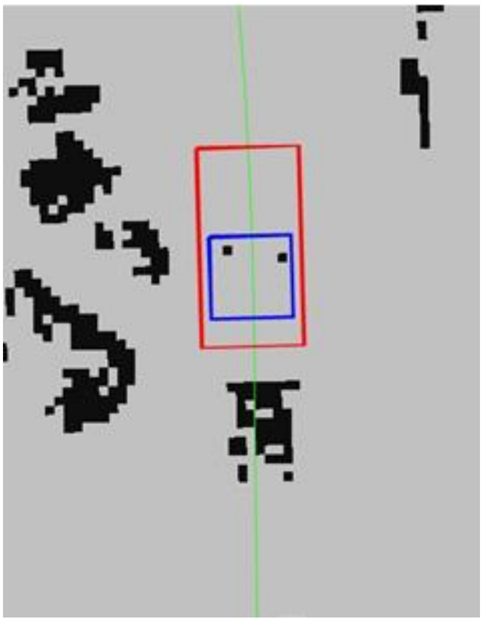


Figure 58 - Cost map showing obstacles (black), the planned path (green), the chassis zone (blue) representing the physical dimensions of the robot and the danger zone (red).

The task of the Collision Avoidance Function (CAF) is to detect obstacles and initiate measures to avoid a collision. Four Lidar sensors, which are mounted on the outer corners of the robot, enable an all-round view with at least double redundancy at about 50 meters. In a preprocessing step the point cloud data from the sensors is filtered and fused using the Point Cloud Library (PCL) [51]. The resulting fused point cloud is mapped onto a two-dimensional grid, called cost map, with occupancy values for each grid cell. Figure 58 shows a visualization of the cost map. Based on the cost map, an algorithm calculates the distance to the closest obstacle in the movement direction, and triggers and emergency brake if the obstacle distance is inside a defined danger zone around the robot.

6.1.4 SPIDER Path Tracking Function

The Path Tracking Function (PTF) is intended to follow a pre-defined global path in a precise manner. A global path is an ordered list of waypoints, which shall be touched by the robot. A waypoint is defined by the coordinates, target speed and orientation of the vehicle. When the PTF is activated by the human operator, the function calculates a trajectory to the next waypoint from the current location of the robot. From this trajectory the required speed and direction is computed and forwarded to the motion control unit of the SPIDER. The underlying control algorithm, used by the PTF, is a Stanley [52] control approach.

6.2 Use Case Objectives

Currently, the SPIDER is only available as a prototype. The goal of Virtual Vehicle Research, in cooperation with its partners, is clearly to develop the SPIDER as a product. This also affects the computing units used, as described in subsection 6.1.1.

To perform rapid tests in the automotive field, the SPIDER has a modular design which is easily expandable. Nevertheless, the functions of the test platform must ensure a safe ride. This requires the implementation of a safety and security concept that includes computationally intensive and at the same time safety-critical algorithms such as obstacle avoidance or path planning. For the execution of these algorithms, an industrial PC with GPU support is required on the SPIDER prototype, which is too maintenance-intensive and expensive for use in product development.

The FRACTAL platform based on RISC-V offers interesting concepts for safety, security, and processing of AI algorithms with hardware support. In addition, the architecture of the FRACTAL nodes enables a similarly modular structure of the software architecture (fractality).

In the implementation of UC7, relevant parts are removed from the existing SPIDER architecture and implemented as FRACTAL nodes. The first function is the CAF, which is of relevance for the safety concept due to its task. By replacing the CAF, see subsection 6.1.3, as a FRACTAL node, it should be shown that it is possible to implement safety and security-relevant deterministic functions with high computing effort using the FRACTAL platform. As a second function, the existing PTF, see subsection 6.1.4, is embedded in a FRACTAL node with a new AI-based concept. With this new concept, we expect not only an improvement of the algorithm with reduced computing power, but also the integration of an AI function into the safety and security concept.

6.2.1 Collision Avoidance Function Objectives

The aim of the CAF is to avoid damage and most importantly human harm by preventing collisions or reducing the impact speed in the event of an unavoidable crash. The CAF uses environmental sensors, which constantly measure the distance to surrounding objects, and initiates speed limits or emergency stops in case of objects getting too close to the SPIDER.

Due to its function, the CAF is an important component in the safety concept of the SPIDER. In the FRACTAL context, the function is used as an example to show how safety-relevant functions can be implemented as FRACTAL nodes. For this, time-critical behavior, monitoring, and redundant execution are required. The successful porting of the CAF is a necessary step to drive the development of the SPIDER functions towards certification and series development.

6.2.2 Path Tracking Function Objectives

The objective for the PTF of the Use Case is to develop a path tracking controller for the SPIDER incorporating a reactive obstacle avoidance strategy. The latter means that if an obstacle is detected in the vehicle's surrounding an evasion maneuver must be initiated to avoid a collision.

The scientific literature knows several non-data-driven methods for the solution of the path tracking and obstacle avoidance problem. Popular and well-known path tracking controllers are for example the "Pure pursuit controller", the "Carrot chasing controller", or the "Stanley controller". For details on the functioning of these algorithms see for example the works of Gutièrrez et al. [53], Perez-Leon et al. [54] Samuel et al. [55], and Hoffmann et al. [56] Approaches for the design of obstacle avoidance controllers, such as the artificial potential field method, can be found in the works of Rostami et al. [57], Wiig et al. [58] and Leca et al. [59] However, we decided to follow an ML approach to solve the task. The reason for this decision is twofold. On one hand, it seemed to us to be difficult to appropriately tune and coordinate a combined controller consisting of a path tracking and collision avoidance component. On the other hand, an ML solution promises a lower computational effort at runtime.

The controller, which is represented by means of an artificial neural network, is designed in such a way that it takes vehicle state data and cost maps as input and outputs control values affecting the linear and angular velocities of the vehicle. The vehicle state comprises information on the current position of the vehicle, its heading, and its linear and angular velocities. The cost map is a 2D occupancy grid representing the immediate surrounding of the SPIDER provided by the vehicle's perception system.

The underlying neural network is composed of two input streams. These streams process the two different kinds of input data, vehicle state and cost map, by means of at most two network layers and are merged in a common network layer. The overall network consists of these input streams, one output layer and no more than three hidden layers. The use of convolutional layers is omitted. The training of the neural network is based on a Python driving simulation and was implemented using the Python packages Keras [60] and Tensorflow [61]. Even though the network is comparable small a GPU supported computer is used to speed up the training procedure.

Besides the scientific and technical appeal of implementing a ML controller on the FRACTAL platform, we expect a cost saving from implementing this kind of a controller on a RISC V based system compared to implementing it on an industrial PC with GPU. This point is essential for the further development of SPIDER to a commercial product.

6.3 State of the art

The SPIDER platform is used as mobile hardware-in-the-loop test platform for testing automated driving systems. Schwarzl et al. [62] introduce the topic of testing in the automotive industry with focus of the SPIDER robot. The focus of UC7 is on the development of safe and secure functions, especially AI functions. Accordingly, the following subchapters provide an insight into standards and technologies in the field of safety, security, and AI in the context of SPIDER in the automotive testing area.

6.3.1 Functional Safety

Functional Safety is the part of the overall safety of a system or piece of equipment that depends on the system or equipment operating correctly in response to its inputs. This includes the safe management of likely operator errors, hardware and software failures and any changes in the environmental conditions.

Looking at Functional Safety standards the IEC 61508 "Functional safety of electrical/electronic/programmable electronic (E/E/PE) safety-related systems" [63] is a basic Functional Safety standard applicable to all kinds of industry. IEC 61508 is a domain independent, generic standard for Functional Safety of these E/E/PE safety-related systems, which provides a basic guidance how to deal with and how to achieve Functional Safety for such systems.

From this basic standard numerous domain specific standards have been derived. The ISO 26262 "Road vehicles – Functional safety" [64] represents the automotive specialization of IEC 61508 and provide a framework for achieving Functional Safety for electrical and/or electronic(E/E) systems in road vehicles. The spider is developed according to the standard ISO 26262.

Another standard that is considered for the development of the spider is ISO/FDIS 21448 "Road vehicles — Safety of the intended functionality" [65]. The Safety of the intended functionality (SOTIF) is defined as "the absence of unreasonable risk due to hazards resulting from functional insufficiencies of the intended functionality, or by reasonably foreseeable misuse by persons". The standard ISO 21448 provides guidance on the applicable design and verification and validation measures needed to achieve the SOTIF. It does not apply to cases covered by the ISO 26262 or to hazards directly caused by the system technology.

For the autonomous robot SPIDER also the technical report ISO/TR 4804 "Road vehicles — Safety and cybersecurity for automated driving systems — Design, verification and validation" [66] is relevant and must be considered. The standard provides an overview and guidance of the steps for developing and validating an automated vehicle equipped with a safe automated driving system. It considers safety by design, verification and validation methods for automated driving focused on SAE level 3 and level 4 vehicles according to ISO/SAE PAS 22736 [67]. The ISO/TR 4804 is an informative technical report, and it will be elaborated as a normative standard technical specification in the WG13 in the new document ISO/AWI TS 5083 "Road vehicles — Safety for automated driving systems — Design, verification and validation" [68], which is under elaboration. Virtual Vehicle is an active participant and contributor in several ISO working groups related to functional safety and automated driving.

6.3.2 Security

In the past security in the automotive industry was mainly focused on the security of electronic control units (ECU). With first reports on vulnerabilities [69], the whole system got attraction in the evaluation of security. To cover these recent developments, the standard ISO/SAE 21434 [70] was introduced. The standard specifies guidelines for cybersecurity risk management of series production road vehicle E/E systems. The document describes the relevant security tasks for the whole development pipeline of the vehicle. For the execution of UC7 we are primarily interested in the *Threat Analysis and Risk Assessment Methods* and the related phases *Concept* and *Product Development*.

6.3.3 Artificial Intelligence

In recent years, a trend towards the development of AI-based control units has been evident in the automotive sector. In this context, approaches from the field of machine learning (ML) have attracted particular interest.

According to Russell and Horvig [71], Nilsson [72] there is no clear definition of the term artificial intelligence (AI). Based on the definitions given in these textbooks, AI may be described as the scientific field dealing with the development of entities which function appropriately and with foresight in their environment. According to this definition, systems that can perceive their environment and perform actions or make decisions based on this perception can be identified as AI-systems. As an example, consider a system for the autonomous detection and recognition of traffic signs – see for instance the approach provided by Tabernik and Skocaj [73]. Recently, so called machine learning (ML) solutions or more general data-driven AI solutions have attracted particular attention. Following Jordan and Mitchell [74], these approaches are characterized by their ability to automatically extract correlations or generate knowledge from large amounts of data to improve performance in regard of a complex decision or control problem. Systems of this type are often referred to as learning systems. This process of generating knowledge or learning is often represented by a numerical minimization problem of an error or cost functional. If a learning system is modeled by means of artificial neural network (ANN), it can be interpreted as a parameter-dependent function. Consequently, learning from data means in this context to fit the network parameters best to the available data in terms of the underlying minimization problem.

The field of ML can be roughly divided into the three subfields 'supervised learning', 'unsupervised learning' and 'reinforcement learning'. The area of supervised learning basically deals with deriving a classifier from a set of labeled data. In contrast, approaches from the area of unsupervised learning dispense with the assumption that classified data are available and aim to learn patterns and structures away from a known classification or data noise. The third area, reinforcement learning, can be seen as a hybrid of the above paradigms. Like in the setting of unsupervised learning, RL methods don't make use of labeled data. However, through repeated interaction of the learning system with its environment, the system receives positive or negative feedback in terms of numerical rewards. RL-algorithms seek to derive a control strategy that generates optimal feedback over a given time horizon. This basic principle motivates the use of RL methods especially for finding control units in robotics. According to Kiran et al. [75], the character and the requirements of autonomous driving scenarios encourages the usage of RL approaches also in the automotive sector. The use of an RL approach to develop a data-driven controller for the SPIDER thus appears to be an appropriate choice. However, as shown in the work of Bojarski et al [76], not only RL approaches are suitable for this purpose.

The choice to use RL for the development of the sought controller requires to define a numerical reward function which is aligned with its intensions. In the given case this means that actions which avoid collision with obstacles and let the vehicle follow the given path should be rewarded. An appropriate and balanced choice of reward strategy is critical. If too much emphasis is placed on collision avoidance, then the incentive to move towards waypoints along the path may be too low. Conversely, if path following is prioritized too much, there is a risk that the collision avoidance task will be neglected. The issue of aligning the reward function to the developers intend is well known and is called value alignment – see for example the work of Taylor et al. [77] and as well Amodei et al. [78]

6.4 Main contributions expected from FRACTAL

The contributions expected from FRACTAL can be derived from the Use Case objectives described in section 6.2. For the productive use of an adaptive mobile robot (such as the SPIDER), a framework is needed that supports computationally intensive algorithms and AI functions while still allowing safe execution. The FRACTAL project provides such a framework with the pillars and objectives listed in Table 2. The Use Case is not connected to a cloud and therefore focused on Pillars 1-3.

The main impact to the SPIDER application is expected from Pillar 1. UC7 is planned to be implemented on a RISC-V based NOEL-V processor model running Linux operating system. The platform shall replace an industrial PC in the SPIDER hardware setup. Various FRACTAL components provided within Pillar 1 will support the realization of UC7. Those components include monitoring units and a diverse redundancy library for safety needs of the functions, and an AI accelerator for performance needs of the used AI models.

Pillar 2 contributes to UC7 with further components to the safety and security requirements of the functions by adding services at FPGA and application level.

The components coming from Pillar 3 are providing services for deployment, like Docker, and validation, like Jupyter, to the Use Case.

6.5 Evaluation of the implementation results

This section defines the KPIs defined for UC7 implementation. These KPIs are classified into three groups:

- KPIs for Implementation Plan Task;
- KPIs for FRACTAL Objectives related to FRACTAL Pillars;
- KPIs for UC Features.

For each KPI, an *Identifier*, a *Description* and the type of result *Value* is defined. The *Test* to be performed for the KPI will be defined later in the Justification Plan, therefore is marked as TBD (To Be Defined).

Next subsections describe in some detail the three groups of KPIs.

6.5.1 KPI for Implementation Plan Tasks

This section defines the ***KPIs defined for the Implementation Tasks***. Figure 59 shows the complete list of KPIs defined for the Implementation Tasks of UC7.

KPI UC for Implementation Plan				
KPI ID	**Description**	**Value**	**Test**	**Comment**
UC7_KPI_IP_01	All subtask success	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_02	Linux on NOEL-V is booting on FPGA	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_03	Simple publisher/subscriber example is running on target platform	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_04	Max data transfer rate deviation of 10 Hz	1 Hz	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_05	All subtask success	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_06	Simulated robot is following trajectory and avoiding obstacles	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_07	Avg. Path Proximity in meter	< 1m	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_08	Collision free rate	> 95%	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_09	Valid ONNX model	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_10	Unit test coverage of PTF	> 75 %	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_11	Unit test coverage of CAF	> 75 %	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_12	Loop rate of CAF function	>= 10 Hz	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_13	Resource monitoring tests in simulation sucessfull	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_14	Redundancy library tests in simulation sucessfull	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_15	All subtask success	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_16	Functions on target platform running with sensor data from 3d simulation	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_17	Functions on target platform running with sensor data from real world test	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_18	Metrics calculated with Jupyter availaible	Proximity, Collision rate, Time consumption, Loop rate	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_Req_01	Processing time of costmap distance	< 100ms	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_Req_02	SPIDER stops in defined emergency situation	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_Req_03	Avg. Path Proximity in meter of the PTF node	< 1m	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_Req_04	Collision free rate of the PTF node	> 95%	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_Req_05	SPIDER stops at connection loss to edge nodes	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_Req_06	SPIDER stops at timeout of edge nodes	200 ms	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_Req_07	Update rate of costmap input data to edge nodes	> 9 Hz	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_Req_08	Edge nodes can exchange data via TCP/UDP with SPIDER	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_Req_09	ROS2 stack installed on target platform	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_Req_10	Library for diverce redundancy is build on target platform	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_Req_11	LEDEL library build for target platform	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_Req_12	Resource monitoring library build for target platform	True/False	TBD	*Defined for Implementation Tasks*
UC7_KPI_IP_Req_13	Hardware accelerator for NN model of UC7 integrated to target platform	True/False	TBD	*Defined for Implementation Tasks*

Figure 59 - KPIs for UC7 Implementation Plan Tasks

The KPIs are divided into two subgroups:

- ***KPIs specifically defined for each Task*** – These KPIs have been defined to check the success of the task. When possible, they are defined as a numerical criterion (i.e.,inference time < 100 ms), otherwise they are defined as a True/False indicating that the task finished successfully.

- ***KPIs related to Tasks, allowing checking the Requirements* defined by *the UC in the general Excel defined in WP1 FRACTAL - Requirements_KPIs_Components.xlsx*** (see Tab Requirements) - These KPIs have been defined taking into account the general requirements posted by the Use Case. These KPIs are defined as a True/False value indicating that the task finished and allows checking whether the requirements are met.

6.5.2 KPI for FRACTAL Objectives related to FRACTAL Pillars

KPIs defined to measure ***how the Implementation Tasks contribute to demonstrate the FRACTAL Objectives*** (Related to Pillars and found in the FRACTAL proposal, Section 1.1.2.). Figure 60 shows the complete list of KPIs defined for this purpose.

KPI for Fractal Objective (an related Pillar)				Helps to demonstrate the following Fractal Specific Objective		
KPI ID	**Description**	**Value**	**Test**	**ID Obj**	**Description**	**Relates to Pillar**
UC7_KPI_FO_01	FRACTAL path tracking node accelerated to perform with a high frequency.	>= 10 Hz	TBD	O1	Design and Implement an Open-Safe-Reliable Platform to Build Cognitive Edge Nodes of Variable Complexity	Pillar 1 (WP3) - Open-Safe-Realiable and low power node architecture.
UC7_KPI_FO_02	Tests in simulation for redundant execution and monitoring succeed.	True / False	TBD	O2	Guarantee extra-functional properties (dependability, security, timeliness and energy-efficiency) of FRACTAL nodes and systems built using FRACTAL nodes (i.e., FRACTAL systems).	Pillar 2 (WP4) - Low power, safety, security and high-preformance trade-off.
UC7_KPI_FO_03	FRACTAL path tracking nodes AI model generates a collision free path with at an acceptable path proximity.	< 1m	TBD	O3	Evaluate and validate the analytics approach by means of AI to help the identification of the largest set of working conditions still preserving safe and secure operational behaviors.	Pillar 3 (WP5) - Cognitive & Autonomous Node.
UC7_KPI_FO_04	Framework for platform independent development and verification of node functions availaible.	True / False	TBD	O4	To integrate fractal communication and remote management features into FRACTAL nodes	Pillar 4 (WP6) - Mutable and fractal communications.

Figure 60 - KPIs for UC7 Implementation Plan to measure the contribution to FRACTAL Objectives

6.5.3 KPI for UC Features

KPIs defined to measure ***how the Implementation Tasks contribute to demonstrate the UC Features*** (defined in the Tab *FRACTAL Features* in the general Excel defined in *WP1 FRACTAL - Requirements_KPIs_Components.xlsx*). Figure 61 shows the complete list of KPIs defined for this purpose.

KPI for UC Feature				Helps to demonstrate the following UC Feature	
KPI ID	**Description**	**Value**	**Test**	**ID Feat**	**Description**
UC7_KPI_FT_01	Target platform supports ONNX.	True / False	TBD	F1_VIF	ADAPTABILITY - AI - SW - INFERENCE - MODEL - FORMAT - ONNX
UC7_KPI_FT_02	Path tracking function AI model executed at node level.	True / False	TBD	F2_VIF	ADAPTABILITY - AI - SW - INFERENCE - LOCATION-NODE
UC7_KPI_FT_03	Reinforcement learning approach trained model path proximity	< 1m	TBD	F3_VIF	ADAPTABILITY - AI - SW - LEARNING / TRAINING -
UC7_KPI_FT_04	CNN path proximity	< 1m	TBD	F4_VIF	ADAPTABILITY - AI - SW - LEARNING / TRAINING -
UC7_KPI_FT_05	LEDEL library availible for target platform	True / False	TBD	F5_VIF	ADAPTABILITY - AI - SW - LIBRARY - LEDEL
UC7_KPI_FT_06	Sensor data from test drives can be stored on hard drive	True / False	TBD	F6_VIF	ADAPTABILITY - DATA ORCHESTRATION -DATA SET - STORAGE
UC7_KPI_FT_07	Frame rate of collision avoidance function	>= 10Hz	TBD	F7_VIF	RELIABILITY - RESPONSE TIME - FRAME RATE
UC7_KPI_FT_08	Switch to emergency state at time exceedance of AI function	True / False	TBD	F8_VIF	SAFETY - MONITORING - AI ENGINES
UC7_KPI_FT_09	Switch to emergency state at time exceedance of safety relevant function	True / False	TBD	F9_VIF	SAFETY - MONITORING - PERFORMANCE
UC7_KPI_FT_10	Safety relevant processes run redundant on different cores	True / False	TBD	F10_VIF	SAFETY - REDUNDANCY - PROCESSES
UC7_KPI_FT_11	Switch to emergency state at fault detected by diverse redundancy model	True / False	TBD	F11_VIF	SAFETY - REDUNDANCY - DIVERSE REDUNDANCY
UC7_KPI_FT_12	Switch to emergency state at fault detected in the communication messages	True / False	TBD	F12_VIF	SAFETY - REDUNDANCY - COMMUNICATION MESSAGES
UC7_KPI_FT_13	Safety concept according IS 26262 availaible	True / False	TBD	F13_VIF	SAFETY - REGULATION - ISO 26262
UC7_KPI_FT_14	Target platform supports ONNX.	True / False	TBD	F14_VIF	LOW POWER - AI - LIBRARY - MODELS - ONNX
UC7_KPI_FT_15	Lidar sensor messages availaible at target platform at data rate.	10 Hz	TBD	F15_VIF	CONTEXT AWARENESS - SENSORS - LIDAR
UC7_KPI_FT_16	Path planning node tested in target platform on proving ground	True / False	TBD	F16_VIF	CONTEXT AWARENESS - ACTIONS - AI TRIGGERED - PATH PLANNING
UC7_KPI_FT_17	Security assesment according ISO SAE 21434 availaible	True / False	TBD	F17_VIF	SECURITY - REGULATION - ISO SAE 21434
UC7_KPI_FT_18	Max data transfer rate with ethernet, deviation of 10 Hz	1 Hz	TBD	F18_VIF	FRACTALITY - COMMUNICATION / CONNECTIVITY - TECHNOLOGIES - ETHERNET
UC7_KPI_FT_19	Target RISC-V hardware platform based on NOEL-V availaible	True / False	TBD	F19_VIF	OTHER: NON-FUNCTIONAL - PLATFORM - NOEL-V RISC-V
UC7_KPI_FT_20	Linux operating system running on target platform	True / False	TBD	F20_VIF	OTHER: NON-FUNCTIONAL - OS - LINUX

Figure 61 - KPIs for UC7 Implementation Plan to measure the contribution to FRACTAL Features

6.6 Implementation plan

6.6.1 Architecture

6.6.1.1 FRACTAL Big Picture

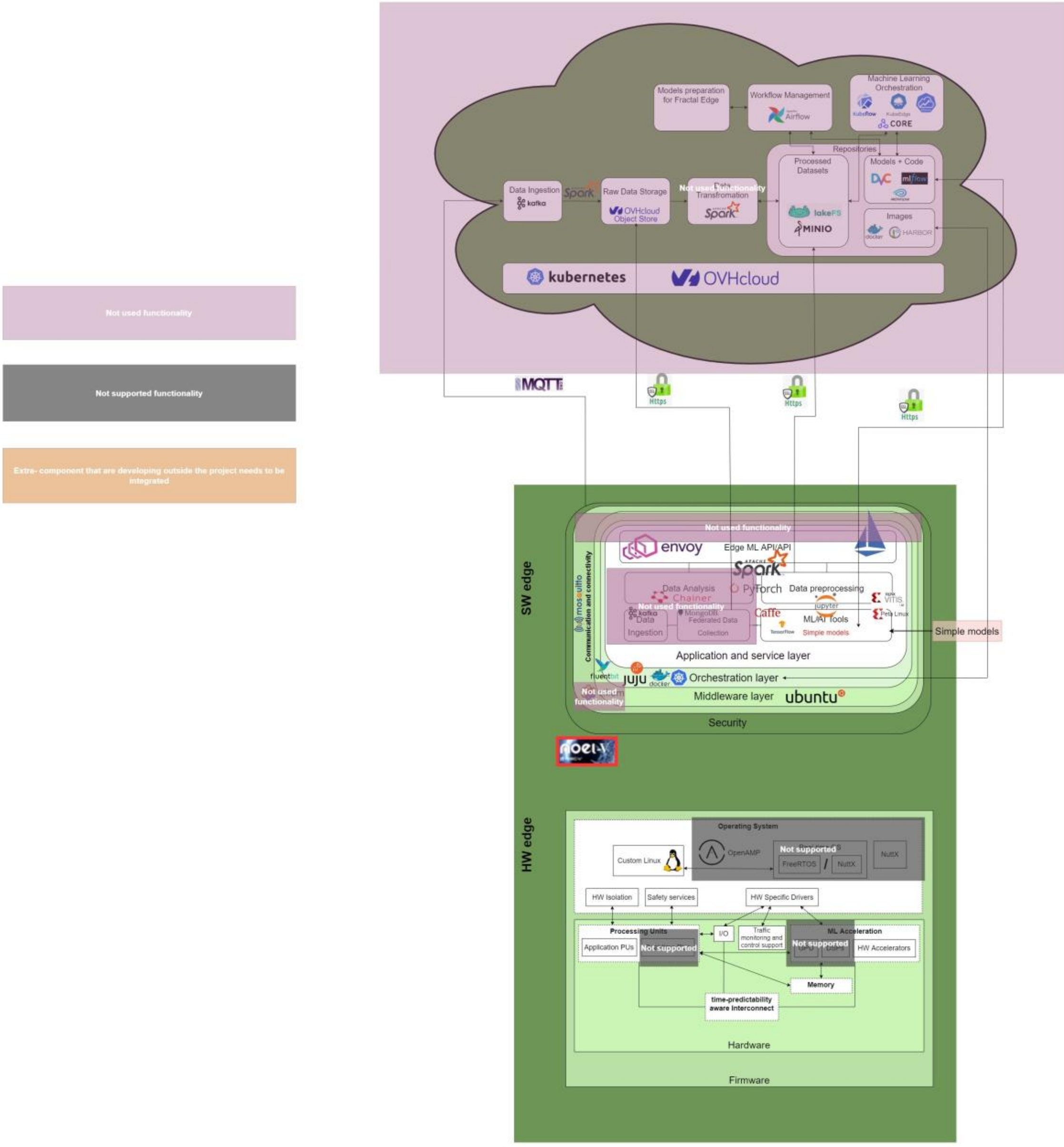


Figure 62 - FRACTAL Big Picture Instantiation for UC7

The contribution of UC7 is to be allocated in the FRACTAL SW edge and HW edge. ML/AI tools are used to deploy a decision-making function on a NOEL-V platform for the control the mobile robot SPIDER. In addition, the redundancy and monitoring libraries are used to redundantly execute and monitor safety relevant functions.

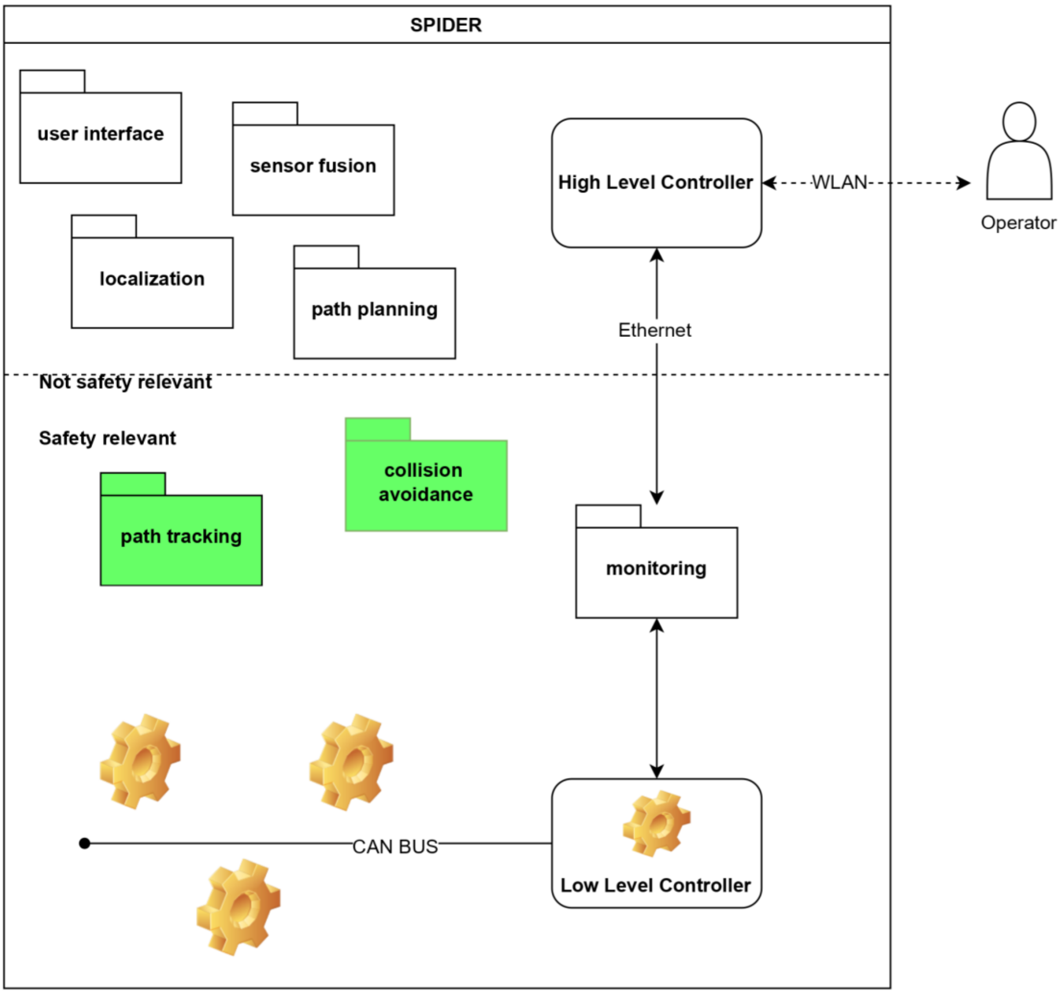


Figure 63 - Architecture of the SPIDER

6.6.2 Tasks

6.6.2.1 Chronogram

Implementation Plan																			
Task ID	**Description**	M19	M20	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30	M31	M32	M33	M34	M35	M36
UC7_T1	**Environment Setup (NOEL-V node)**																		
UC7_T1_1	Setup hardware (NOEL-V)																		
UC7_T1_2	Install sample C++ ROS2 node																		
UC7_T1_3	Connect node to SPIDER via ethernet																		
UC7_T2	**Function Implementation**																		
UC7_T2_1	Development of NN for PTF with Python Keras																		
UC7_T2_2	Training of model																		
UC7_T2_3	Port NN to ONNX																		
UC7_T2_4	Implementation of PTF with ROS2 and LEDDL on NOEL-V																		
UC7_T2_5	Porting CAF to NOEL-V																		
UC7_T2_6	Integration of resource monitoring																		
UC7_T2_7	Integratinon of Redundancy library																		
UC7_T3	**System Evaluation**																		
UC7_T3_1	Build simulation and test functions																		
UC7_T3_2	Real world tests																		
UC7_T3_3	Evaluation with Jupiter (metrics calulation)																		

Figure 64 - UC Implementation Plan Chronogram

6.6.2.2 Task UC7_T1 - Environment Setup (NOEL-V node)

6.6.2.2.1 Sub Task UC7_T1_1 - Setup hardware (NOEL-V)

This task deals with the preparation of the target platform (Xilinx VCU118) by flashing the corresponding bitfile from the SELENE [79] project.

6.6.2.2.2 Sub Task UC7_T1_2 - Install sample C++ ROS2 node

This task deals with the installation of the operating system (Linux) including ROS 2 packages based on the ISAR [80] layer. Dummy nodes are used to check the communication on the target platform.

6.6.2.2.3 Sub Task UC7_T1_3 - Connect node to SPIDER via ethernet

This task deals with the integration of the target platform (configured as in the previous tasks) into the SPIDER network environment. Dummy nodes are used to check if the communication works in the SPIDER network environment.

6.6.2.3 Task UC7_T2 – Function Implementation

6.6.2.3.1 Sub Task UC7_T2_1 - Development of NN for PTF with Python Keras

Based on a Python driving simulation a neural network-based controller for the SPIDER incorporating the capability of following a predefined path and evading static as well as dynamic obstacles shall be developed. The neural network is implemented using the Python library Keras.

6.6.2.3.2 Sub Task UC7_T2_2 – Training of model

For the training of the model algorithms from the field of Reinforcement Learning are used. Thus, due to repeated interaction of the virtual model of the robot with its simulated environment a control strategy maximizing the robot's performance (measured by means of a suitable reward function) is extracted.

6.6.2.3.3 Sub Task UC7_T2_3 - Port NN to ONNX

Using Python libraries like keras2onnx [81] the Keras implementation of the neural network, the controller is based on, is exported into the ONNX format. This allows the usage of the neural network in EDDL, respectively LEDEL.

6.6.2.3.4 Sub Task UC7_T2_4 - Implementation of PTF with ROS2 and LEDEL on NOEL-V

The neural network-based path tracking function (represented by the ONNX model generated in the previous task) is integrated into the ROS environment. Additionally, the evaluation of the function is offloaded to the hardware accelerator.

6.6.2.3.5 Sub Task UC7_T2_5 - Porting CAF to NOEL-V

The CAF is ported from ROS 1 to ROS 2 (running on RISC-V/NOEL-V).

6.6.2.3.6 Sub Task UC7_T2_6 - Integration of resource monitoring

Using the FRACTAL PMU the safety relevant functions of the SPIDER are monitored.

6.6.2.3.7 Sub Task UC7_T2_7 - Integration of Redundancy library

Using the FRACTAL redundancy library the safety relevant functions of the SPIDER are spawned redundantly.

6.6.2.4 Task UC7_T3 – System Evaluation

6.6.2.4.1 Sub Task UC7_T3_1 – Build simulation and test functions

With the target platform in the loop (HiL), using Gazebo [82] on the host system, the robot is simulated, and tests of the path tracking function (PTF) and the collision avoidance function (CAF) are performed.

6.6.2.4.2 Sub Task UC7_T3_2 - Real world tests

Provided the performance of the FPGA board is good enough, the same tests as in Sub Task UC7_T3_1 are performed on the robot.

6.6.2.4.3 Sub Task UC7_T3_3 - Evaluation with Jupyter (metrics calculation)

For the evaluation of the CAF and the PT function suitable metrics are defined. Given a dataset stemming from the real world tests, the metric values are computed in a reproduceable manner using Jupyter notebooks.

In the given context the following metrics are considered:

- Average proximity to path: See UC7_KPI_IP_07.
- Collision free rate: See UC7_KPI_IP_08.
- Time consumption: Measurement of execution time of particular functions.
- Loop rate: Analysis of the frequency at which particular functions get called.

6.6.3 Components

This section summarizes the components involved in the Implementation Plan.

Components are divided into two groups:

- **Components produced by the Implementation Plan** *resulting from executing the Implementation Plan*.

- **FRACTAL components needed to execute the Implementation Plan** (from WP3, WP4, WP5, WP6) that are needed to execute the Implementation Plan.

The following two subsections list these components.

6.6.3.1 Components produced by the Implementation Plan

UC Components		
KPI ID	**Name**	**Description**
UC7_CMP_01	Path Optimization	Neural network for calulating optimal path, based on a planned trajectory and obstacles
UC7_CMP_02	Path Tracking Function	Control algorithm to follow a planned trajectory and avoid obstacles including UC7_CMP_01
UC7_CMP_03	Collision Avoidance Function	Safety relevant function for switching to failsafe mode in case of obstacles in danger zone
UC7_CMP_04	3D Simulation	3D simulation based on Gazebo for testing of developed and integrated functions
UC7_CMP_05	Integrated Demonstration Software on Target	Demonstrating software running on NOEL-V platform integrated to SPIDER

Figure 65 - Components created in UC 7

6.6.3.2 FRACTAL components needed to execute the Implementation Plan

FRACTAL Components needed by the UC		
KPI ID	**Name**	**Description**
WP3T31-01	Edge-oriented monitoring unit	AXI-compliant statistics unit to support safety measures and validation in the context of edge systems.
WP3T34-01	Driver for the edge-oriented monitoring unit	Driver for the statistics unit supporting safety measures and validation in edge systems.
WP3T34-02	Drivers for the SW diverse redundancy library	Driver to read PMCs (Performance Monitoring Counters) from a remote core, and to issue SIG_STOP and SIG_CONT signals to remote cores.
WP3T31-02	Interconnect to support Accelerators integration	Interconnect: AXI pulp library Integration.
WP3T31-03	Safety and security hardware support	Extensions to the interconnect and other NOEL-V components for Security and Safety.
WP3T32-06	Redundant Acceleration Scheme	Integration of a redundant AI inference accelerator in the platform.
WP3T35-02	Accelerator Adaptation to AI library	Implementing support for missing functionalities/layers and data formats.
WP3T35-03	LEDEL (Low Energy EDDL)	EDDL integration on NOEL-V.
WP4T43-03	SW diverse redundancy library	Library allowing to run a task redundantly in two RISC-V cores enforcing some staggering among them to avoid common cause faults.
WP4T43-01	Performance monitoring services	Services to configure the multicore-aware monitoring unit and retrieve information on the multicore interference observed.
WP4T44-03	Safety Analysis	Safety concept by performing a Hazard and Risk Analysis (HARA) within the scope of the concept phase of ISO 26262 (item definition, hazard analysis, risk assessment and functional safety concept), in context of VAL_UC7.
WP4T44-04	Security Assessment	Security assesment by performing a Threat Analysis and Risk Assessment (TARA), covered by the ISO SAE 21434 standard, in context of VAL_UC7.
WP6T61-03-04	Jupyter	Jupyter Notebook is a web-based interactive computing platform.

Figure 66 - Components needed to execute the implementation plan of UC 7

6.6.4 Traceability relationships of Tasks-Components-KPIs

This section links together tasks, components and KPIs. For each Task, the following traceability-relationships are given:

- ***Components***
  - IN Components – Input components needed by the task.
  - OUT Components – Output components produced by the task.
- ***KPIs for UC Implementation Plan***
- ***KPIs for FRACTAL Objectives & Features***

6.6.4.1 UC7_T1 - Environment Setup (NOEL-V node)

The traceability relationships in regard of the Sub Tasks belonging to Task UC7_T1 are listed in this section.

6.6.4.1.1 Sub Task UC7_T1_1 - Setup hardware (NOEL-V)

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
		UC7_KPI_IP_02	Linux on NOEL-V is booting on FPGA	True/False	UC7_KPI_FT_19 UC7_KPI_FT_20	Target RISC-V hardware platform based on NOEL-V availaible Linux operating system running on target platform	True/False True/False

Figure 67 - Sub Task UC7_T1_1 traceability relationship

6.6.4.1.2 Sub Task UC7_T1_2 - Install sample C++ ROS2 node

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
		UC7_KPI_IP_03 UC7_KPI_IP_Req_09	Simple publisher/subscriber example is running on target platform ROS2 stack installed on target platform	True/False True/False	UC7_KPI_FT_19 UC7_KPI_FT_20	Target RISC-V hardware platform based on NOEL-V availaible Linux operating system running on target platform	True/False True/False

Figure 68 - Sub Task UC7_T1_2 traceability relationship

6.6.4.1.3 Sub Task UC7_T1_3 - Connect node to SPIDER via ethernet

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
		UC7_KPI_IP_04 UC7_KPI_IP_Req_08	Max data transfer rate deviation of 10 Hz Edge nodes can exchange data via	1 Hz True/False	UC7_KPI_FT_18	Max data transfer rate with ethernet, deviation of 10 Hz	True/False

Figure 69 - Sub Task UC7_T1_3 traceability relationship

6.6.4.2 UC7_T2 - Function Implementation

The traceability relationships in regard of the Sub Tasks belonging to Task UC7_T2 are listed in this section.

6.6.4.2.1 Sub Task UC7_T2_1 - Development of NN for PTF with Python Keras

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
	UC7_CMP_01	UC7_KPI_IP_06	Simulated robot is following trajectory and avoiding obstacles.	True / False	UC7_KPI_FT_03	Reinforcement learning approach trained model path proxitimy.	< 1m

Figure 70 - Sub Task UC7_T2_1 traceability relationship

6.6.4.2.2 Sub Task UC7_T2_2 - Training of model

Components		KPIs for UC Implementation Plan			ʼIs for Fractal Objectives & Featur		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
	UC7_CMP_01	UC7_KPI_IP_07 UC7_KPI_IP_08	Avg. Path Proxitimy in meter Collision free rate	<1m > 95%	UC7_KPI_FO_03 UC7_KPI_FT_04	PTF AI model path proximity UC7_KPI_FT_04	< 1m < 1m

Figure 71 - Sub Task UC7_T2_2 traceability relationship

6.6.4.2.3 Sub Task UC7_T2_3 - Port NN to ONNX

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
UC7_CMP_01 WP3T35-02		UC7_KPI_IP_09	Valid ONNX model	True / False	UC7_KPI_FT_01 UC7_KPI_FT_14	Target platform supports ONNX. Target platform supports ONNX.	True/False True/False

Figure 72 - Sub Task UC7_T2_3 traceability relationship

6.6.4.2.4 Sub Task UC7_T2_4 - Implementation of PTF with ROS2 and LEDDL on NOEL-V

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
UC7_CMP_01 UC7_CMP_04 WP3T31-02 WP3T35-02 WP3T35-03 WP4T44-03 WP4T44-04	UC7_CMP_02	UC7_KPI_IP_10 UC7_KPI_IP_Req_03 UC7_KPI_IP_Req_04 UC7_KPI_IP_Req_07 UC7_KPI_IP_Req_11 UC7_KPI_IP_Req_13	Unit test coverage of PTF Avg. Path Proximity in meter of the PTF node Collision free rate of the PTF node Update rate of costmap input data to edge nodes LEDEL library build for target platform Hardware accelerator for NN model of UC7 integrated to target platform	> 75% < 1 m > 95 % > 49 Hz True/False True/False	UC7_KPI_FO_01 UC7_KPI_FO_03 UC7_KPI_FT_02 UC7_KPI_FT_05 UC7_KPI_FT_08 UC7_KPI_FT_15 UC7_KPI_FT_17	FRACTAL path tracking node accelerated to perform with a high frequency. PTF AI model path proximity Path tracking function AI model executed at node level. LEDEL library availaible for target platform Switch to emergency state at time exceedance of AI function Lidar sensor messages available at target platform at data rate. Security assesment according ISO SAE 21434 availaible	>= 10 Hz < 1m True/False True/False True/False 10 Hz True/False

Figure 73 - Sub Task UC7_T2_4 traceability relationship

6.6.4.2.5 Sub Task UC7_T2_5 - Porting CAF to NOEL-V

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
UC7_CMP_04 WP3T32-06 WP4T43-03 WP4T43-01 WP4T44-03 WP4T44-04	UC7_CMP_03	UC7_KPI_IP_11 UC7_KPI_IP_12 UC7_KPI_IP_Req_01 UC7_KPI_IP_Req_02 UC7_KPI_IP_Req_07	Unit test coverage of CAF Loop rate of CAF function Processing time of costmap distance SPIDER stops in defined emergency situation Update rate of costmap input data to edge nodes	> 75% >= 10 Hz < 100 ms True/False > 49 Hz	UC7_KPI_FT_07 UC7_KPI_FT_09 UC7_KPI_FT_13 UC7_KPI_FT_15 UC7_KPI_FT_17	Frame rate of collision avoidance function Switch to emergency state at time exceedance of safety relevant function Safety concept according IS 26262 availaible Lidar sensor messages availaible at target platform at data rate. Security assesment according ISO SAE 21434 availaible	>= 10Hz True/False True/False 10 Hz True/False

Figure 74 - Sub Task UC7_T2_5 traceability relationship

6.6.4.2.6 Sub Task UC7_T2_6 - Integration of resource monitoring

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
UC7_CMP_02 UC7_CMP_04 WP3T31-01 WP3T34-01 WP4T43-01		UC7_KPI_IP_13 UC7_KPI_IP_Req_12	Resource monitoring tests in simulation sucessfull Resource monitoring library build for target platform	True/False True/False	UC7_KPI_FT_09 UC7_KPI_FT_12 UC7_KPI_FT_13	Switch to emergency state at time exceedance of safety relevant function Switch to emergency state at fault detected in the communication messages Safety concept according IS 26262 availaible	True/False True/False True/False

Figure 75 - Sub Task UC7_T2_6 traceability relationship

6.6.4.2.7 Sub Task UC7_T2_7 - Integration of Redundancy library

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
UC7_CMP_02 UC7_CMP_04 WP3T34-02 WP3T32-06 WP4T43-03		UC7_KPI_IP_14 UC7_KPI_IP_Req_10	Redundancy library tests in simulation sucessfull Library for diverce redundancy is build on target platform	True/False True/False	UC7_KPI_FT_10 UC7_KPI_FT_11 UC7_KPI_FT_13	Safety relevant processes run redundant on different cores Switch to emergency state at fault detected by diverse redundancy model Safety concept according IS 26262 availaible	True/False True/False True/False

Figure 76 - Sub Task UC7_T2_7 traceability relationship

6.6.4.3 UC7_T3 – System Evaluation

The traceability relationships in regard of the Sub Tasks belonging to Task UC7_T3 are listed in this section.

6.6.4.3.1 Sub Task UC7_T3_1 - Build simulation and test functions

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
UC7_CMP_01 UC7_CMP_02 UC7_CMP_03 WP3T31-01 WP3T34-01 WP3T34-02 WP3T35-03 WP4T43-03 WP4T43-01	UC7_CMP_04	UC7_KPI_IP_17 UC7_KPI_IP_Req_03 UC7_KPI_IP_Req_04	Functions on target platform running with sensor data from 3d simulation Avg. Path Proxitimy in meter of the PTF node Collision free rate of the PTF node	True/False < 1 m > 95 %	UC7_KPI_FO_02 UC7_KPI_FO_03	Tests in simulation for redundant execution and monitoring succeed. PTF AI model path proximity	True/False < 1m

Figure 77 - Sub Task UC7_T3_1 traceability relationship

6.6.4.3.2 Sub Task UC7_T3_2 - Real world tests

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
UC7_CMP_01 UC7_CMP_02 UC7_CMP_03 WP3T31-01 WP3T34-01 WP3T34-02 WP3T31-02 WP3T31-03 WP3T32-06 WP3T35-02 WP3T35-03 WP4T43-03 WP4T43-01	UC7_CMP_05	UC7_KPI_IP_18 UC7_KPI_IP_Req_03 UC7_KPI_IP_Req_04 UC7_KPI_IP_Req_05 UC7_KPI_IP_Req_06	Functions on target platform running with sensor data from real world test Avg. Path Proxitimy in meter of the PTF node Collision free rate of the PTF node SPIDER stops at connection loss to edge nodes SPIDER stops at timeout of edge nodes	True/False < 1 m > 95 % True/False 200 ms	UC7_KPI_FO_03 UC7_KPI_FT_06 UC7_KPI_FT_16	PTF AI model path proximity Sensor data from test drives can be stored on hard drive Path planning node tested in target platform on proving ground	< 1m True/False True/False

Figure 78 - Sub Task UC7_T3_2 traceability relationship

6.6.4.3.3 Sub Task UC7_T3_3 - Evaluation with Jupiter (metrics calculation)

Components		KPIs for UC Implementation Plan			KPIs for Fractal Objectives & Features		
IN Comp	OUT Comp	KPI ID	KPI Description	Value	KPI ID	KPI Description	Value
UC7_CMP_01 UC7_CMP_02 UC7_CMP_03 WP3T31-01 WP3T34-01 WP3T34-02 WP3T31-02 WP3T31-03 WP3T32-06 WP3T35-02 WP3T35-03 WP4T43-04 WP4T43-01 WP6T61-03-04		UC7_KPI_IP_19 UC7_KPI_IP_Req_03 UC7_KPI_IP_Req_04	Metrics calculated with Jupyter availaible Avg. Path Proximity in meter of the PTF node Collision free rate of the PTF node	Proximity, Collision rate, Time consumption, Loop rate < 1 m > 95 %	UC7_KPI_FO_01 UC7_KPI_FO_03 UC7_KPI_FT_06 UC7_KPI_FT_07	FRACTAL path tracking node accelerated to perform with a high frequency. PTF AI model path proximity Sensor data from test drives can be stored on hard drive Frame rate of collision avoidance function	>= 10 Hz < 1m True/False >= 10Hz

Figure 79 - Sub Task UC7_T3_3 traceability relationship

6.7 Justification plan

6.7.1 KPI evaluation method

6.7.1.1 KPI for Implementation Plan

6.7.1.1.1 UC7_KPI_IP_01

- ***Description***: All subtask success.
- ***Result type***: True/False
- ***Evaluation method***: The KPI is True when all subtasks under the task of environment setup succeed.

6.7.1.1.2 UC7_KPI_IP_02

- ***Description***: Linux on NOEL-V is booting on FPGA.
- ***Result type***: True/False
- ***Evaluation method***: The KPI is True, if the target platform is flashed with the SELENE system on chip (SOC) and Linux OS is booted with GRMON and ready for log in.

6.7.1.1.3 UC7_KPI_IP_03

- ***Description***: Simple publisher/subscriber example is running on target platform.
- ***Result type***: True/False
- ***Evaluation method***: ROS2 has to be installed on the target platform. Its correct installation is verified by means of a simple publisher/subscriber example provided in package demo_nodes_cpp [83]. The KPI is True if the C++ nodes implementing the publisher and the subscriber from the abovementioned package can be compiled and are running on the target platform. The communication between these nodes is verified in the terminal window.

6.7.1.1.4 UC7_KPI_IP_04

- ***Description***: Max data transfer rate deviation of 10 Hz
- ***Result type***: 1 Hz
- ***Evaluation method***: The nominal value for the frequency of the data transmission, i.e., the rate with which ROS nodes exchange data, is 10 Hz. The test in regard of this KPI is passed if the frequency with which nodes in the ROS network communicate with a frequency larger than 9 Hz and less than 11 Hz. For the test the quality of service [84] functionality of ROS 2 is used.

6.7.1.1.5 UC7_KPI_IP_05

- ***Description***: All subtask success
- ***Result type***: True/False
- ***Evaluation method***: The KPI is True when all subtasks under the task of function implementation succeed.

6.7.1.1.6 UC7_KPI_IP_06

- ***Description***: Simulated robot is following trajectory and avoiding obstacles.
- ***Result type***: True/False
- ***Evaluation method:*** The KPI is True, if in predefined Gazebo scenarios the robot, controlled by the ML decision-making function, can autonomously follow a path and avoid collisions with obstacles by initiating appropriate evasion maneuvers.

6.7.1.1.7 UC7_KPI_IP_07

- ***Description***: Avg. Path Proximity in meter
- ***Result type***: <1 m
- ***Evaluation method***: A fixed set of Gazebo test scenarios is used to test this KPI. The test in regard of this KPI is passed, if the ML decision-making function creates in each of the test scenarios a trajectory such that the average distance of the robot's position to way points along a path is less than 1 m.

6.7.1.1.8 UC7_KPI_IP_08

- ***Description***: Collision free rate
- ***Result type***: > 95%
- ***Evaluation method***: A fixed set of obstacle-rich Gazebo test scenarios is used to test this KPI. The test in regard of this KPI is passed, if the ratio of the number of controls provided by the ML decision-making function and executed by the robot leading to a collision and the total number of applied controls provided by the ML decision-making function is less than 0.05.

6.7.1.1.9 UC7_KPI_IP_09

- ***Description***: Valid ONNX model
- ***Result type***: True/False
- ***Evaluation method***: The KPI is True, if the conversion of the Keras neural network model into a ONNX neural network model is successful, i.e., on a set of predefined input data both models generate the same output.

6.7.1.1.10 UC7_KPI_IP_10

- ***Description***: Unit test coverage of PTF
- ***Result type***: > 75 %
- ***Evaluation method***: For the evaluation of the unit test coverage the tool GNU gcov [85] is used. The test in regard of this KPI is passed, if the unit test coverage of the code package incorporating the ML path tracking function is larger than 75 %.

6.7.1.1.11 UC7_KPI_IP_11

- ***Description***: Unit test coverage of CAF
- ***Result type***: > 75 %
- ***Evaluation method***: As in the case of UC7_KPI_IP_10 the tool GNU gcov is used to determine the unit test coverage. The test regarding this KPI is passed if the unit test coverage of the code package incorporating the collision avoidance function is larger than 75 %.

6.7.1.1.12 UC7_KPI_IP_12

- ***Description***: Loop rate of CAF function
- ***Result type***: >= 10 Hz
- ***Evaluation method***: With the collision avoidance function operating on maximal frequency and the inputs to the collision avoidance function sent with fixed frequency, the quality-of-service [86] functionality of ROS 2 (on the host system) is used to evaluate the actual loop rate on the target platform. The test in regard of this KPI is passed if the measured loop rate of the collision avoidance is not less than 10 Hz.

6.7.1.1.13 UC7_KPI_IP_13

- ***Description***: Resource monitoring tests in simulation successful
- ***Result type***: True/False
- ***Evaluation method***: The evaluation is based on the monitoring of the impact of an artificial generated computation load. For this purpose, the CAF and the PTF are each executed in one core. In a third core an artificial computational effort is generated. The KPI is True if the impact of this computational effort (in regard of timing inference) can be monitored by means of the FRACTAL monitoring node.

6.7.1.1.14 UC7_KPI_IP_14

- ***Description***: Redundancy library tests in simulation successful
- ***Result type***: True/False
- ***Evaluation method***: According to the implementation of the functional safety requirements the safety relevant functions shall be spawned redundantly. The KPI is True if the monitor of the redundancy library shows that in simulation all the safety relevant functions are spawned redundantly.

6.7.1.1.15 UC7_KPI_IP_15

- ***Description***: All subtask success
- ***Result type***: True/False
- ***Evaluation method***: The KPI is True when all subtasks under the task of system evaluation succeed.

6.7.1.1.16 UC7_KPI_IP_16

- ***Description***: Functions on target platform running with sensor data from 3d simulation
- ***Result type***: True/False
- ***Evaluation method***: By means of the Gazebo simulation of the robot, data (e.g., occupancy grid, odometry) is generated and sent to the target platform vis a ROS 2 bridge. The KPI is True if the tests regarding the CAF and the ML decision-making function, given the data collected in Gazebo, are passed.

6.7.1.1.17 UC7_KPI_IP_17

- ***Description***: Functions on target platform running with sensor data from real world test
- ***Result type***: True/False

- ***Evaluation method***: Data measured by the sensors (mounted on the robot) is sent to the target platform. The KPI is True if the tests regarding the CAF and the ML decision-making function, given data collected by the robot, are passed.

6.7.1.1.18 UC7_KPI_IP_18

- ***Description***: Metrics calculated with Jupyter available
- ***Result type***: "Proximity, Collision rate, Time consumption, Loop rate"
- ***Evaluation method***: The data collected during the execution of the tests corresponding to UC7_KPI_IP_ and UC7_KPI_IP_ is processed using Jupyter notebooks. Proximity to path (see UC7_KPI_IP_07), collision rate (see UC7_KPI_IP_08), average time consumption of the execution of CAF and PTF, and loop rate (see UC7_KPI_IP_12) are computed.

6.7.1.2 KPI for FRACTAL Objectives

6.7.1.2.1 UC7_KPI_FO_01

- **Description**: FRACTAL path tracking node accelerated to perform with a high frequency.
- **Result type**: >= 10 Hz
- **Helps to demonstrate FRACTAL Objective:** O1 - Design and Implement an Open-Safe-Reliable Platform to Build Cognitive Edge Nodes of Variable Complexity
- **Relates to FRACTAL Pillar:** Pillar 1 (WP3) - Open-Safe-Reliable and low power node architecture.
- **Evaluation method**: By offloading the ML path tracking function to the hardware accelerator, a performance increase is expected. The loop time of the execution of the function on CPU is compared with the loop time of the execution of the function on the accelerator. The test in regard of this KPI is passed if the application of the ML path tracking function on the hardware accelerator leads to an increase in performance corresponding to a frequency not less than 10 Hz.

6.7.1.2.2 UC7_KPI_FO_02

- **Description**: Tests in simulation for redundant execution and monitoring succeed.
- **Result type**: True/False
- **Helps to demonstrate FRACTAL Objective:** O2 - Guarantee extra-functional properties (dependability, security, timelines and energy-efficiency) of FRACTAL nodes and systems built using FRACTAL nodes (i.e., FRACTAL systems).
- **Relates to FRACTAL Pillar:** Pillar 2 (WP4) - Low power, safety, security and high-performance trade-off.
- **Evaluation method**: This KPI is tested by means of software fault injection in one of the instances of the redundant threads. The KPI is True if the redundancy monitor successfully reports these fault injections.

6.7.1.2.3 UC7_KPI_FO_03

- **Description**: FRACTAL path tracking nodes AI model generates a collision free path with an acceptable path proximity.
- **Result type**: < 1 m
- **Helps to demonstrate FRACTAL Objective:** O3 - Evaluate and validate the analytics approach by means of AI to help the identification of the largest set of working conditions still preserving safe and secure operational behaviors
- **Relates to FRACTAL Pillar:** Pillar 3 (WP5) - Cognitive & Autonomous Node.
- **Evaluation method**: Provided the path to be followed is free of obstacles, the ML path tracking function shall have a sufficient path tracking accuracy. For this purpose, this path tracking function is applied to a variety of obstacle free Gazebo scenarios. The test in regard of this KPI is passed, if in each of the test scenarios the average distance of the robot's position to way points along a path is less than 1 m (see UC7_KPI_IP_07).

6.7.1.2.4 UC7_KPI_FO_04

- **Description**: Framework for platform independent development and verification of node functions available.
- **Result type**: True/False
- **Helps to demonstrate FRACTAL Objective:** O4 - To integrate FRACTAL communication and remote management features into FRACTAL nodes
- **Relates to FRACTAL Pillar:** Pillar 4 (WP6) - Mutable and FRACTAL communications.
- **Evaluation method**: The development, the execution and as well the performance analysis of ML path tracking function shall be independent of the underlying platform. The KPI is True, if the training and evaluation suite of the machine learning model can be deployed in a docker container and if the training progress and the training results can be evaluated by means of Jupyter notebooks.

6.7.1.3 KPI for FRACTAL Features

6.7.1.3.1 UC7_KPI_FT_01

- **Description**: Target platform supports ONNX.
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F1_VIF - ADAPTABILITY - AI - SW - INFERENCE - MODEL - FORMAT - ONNX
- **Evaluation method**: The KPI is True, if a dummy ONNX model can be loaded and is runnable.

6.7.1.3.2 UC7_KPI_FT_02

- **Description**: Path tracking function AI model executed at node level.
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F2_VIF - ADAPTABILITY - AI - SW - INFERENCE - LOCATION - NODE
- **Evaluation method**: The KPI is True if the ROS node incorporating the ML decision-making function can be executed at the target platform.

6.7.1.3.3 UC7_KPI_FT_03

- **Description**: Reinforcement learning approach trained model path proximity
- **Result type**: < 1 m
- **Helps to demonstrate UC Feature**: F3_VIF - ADAPTABILITY - AI - SW - LEARNING / TRAINING - PARADIGM - REINFORCEMENT LEARNING
- **Evaluation method**: Reinforcement Learning techniques are used to develop a controller (based on a neural network) for the SPIDER enabling the robot to follow a predefined path avoiding collisions with obstacles. This ML decision-making function is evaluated in a variety of Gazebo scenarios. The test in regard of this KPI is passed if the ML decision-making function creates in each of the test scenarios trajectories such that the average distance of the robot's position to way points along the path is less than 1 m.

6.7.1.3.4 UC7_KPI_FT_04

- **Description**: Reinforcement learning approach trained collision avoidance
- **Result type**: >95 %
- **Helps to demonstrate UC Feature**: F4_VIF - ADAPTABILITY - AI - SW - LEARNING / TRAINING - ALGORITHMS – REINFORCEMENT LEARNING
- **Evaluation method**: The decision-making function based on a neural network and trained by means of Reinforcement Learning shall react on static and dynamic obstacles appearing in the surrounding of the robot. The test in regard of this KPI is passed, if in predefined set of obstacle rich Gazebo scenarios, the decision-making function reaches a collision rate which is less than 5 % (see UC7_KPI_IP_08).

6.7.1.3.5 UC7_KPI_FT_05

- **Description**: LEDEL library available for target platform
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F5_VIF - ADAPTABILITY - AI - SW - LIBRARY - LEDEL
- **Evaluation method**: The KPI is True if the neural network model of the decision-making function (incorporating path following and collision avoidance) can be deployed on the target platform and application on test data set gives the same result as the corresponding Keras model on CPU level.

6.7.1.3.6 UC7_KPI_FT_06

- **Description**: Sensor data from test drives can be stored on hard drive
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F6_VIF - ADAPTABILITY - DATA ORCHESTRATION -DATA SET - STORAGE
- **Evaluation method**: Sensor data from test drives can be stored using the rosbag2 [87] package. The KPI is True if the sensor data can be stored as rosbags using the rosbag2 package on the hard drive of the SPIDER.

6.7.1.3.7 UC7_KPI_FT_07

- **Description**: Frame rate of collision avoidance function
- **Result type**: >= 10 Hz

- **Helps to demonstrate UC Feature**: F7_VIF - RELIABILITY - RESPONSE TIME - FRAME RATE
- **Evaluation method**: As in the evaluation of UC7_KPI_IP_12 the quality-of-service functionality of ROS 2 is used to test this KPI. With the collision avoidance function operating on maximal frequency on the target platform and the inputs to the collision avoidance function are sent with fixed frequency, the quality-of-service functionality of ROS 2 (on the host system) is used to evaluate the actual loop rate on the target platform. The test in regard of this KPI is passed if the measured loop rate of the collision avoidance is not less than 10 Hz.

6.7.1.3.8 UC7_KPI_FT_08

- **Description**: Switch to emergency state at time exceedance of AI function
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F8_VIF - SAFETY - MONITORING - AI ENGINES
- **Evaluation method**: The KPI is True, if stalling the execution of the ML decision making function results in a timeout, which triggers the safety mechanism and causes an emergency brake.

6.7.1.3.9 UC7_KPI_FT_09

- **Description**: Switch to emergency state at time exceedance of safety relevant function
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F9_VIF - SAFETY - MONITORING - PERFORMANCE
- **Evaluation method**: The KPI is True if stalling the PTF or CAF (independent of each other) results in a timeout, which, triggered by the safety mechanism, causes an emergency brake.

6.7.1.3.10 UC7_KPI_FT_10

- **Description**: Safety relevant processes run redundant on different cores
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F10_VIF - SAFETY - REDUNDANCY - PROCESSES
- **Evaluation method**: This KPI is True if safety relevant processes (CAF and PTF) can be spawned redundantly on different cores.

6.7.1.3.11 UC7_KPI_FT_11

- **Description**: Switch to emergency state at fault detected by diverse redundancy model
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F11_VIF - SAFETY - REDUNDANCY - DIVERSE REDUNDANCY
- **Evaluation method**: The KPI is True, if stalling of a redundant process is detected by the redundancy monitor, the safety mechanism is triggered, and an emergency brake is initiated.

6.7.1.3.12 UC7_KPI_FT_12

- **Description**: Switch to emergency state at fault detected in the communication messages
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F12_VIF - SAFETY - REDUNDANCY - COMMUNICATION MESSAGES
- **Evaluation method**: The KPI is True, if communication in the middle layer is interrupted by fault injection the safety mechanism is triggered and an emergency brake initiated.

6.7.1.3.13 UC7_KPI_FT_13

- **Description**: Safety concept according to ISO 26262 available
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F13_VIF - SAFETY - REGULATION - ISO 26262
- **Evaluation method**: The KPI is True, if the functional safety concept for the SPIDER according to ISO 26262 is available.

6.7.1.3.14 UC7_KPI_FT_14

- **Description**: Target platform supports ONNX.
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F14_VIF - LOW POWER - AI - LIBRARY - MODELS - ONNX
- **Evaluation method**: The KPI is True, if a dummy ONNX model can be loaded and is runnable.

6.7.1.3.15 UC7_KPI_FT_15

- **Description**: Lidar sensor messages available at target platform at data rate.
- **Result type**: 10 Hz
- **Helps to demonstrate UC Feature**: F15_VIF - CONTEXT AWARENESS - SENSORS - LIDAR
- **Evaluation method**: For the evaluation of this KPI the quality-of-service (QoS) functionality of ROS2 is used. The test in regard of this KPI is passed, if the lidar data arrives with a frequency of 10 Hz on the target platform, provided the data arrives at a frequency of 10 Hz at the host system.

6.7.1.3.16 UC7_KPI_FT_16

- **Description**: Path tracking node tested in target platform on proving ground
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F16_VIF - CONTEXT AWARENESS - ACTIONS - AI TRIGGERED - PATH PLANNING
- **Evaluation method**: The path tracking node based on the ML approach shall be tested in real world scenarios. For this purpose, a fixed set of paths to be followed is introduced. The KPI is True, if the ML based path tracking function deployed on the target platform can be applied in real world scenarios and provides a trajectory satisfying UC7_KPI_IP_07.

6.7.1.3.17 UC7_KPI_FT_17

- **Description**: Security assessment according ISO SAE 21434 available
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F17_VIF - SECURITY - REGULATION - ISO SAE 21434
- **Evaluation method**: The KPI is True if the security assessment according to ISO SAE 21434 for the SPIDER is available.

6.7.1.3.18 UC7_KPI_FT_18

- **Description**: Max data transfer rate with ethernet, deviation of 10 Hz
- **Result type**: 1 Hz
- **Helps to demonstrate UC Feature**: F18_VIF - FRACTALITY - COMMUNICATION / CONNECTIVITY - TECHNOLOGIES - ETHERNET
- **Evaluation method**: The CAF and the PTF shall work with a frequency of 10 Hz. Thus, the input data to these functions has to be provided with the same rate. The test in regard of this KPI is passed, if data between host system and target platform can be exchanged with a rate larger than 9 Hz and less than 11 Hz. For the evaluation of the exchange rate the quality-of-service functionality is used.

6.7.1.3.19 UC7_KPI_FT_19

- **Description**: Target RISC-V hardware platform based on NOEL-V available
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F19_VIF - OTHER: NON-FUNCTIONAL - PLATFORM - NOEL-V RISC-V
- **Evaluation method**: The KPI is True if the target RISC-V platform based on NOEL-V is available and bitfile from SELENE can be flashed.

6.7.1.3.20 UC7_KPI_FT_20

- **Description**: Linux operating system running on target platform
- **Result type**: True/False
- **Helps to demonstrate UC Feature**: F20_VIF - OTHER: NON-FUNCTIONAL - OS - LINUX
- **Evaluation method**: The KPI is True if the Linux OS can be booted and is ready for log in.

6.7.2 Use Case Requirement Validation methods

6.7.2.1.1 UC7_KPI_IP_Req_01

- **Description**: Processing time of costmap distance.
- **Result type**: < 100 ms
- **Evaluation method**: Costmaps are generated due to sensor fusion on the host system and then sent to the target platform, where the costmap is used to compute the distances to obstacles. The test is passed in regard of this KPI if the computation of the distance to obstacles takes less than 100 ms. For the evaluation of the computation time the quality-of-service functionality of ROS 2 is used.

6.7.2.1.2 UC7_KPI_IP_Req_02

- **Description**: SPIDER stops in defined emergency situation.
- **Result type**: True/False
- **Evaluation method**: To evaluate this KPI, the robot is steered towards a wall in the simulation. The KPI is True if the robot, triggered by the safety mechanism, initiates an emergency brake in the previously described situation.

6.7.2.1.3 UC7_KPI_IP_Req_03

- **Description**: Avg. Path Proximity in meter of the PTF node
- **Result type**: < 1 m
- **Evaluation method**: The test in regard of this KPI is True if the ROS node incorporating the neural network based decision-making function passes the test corresponding to UC7_KPI_IP_07.

6.7.2.1.4 UC7_KPI_IP_Req_04

- **Description**: Collision free rate of the PTF node.
- **Result type**: > 95 %
- **Evaluation method**: The test in regard of this KPI is passed if the ROS node incorporating the neural network based decision-making function passes the test corresponding to UC7_KPI_IP_08.

6.7.2.1.5 UC7_KPI_IP_Req_05

- **Description**: SPIDER stops at connection loss to edge nodes.
- **Result type**: True/False
- **Evaluation method**: The KPI is True if, triggered by the safety mechanism, an emergency brake is initiated after the connection to the target platform is interrupted software-wise.

6.7.2.1.6 UC7_KPI_IP_Req_06

- **Description**: SPIDER stops at timeout of edge nodes.
- **Result type**: 200 ms
- **Evaluation method**: For the evaluation of this KPI a power loss is simulated. The KPI is True, if triggered by the safety mechanism, an emergency brake is initiated after the edge node is shut down.

6.7.2.1.7 UC7_KPI_IP_Req_07

- **Description**: Update rate of costmap input data to edge nodes.
- **Result type**: > 9 Hz
- **Evaluation method**: Given the sensor data as input, the costmap shall be computed and provided at a rate larger than 9 Hz. The test in regard of this KPI is passed if for a given test set of sensor data costmaps can be generated at a rate larger than 9 Hz. For the evaluation the quality-of-service functionality of ROS 2 is used.

6.7.2.1.8 UC7_KPI_IP_Req_08

- **Description**: Edge nodes can exchange data via TCP/UDP with SPIDER
- **Result type**: True/False

- **Evaluation method**: This KPI is evaluated using the command line function netstat [88]. The KPI is True if the output of netstat (terminal window) confirms the existence of a TCP or UDP connection between the edge node and the SPIDER.

6.7.2.1.9 UC7_KPI_IP_Req_09

- **Description**: ROS2 stack installed on target platform
- **Result type**: True/False
- **Evaluation method**: The KPI is True if the ROS 2 software stack [89] is installed on the target platform and UC7_KPI_IP_03 is satisfied.

6.7.2.1.10 UC7_KPI_IP_Req_10

- **Description**: Library for diverse redundancy is built on target platform
- **Result type**: True/False
- **Evaluation method**: This KPI is True if the FRACTAL redundancy library is available and is built on the target platform.

6.7.2.1.11 UC7_KPI_IP_Req_11

- **Description**: LEDEL library is built for target platform
- **Result type**: True/False
- **Evaluation method**: The KPI is True if the neural network model of the decision-making function (incorporating path following and collision avoidance) can be deployed on the target platform and application on the test data set gives the same result as the corresponding Keras model on CPU level.

6.7.2.1.12 UC7_KPI_IP_Req_12

- **Description**: Resource monitoring library build for the target platform.
- **Result type**: True/False
- **Evaluation method**: This KPI is True if the FRACTAL resource monitoring library is available and is built on the target platform.

6.7.2.1.13 UC7_KPI_IP_Req_13

- **Description**: Hardware accelerator for NN model of UC7 integrated to target platform
- **Result type**: True/False
- **Evaluation method**: The KPI is True if the hardware accelerator is integrated to the target platform and the neural network of UC7 is deployed on the hardware accelerator.

6.7.3 Components Validation

6.7.3.1 Case Specific Components

6.7.3.1.1 UC7_CMP_01- Path Optimization

Neuronal network for calculating optimal path based on a planned trajectory and obstacles

- UC7_KPI_IP_06 - Simulated robot is following trajectory and avoiding obstacles

- UC7_KPI_IP_07 - Avg. Path Proximity in meter
- UC7_KPI_IP_08 - Collision free rate

6.7.3.1.2 UC7_CMP_02- Path Tracking Function

Control algorithm to follow a planned trajectory and avoid obstacles including UC7_CMP_01

- UC7_KPI_IP_10 - Unit test coverage of PTF
- UC7_KPI_IP_Req_03 - Avg. Path Proximity in meter of the PTF node
- UC7_KPI_IP_Req_04 - Collision free rate of the PTF node
- UC7_KPI_IP_Req_07 - Update rate of costmap input data to edge nodes
- UC7_KPI_IP_Req_11 - LEDEL library build for target platform
- UC7_KPI_IP_Req_13 - Hardware accelerator for NN model of UC7 integrated to target platform

6.7.3.1.3 UC7_CMP_03- Collision Avoidance Function

Safety relevant function for switching to failsafe mode in case of obstacles in danger zone

- UC7_KPI_IP_11 - Unit test coverage of CAF
- UC7_KPI_IP_12 - Loop rate of CAF function
- UC7_KPI_IP_Req_01 – Processing time of costmap distance
- UC7_KPI_IP_Req_02 – SPIDER stops in defined emergency situation
- UC7_KPI_IP_Req_07 - Update rate of costmap input data to edge nodes

6.7.3.1.4 UC7_CMP_04- 3D Simulation

3D simulation based on Gazebo for testing of developed and integrated functions

- UC7_KPI_IP_17 - Functions on target platform running with sensor data from 3d simulation
- UC7_KPI_IP_Req_03 - Avg. Path Proximity in meter of the PTF node
- UC7_KPI_IP_Req_04 - Collision free rate of the PTF node

6.7.3.1.5 UC7_CMP_05- Integrated Demonstration Software on Target

Demonstrating software running on NOEL-V platform integrated to SPIDER

- UC7_KPI_IP_18 - Functions on target platform running with sensor data from real world test
- UC7_KPI_IP_Req_03 - Avg. Path Proximity in meter of the PTF node
- UC7_KPI_IP_Req_04 - Collision free rate of the PTF node
- UC7_KPI_IP_Req_05 - SPIDER stops at connection loss to edge nodes
- UC7_KPI_IP_Req_06 - SPIDER stops at timeout of edge nodes

6.7.3.2 FRACTAL Common Components

6.7.3.2.1 WP3T31-01 - Edge-oriented monitoring unit

AXI-compliant statistics unit to support safety measures and validation in the context of edge systems.

- UC7_CMP_04 - 3D Simulation
- UC7_CMP_05 - Integrated Demonstration Software on Target

6.7.3.2.2 WP3T34-01 - Driver for the edge-oriented monitoring unit

Driver for the statistics unit supporting safety measures and validation in edge systems.

- UC7_CMP_04 - 3D Simulation
- UC7_CMP_05 - Integrated Demonstration Software on Target

6.7.3.2.3 WP3T34-02 - Drivers for the SW diverse redundancy library

Driver to read PMCs (Performance Monitoring Counters) from a remote core, and to issue SIG_STOP and SIG_CONT signals to remote cores.

- UC7_CMP_04 - 3D Simulation
- UC7_CMP_05 - Integrated Demonstration Software on Target

6.7.3.2.4 WP3T31-02 - Interconnect to support Accelerators integration

Interconnect: AXI pulp library Integration.

- UC7_CMP_02 - Path Tracking Function
- UC7_CMP_05 - Integrated Demonstration Software on Target

6.7.3.2.5 WP3T31-03 - Safety and security hardware support

Extensions to the interconnect and other NOEL-V components for Security and Safety.

- UC7_CMP_05 - Integrated Demonstration Software on Target

6.7.3.2.6 WP3T32-06 - Redundant Acceleration Scheme

Integration of a redundant AI inference accelerator in the platform.

- UC7_CMP_03 - Collision Avoidance Function
- UC7_CMP_05 - Integrated Demonstration Software on Target

6.7.3.2.7 WP3T35-02 - Accelerator Adaptation to AI library

Implementing support for missing functionalities/layers and data formats.

- UC7_CMP_02 - Path Tracking Function
- UC7_CMP_05 - Integrated Demonstration Software on Target

6.7.3.2.8 WP3T35-03 - LEDEL (Low Energy EDDL)

EDDL integration on NOEL-V.

- UC7_CMP_02 - Path Tracking Function
- UC7_CMP_04 - 3D Simulation
- UC7_CMP_05 - Integrated Demonstration Software on Target

6.7.3.2.9 WP4T43-03 - SW diverse redundancy library

Library allowing to run a task redundantly in two RISC-V cores enforcing some staggering among them to avoid common cause faults.

- UC7_CMP_04 - 3D Simulation
- UC7_CMP_05 - Integrated Demonstration Software on Target

6.7.3.2.10 WP4T43-01 - Performance monitoring services

Services to configure the multicore-aware monitoring unit and retrieve information on the multicore interference observed.

- UC7_CMP_03 - Collision Avoidance Function
- UC7_CMP_04 - 3D Simulation
- UC7_CMP_05 - Integrated Demonstration Software on Target

6.7.3.2.11 WP4T44-03 - Safety Analysis

Safety concept by performing a Hazard and Risk Analysis (HARA) within the scope of the concept phase of ISO 26262 (item definition, hazard analysis, risk assessment and functional safety concept), in context of UC7.

- UC7_CMP_02 - Path Tracking Function
- UC7_CMP_03 - Collision Avoidance Function

6.7.3.2.12 WP4T44-04 - Security Assessment

Security assessment by performing a Threat Analysis and Risk Assessment (TARA), covered by the ISO SAE 21434 standard, in context of UC7.

- UC7_CMP_02 - Path Tracking Function
- UC7_CMP_03 - Collision Avoidance Function

6.7.3.2.13 WP6T61-02-01 - Docker

Platform-as-a-Service product that uses OS-level virtualization to deliver software in packages called containers.

6.7.3.2.14 WP6T61-03-04 - Jupyter

Jupyter Notebook is a web-based interactive computing platform.

7 VAL-UC8 Improve the performance of autonomous shuttles for moving goods in a warehouse

Shuttle technologies are gaining increasing interest as an automated storage and retrieval system (AS/RS) solution in the intralogistics industry. Customer requirements are variable and require expertise, especially in the application of their variaty of storage goods and physical properties like size and weight. These requirements demand a flexible and stable system for consistent throughput, which a shuttle system can provide.

A look at the shuttle as component shows that it is an automated guided vehicle used in a sheet metal high bay warehouse. The movement is limited in the horizontal to the guide rails in one axis and in the other axis to store and retrieve containers by the two belts of the load handling device, like in Figure 80 shown.

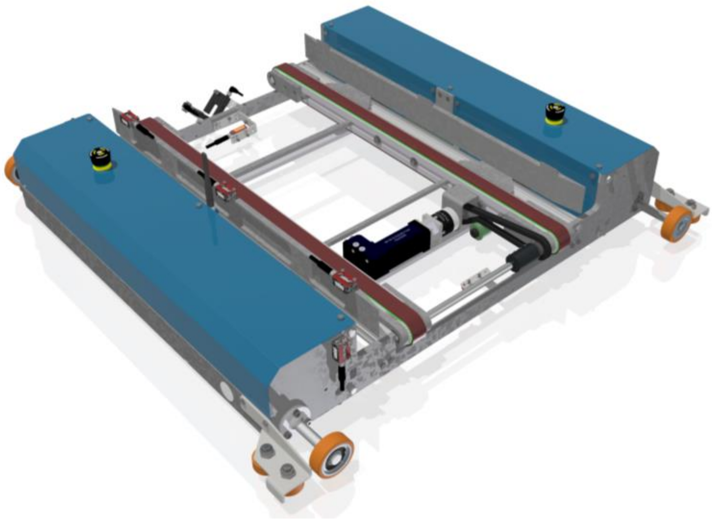


Figure 80 - CAD rendering of a shuttle

In this use case, the test setup, shown in Figure 81 is set up in-house and consists of a small warehouse, two shuttles and two elevators. This warehouse has 7 levels obstructed and a maximum storage capacity of 210 storage places distributed over a length of 7 m.

Figure 81 - 1. CAD rendering of the test setup - side view

On both sides of the warehouse, the lifts are placed in front of the aisle. The shuttles can travel in one axis within this warehouse through guided rails in a level and can switch the current level by the lifts, so the system offers the possibility for the shuttles to move in two axes. To move containers into or out of the system, conveyor sections will be placed near the hoist frame of the lifts. This moving process of picking up or depositing a container is accomplished directly from the shuttles to the conveyor sections and has no buffering capability planned.

From the top view of the test setup, shown in Figure 82, the material flow between inbound and outbound is presented by the yellow arrows at the associated conveyor sections left from both lifts. The path between the two conveyor sections is connected by a technically slimmed down version of a shuttle and consists of a workplace to pick up or store items in the containers. For ease of use, the workplace is equipped with a workstation, where orders for the system can be sent.

The commonly used industrial grade single board computers in UC8 will be replaced in the equipment by the FRACTAL nodes. The previous platform is based on a Windows CE/IPC and will be ported to the new nodes. Due to the gain in computing power, services will be offloaded on the FRACTAL edges. The use of the FRACTAL project is intended to improve adaptability and reliability regarding functional safety, usability, and especially performance. Nodes with AI resources form the basis for later developments in the field of swarm intelligence as part of the exploitation of project results.

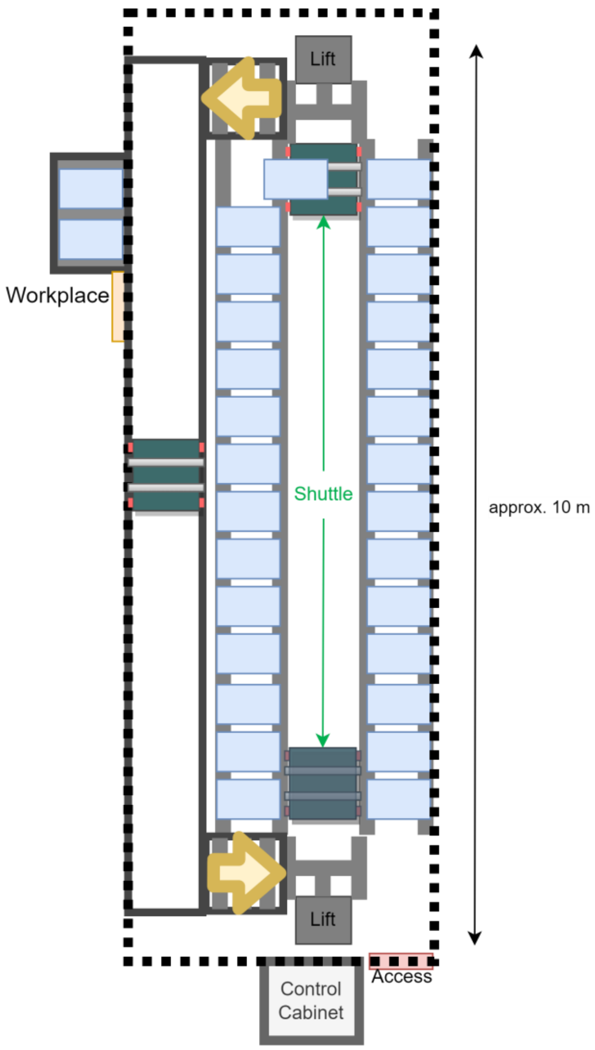


Figure 82 - Top view of the test setup for UC8

7.1 Before FRACTAL

This section examines the configuration of the underlying shuttle technology and explains the behavior about control and communication. Firstly, the system architecture will be explained and then the base concepts for the message flow,

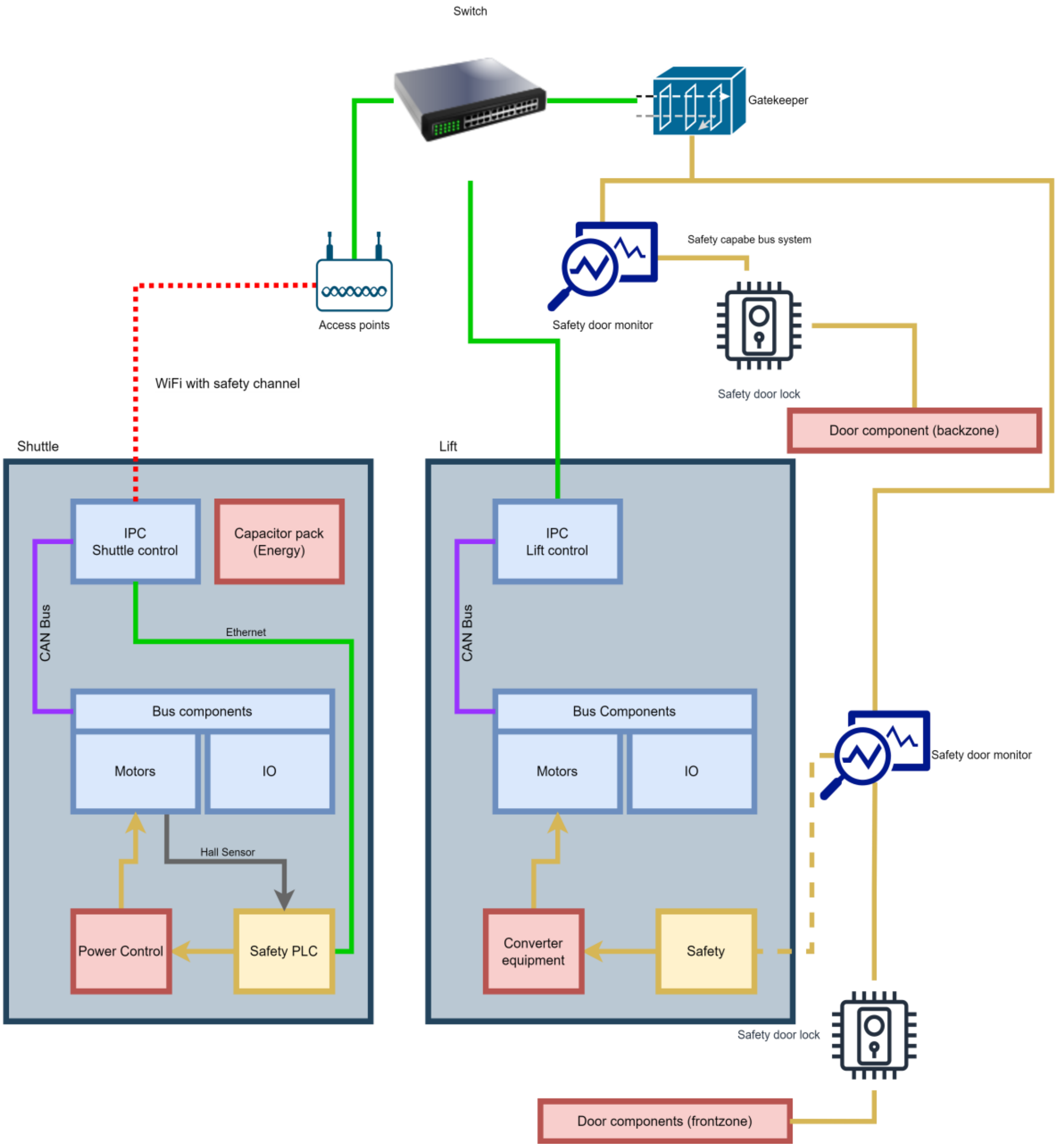


Figure 83 - State of the art - shuttle technology from the field level perspective

functional safety with maintenance levels, and the electrical parts of the two core components, i.e., the shuttle and the lift.

The system architecture of the field level is shown in Figure 83, where the components are listed and the connections between them are pointed out. Almost every component of this architecture can appear multiple times and it is described in singular for an explanation.

From top to down, the network structure begins with a network switch, which is isolated from the internet, and remote access is only provided by a VPN for the material handling equipment manufacturer. Warehouse-related orders are shared over the local network and consist of tasks for shuttle and lift. Wireless communication is commonly realized by Wi-Fi in the 5 GHz frequency band; other components are connected via ethernet or proprietary bus systems. The typical telegram flow starts with the material flow controller, which gets an order for a specific container with the instruction for storage or retrieval. The controller generates jobs of that order for the system and distributes them in the local network.

7.1.1 Hardware

A look at the shuttle block diagram reveals the typical electrical components listed in Table 6 with a brief description.

<u>Capacitor pack</u> Energy source for the shuttle. Image exemplary.	
<u>IPC with enclosure</u> Single board computer with Wi-Fi and CAN interface for the shuttle or lift control service. Originally used with *Windows Embedded CE* operating system.	
<u>Motors</u> Brushless DC motors with controller and CAN interface for shuttle and load handling device motion.	
<u>Safety PLC</u> The failsafe programmable logic controller monitors the status of the shuttle and misbehavior leads to a power supply shutdown for the motors. Image exemplary.	

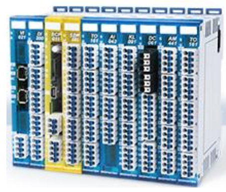
<u>Power control</u> Power contactors are used to control the power supply of the rotary equipment by the safety plc.	
<u>IO System Node</u> Modular IO system with CAN interface for flexible extension of inputs and outputs. Image exemplary.	
<u>Sensors</u> Commonly light barriers and inductive probes are built into the shuttles. Positioning in terms of any motion tasks requires sensors.	 

Table 6 - List of electrical components of a shuttle

Lift components are similar to the shuttle components and only differ in some points, which are listed in Table 7 with a brief description.

<u>Motor</u> The motor moves the hoist frame in vertical direction and is used to change the shuttle level position.	
<u>Safety</u> Safety relay for the safe operation of the motor. The main task is the initiation into the controlled stop of the machine in case of an emergency stop. Image exemplary.	

Table 7 - Lift of electrical components of a lift

7.2 Use Case Objectives

FRACTAL technology will be implemented in a warehouse to develop intelligent shuttles based on cognitive computing for swarm intelligence, improving availability, throughput, and reliability.

Following long-term goals are expected from the project:

- Adaptivity:
  The shuttle system should adapt autonomously to new situations within the warehouse.
- Energy optimization and improved strategy for warehouse locations:
  By optimizing the location of high-speed goods and their distribution; jams shall be avoided and the efficiency of retrieving goods improved.
- Route optimization:
  Aggregated data on route patterns and delivery efficiency will be used by the AI application to achieve higher throughput for the warehouse.
- Increase pickup order productivity:
  Use of optimized strategies for system-driven picking based on the accumulated picking list.
- Defined bulk order fulfillment:
  Mass dispatch information, including the expected schedule is passed to the swarm. The swarm resolves the solutions to be delivered as specified.

The goal is to improve the warehouse throughput, as delays in warehouse operations are undesirable and potentially critical, because they have a domino effect on the whole supply chain. The handling, storage and retrieval of warehouse goods by automated shuttles will be optimized using artificial intelligence techniques. AI will organize and analyze the generated data sets in an optimal way to improve warehouse throughput.

The automated shuttle systems will operate as agents of an intelligent swarm system to improve its reliability. Real-time information (e.g., diagnosis, battery status, task) hosted on the shuttle operations will be registered and flow into the evaluation. The FRACTAL node will thereby meet the computational requirements at high energy efficiency.

The shuttles will be based on edge nodes that process real-time information at high speed via integrated filters. Task processing will be shifted from the material flow controller to the edge nodes with local decision-making capabilities (e.g., routing and sequencing), and the system will minimize human interruptions due to errors.

The warehouse system is expected to use new data flows to optimize warehouse throughput. The following benefits are expected from swarm functions:

1. Autonomy and adaptivity:
   The shuttles in the swarm are expected to cooperate autonomously to achieve a common goal of providing high and reliable warehouse throughput

for the transported goods. The FRACTAL-based shuttle adapts to pending orders and failures so that tasks are completed even in the event of resource failure (e.g., shuttle failure, elevator failure, track failure).

2. Obstacle avoidance and removal:
   The swarm adapts by avoiding obstacles, selecting alternative paths, and contributing to the removal of obstacles when possible.
3. Improved availability:
   Path planning algorithms based on swarm intelligence are introduced to ensure fast delivery of goods in the overall warehouse in case of faulty and degraded resources (e.g. faulty lane, faulty lift, faulty shuttle).
4. Safety:
   The shuttles in the swarm will cooperate to support safety-critical scenarios, e.g. avoiding collisions between shuttles and human engineers during online maintenance, which is also highly desirable.

By endowing the shuttle system with swarm intelligence, the system is not limited to fixed routes and the resource constraints of the central computing servers. The shuttles interact autonomously with each other and with infrastructures such as elevators and conveyors. The swarm capabilities enable the overall system to adapt its capacity to fluctuations between storage and transport processes.

The need for the FRACTAL node is justified by the high computational requirements of the AI-based swarm intelligence algorithms. These tasks consume resources, especially energy and memory. The proposed FRACTAL nodes provide a secure and energy-efficient solution with the ability to host cognitive functions, making them a suitable candidate for use. The following inputs from the project are relevant or serve to demonstrate this Use Case:

- Specification and Methodology.
- Cognitive computing capabilities of the FRACTAL node (for swarm intelligence).
- Techniques for functional safety, energy efficiency, reliability, and real-time capability.

7.3 State of the art

According to the state of the art, the system is first described from the point of view of warehouse logistics and a clear boundary to the customer-oriented solution from the material handling equipment manufacturer is shown. Common practices in consideration of electronics platforms, as well as communication and functional safety are described here.

7.3.1 System architecture

For AS/RS solutions based on shuttle technology, the system architecture for every project is nearly identical and can be illustrated in a pyramid (Figure 84), whereby

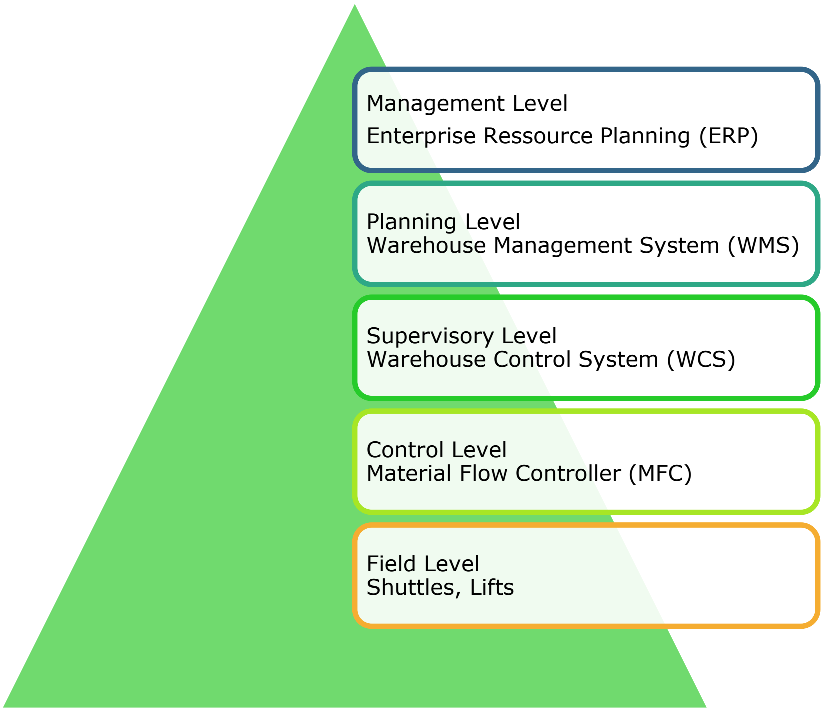


Figure 84 - Shuttle system adapted from automation pyramid

the dependencies in terms of complexity and the degree of automation define, which of these levels were used and how.

Depending on the system size and customer requirements the WMS layer is part of the ERP or works in big size wholesale and distribution businesses with extended functionalities as a standalone system. In general, levels 2-4 in real software solutions are blurred from an external point of view and do not have a hard separation.

The WMS layer is used to manage within a warehouse and is a part of whole supply chain management. The supervisory level is only used in medium-complex distribution centers with a high degree of automation when the basic routing of the goods up to the conveyor systems can no longer be handled by the control level (MFC). In the last level the hardware components are located, which can be divided in two major groups, the shuttles, and the lifts. Both groups include a variety of configurations depending on customer requirements, especially throughput as the major decision factor for the best configuration.

The shuttles used in today's high-bay warehouses can be divided into three categories: one-dimensional (1-D), two-dimensional (2-D) and three-dimensional (3-D) shuttles. The 1-D shuttles travel in a single aisle in the warehouse. In a warehouse, a given aisle often contains multiple levels. A shuttle that travels only in a particular aisle on a fixed level is called level captive. In 2D, an elevator system is used to switch between levels. This causes the shuttle system to move in both horizontal and vertical directions. The 3D system allows the shuttle to travel through multiple lanes during operation and change levels simultaneously. The 3D system refers to the entire warehouse.

Analogous to the shuttle categories, the tray depth is typically in three grades applicable. Single, doubleand triple deep storages are offered in the market and are decisive for the load handling device in the shuttle and lift.

As a further adjusting screw, the number of storages and hoist frames can vary regarding the lift, as e.g. in another setup the shuttles stay level captive and only container lifts are used. The number of storages per hoist frame can be single and double deep, just as the hoist frame itself can be single or twice per lift.

In the case of the FRACTAL project, the most probable solution was chosen, so that single-deep storage with 2-D shuttles and relatively low throughput serves as a reference.

7.3.2 Electronic platforms in shuttles

The requirements for the electronics platform of a shuttle must provide the necessary interfaces to input/output devices, e.g., to control the motor, read signals, and for user interfaces. Real-time capability must be supported in order to control the shuttle in real-time and to ensure functional safety (e.g., no collisions between shuttles). In addition, wireless interfaces are required to realize communication with the material flow controller as well as with the user interface. Other relevant non-functional properties include availability, reliability, and energy efficiency.

Existing platforms use various single-board computers usually with ARM processors and embedded operating systems such as Windows CE as well as bus systems to interact with input/output devices. There also exists solutions in the market with microprocessors that directly control the inputs/ outputs. Various communication protocols such as CAN, Ethernet, Wi-Fi and SRD-based protocols (868-MHz) are used in shuttles.

Today, order packages for shuttles are mainly planned and managed centrally. Central control is handled by the warehouse management system, which coordinates orders to the shuttles via the material flow controller.

7.3.3 Communication

Current systems use various wireless communication technologies in warehouse applications such as Wi-Fi (2.4 or 5 GHz) or other IEEE 802.15.4 based technologies in the ISM radio band (2.4 GHz or 868 MHz for Europe). In order to ensure sufficient coverage in the storage rack, access points with partially directional antennas are usually distributed in a high density. Wireless solutions, especially Wi-Fi, have several disadvantages such as high installation costs, maintenance costs and error-proneness, with consequences of telegram loss during operation. In many cases, the same frequency bands/channels are shared between the shuttles and other operating departments of the customer. This raises security concerns and makes it difficult to diagnose faults in the network and requires coordination with the customer regarding channel overlaps. Roaming between access points in turn means, that continuous roaming behavior is associated with a high risk of communication disruptions. Shuttles are in motion and continuously generate different network topologies. Strict latency requirements are necessary for safety-critical functions, and the shuttles operate in a demanding environment subject to vibration, dust, signal interference from motors and stored goods, and so on. Assurance of fault tolerance as well as determinism in terms of latency and jitter is a mandatory requirement for functional safety and ensuring high plant availability. The wireless communication network of a warehouse shuttle system requires consideration of dynamic topologies of mobile nodes, signal interference, path loss, limited energy, and changing positions of shuttles.

In the warehouse environment, network topology changes and sensor node failures can cause fluctuations in network connectivity. In addition, sensors are often exposed to interference, high humidity, vibration, dirt, and dust. These harsh environmental conditions also cause variations in network connectivity with variable link capacity. Interference detection techniques have been developed previously to counteract harsh industrial environments.

7.3.4 Functional safety

Another challenge today is limited availability due to various types of errors. These include wedged stored goods, communication breakdowns, defective hardware components in shuttles and contamination. Thus, access to the system must be created and certain parts of the warehouse must be blocked. Today, fault diagnosis is carried out via sequence control and manual maintenance by an operator.

For manual maintenance, the system access concept is described for better understanding by means of Figure 83. On the right side of the network switch is the gatekeeper, which supervises all functional safety-related actions, this includes the safety door monitors and parts below in that structure. The safety door monitors, which can manage a certain number of door locks, communicate with the gatekeeper when an access is requested. Door monitors are placed in the front and in the back zone. An access depends on the location and is provided from both sides of the rack. The front zone is defined by the position of the lift and differs only in connection to the safety relay of the lift block. In the back zone is typically the direct access to the shuttle area. That concept allows battery-powered AGVs to exclude from a certain block in the system for maintenance purposes, while the rest of the system stays operational, as one block per aisle covers up to 7 shuttle levels and is called maintenance level.

To gain access, users have to make a request at the desired maintenance level. In the case of the back zone, this request is used to check the location of the shuttles near or in the maintenance area from the gatekeeper. These shuttles get a message on the safety level, to leave the area. As the system utilize a time delay for the access, the shuttles have a defined time to finish a task and then leave the area. When the time has passed and the affected shuttles are still in the area, the safety PLC in the shuttle triggers the power control and interrupts the supply of the motors by the message of the gatekeeper. The shuttle state is submitted back to the gatekeeper. After this process, the monitor gets a message from the gatekeeper, releases the door lock and the user gains the requested access.

Accessing from the front zone depends on the configuration of the passageway to the shuttle area and turn off in a controlled manner the whole aisle or only the affected lift.

7.4 Main contributions expected from FRACTAL

Two advanced functions have emerged from the FRACTAL components, which will be explained in the following sub chapters. Regarding the FRACTAL Pillars, the implementation finds its approach from the supervisory level down to the field level in form of the targeted swarm intelligence and the extension of the safety concept. The connection between the pillars and the functions will be justified, after the introduction of the hardware implementation.

In the test setup, FRACTAL edge nodes will be implemented in the control cabinet for the lifts, but also in each shuttle, like shown in Figure 85. The size of the Versal-board was unsuitable for the shuttle, therefore there was a deviation in the board selection and the choice was made to use the Kria KV260 boards instead of Versal. These FPGAs use a Zynq® UltraScale+™ MPSoC with enough power for the shuttle control services and for additional computational capabilities for the FRACTAL components. For communication between the edge nodes, a time-triggered network will be used, which is set up on Wi-Fi technology. To utilize low power services from the FRACTAL, the time-triggered network on chip will be implemented in both boards

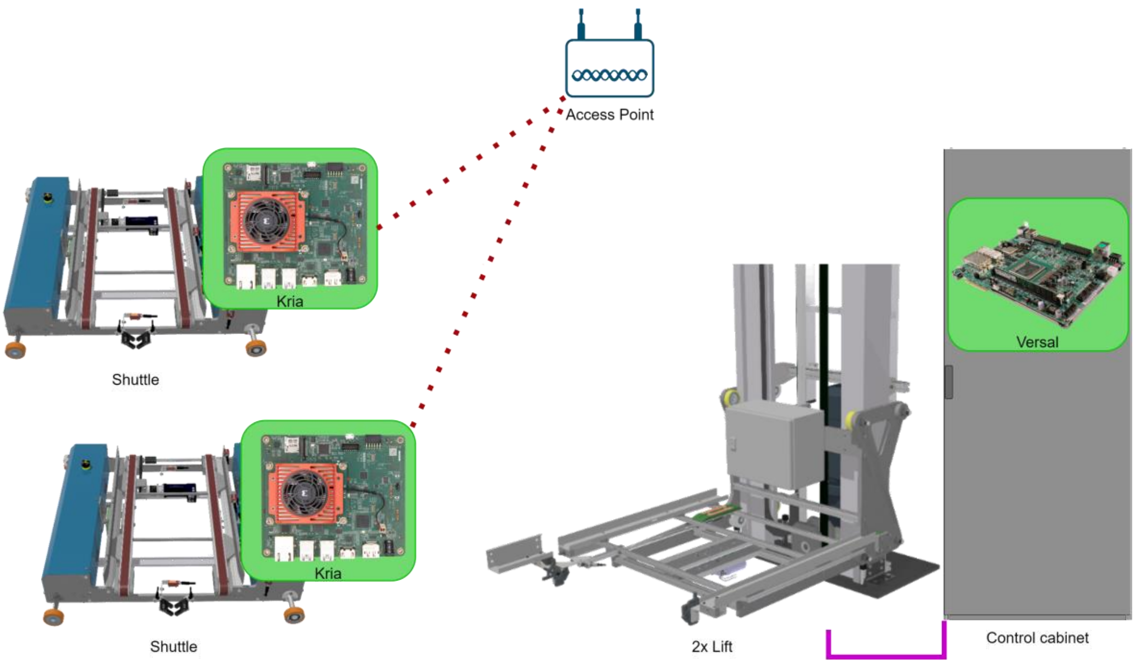


Figure 85 - FRACTAL edge nodes installation locations

with predefined scenarios and access to dynamic voltage frequency scaling and the option to disable unused hardware blocks.

The shuttle edge nodes will be extended with cameras to become cognitive and autonomous nodes and will be implemented in the functional safety extension under consideration of the applied harmonized standard DIN EN ISO 3691-4:2020-11.

Regarding the hardware setup, both boards provide an open, safe, and reliable node architecture with low power capabilities and still high performance. The real-time capable time-triggered on- and off-chip communication covers a lot of the desired use case objectives and offers a lot of possibilities, especially in new approaches, like the first steps in decentralization of warehouse applications.

7.4.1 Swarm intelligence

Shuttles with AI resources and FPGAs that would support the stated goals in terms of swarm intelligence with neural networks in an energy-efficient, real-time environment are not known yet. Existing solutions for swarm intelligence concepts (e.g., Knapp [90]) are only limited to route planning.

Tasks/ jobs are generated centralized in the control level for all field components. Future requirements like flexibility, scalability and robustness could be improved by decentralization and customization of the state-of-the-art topology. The AI supported swarm intelligence concentrates tasks from the supervisory and control level in the field level. Offloading computationally intensive tasks like path planning of the MFC in the edge nodes and the segmentation in swarm blocks shall improve the performance of the system. A synergy effect is created in terms of scalability of such systems, as the coordination of optimized paths and storage strategies will be calculated in a lower level and distributed for each swarm.

In concrete terms, this approach means to gain more capabilities regarding adaptivity and reliability by applying WP4's meta scheduler from the node level to the application level. In UC8, this scheduler will be implemented in the Versal board, to deal with different scenarios as a solution model with hybrid approach, where exactly this separation and offloading is described in [91]. Due to the high amount of possible scenarios, during the FRACTAL project there will be a limitation of three scenarios with the greatest added value or the highest probability, as listed below.

- Weight distribution – weight flow optimization
  To optimize the energy consumption, it can be helpful to store containers with the max. specified weight near the outbound to shorten the transportation path of the shuttles or distribute these containers in the lower level near the conveyor section heights, to reduce the lift movements. The weight flow of containers over the rack shall be minimized and the capacitor pack charging time kept low as possible. Especially in long systems the capacitor pack as energy source discharges to its lower design limits.
- Priority flag sorting
  Sorting the containers by the deposited priority flags when entering the system could gain the throughput. E.g., Containers get a flag with numbers from 1 to 3, the higher the value of the flag, the higher the priority. So, the highest value will be stored near the outbound lane of the rack, to get the fastest pick of recurrent items ordered from the operators in the shortest physical process time. This kind of sorting containers can also be applied to the inbound.
- Obstacle avoidance
  By detecting an obstacle, the orchestrator reschedules the tasks of the swarm and keeps that path blocked until the error is corrected. This scenario has a high probability and occurs often in already implemented systems. Typically,

a shuttle has a misbehavior and could not pick a container out of the rack, or the container was stored misaligned and now blocks the path, just like another shuttle, which already turned off for energy saving purposes. Additionally, maintenance staff could enter the system and shall be identified as an obstacle. Verification will be included in the communication between the shuttle edge nodes and the lift node.

The task allocation is based on the meta-heuristic population method "Ant Colony Optimization" and shall find the optimal path to the desired destination for each shuttle in a short inference time. In a failure scenario the best possible solution will be computed every time for every shuttle. This behavior is required due to overlapping shuttle time frames for each location in the warehouse, if only one shuttle would get the correction.

By applying the meta scheduler in different levels of the edge node, the potential fractality of UC8 is demonstrated.

7.4.2 Functional safety extension

In the functional safety extension, an approach is chosen that allows under the requirements of ISO 61508 the implementation of safety-related building blocks in the edge nodes, as envisaged by the FRACTAL project. Compared to the old concept, it will be possible to access to the rack without safety door locks, nor physical barrier ahead of each maintenance level will be installed.

The time-triggered network-on-chip (TTNoC) for the edge nodes provides the ability for time-triggered off-chip communication. Functional safety relevant communication between the edge nodes utilizes this function and exchanges telegrams of the current status from each edge node. The degradation of single edge nodes will be used, to decrease specific functions in operational mode on application level, as well for on-chip operations in the node level. Restrictions in function or limitation of velocity or acceleration are possible solutions and shall extend overall availability in the warehouse.

On the other side, the shuttles will be extended with camera systems to use person detection, more precisely human body detection on the edge node. To accomplish this approach, it is crucial to implement a fail-safe evaluation algorithm and connection this to the existing failsafe PLC. In the case of the detection model, neural networks in an embedded system with restricted energy sources will be implemented and evaluated. This evaluation algorithm is used to calculate the distance between the detected object and the shuttle in two zones, the danger, and the warning zone. The behavior in case of detection will be predefined. If the object is in the warning zone, the target velocity will be degraded to 0.3 m/s as specified in the standard DIN EN ISO 3691-4 and monitored by the built-in failsafe-PLC. By entering in the danger zone, the power supply of the motors from the affected shuttle will be turned off. To avoid entering the danger zone, the shuttle edge nodes with the detected object send a request to reschedule the tasks. When the request is rejected, the shuttle edge node will go in a kind of waiting mode to save energy, until the detected object disappears again and will acknowledge the orchestrator about the change of status to schedule new tasks for the swarm.

A more detailed description of this subsection can be found in D4.4, where the risk assessment according to the standard DIN EN ISO 3691-4:2020-11 was done for implementation purposes. The applied standard is used for safety requirements and verification of driverless industrial trucks since the shuttle is treated in Germany under this standard. Regarding the pillars, a safe and reliable node with cognitive capabilities is required to accomplish this extension and will be provided in the FRACTAL project. The communication between the edge nodes is mutable through its underlying hardware architecture and complies with the pillar of FRACTAL communication by time-triggered components.

7.4.3 Cloud services

The implementation of a service orchestrator in the cloud enables the benefits of a fleet management system. As updates of control services were done in the past manually, it would be possible to manage equipment control services and AI models as well from the cloud with FRACTAL components. Especially AI models require version controlling and fast reaction times to prevent or fix misbehavior in customer systems. Individual system properties from the customer specifications must be assigned to the respective project, as this information is required for the control services and the swarm intelligence to make physically feasible tasks. The initiation of an update could be realized via VPN and would reduce downtimes significantly, as well as the supporting time of employees.

Implementation in the test setup will be realized by a locally installed server with ethernet access to the FRACTAL network. After the verification of AI models and control services, the migration of data to the cloud will start. Summarized, the management of services, models, and data sets shall be accomplished in the cloud services as projects will reflect single or multiple swarms per customer. In that configuration, single core components will be mutable provided in a high flexible environment with small management effort.

7.5 Evaluation of the implementation results

This section defines the KPIs defined for UC8 implementation. These KPIs are classified into three groups:

- KPIs for Implementation Plan Task
- KPIs for FRACTAL Objectives related to FRACTAL Pillars
- KPIs for UC Features

For each KPI, an *Identifier*, a *Description* and the type of result *Value* is defined. The *Test* to be performed for the KPI will be defined later in the Justification Plan, therefore is marked as TBD.

The next subsections describe in some detail the three groups of KPIs.

7.5.1 KPI for Implementation Plan Tasks

This section defines the ***KPIs defined for the Implementation Tasks***. Figure 86 shows the complete list of KPIs defined for the Implementation Tasks of UC8.

The KPIs are divided into two subgroups, which are derived from the implementation plan and the requirements:

KPI UC for Implementation Plan				
KPI ID	**Description**	**Value**	**Test**	**Comment**
UC8_KPI_IP_01	All subtask success - Versal node	True/False	TBD	*Defined for Implementation Tasks*
UC8_KPI_IP_02	Duty cycle of control software (Versal) with target design	< 20 ms	TBD	*duty cycle*
UC8_KPI_IP_03	Build AA - shuttle orchestrator for target (Versal - ARM)	True/ False	TBD	*Defined for Implementation Tasks*
UC8_KPI_IP_04	Inference time of predictions - shuttle orchestrator (Versal - ARM)	< 2 s	TBD	*avg. job execution time*
UC8_KPI_IP_05	All subtask success - Kria node	True/False	TBD	*Defined for Implementation Tasks*
UC8_KPI_IP_06	Duty cycle of control software (Kria) with target design	< 20 ms	TBD	*duty cycle*
UC8_KPI_IP_07	Build OpenCV for target (Kria - ARM) success	True/ False	TBD	*Defined for Implementation Tasks*
UC8_KPI_IP_08	Build demonstration software on target success	True/ False	TBD	*Defined for Implementation Tasks*
UC8_KPI_IP_09	Build object detection application success	True/ False	TBD	*Defined for Implementation Tasks*
UC8_KPI_IP_10	Model accuracy of the object detection	> 95 %	TBD	*Defined for Implementation Tasks*
UC8_KPI_IP_11	Build zone evaluation logic application success	True/ False	TBD	*Defined for Implementation Tasks*
UC8_KPI_IP_12	Inference time of object detection	< 100 ms	TBD	*Defined for Implementation Tasks*
UC8_KPI_IP_13	Failure rate of connection between FPGA and safety plc	%	TBD	*Defined for Implementation Tasks*
UC8_KPI_IP_14	Setup cloud service orchestrator success	True/ False	TBD	*Defined for Implementation Tasks*
UC8_KPI_IP_Req_01	The edge node should have following hardware specification: - at least 2 cores @ 800 MHz - at least 4 GB RAM - at least eMMC Memory or similar.	True/False	TBD	*Comming from UC Requirements*
UC8_KPI_IP_Req_02	These communication protocols shall be used from Linux OS: - MQTT over WiFi mesh network for communication between nodes - CAN Bus for internal communication.	True/False	TBD	*Comming from UC Requirements*
UC8_KPI_IP_Req_03	The edge node shall provide enough interfaces for two cameras.	True/False	TBD	*Comming from UC Requirements*
UC8_KPI_IP_Req_04	The edge node shall be capable to detect objects (human body and other obstacles) from video input stream of the provided cameras and	True/False	TBD	*Comming from UC Requirements*
UC8_KPI_IP_Req_05	The edge node shall be able to use an adaptive orchestrator (scheduler) for storing strategies and optimized pathfinding for each shuttle depending on material (weight, type), frequency of requests, division of	True/False	TBD	*Comming from UC Requirements*
UC8_KPI_IP_Req_06	The edge node shall offer optimized pathfinding: Improving path of the shuttles, for different scenarios; obstacle in same layer; malfunction of	True/False	TBD	*Comming from UC Requirements*
UC8_KPI_IP_Req_07	The node shall feature Linux operating system with real time capability	True/False	TBD	*Comming from UC Requirements*
UC8_KPI_IP_Req_08	Safety wireless communication should be over a black channel (ASIL 3,	True/False	TBD	*Comming from UC Requirements*
UC8_KPI_IP_Req_09	For the edge nodes a cross compiler shall be available to port control	True/False	TBD	*Comming from UC Requirements*
UC8_KPI_IP_Req_10	The edge node shall support libraries, like Tensorflow/ Keras.	True/False	TBD	*Comming from UC Requirements*

Figure 86 - KPIs for UC8 Implementation Plan Tasks

- ***KPIs specifically defined for each Task*** – These KPIs have been defined to check the success of the task. *When possible, they are defined as a numerical criterion* (i.e., inference time < 100 ms), otherwise they are defined as a True/False indicating that the task finished successfully.

- ***KPIs related to Tasks, allowing checking the Requirements* defined by the UC *in the general Excel defined in WP1 FRACTAL - Requirements_KPIs_Components.xlsx*** (see Tab Requirements) - These KPIs have been defined taking into accounts the general requirements posted by the Use Case. These KPIs are defined as a True/False value indicating that *the task finished* and allows checking whether *the requirements is met.*

7.5.2 KPI for FRACTAL Objectives related to FRACTAL Pillars

KPIs defined to measure ***how the Implementation Tasks contribute to demonstrate the FRACTAL Objectives*** (Related to Pillars and found in the FRACTAL proposal, Section 1.1.2.). Figure 87 shows the complete list of KPIs defined for this purpose.

KPI for Fractal Objective (an related Pillar)				Helps to demonstrate the following Fractal Specific Objective		
KPI ID	**Description**	**Value**	**Test**	**ID Ob**	**Description**	**Relates to Pillar**
UC8_KPI_FO_00	Fractal technology helps to improve the state of the art in the intralogistics industry	True/False	TBD			
UC8_KPI_FO_01	Cycle time of services on edge node with accelerated orchestrator implemented and running. (VERSAL)	< 20 ms	TBD	O1	Design and Implement an Open-Safe-Reliable Platform to Build Cognitive Edge Nodes of Variable Complexity	Pillar 1 (WP3) - Open-Safe-Realiable and low power node architecture.
UC8_KPI_FO_02	Cycle time of services on edge node with accurate cognitive AI application implemented and running. (KRIA)	< 20 ms	TBD	O1	Design and Implement an Open-Safe-Reliable Platform to Build Cognitive Edge Nodes of Variable Complexity	Pillar 1 (WP3) - Open-Safe-Realiable and low power node architecture.
UC8_KPI_FO_03	Self-sufficient decisions for each shuttle in respect to functional safety and additional degradation steps. High accuracy in detection is required.	> 95 %	TBD	O2	Guarantee extra-functional properties (dependability, security, timeliness and energy-efficiency) of FRACTAL nodes and systems built using FRACTAL nodes (i.e., FRACTAL systems).	Pillar 2 (WP4) - Low power, safety, security and high-preformance trade-off.
UC8_KPI_FO_04	Real-time inference for meta scheduler, which can react on various pre-defined events and make safe decisions for pathfinding and storage strategies for different goods.	< 2 s	TBD	O3	Evaluate and validate the analytics approach by means of AI to help the identification of the largest set of working conditions still preserving safe and secure operational behaviors	Pillar 3 (WP5) - Cognitive & Autonomous Node.
UC8_KPI_FO_05	Real-time inference for object detection on edge node with all services and accelerators implemented.	10 fps	TBD	O3	Evaluate and validate the analytics approach by means of AI to help the identification of the largest set of working conditions still preserving safe and secure operational behaviors	Pillar 3 (WP5) - Cognitive & Autonomous Node.
UC8_KPI_FO_06	Safe wireless communication between nodes.	% telegram losses	TBD	O4	To integrate fractal communication and remote management features into FRACTAL nodes	Pillar 4 (WP6) - Mutable and fractal communications.

Figure 87 - KPIs for UC8 Implementation Plan to measure the contribution to FRACTAL Objectives

7.5.3 KPI for UC Features

KPIs defined to measure ***how the Implementation Tasks contribute to demonstrate the UC Features*** (defined in the Tab *FRACTAL Features* in the general Excel defined in *WP1 FRACTAL - Requirements_KPIs_Components.xlsx*). Figure 88 shows the complete list of KPIs defined for this purpose.

KPI for UC Feature				Helps to demonstrate the following UC Feature	
KPI ID	**Description**	**Value**	**Test**	**ID Feat**	**Description**
UC8_KPI_FT_01	Edge node has CAN Bus connectivity	True/False	TBD	F1_BEEA	ADAPTABILITY - EXTENSIBILITY - PORT CONNECTION - CAN BUS
UC8_KPI_FT_02	Edge node has AI/ ML accelerator	True/False	TBD	F2_BEEA	ADAPTABILITY - AI - HW - AI/ ML ACCELERATOR
UC8_KPI_FT_03	Edge node is capable of real time applications and process camera streams in real-time	True/False	TBD	F3_BEEA	ADAPTABILITY - AI - SW - INFERENCE - REALTIME
UC8_KPI_FT_04	The AI model are located in the node	True/False	TBD	F4_BEEA	ADAPTABILITY - AI - SW - INFERENCE - LOCATION - NODE
UC8_KPI_FT_05	The AI models will be prepared for the VERSAL	True/False	TBD	F5_BEEA	ADAPTABILITY - AI - SW - INFERENCE - MODEL - FORMAT -
UC8_KPI_FT_06	AI models will be trained in the cloud and then deployed on the node	True/False	TBD	F6_BEEA	ADAPTABILITY - AI - SW - LEARNING/ TRAINING - LOCATION - CLOUD
UC8_KPI_FT_07	AI models will be trained on a device and then deployed on the node	True/False	TBD	F7_BEEA	ADAPTABILITY - AI - SW - LEARNING/ TRAINING - LOCATION - OTHER
UC8_KPI_FT_08	The AI models use supervised learning for training	True/False	TBD	F8_BEEA	ADAPTABILITY - AI - SW - LEARNING/ TRAINING - PARADIGM - SUPERVISED LEARNING
UC8_KPI_FT_09	Vitis is able to import and execute YOLO algorithms for KRIA platform	True/False	TBD	F9_BEEA	ADAPTABILITY - AI - SW - LEARNING/ TRAINING - ALGORITHMS - YOLO
UC8_KPI_FT_10	Vitis is able to import and deploy convolutional neural networks for KRIA platform	True/False	TBD	F10_BEEA	ADAPTABILITY - AI - SW - LEARNING/ TRAINING - ALGORITHMS - CNN
UC8_KPI_FT_11	Vitis is able to import and deploy artificial neural networks for Versal platform	True/False	TBD	F11_BEEA	ADAPTABILITY - AI - SW - LEARNING/ TRAINING - ALGORITHMS - ANN
UC8_KPI_FT_12	Vitis is able to import and deploy graph neural networks for Versal platform	True/False	TBD	F12_BEEA	ADAPTABILITY - AI - SW - LEARNING/ TRAINING - ALGORITHMS - GNN
UC8_KPI_FT_13	Edge node provides the library Tensorflow - Keras	True/False	TBD	F13_BEEA	ADAPTABILITY - AI - SW - LIBRARY - TENSORFLOW - KERAS
UC8_KPI_FT_14	Edge node provides the library OpenCV	True/False	TBD	F14_BEEA	ADAPTABILITY - AI - SW - LIBRARY - OPENCV
UC8_KPI_FT_15	Edge node provides the library NumPy	True/False	TBD	F15_BEEA	ADAPTABILITY - AI - SW - LIBRARY - NUMPY
UC8_KPI_FT_16	Edge node provides the library PyTorch	True/False	TBD	F16_BEEA	ADAPTABILITY - AI - SW - LIBRARY - PYTORCH
UC8_KPI_FT_17	Service orchestration part of the fleet management system	True/False	TBD	F17_BEEA	ADAPTABILITY - SERVICES ORCHESTRATION
UC8_KPI_FT_18	Edge node adapts to various predefined scenarios	True/False	TBD	F18_BEEA	ADAPTABILITY - OPERATION MODE CHANGE - METASCHEDULING - SYSTEM RECONFIGURATION
UC8_KPI_FT_19	Edge node is fault tolerant	True/False	TBD	F19_BEEA	ADAPTABILITY - OPERATION MODE CHANGE - METASCHEDULING - FAULT TOLERANCE
UC8_KPI_FT_20	Edge node adapts to required load level with different low power approaches	True/False	TBD	F20_BEEA	ADAPTABILITY - OPERATION MODE CHANGE - METASCHEDULING - LOW POWER
UC8_KPI_FT_21	AI model for object detection have to be validated concerning the accuracy	> 95 %	TBD	F21_BEEA	RELIABILITY - AI MODEL - ACCURACY / VALIDATION
UC8_KPI_FT_22	TT off chip comm. required for safe communication between the edge node	True/False	TBD	F22_BEEA	SAFETY - TIME TRIGGERED COMMUNICATION - OFF CHIP
UC8_KPI_FT_23	TT on chip comm. required for safety monitoring the node level of an edge node	True/False	TBD	F23_BEEA	SAFETY - TIME TRIGGERED COMMUNICATION - ON CHIP
UC8_KPI_FT_24	Safety service is required for evaluation of the object detection	True/False	TBD	F24_BEEA	SAFETY - MONITORING - CORES
UC8_KPI_FT_25	Self testing for the TTNOC on the edge	True/False	TBD	F25_BEEA	SAFETY - SELF TESTING - BUILT-IN SELF TEST ON ADAPTIVE TIME TRIGGERED NOC
UC8_KPI_FT_26	Scheduling services on node level to provide fail-safe operation	True/False	TBD	F26_BEEA	SAFETY - FAIL-SAFE - SCHEDULING SERVICES
UC8_KPI_FT_27	Safe wireless communication between nodes	True/False	TBD	F27_BEEA	SAFETY - REDUNDANCY - COMMUNICATION MESSAGES
UC8_KPI_FT_28	Safety service is required for evaluation of the object detection	True/False	TBD	F28_BEEA	SAFETY - REALTIME AWARE - NON-INTERRUPTABLE PROCESSES
UC8_KPI_FT_29	Scheduling services on node level to provide fail-safe operation	True/False	TBD	F29_BEEA	SAFETY - REALTIME AWARE - HW FAILURE INTERRUPTS
UC8_KPI_FT_30	Edge node must provide a degration level for processes	True/False	TBD	F30_BEEA	SAFETY - PROCESS SCHEDULING - SYSTEM DEGRADATION
UC8_KPI_FT_31	Safety Regulation ISO 61508 Generic	True/False	TBD	F31_BEEA	SAFETY - REGULATION - ISO 61508 - Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems
UC8_KPI_FT_32	Part of the meta scheduling approach	True/False	TBD	F32_BEEA	LOW POWER - SCHEDULING SERVICES
UC8_KPI_FT_33	Battery level of the shuttle will be tracked for data collection	True/False	TBD	F33_BEEA	CONTEXT-AWARENESS - SENSORS - BATTERY LEVEL
UC8_KPI_FT_34	Shuttle edge node requires cameras for environmental awareness	10 fps	TBD	F34_BEEA	CONTEXT-AWARENESS - SENSORS - CAMERA
UC8_KPI_FT_35	Shuttle edge node utilizes sensors for positioning in the racking	True/False	TBD	F35_BEEA	CONTEXT-AWARENESS - SENSORS - POSITION
UC8_KPI_FT_36	Shuttle edge node utilizes sensors for fine positioning to the totes	True/False	TBD	F36_BEEA	CONTEXT-AWARENESS - SENSOR NETWORK - RELATIVE POSITION
UC8_KPI_FT_37	AI model for object detection via cameras for the shuttles	True/False	TBD	F37_BEEA	CONTEXT-AWARENESS - ACTIONS - OBJECT DETECTION
UC8_KPI_FT_38	AI model for object detection triggers on detection and generates an alarm	True/False	TBD	F38_BEEA	CONTEXT-AWARENESS - ACTIONS - AI TRIGGERED - ALARM
UC8_KPI_FT_39	Deployed design and models has to be verified during boot process	True/False	TBD	F39_BEEA	SECURITY - BOOT - FIRMWARE VERIFICATION
UC8_KPI_FT_40	Connection to higher-level processes, such as the mfc or for downloading diagnose data	True/False	TBD	F40_BEEA	FRACTALITY - COMMUNICATION / CONNECTIVITY - TECHNOLOGIES - ETHERNET
UC8_KPI_FT_41	Connection between nodes, Versal <--> Kria	True/False	TBD	F41_BEEA	FRACTALITY - COMMUNICATION / CONNECTIVITY - TECHNOLOGIES - WIFI
UC8_KPI_FT_42	Data protocoll between nodes will be MQTT	True/False	TBD	F42_BEEA	FRACTALITY - COMMUNICATION / CONNECTIVITY - DATA PROTOCOLS - MQTT
UC8_KPI_FT_43	Fleet management system service orchestration	True/False	TBD	F43_BEEA	FRACTALITY - ORCHESTRATION - SERVICES
UC8_KPI_FT_44	Fleet management system data orchestration	True/False	TBD	F44_BEEA	FRACTALITY - ORCHESTRATION - DATA
UC8_KPI_FT_45	Fleet management system model orchestration	True/False	TBD	F45_BEEA	FRACTALITY - ORCHESTRATION - MODEL
UC8_KPI_FT_46	Hierarchical architecture on system level of the edge nodes	True/False	TBD	F46_BEEA	FRACTALITY - HIERARCHICAL ARCHITECTURE
UC8_KPI_FT_47	Versal node will be implemented in the lift node	True/False	TBD	F47_BEEA	OTHER: NON-FUNCTIONAL - PLATFORM (SELECT ONE) - VERSAL - ARM
UC8_KPI_FT_48	Kria node (Zynq Ultrascale + MPSoC) will be implented in the shutte nodes	True/False	TBD	F48_BEEA	OTHER: NON-FUNCTIONAL - PLATFORM (SELECT ONE) - ZYNQ ULTRASCALE+ (VERSAL ALTERNATIVE)
UC8_KPI_FT_49	Edge nodes execute a Linux OS	True/False	TBD	F49_BEEA	OTHER: NON-FUNCTIONAL - OS - LINUX

Figure 88 - KPIs for UC8 Implementation Plan to measure the contribution to FRACTAL Features

7.6 Implementation plan

7.6.1 Architecture

7.6.1.1 FRACTAL Big Picture

The use case integrates several FRACTAL components from the edge and from the cloud. Those components can be seen in the Figure 89 within the context of FRACTAL Big Picture representation.

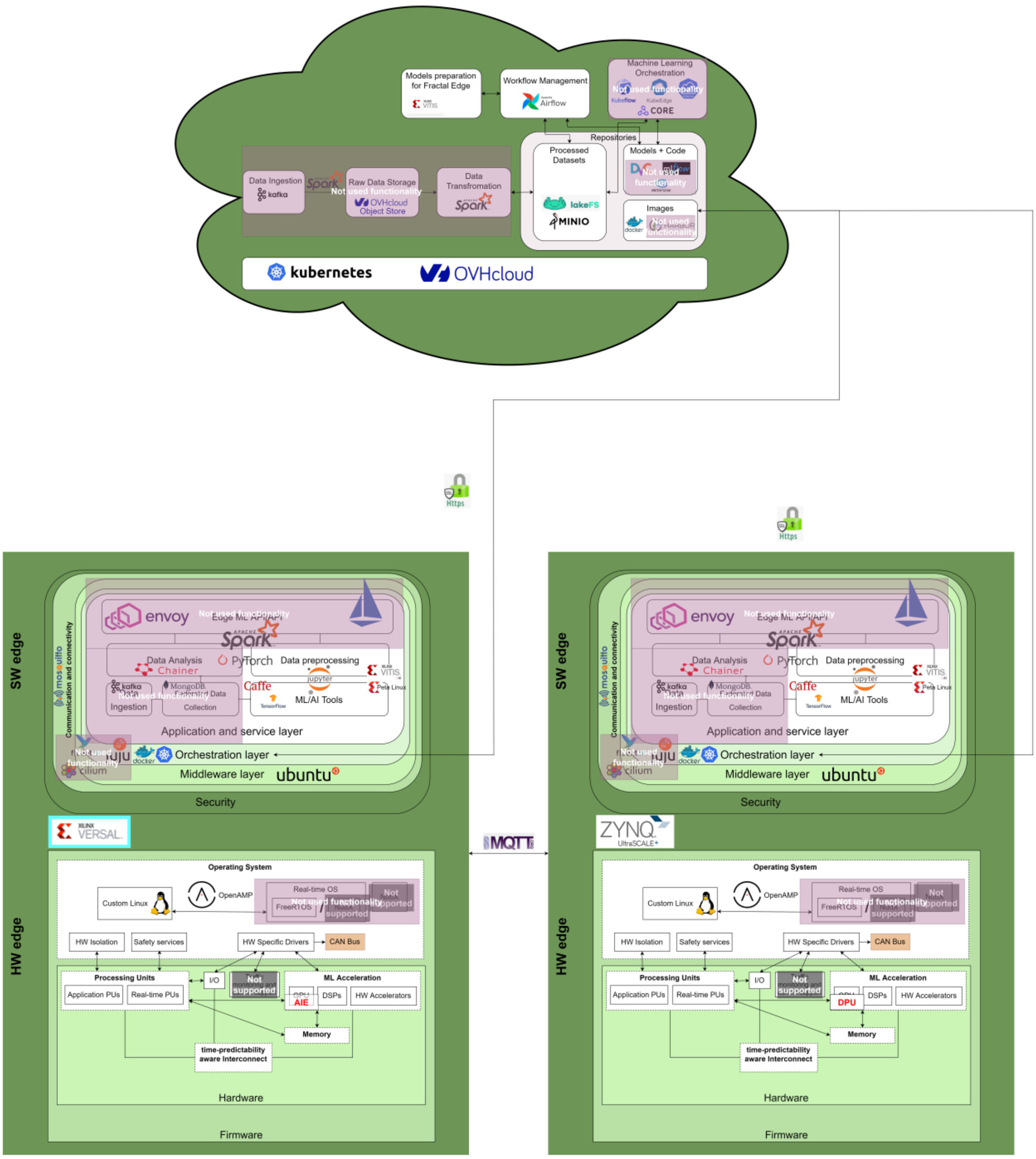


Figure 89 - Fractal Big Picture Instantiation for UC8

The main components for UC8 are implemented in the edge nodes and related to the swarm intelligence and the extended functional safety aspect. On the other side cloud components bring the opportunity for a fleet management system, to manage single or multiple swarms per customer with custom specifications. An architecture for the test setup is shown in Figure 90 with green boxes for the FRACTAL components.

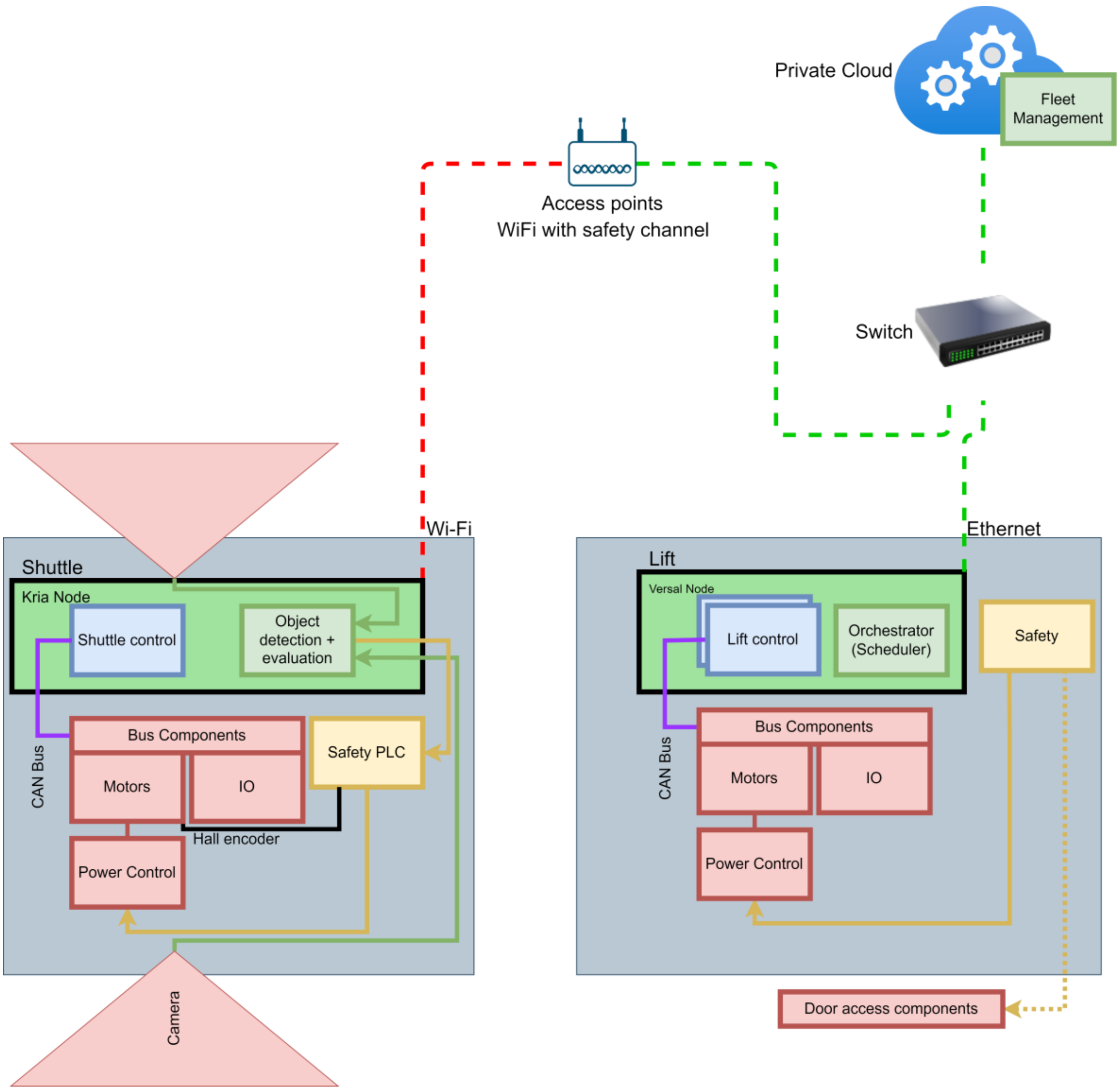


Figure 90 - UC8 test setup architecture with fractal components

7.6.2 Tasks

7.6.2.1 Chronogram

Figure 91 shows the implementation plan tasks and chronogram for UC8. It is basically divided into four main tasks with subtasks. Sections to follow describe the tasks in some detail.

Implementation Plan																			
Task ID	**Description**	M19	M20	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30	M31	M32	M33	M34	M35	M36
UC8_T1	**Basic target environment setup (Versal node and Kria node)**																		
UC8_T1_1	Prepare hardware setup for Vitis AI on target (Versal node)						■	■	■										
UC8_T1_2	Build AA - shuttle orchestrator for target (Versal - ARM)							■	■	■									
UC8_T1_3	Test AA - shuttle orchestrator for target (Versal - ARM)							■	■	■	■	■	■						
UC8_T1_4	Build shuttle orchestrator application												■	■					
UC8_T1_5	Prepare hardware setup for Vitis AI on target (Kria node)					■	■	■	■										
UC8_T1_6	Build object detection model for target (Kria - ARM)							■	■	■									
UC8_T1_7	Test object detection model on target								■	■	■								
UC8_T1_8	Build zone evaluation logic application									■	■	■	■						
UC8_T1_9	Setup cloud service orchestrator										■	■	■	■					
UC8_T1_10	Build demonstration software for test setup												■	■					
UC8_T2	**Preparation**																		
UC8_T2_1	Model training (Versal node) - Orchestrator						■	■	■	■									
UC8_T2_2	Model training (Kria node) - Object detection						■	■	■	■									
UC8_T3	**Integration**																		
UC8_T3_1	Integration of HW and SW base functionalities in the test setup													■	■	■	■		
UC8_T3_2	Test basic functionalities (shuttle control, lift control, interfaces)													■	■	■			
UC8_T3_3	Test extended functionalities (FRACTAL edge components)													■	■	■			
UC8_T3_4	Test cloud services															■	■		
UC8_T4	**System Evaluation/ Benchmark**																		
UC8_T4_1	Metrics Calculation														■	■	■	■	■

Figure 91 - UC8 implementation plan

7.6.2.2 Task UC8_T1 - Basic target environment setup (Versal node and Kria node)

The sub tasks under this task are devoted to implement the prime basic scenario of UC8 that involves only the Edge Versal based node of FRACTAL.

7.6.2.2.1 Sub Task: UC8_T1_1 - Prepare hardware setup for Vitis AI on target (Versal node)

This task consists of preparing the hardware design and then testing for the versal edge node in UC8. The focus is on preparing hardware interfaces of the board in Vivado, which are required for the basic operation and the successful import in Vitis AI.

7.6.2.2.2 Sub Task: UC8_T1_2 – Build AA - shuttle orchestrator for target (Versal - ARM)

Build and deploy of the orchestrator on versal board.

7.6.2.2.3 Sub Task: UC8_T1_3 – Test AA - shuttle orchestrator for target (Versal - ARM)

Test of the orchestrator and predictions check.

7.6.2.2.4 Sub Task: UC8_T1_4 – Build shuttle orchestrator application

Finalizing orchestrator model and merging with hardware build, including control services.

7.6.2.2.5 Sub Task: UC8_T1_5 – Prepare hardware setup for Vitis AI on target (Kria node)

This task consists of preparing the hardware design and then testing for the kria edge node in UC8. The focus is on preparing hardware interfaces of the board in Vivado, which are required for the basic operation and the successful import in Vitis AI.

7.6.2.2.6 Sub Task: UC8_T1_6 – Build object detection model for target (Kria - ARM)

Build and deploy object detection model on the kria board.

7.6.2.2.7 Sub Task: UC8_T1_7 – Test object detection model on target

Test of the object detection model on the kria board. Internal comparison of YOLO and CNN.

7.6.2.2.8 Sub Task: UC8_T1_8 – Build zone evaluation logic application

Build zone evaluation logic for object detection model on the kria board.

7.6.2.2.9 Sub Task: UC8_T1_9 – Setup cloud service orchestrator

Provide private cloud and setup of the fleet management system.

7.6.2.2.10 Sub Task: UC8_T1_10 - Build demonstration software for test setup

Merging of components.

- Versal board: Orchestrator and control services based on own hardware design.
- Kria board: Object detection + evaluation and control services based on own hardware design.
- Cloud: Fleet management system preparation for the test setup.

7.6.2.3 Task UC8_T2 – Preparation

7.6.2.3.1 Sub Task: UC8_T2_1 - Model training (Versal node) – Orchestrator

Training of the model with self-generated data over the test setup and the scenario generator.

7.6.2.3.2 Sub Task: UC8_T2_2 - Model training (Kria node) - Object detection

Training of the model with public datasets.

7.6.2.4 Task UC8_T3 – Integration

7.6.2.4.1 Sub Task: UC8_T3_1 - Integration of HW and SW base functionalities in the test setup

Integration of demonstration software and implementation of versal board in the control cabinet. Same for kria board, regarding the shuttles. Integration of the fleet management system in the edge nodes as well.

7.6.2.4.2 Sub Task: UC8_T3_2 - Test basic functionalities (shuttle control, lift control, interfaces)

Testing of all basic functionalities, to ensure the core functions are implemented successfully before exploring the extended functionalities.

7.6.2.4.3 Sub Task: UC8_T3_3 - Test extended functionalities (FRACTAL edge components)

Testing of FRACTAL edge components in the test setup.

7.6.2.4.4 Sub Task: UC8_T3_4 - Test cloud services

Testing of FRACTAL cloud components in the test setup.

7.6.2.5 Task UC8_T4 - System Evaluation/ Benchmark

7.6.2.5.1 Sub Task: UC8_T4_1 - Metrics Calculation

This task consists of calculating the metrics results. The Test of the KPI associated to this task will consist of several criterions, not only accuracy and fps, but also criterions such as safety, updating from cloud, model management, system throughput etc.

7.6.3 Components

This section summarizes the components involved in the Implementation Plan. All the components listed here have been extracted from Tab ***Components*** in the general Excel defined in WP1 ***FRACTAL - Requirements_KPIs_Components.xlsx.***

Components are basically divided into two groups:

- ***Components produced by the UC*** *resulting from executing the Implementation Plan*.
- ***Common FRACTAL Components*** (from WP3, WP4, WP5, WP6) that are needed to execute the Implementation Plan.

Following two sub sections list these components.

7.6.3.1 Components produced by the Implementation Plan

These components (Figure 92), *are* ***produced by executing the tasks*** of the Implementation Plan.

UC Components		
KPI ID	Name	Description
UC8_CMP_01	Hardware design with CAN Bus connectivity (VERSAL and KRIA)	FPGA Hardware design ready for Internal communication between components like IOs and motors.
UC8_CMP_02	Evaluation of object detection	Safety relevant evaluation logic and connection to the safety plc
UC8_CMP_03	AI accelerated orchestrator/ scheduler	Warehouse optimization and pathfinding based on the metascheduler
UC8_CMP_04	Cloud service orchestration	Fleet management system
UC8_CMP_05	Integrated demonstration software on target	Demonstration of software running in the test setup

Figure 92 - Components produced by the execution of UC8 Implementation Plan

7.6.3.2 FRACTAL components needed to execute the Implementation Plan

These components (Figure 93), *are* ***Common FRACTAL Components*** (from WP3, WP4, WP5, WP6) that are ***needed to execute the Implementation Plan***.

FRACTAL Components needed by the UC	
KPI ID	**Name**
WP3T32-10	VERSAL accelerator building-blocks
WP3T34-03	Versal Model deployment layer
WP4T41-02	HATMA
WP4T41-04	Versal RPU access for Power Services
WP4T41-05	Agreement protocol for Low-Power Services
WP4T41-06	Versal Isolation Design - Functional Safety
WP4T42-02	Versal RPU access to AI acceleration
WP4T42-03	Scenario Generator
WP4T42-04	GA-Scheduler
WP4T42-05	AI-Scheduler Model
WP4T42-06	Schedule Verifier
WP4T42-07	Hierarchical Metascheduler
WP4T43-04	ATTNoC
WP4T43-06	FPGA Fault-injector
WP4T43-08	Seamless redundancy for ATTNoC
WP4T43-11	Time-Triggered Extension Layer for VERSAL NoC
WP4T43-13	Safety Analysis
WP4T44-02	OS Security Layer
WP5T52-04-05	Datasets version control
WP5T52-04-07	Images repository
WP5T52-05-02	Data pipelines and workflows orchestrator
WP5T52-06-01	Model preparation for Fractal Edge (Versal Xilinx
WP5T54-01-01	MLBuffet
WP5T54-02-02	Kubernetes-based Container Orchestrators for tl

Figure 93 - Common FRACTAL components from WP3, WP4, WP5, WP6 needed to execute UC8 Implementation Plan.

7.6.4 Traceability relationships of Tasks-Components-KPIs

Finally, this section ***links together*** *tasks*, *components* and *KPIs*. For each Task, the following traceability-relationships are given:

- ***Components***
  - IN Components – Input components needed by the task.
  - OUT Components – Output components produced by the task.
- ***KPIs for UC Implementation Plan***
- ***KPIs for FRACTAL Objectives & Features***

Following sub sections detail this information for each task.

7.6.4.1 T1 - Prepare hardware setup for Vitis AI on target (Versal node)

7.6.4.1.1 Sub Task: UC8_T1_1 - Prepare hardware setup for Vitis AI on target (Versal node)

Figure 94 shows traceability relationships for Task UC8_T1_1:

IN Comp	OUT Comp	KPI ID	KPI Description	Value
WP3T32-10 WP3T34-03 WP4T41-02 WP4T41-04 WP4T41-05 WP4T42-02 WP4T42-03 WP4T42-04 WP4T42-05 WP4T42-06 WP4T42-07 WP4T43-04 WP4T43-08 WP4T43-11 WP4T43-13 WP4T44-02	UC8_CMP_01	UC8_KPI_IP_02	Duty cycle of control software (Versal) with target design	< 20 ms
		UC8_KPI_IP_Req_01	"The edge node should have following hardware specification: - at least 2 cores @ 800 MHz - at least 4 GB RAM - at least eMMC Memory or similar."	True/False
		UC8_KPI_IP_Req_02	"These communication protocols shall be used from Linux OS: - MQTT over WiFi mesh network for communication between nodes - CAN Bus for internal communication."	True/False
		UC8_KPI_IP_Req_07	The node shall feature Linux operating system with real time capability (e.g. time-triggered communication capabilities).	True/False
		UC8_KPI_IP_Req_08	Safety wireless communication should be over a black channel (ASIL 3, ISO 26262) between nodes.	True/False
		UC8_KPI_IP_Req_09	For the edge nodes a cross compiler shall be available to port control software.	True/False

KPI ID	KPI Description	Value	Test
UC8_KPI_FT_01	Edge node has CAN Bus connectivity	True/False	Meassure duty cycle
UC8_KPI_FT_05	The AI models will be prepared for the VERSAL platform	True/False	
UC8_KPI_FT_11	Vitis is able to import and deploy artificial neural networks for Versal platform	True/False	
UC8_KPI_FT_12	Vitis is able to import and deploy graph neural networks for Versal platform	True/False	
UC8_KPI_FT_18	Edge node adapts to various predefined scenarios	True/False	
UC8_KPI_FT_19	Edge node is fault tolerant	True/False	
UC8_KPI_FT_20	Edge node adapts to required load level with different low power approaches	True/False	
UC8_KPI_FT_22	Required for safe communication between the edge node	True/False	
UC8_KPI_FT_23	Required for safety monitoring the node level of an edge node	True/False	
UC8_KPI_FT_25	Self testing for the TTNOC on the edge	True/False	
UC8_KPI_FT_27	Safe wireless communication between nodes	True/False	
UC8_KPI_FT_29	Scheduling services on node level to provide fail-safe operation	True/False	
UC8_KPI_FT_30	Edge node must provide a degration level for processes	True/False	
UC8_KPI_FT_32	Part of the meta scheduling approach	True/False	
UC8_KPI_FT_39	Deployed design and models has to be verified during boot process	True/False	
UC8_KPI_FT_40	Connection to higher-level processes, such as the mfc or for downloading diagnose data	True/False	
UC8_KPI_FT_41	Connection between nodes, Versal <--> Kria	True/False	
UC8_KPI_FT_42	Data protocoll between nodes will be MQTT	True/False	
UC8_KPI_FT_46	Hierarchical architecture on system level of the edge nodes	True/False	
UC8_KPI_FT_47	Versal node will be implemented in the lift node	True/False	
UC8_KPI_FT_49	Edge nodes execute a Linux OS	True/False	
UC8_KPI_FO_01	Cycle time of services on edge node with accelerated orchestrator implemented and running. (VERSAL)	< 20 ms	

Figure 94 - Task UC8_T1_1 traceability relationship between task, components and KPIs

7.6.4.1.2 Sub Task: UC8_T1_2 – Build AA - shuttle orchestrator for target (Versal - ARM)

Figure 95 shows traceability relationships for Task UC8_T1_2:

IN Comp	UT Com	KPI ID	KPI Description	Value
WP4T41-02 WP4T41-04 WP4T42-02 WP4T42-03 WP4T42-04 WP4T42-05 WP4T42-06 WP4T42-07 WP4T43-04 WP4T43-06 WP4T43-08 WP4T43-11 WP5T54-02-02		UC8_KPI_IP_03	Build AA - shuttle orchestrator for target (Versal - ARM)	True/False
		UC8_KPI_IP_Req_05	The edge node shall be able to use an adaptive orchestrator (scheduler) for storing strategies and optimized pathfinding for each shuttle depending on material (weight, type), frequency of requests, division of same type in different levels for alternative access/ faster access on big order amount.	True/False
		UC8_KPI_IP_Req_06	The edge node shall offer optimized pathfinding: Improving path of the shuttles, for different scenarios; obstacle in same layer; malfunction of a shuttle; avoiding crossing in same level. The node shall feature Linux operating system with real time capability (e.g. time-triggered communication capabilities).	True/False
		UC8_KPI_IP_Req_10	The edge node shall support libraries, like Tensorflow/ Keras.	True/False

KPI ID	KPI Description	Value	Test
UC8_KPI_FT_02	Edge node has AI/ ML accelerator	True/False	True/False
UC8_KPI_FT_04	The AI model are located in the node	True/False	
UC8_KPI_FT_05	The AI models will be prepared for the VERSAL platform	True/False	
UC8_KPI_FT_06	AI models will be trained in the cloud and then deployed on the node	True/False	
UC8_KPI_FT_07	AI models will be trained on a device and then deployed on the node	True/False	
UC8_KPI_FT_08	The AI models use supervised learning for training	True/False	
UC8_KPI_FT_11	Vitis is able to import and deploy artificial neural networks for Versal platform	True/False	
UC8_KPI_FT_12	Vitis is able to import and deploy graph neural networks for Versal platform	True/False	
UC8_KPI_FT_13	Edge node provides the library Tensorflow - Keras	True/False	
UC8_KPI_FT_15	Edge node provides the library NumPy	True/False	
UC8_KPI_FT_16	Edge node provides the library PyTorch	True/False	
UC8_KPI_FT_18	Edge node adapts to various predefined scenarios	True/False	
UC8_KPI_FT_22	Required for safe communication between the edge node	True/False	
UC8_KPI_FT_23	Required for safety monitoring the node level of an edge node	True/False	
UC8_KPI_FT_41	Connection between nodes, Versal <--> Kria	True/False	
UC8_KPI_FT_42	Data protocoll between nodes will be MQTT	True/False	
UC8_KPI_FT_46	Hierarchical architecture on system level of the edge nodes	True/False	

Figure 95 - Task UC8_T1_2 traceability relationship between task, components and KPIs

7.6.4.1.3 Sub Task: UC8_T1_3 – Test AA - shuttle orchestrator for target (Versal - ARM)

Figure 96 shows traceability relationships for Task UC8_T1_3:

IN Comp	UT Com	KPI ID	KPI Description	Value
WP4T41-02 WP4T41-04 WP4T42-02 WP4T42-03 WP4T42-04 WP4T42-05 WP4T42-06 WP4T42-07 WP4T43-04 WP4T43-06 WP4T43-08 WP4T43-11 WP5T54-02-02		UC8_KPI_IP_04	Inference time of predictions - shuttle orchestrator (Versal - ARM)	< 2 s
		UC8_KPI_IP_Req_05	The edge node shall be able to use an adaptive orchestrator (scheduler) for storing strategies and optimized pathfinding for each shuttle depending on material (weight, type), frequency of requests, division of same type in different levels for alternative access/ faster access on big order amount. The edge node shall offer optimized pathfinding: Improving path of the shuttles, for different scenarios; obstacle in same layer; malfunction of a shuttle; avoiding crossing in same level.	True/False
		UC8_KPI_IP_Req_06	The node shall feature Linux operating system with real time capability (e.g. time-triggered communication capabilities).	True/False
		UC8_KPI_IP_Req_10	The edge node shall support libraries, like Tensorflow/ Keras.	True/False

KPI ID	KPI Description	Value	Test
UC8_KPI_FT_02	Edge node has AI/ ML accelerator	True/False	True/False
UC8_KPI_FT_04	The AI model are located in the node	True/False	
UC8_KPI_FT_05	The AI models will be prepared for the VERSAL platform	True/False	
UC8_KPI_FT_06	AI models will be trained in the cloud and then deployed on the node	True/False	
UC8_KPI_FT_07	AI models will be trained on a device and then deployed on the node	True/False	
UC8_KPI_FT_08	The AI models use supervised learning for training	True/False	
UC8_KPI_FT_11	Vitis is able to import and deploy artificial neural networks for Versal platform	True/False	
UC8_KPI_FT_12	Vitis is able to import and deploy graph neural networks for Versal platform	True/False	
UC8_KPI_FT_13	Edge node provides the library Tensorflow - Keras	True/False	
UC8_KPI_FT_15	Edge node provides the library NumPy	True/False	
UC8_KPI_FT_16	Edge node provides the library PyTorch	True/False	
UC8_KPI_FT_18	Edge node adapts to various predefined scenarios	True/False	
UC8_KPI_FT_22	Required for safe communication between the edge node	True/False	
UC8_KPI_FT_23	Required for safety monitoring the node level of an edge node	True/False	
UC8_KPI_FT_41	Connection between nodes, Versal <--> Kria	True/False	
UC8_KPI_FT_42	Data protocoll between nodes will be MQTT	True/False	
UC8_KPI_FT_46	Hierarchical architecture on system level of the edge nodes	True/False	

Figure 96 - Task UC8_T1_3 traceability relationship between task, components and KPIs

7.6.4.1.4 Sub Task: UC8_T1_4 – Build shuttle orchestrator application

Figure 97 shows traceability relationships for Task UC8_T1_4:

IN Comp	UT Com	KPI ID	KPI Description	Value
WP4T41-02 WP4T41-04 WP4T42-02 WP4T42-03 WP4T42-04 WP4T42-05 WP4T42-06 WP4T42-07 WP4T43-04 WP4T43-08 WP4T43-11	UC8_CMP_03	UC8_KPI_IP_01 UC8_KPI_IP_Req_05 UC8_KPI_IP_Req_06	All subtask success - Versal node The edge node shall be able to use an adaptive orchestrator (scheduler) for storing strategies and optimized pathfinding for each shuttle depending on material (weight, type), frequency of requests, division of same type in different levels for alternative access/ faster access on big order amount. The edge node shall offer optimized pathfinding: Improving path of the shuttles, for different scenarios; obstacle in same layer; malfunction of a shuttle; avoiding crossing in same level. The node shall feature Linux operating system with real time capability (e.g. time-triggered communication capabilities).	True/False True/False True/False

KPI ID	KPI Description	Value	Test
_KPI_FO_04	Real-time inference for meta scheduler, which can react on various pre-defined events and make safe decisions for pathfinding and storage strategies for different goods.	< 2 s	Meassure inference time

Figure 97 - Task UC8_T1_4 traceability relationship between task, components and KPIs

7.6.4.1.5 Sub Task: UC8_T1_5 – Prepare hardware setup for Vitis AI on target (Kria node)

Figure 98 shows traceability relationships for Task UC8_T1_5:

IN Comp	UT Comp	KPI ID	KPI Description	Value
WP3T34-03 WP4T41-04 WP4T41-05 WP4T41-06 WP4T42-02 WP4T43-04 WP4T43-06 WP4T43-08 WP4T43-11 WP4T43-13 WP4T44-02	UC8_CMP_01	UC8_KPI_IP_06	Duty cycle of control software (Kria) with target design.	< 20 ms
		UC8_KPI_IP_Req_01	"The edge node should have following hardware specification: - at least 2 cores @ 800 MHz - at least 4 GB RAM - at least eMMC Memory or similar."	True/False
		UC8_KPI_IP_Req_02	"These communication protocols shall be used from Linux OS: - MQTT over WiFi mesh network for communication between nodes - CAN Bus for internal communication."	True/False
		UC8_KPI_IP_Req_07	The node shall feature Linux operating system with real time capability (e.g. time-triggered communication capabilities).	True/False
		UC8_KPI_IP_Req_08	Safety wireless communication should be over a black channel (ASIL 3, ISO 26262) between nodes.	True/False
		UC8_KPI_IP_Req_09	For the edge nodes a cross compiler shall be available to port control software.	True/False

KPI ID	KPI Description	Value	Test
UC8_KPI_FT_01	Edge node has CAN Bus connectivity	True/False	
UC8_KPI_FT_03	Edge node is capable of real time applications and process camera streams in real-time	True/False	
UC8_KPI_FT_13	Edge node provides the library Tensorflow - Keras⍰	True/False	
UC8_KPI_FT_22	Required for safe communication between the edge node	True/False	
UC8_KPI_FT_23	Required for safety monitoring the node level of an edge node	True/False	
UC8_KPI_FT_24	Safety service is required for evaluation of the object detection	True/False	
UC8_KPI_FT_25	Self testing for the TTNOC on the edge	True/False	
UC8_KPI_FT_26	Scheduling services on node level to provide fail-safe operation	True/False	
UC8_KPI_FT_27	Safe wireless communication between nodes	True/False	
UC8_KPI_FT_28	Safety service is required for evaluation of the object detection	True/False	
UC8_KPI_FT_29	Scheduling services on node level to provide fail-safe operation	True/False	
UC8_KPI_FT_30	Edge node must provide a degration level for processes	True/False	Meassure duty cycle
UC8_KPI_FT_31	Safety Regulation ISO 61508 Generic	True/False	
UC8_KPI_FT_33	Battery level of the shuttle will be tracked for data collection	True/False	
UC8_KPI_FT_34	Shuttle edge node requires cameras for environmental awareness	10 fps	
UC8_KPI_FT_35	Shuttle edge node utilizes sensors for positioning in the racking	True/False	
UC8_KPI_FT_36	Shuttle edge node utilizes sensors for fine positioning to the totes	True/False	
UC8_KPI_FT_39	Deployed design and models has to be verified during boot process	True/False	
UC8_KPI_FT_41	Connection between nodes, Versal <--> Kria⍰	True/False	
UC8_KPI_FT_42	Data protocoll between nodes will be MQTT⍰	True/False	
UC8_KPI_FT_46	Hierarchical architecture on system level of the edge nodes	True/False	
UC8_KPI_FT_48	Kria node (Zynq Ultrascale + MPSoC) will be implemented in the shuttle nodes	True/False	
UC8_KPI_FT_49	Edge nodes execute a Linux OS	True/False	
UC8_KPI_FO_02	Cycle time of services on edge node with accurate cognitive AI application implemented and running. (KRIA)	< 20 ms	

Figure 98 - Task UC8_T1_5 traceability relationship between task, components and KPIs

7.6.4.1.6 Sub Task: UC8_T1_6 – Build object detection model for target (Kria - ARM)

Figure 99 shows traceability relationships for Task UC8_T1_6:

KPI ID	KPI Description	Value
UC8_KPI_IP_07	Build OpenCV for target (Kria - ARM) success	True/False
UC8_KPI_IP_09	Build object detection application success	True/False
UC8_KPI_IP_Req_04	The edge node shall be capable to detect objects (human body and other obstacles) from video input stream of the provided cameras and evaluate the detected object to generate a safe output, if the obstacle is in a defined range of the shuttle.	True/False
UC8_KPI_IP_Req_10	The edge node shall support libraries, like Tensorflow/ Keras.	True/False

KPI ID	KPI Description	Value	Test
UC8_KPI_FT_02	Edge node has AI/ ML accelerator	True/False	
UC8_KPI_FT_03	Edge node is capable of real time applications and process camera streams in real-time	True/False	
UC8_KPI_FT_04	The AI model are located in the node	True/False	
UC8_KPI_FT_05	The AI models will be prepared for the VERSAL platform	True/False	
UC8_KPI_FT_06	AI models will be trained in the cloud and then deployed on the node	True/False	
UC8_KPI_FT_07	AI models will be trained on a device and then deployed on the node	True/False	
UC8_KPI_FT_08	The AI models use supervised learning for training	True/False	
UC8_KPI_FT_09	Vitis is able to import and execute YOLO algorithms for KRIA platform	True/False	
UC8_KPI_FT_10	Vitis is able to import and deploy convolutional neural networks for KRIA platform	True/False	
UC8_KPI_FT_13	Edge node provides the library Tensorflow - Keras	True/False	
UC8_KPI_FT_14	Edge node provides the library OpenCV	True/False	
UC8_KPI_FT_21	AI model for object detection have to be validated concerning the accuracy	> 95 %	Accuracy Inference time
UC8_KPI_FT_22	Required for safe communication between the edge node	True/False	
UC8_KPI_FT_23	Required for safety monitoring the node level of an edge node	True/False	
UC8_KPI_FT_24	Safety service is required for evaluation of the object detection	True/False	
UC8_KPI_FT_28	Safety service is required for evaluation of the object detection	True/False	
UC8_KPI_FT_31	Safety Regulation ISO 61508 Generic	True/False	
UC8_KPI_FT_34	Shuttle edge node requires cameras for environmental awareness	10 fps	
UC8_KPI_FT_37	AI model for object detection via cameras for the shuttles	True/False	
UC8_KPI_FT_38		True/False	

Figure 99 - Task UC8_T1_6 traceability relationship between task, components and KPIs

7.6.4.1.7 Sub Task: UC8_T1_7 – Test object detection model on target

Figure 100 shows traceability relationships for Task UC8_T1_7:

IN Comp	UT Com	KPI ID	KPI Description	Value
		UC8_KPI_IP_10	Model accuracy of the object detection	> 95 %
WP3T34-03 WP4T42-02 WP4T43-04 WP4T43-08 WP4T43-11 WP4T43-13		UC8_KPI_IP_Req_04	The edge node shall be capable to detect objects (human body and other obstacles) from video input stream of the provided cameras and evaluate the detected object to generate a safe output, if the obstacle is in a defined range of the shuttle.	True/False
		UC8_KPI_IP_Req_10	The edge node shall support libraries, like Tensorflow/ Keras.	True/False

KPI ID	KPI Description	Value	Test
UC8_KPI_FT_02 UC8_KPI_FT_03 UC8_KPI_FT_04 UC8_KPI_FT_05 UC8_KPI_FT_06 UC8_KPI_FT_07 UC8_KPI_FT_08 UC8_KPI_FT_09 UC8_KPI_FT_10 UC8_KPI_FT_13 UC8_KPI_FT_14 UC8_KPI_FT_21 UC8_KPI_FT_22 UC8_KPI_FT_23 UC8_KPI_FT_24 UC8_KPI_FT_28 UC8_KPI_FT_31 UC8_KPI_FT_34 UC8_KPI_FT_37 UC8_KPI_FT_38	Edge node has AI/ ML accelerator Edge node is capable of real time applications and process camera streams in real-time The AI model are located in the node The AI models will be prepared for the VERSAL platform AI models will be trained in the cloud and then deployed on the node AI models will be trained on a device and then deployed on the node The AI models use supervised learning for training Vitis is able to import and execute YOLO algorithms for KRIA platform Vitis is able to import and deploy convolutional neural networks for KRIA platform Edge node provides the library Tensorflow - Keras Edge node provides the library OpenCV AI model for object detection have to be validated concerning the accuracy Required for safe communication between the edge node Required for safety monitoring the node level of an edge node Safety service is required for evaluation of the object detection Safety service is required for evaluation of the object detection Safety Regulation ISO 61508 Generic Shuttle edge node requires cameras for environmental awareness AI model for object detection via cameras for the shuttles	True/False True/False True/False True/False True/False True/False True/False True/False True/False True/False True/False > 95 % True/False True/False True/False True/False True/False 10 fps True/False True/False	Accuracy Inference time

Figure 100 - Task UC8_T1_7 traceability relationship between task, components and KPIs

7.6.4.1.8 Sub Task: UC8_T1_8 – Build zone evaluation logic application

Figure 101 shows traceability relationships for Task UC8_T1_8:

IN Comp	UT Com	KPI ID	KPI Description	Value
WP3T34-03 WP4T42-02 WP4T43-04 WP4T43-08 WP4T43-11 WP4T43-13	UC8_CMP_02	UC8_KPI_IP_05 UC8_KPI_IP_11 UC8_KPI_IP_12 UC8_KPI_IP_13 UC8_KPI_IP_Req_04	All subtask success - Kria node Build zone evaluation logic application success Inference time of object detection Failure rate of connection between FPGA and safety plc The edge node shall be capable to detect objects (human body and other obstacles) from video input stream of the provided cameras and evaluate the detected object to generate a safe output, if the obstacle is in a defined range of the shuttle.	True/ False True/ False < 100 ms % True/ False

KPI ID	KPI Description	Value	Test
UC8_KPI_FO_05	Real-time inference for object detection on edge node with all services and accelerators implemented.	10 fps	Inference time

Figure 101 - Task UC8_T1_8 traceability relationship between task, components and KPIs

7.6.4.1.9 Sub Task: UC8_T1_9 – Setup cloud service orchestrator

Figure 102 shows traceability relationships for Task UC8_T1_9:

IN Comp	UT Com	KPI ID	KPI Description	Value
WP5T52-04-05 WP5T52-04-07 WP5T52-05-02 WP5T52-06-01 WP5T54-01-01 WP5T54-02-02	UC8_CMP_04	UC8_KPI_IP_14	Setup cloud service orchestrator success	True/ False

KPI ID	KPI Description	Value	Test
UC8_KPI_FT_17	Part of the fleet management system	True/False	TBD
UC8_KPI_FT_43	Fleet management system service orchestration	True/False	
UC8_KPI_FT_44	Fleet management system data orchestration	True/False	
UC8_KPI_FT_45	Fleet management system model orchestration	True/False	

Figure 102 - Task UC8_T1_9 traceability relationship between task, components and KPIs

7.6.4.1.10 Sub Task: UC8_T1_10 - Build demonstration software for test setup

Figure 103 shows traceability relationships for Task UC8_T1_10:

IN Comp	UT Com	KPI ID	KPI Description	Value
UC8_CMP_01 UC8_CMP_02 UC8_CMP_03 UC8_CMP_04	UC8_CMP_05	UC8_KPI_IP_08	Build demonstration software on target success	True/ False

KPI ID	KPI Description	Value	Test
UC8_KPI_FO_00	Fractal technology helps to improve the state of the art in the intralogistics industry	True/False	TBD

Figure 103 - Task UC8_T1_10 traceability relationship between task, components and KPIs

7.6.4.2 Task UC8_T2 – Preparation

7.6.4.2.1 Sub Task: UC8_T2_1 - Model training (Versal node) – Orchestrator

Figure 104 shows traceability relationships for Task UC8_T2_1:

IN Comp	UT Com	KPI ID	KPI Description	Value
UC8_CMP_03		UC8_KPI_IP_04	Inference time of predictions - shuttle orchestrator (Versal - ARM)	< 2 s

Figure 104 - Task UC8_T2_1 traceability relationship between task, components and KPIs

There are no KPIs regarding the FRACTAL objectives or features.

7.6.4.2.2 Sub Task: UC8_T2_2 - Model training (Kria node) - Object detection

Figure 105 shows traceability relationships for Task UC8_T2_2:

IN Comp	UT Com	KPI ID	KPI Description	Value
UC8_CMP_02		UC8_KPI_IP_10	Model accuracy of the object detection	> 95 %

Figure 105 - Task UC8_T2_2 traceability relationship between task, components and KPIs

There are no KPIs regarding the FRACTAL objectives or features.

7.6.4.3 Task UC8_T3 – Integration

7.6.4.3.1 Sub Task: UC8_T3_1 - Integration of HW and SW base functionalities in the test setup

Figure 106 shows traceability relationships for Task UC8_T3_1:

IN Comp	UT Com	KPI ID	KPI Description	Value
UC8_CMP_01 UC8_CMP_02 UC8_CMP_03 UC8_CMP_04		UC8_KPI_IP_01 UC8_KPI_IP_05 UC8_KPI_IP_14	All subtask success - Versal node All subtask success - Kria node Setup cloud service orchestrator success	True/False True/False True/ False

Figure 106 - Task UC8_T3_1 traceability relationship between task, components and KPIs

There are no KPIs regarding the FRACTAL objectives or features.

7.6.4.3.2 Sub Task: UC8_T3_2 - Test basic functionalities (shuttle control, lift control, interfaces)

Figure 107 shows traceability relationships for Task UC8_T3_2:

IN Comp	UT Com	KPI ID	KPI Description	Value
UC8_CMP_01		UC8_KPI_IP_02 UC8_KPI_IP_06 UC8_KPI_IP_13	Duty cycle of control software (Versal) with target design Duty cycle of control software (Kria) with target design Failure rate of connection between FPGA and safety plc	< 20 ms < 20 ms %

KPI ID	KPI Description	Value	Test
UC8_KPI_FO_01 UC8_KPI_FO_02 UC8_KPI_FO_06	Cycle time of services on edge node with accelerated orchestrator implemented and running. (VERSAL) Cycle time of services on edge node with accurate cognitive AI application implemented and running. (KRIA) Safe wireless communication between nodes.	< 20 ms < 20 ms % telegram losses	Meassure duty cycle

Figure 107 - Task UC8_T3_2 traceability relationship between task, components and KPIs

7.6.4.3.3 Sub Task: UC8_T3_3 - Test extended functionalities (FRACTAL components)

Figure 108 shows traceability relationships for Task UC8_T3_3:

IN Comp	UT Com	KPI ID	KPI Description	Value
UC8_CMP_01 UC8_CMP_02 UC8_CMP_03		UC8_KPI_IP_04 UC8_KPI_IP_10	Predictions of AA - shuttle orchestrator compared to conventional solution (Versal - ARM) Model accuracy of the object detection	< 2 s > 95 %

KPI ID	KPI Description	Value	Test
UC8_KPI_FO_03	Self-sufficient decisions for each shuttle in respect to functional safety and additional degration steps. High accuracy in detection is required.	> 95 %	TBD
UC8_KPI_FO_04	Real-time inference for meta scheduler, which can react on various pre-defined events and make safe decisions for pathfinding and storage strategies for different goods.	< 2 s	
UC8_KPI_FO_05	Real-time inference for object detection on edge node with all services and accelerators implemented.	10 fps	

Figure 108 - Task UC8_T3_3 traceability relationship between task, components and KPIs

7.6.4.3.4 Sub Task: UC8_T3_4 - Test cloud services

Figure 109 shows traceability relationships for Task UC8_T3_4:

IN Comp	UT Com	KPI ID	KPI Description	Value
UC8_CMP_04		UC8_KPI_IP_14	Setup cloud service orchestrator success	True/ False

KPI ID	KPI Description	Value	Test
UC8_KPI_FO_06	Safe wireless communication between nodes.	% telegram losses	TBD

Figure 109 - Task UC8_T3_4 traceability relationship between task, components and KPIs

7.6.4.4 Task UC8_T4 - System Evaluation/ Benchmark

7.6.4.4.1 Sub Task: UC8_T4_1 - Metrics Calculation

Figure 110 shows traceability relationships for Task UC8_T4_1:

IN Comp	UT Com	KPI ID	KPI Description	Value
UC8_CMP_05		-	-	-

KPI ID	KPI Description	Value	Test
UC8_KPI_FO_00	Fractal technology helps to improve the state of the art in the intralogistics industry	True/False	TBD

Figure 110 - Task UC8_T4_1 traceability relationship between task, components and KPIs

7.7 Justification plan

7.7.1 KPI evaluation method

This section defines the justification methods (like demonstrations, tests, simulations, calculations, etc.) for KPIs evaluation, Use Case Requirements validation and Components validation.

7.7.1.1 KPI for Implementation Plan

7.7.1.1.1 UC8_KPI_IP_01

- ***Description***: All subtask success - Versal node
- ***Result type***: True/False
- ***Evaluation method***: The KPI is True when all subtasks under the task succeed.

7.7.1.1.2 UC8_KPI_IP_02

- **Description**: Duty cycle of control software (Versal) with target design
- **Result type**: < 20 ms
- **Evaluation method**: Measuring the duty cycle, after all FRACTAL components implemented.

7.7.1.1.3 UC8_KPI_IP_03

- **Description**: Build AA - shuttle orchestrator for target (Versal - ARM)
- **Result type**: True/ False
- **Evaluation method**: True if model successful build and deployed on Versal board.

7.7.1.1.4 UC8_KPI_IP_04

- **Description**: Inference time of predictions - shuttle orchestrator (Versal - ARM)
- **Result type**: < 2 s
- **Evaluation method**: KPI is True when the inference time of the orchestrator is below 2s.

7.7.1.1.5 UC8_KPI_IP_05

- **Description**: All subtask success - Kria node
- **Result type**: True/False
- **Evaluation method**: The KPI is True when all subtasks under the task succeed.

7.7.1.1.6 UC8_KPI_IP_06

- **Description**: Duty cycle of control software (Kria) with target design
- **Result type**: < 20 ms
- **Evaluation method**: Measuring the duty cycle, after all FRACTAL components implemented.

7.7.1.1.7 UC8_KPI_IP_07

- **Description**: Build OpenCV for target (Kria - ARM) success
- **Result type**: True/ False
- **Evaluation method**: True if successful build for target.

7.7.1.1.8 UC8_KPI_IP_08

- **Description**: Build demonstration software on target success
- **Result type**: True/ False
- **Evaluation method**: True if all implementation tasks successful.

7.7.1.1.9 UC8_KPI_IP_09

- **Description**: Build object detection application success
- **Result type**: True/ False
- **Evaluation method**: True if model was built successful for target.

7.7.1.1.10 UC8_KPI_IP_10

- **Description**: Model accuracy of the object detection
- **Result type**: > 95 %
- **Evaluation method**: If accuracy of trained model higher then result type, then success.

7.7.1.1.11 UC8_KPI_IP_11

- **Description**: Build zone evaluation logic application success
- **Result type**: True/ False
- **Evaluation method**: True if build and implementation on target successful.

7.7.1.1.12 UC8_KPI_IP_12

- **Description**: Inference time of object detection
- **Result type**: < 100 ms
- **Evaluation method**: If inference time of trained model better or equal to result type, then success.

7.7.1.1.13 UC8_KPI_IP_13

- **Description**: Failure rate of connection between FPGA and safety plc
- **Result type**: %
- **Evaluation method**: Diagnostic coverage > 99.9 %.

7.7.1.1.14 UC8_KPI_IP_14

- **Description**: Setup cloud service orchestrator success
- **Result type**: True/ False
- **Evaluation method**: True if setup successfully implemented in the test setup.

7.7.1.2 KPI for FRACTAL Objectives

7.7.1.2.1 UC8_KPI_FO_00

- **Description**: FRACTAL technology helps to improve the state of the art in the intralogistics industry
- **Result type**: True/False
- **Evaluation method**: After collecting of system metrics, the comparison between state-of-the-art concept and FRACTAL concept will be evaluated. Key indicators for the system will be throughput, MTTF/ MTBF per shuttle, MTTR per shuttle, reached PL of new safety concept and availability of the system in %.

7.7.1.2.2 UC8_KPI_FO_01

- **Description**: Cycle time of services on edge node with accelerated orchestrator implemented and running. (Versal)
- **Result type**: < 20 ms
- **Evaluation method**: Measuring the duty cycle, after all FRACTAL components implemented.

7.7.1.2.3 UC8_KPI_FO_02

- **Description**: Cycle time of services on edge node with accurate cognitive AI application implemented and running. (KRIA)
- **Result type**: < 20 ms
- **Evaluation method**: Measuring the duty cycle, after all FRACTAL components implemented.

7.7.1.2.4 UC8_KPI_FO_03

- **Description**: Self-sufficient decisions for each shuttle in respect to functional safety and additional degradation steps. High accuracy in detection is required.
- **Result type**: > 95 %
- **Evaluation method**: Accuracy of the object detection (person detection) will be crucial for the safety concept and must achieve in the target setup the highest possible accuracy.

7.7.1.2.5 UC8_KPI_FO_04

- **Description**: Real-time inference for meta scheduler, which can react on various pre-defined events and make safe decisions for pathfinding and storage strategies for different goods.
- **Result type**: < 2 s
- **Evaluation method**: The real-time inference of the adapted meta scheduler must be 2 s or lower, to be able to adapt to different scenarios, especially in bigger swarms.

7.7.1.2.6 UC8_KPI_FO_05

- **Description**: Real-time inference for object detection on edge node with all services and accelerators implemented.
- **Result type**: 10 fps
- **Evaluation method**: The real-time inference of the object detection model must achieve at 100 ms or faster, regarding the limited power source.

7.7.1.2.7 UC8_KPI_FO_06

- **Description**: Safe wireless communication between nodes.
- **Result type**: % telegram losses
- **Evaluation method**: Max. 5 telegrams per second are allowed for safety critical communication between edge nodes. Furthermore, the wireless communication must be also robust enough for model update processes from the cloud.

7.7.1.3 KPI for FRACTAL Features

7.7.1.3.1 UC8_KPI_FT_01 - Edge node has CAN Bus connectivity

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - EXTENSIBILITY - PORT CONNECTION - CAN BUS
- **Evaluation method**: Successful implementation of the CAN Bus interface on the FPGAs.

7.7.1.3.2 UC8_KPI_FT_02 - Edge node has AI/ ML accelerator

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - AI - HW - AI/ ML ACCELERATOR
- **Evaluation method**: Utilizing of AI accelerators for AI models.

7.7.1.3.3 UC8_KPI_FT_03 - Edge node is capable of real time applications and process camera streams in real-time

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - AI - SW - INFERENCE - REALTIME
- **Evaluation method**: True if successful implementation of object detection.

7.7.1.3.4 UC8_KPI_FT_04 - The AI model are located in the node

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - AI - SW - INFERENCE - LOCATION - NODE
- **Evaluation method**: True by default, as all AI models will be implemented in edge node.

7.7.1.3.5 UC8_KPI_FT_05 - The AI models will be prepared for the Versal platform

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - AI - SW - INFERENCE - MODEL - FORMAT - VERSAL
- **Evaluation method**: AI models will be prepared in Vitis AI for the Versal platform. True if successful deployment.

7.7.1.3.6 UC8_KPI_FT_06 - AI models will be trained in the cloud and then deployed on the node

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - AI - SW - LEARNING/ TRAINING - LOCATION - CLOUD
- **Evaluation method**: Supervised training and model management in the cloud. True if training in the cloud successful implemented.

7.7.1.3.7 UC8_KPI_FT_07 - AI models will be trained on a device and then deployed on the node

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - AI - SW - LEARNING/ TRAINING - LOCATION - OTHER
- **Evaluation method**: Supervised training on local computer before implementation of cloud will start. True if training successful.

7.7.1.3.8 UC8_KPI_FT_08 - The AI models use supervised learning for training

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - AI - SW - LEARNING/ TRAINING - PARADIGM - SUPERVISED LEARNING
- **Evaluation method**: True by default, as all models are planned for supervised learning.

7.7.1.3.9 UC8_KPI_FT_09 - Vitis is able to import and execute YOLO algorithms for KRIA platform

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - AI - SW - LEARNING/ TRAINING - ALGORITHMS - YOLO
- **Evaluation method**: True if Vitis AI can import YOLO algorithms.

7.7.1.3.10 UC8_KPI_FT_10 - Vitis is able to import and deploy convolutional neural networks for KRIA platform

- **Result type**: True/False

- **Helps to demonstrate UC Feature**: ADAPTABILITY - AI - SW - LEARNING/ TRAINING - ALGORITHMS - CNN
- **Evaluation method**: True if Vitis AI can import CNN models for the Kria platform.

7.7.1.3.11 UC8_KPI_FT_11 - Vitis is able to import and deploy artificial neural networks for Versal platform

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - AI - SW - LEARNING/ TRAINING - ALGORITHMS - ANN
- **Evaluation method**: True if Vitis AI can import ANN models for the Versal platform.

7.7.1.3.12 UC8_KPI_FT_12 - Vitis is able to import and deploy graph neural networks for Versal platform

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - AI - SW - LEARNING/ TRAINING - ALGORITHMS - GNN
- **Evaluation method**: True if Vitis AI can import GNN models for the Versal platform.

7.7.1.3.13 UC8_KPI_FT_13 - Edge node provides the library TensorFlow – Keras

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - AI - SW - LIBRARY - TENSORFLOW - KERAS
- **Evaluation method**: True if Vitis AI provides support for TensorFlow and Keras.

7.7.1.3.14 UC8_KPI_FT_14 - Edge node provides the library OpenCV

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - AI - SW - LIBRARY - OPENCV
- **Evaluation method**: True if Vitis AI provides support for OpenCV or branches of OpenCV.

7.7.1.3.15 UC8_KPI_FT_15 - Edge node provides the library NumPy

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - AI - SW - LIBRARY - NUMPY
- **Evaluation method**: True if Vitis AI provides support for NumPy.

7.7.1.3.16 UC8_KPI_FT_16 - Edge node provides the library PyTorch

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - AI - SW - LIBRARY - PYTORCH
- **Evaluation method**: True if Vitis AI provides support for PyTorch.

7.7.1.3.17 UC8_KPI_FT_17 - Service orchestration part of the fleet management system

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - SERVICES ORCHESTRATION
- **Evaluation method**: True if orchestration of services is available from the cloud.

7.7.1.3.18 UC8_KPI_FT_18 - Edge node adapts to various predefined scenarios

- **Result type**: detection time < 1 ms
- **Helps to demonstrate UC Feature**: ADAPTABILITY - OPERATION MODE CHANGE - METASCHEDULING - SYSTEM RECONFIGURATION
- **Evaluation method**: If HATMA performs a schedule switch when a predefined scenario occurs. How long takes the HATMA adaptation logic to detect the new scenario.

7.7.1.3.19 UC8_KPI_FT_19 - Edge node is fault tolerant

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: ADAPTABILITY - OPERATION MODE CHANGE - METASCHEDULING - FAULT TOLERANCE
- **Evaluation method**: True if system services are still supported in the presence of system faults.

7.7.1.3.20 UC8_KPI_FT_20 - Edge node adapts to required load level with different low power approaches

- **Result type**: >= 75 m
- **Helps to demonstrate UC Feature**: ADAPTABILITY - OPERATION MODE CHANGE - METASCHEDULING - LOW POWER
- **Evaluation method**: Optimization of functions and components on the edge node to accomplish at least 75 m from the designed 100 m of the ultracapacitor pack, regarding the additional components like the camera systems added during the FRACTAL project.

7.7.1.3.21 UC8_KPI_FT_21 - AI model for object detection have to be validated concerning the accuracy

- **Result type**: > 95 %
- **Helps to demonstrate UC Feature**: RELIABILITY - AI MODEL - ACCURACY / VALIDATION
- **Evaluation method**: True if object detection accuracy is higher than 95 %.

7.7.1.3.22 UC8_KPI_FT_22 - TT off chip comm. required for safe communication between the edge nodes

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: SAFETY - TIME TRIGGERED COMMUNICATION - OFF CHIP
- **Evaluation method**: True if safety critical communication between edge nodes is established.

7.7.1.3.23 UC8_KPI_FT_23 - TT on chip comm. required for safety monitoring the node level of an edge node

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: SAFETY - TIME TRIGGERED COMMUNICATION - ON CHIP
- **Evaluation method**: True if TTNoC is implemented on the FPGAs.

7.7.1.3.24 UC8_KPI_FT_24 - Safety service is required for evaluation of the object detection

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: SAFETY - MONITORING - CORES
- **Evaluation method**: True if meta scheduler is implemented on the FPGAs.

7.7.1.3.25 UC8_KPI_FT_25 - Self testing for the TTNOC on the edge

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: SAFETY - SELF TESTING - BUILT-IN SELF TEST ON ADAPTIVE TIME TRIGGERED NOC
- **Evaluation method**: True if TTNoC is implemented in the FPGAs, as it is a part of the TTNoC.

7.7.1.3.26 UC8_KPI_FT_26 - Scheduling services on node level to provide fail-safe operation

- **Result type**: True/False

- **Helps to demonstrate UC Feature**: SAFETY - FAIL-SAFE - SCHEDULING SERVICES
- **Evaluation method**: True if TTNoC is implemented on the FPGAs.

7.7.1.3.27 UC8_KPI_FT_27 - Safe wireless communication between nodes

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: SAFETY - REDUNDANCY - COMMUNICATION MESSAGES
- **Evaluation method**: True if TTNoC is implemented on the FPGAs.

7.7.1.3.28 UC8_KPI_FT_28 - Safety service is required for evaluation of the object detection

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: SAFETY - REALTIME AWARE - NON-INTERRUPTABLE PROCESSES
- **Evaluation method**: True if zone evaluation will be implemented as non-interruptible process.

7.7.1.3.29 UC8_KPI_FT_29 - Scheduling services on node level to provide fail-safe operation

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: SAFETY - REALTIME AWARE - HW FAILURE INTERRUPTS
- **Evaluation method**: True if TTNoC is implemented on the FPGAs.

7.7.1.3.30 UC8_KPI_FT_30 - Edge node must provide a degration level for processes

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: SAFETY - PROCESS SCHEDULING - SYSTEM DEGRADATION
- **Evaluation method**: True if HATMA is implemented on the Versal platform.

7.7.1.3.31 UC8_KPI_FT_31 - Safety Regulation ISO 61508 Generic

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: SAFETY - REGULATION - ISO 61508 - Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems
- **Evaluation method**: True if implementation of safety concept extension successful.

7.7.1.3.32 UC8_KPI_FT_32 - Part of the meta scheduling approach

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: LOW POWER - SCHEDULING SERVICES
- **Evaluation method**: True if meta scheduler is implemented on the Versal platform.

7.7.1.3.33 UC8_KPI_FT_33 - Battery level of the shuttle will be tracked for data collection

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: CONTEXT-AWARENESS - SENSORS - BATTERY LEVEL
- **Evaluation method**: True by default from shuttle control services.

7.7.1.3.34 UC8_KPI_FT_34 - Shuttle edge node requires cameras for environmental awareness

- **Result type**: 10 fps
- **Helps to demonstrate UC Feature**: CONTEXT-AWARENESS - SENSORS - CAMERA
- **Evaluation method**: True if hardware implementation of cameras on Kria platform successful.

7.7.1.3.35 UC8_KPI_FT_35 - Shuttle edge node utilizes sensors for positioning in the racking

- **Result type**: True/False

- **Helps to demonstrate UC Feature**: CONTEXT-AWARENESS - SENSORS - POSITION
- **Evaluation method**: True by default from shuttle and lift control services.

7.7.1.3.36 UC8_KPI_FT_36 - Shuttle edge node utilizes sensors for fine positioning to the totes

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: CONTEXT-AWARENESS - SENSOR NETWORK - RELATIVE POSITION
- **Evaluation method**: True by default from shuttle control services.

7.7.1.3.37 UC8_KPI_FT_37 - AI model for object detection via cameras for the shuttles

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: CONTEXT-AWARENESS - ACTIONS - OBJECT DETECTION
- **Evaluation method**: True if zone evaluation will be implemented successful.

7.7.1.3.38 UC8_KPI_FT_38 - AI model for object detection triggers on detection and generates an alarm

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: CONTEXT-AWARENESS - ACTIONS - AI TRIGGERED - ALARM
- **Evaluation method**: True if connection between object detection and zone evaluation successful.

7.7.1.3.39 UC8_KPI_FT_39 - Deployed design and models has to be verified during boot process

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: SECURITY - BOOT - FIRMWARE VERIFICATION
- **Evaluation method**: True by version controlling in the cloud and the verification on the edge nodes.

7.7.1.3.40 UC8_KPI_FT_40 - Connection to higher-level processes, such as the mfc or for downloading diagnose data

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: FRACTALITY - COMMUNICATION / CONNECTIVITY - TECHNOLOGIES - ETHERNET
- **Evaluation method**: True by hardware design.

7.7.1.3.41 UC8_KPI_FT_41 - Connection between nodes, Versal <--> Kria

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: FRACTALITY - COMMUNICATION / CONNECTIVITY - TECHNOLOGIES - WIFI
- **Evaluation method**: True by hardware design.

7.7.1.3.42 UC8_KPI_FT_42 - Data protocol between nodes will be MQTT

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: FRACTALITY - COMMUNICATION / CONNECTIVITY - DATA PROTOCOLS - MQTT
- **Evaluation method**: True if MQTT protocol is utilized between edge nodes.

7.7.1.3.43 UC8_KPI_FT_43 - Fleet management system service orchestration

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: FRACTALITY - ORCHESTRATION - SERVICES

- **Evaluation method**: True if service orchestrator can manage equipment control services.

7.7.1.3.44 UC8_KPI_FT_44 - Fleet management system data orchestration

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: FRACTALITY - ORCHESTRATION - DATA
- **Evaluation method**: True if service orchestrator can manage data sets and customer specific data.

7.7.1.3.45 UC8_KPI_FT_45 - Fleet management system model orchestration

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: FRACTALITY - ORCHESTRATION - MODEL
- **Evaluation method**: True if service orchestrator can manage and train AI models.

7.7.1.3.46 UC8_KPI_FT_46 - Hierarchical architecture on system level of the edge nodes

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: FRACTALITY - HIERARCHICAL ARCHITECTURE
- **Evaluation method**: True if hierarchical is established from cloud down to the Kria board.

7.7.1.3.47 UC8_KPI_FT_47 - Versal node will be implemented in the lift node

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: OTHER: NON-FUNCTIONAL - PLATFORM (SELECT ONE) - VERSAL - ARM
- **Evaluation method**: True if implementation of Versal edge node for the lifts is successful.

7.7.1.3.48 UC8_KPI_FT_48 - Kria node (Zynq Ultrascale + MPSoC) will be implemented in the shuttle nodes

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: OTHER: NON-FUNCTIONAL - PLATFORM (SELECT ONE) - ZYNQ ULTRASCALE+ (VERSAL ALTERNATIVE)
- **Evaluation method**: True if implementation of Kria edge nodes for the shuttles is successful.

7.7.1.3.49 UC8_KPI_FT_49 - Edge nodes execute a Linux OS

- **Result type**: True/False
- **Helps to demonstrate UC Feature**: OTHER: NON-FUNCTIONAL - OS - LINUX
- **Evaluation method**: True if control services are verified in its functionality.

7.7.2 Use Case Requirement Validation methods

Use case requirements validation methods are defined under the KPI defined for Use Case Requirements.

7.7.2.1.1 UC8_KPI_IP_Req_01

- **Description**: The edge node should have followed hardware specification:
  - at least 2 cores @ 800 MHz
  - at least 4 GB RAM
  - at least eMMC Memory or similar.
- **Result type**: True/False
- **Evaluation method**: True by development board properties.

7.7.2.1.2 UC8_KPI_IP_Req_02

- **Description**: These communication protocols shall be used from Linux OS:
  - MQTT over WiFi mesh network for communication between nodes
  - CAN Bus for internal communication.
- **Result type**: True/False
- **Evaluation method**: True if hardware design for Versal and Kria boards successful.

7.7.2.1.3 UC8_KPI_IP_Req_03

- **Description**: The edge node shall provide enough interfaces for two cameras.
- **Result type**: True/False
- **Evaluation method**: True by development board properties.

7.7.2.1.4 UC8_KPI_IP_Req_04

- **Description**: The edge node shall be capable to detect objects (human body and other obstacles) from video input stream of the provided cameras and evaluate the detected object to generate a safe output, if the obstacle is in a defined range of the shuttle.
- **Result type**: True/False
- **Evaluation method**: True if object detection and evaluation are implemented on the Kria edge node successful.

7.7.2.1.5 UC8_KPI_IP_Req_05

- **Description**: The edge node shall be able to use an adaptive orchestrator (scheduler) for storing strategies and optimized pathfinding for each shuttle depending on material (weight, type), frequency of requests, division of same type in different levels for alternative access/ faster access on big order amount.
- **Result type**: True/False
- **Evaluation method**: True if orchestrator is implemented on the Versal edge node successful.

7.7.2.1.6 UC8_KPI_IP_Req_06

- **Description**: The edge node shall offer optimized pathfinding: Improving path of the shuttles, for different scenarios; obstacle in same layer; malfunction of a shuttle; avoiding crossing in same level.
- **Result type**: True/False
- **Evaluation method**: True if orchestrator is implemented on the Versal edge node successful.

7.7.2.1.7 UC8_KPI_IP_Req_07

- **Description**: The node shall feature Linux operating system with real time capability (e.g., time-triggered communication capabilities).
- **Result type**: True/False
- **Evaluation method**: True by TTNoC implementation.

7.7.2.1.8 UC8_KPI_IP_Req_08

- **Description**: Safety wireless communication should be over a black channel (ASIL 3, ISO 26262) between nodes.
- **Result type**: True/False
- **Evaluation method**: True by TTNoC implementation and the realization of TT off chip communication capabilities.

7.7.2.1.9 UC8_KPI_IP_Req_09

- **Description**: For the edge nodes a cross compiler shall be available to port control software.
- **Result type**: True/False
- **Evaluation method**: True if control software builds for Versal and Kria are successful.

7.7.2.1.10 UC8_KPI_IP_Req_10

- **Description**: The edge node shall support libraries, like TensorFlow/ Keras.
- **Result type**: True/False
- **Evaluation method**: True if model implementation through Vitis AI successful.

7.7.3 Components Validation

Components used by the Use Case can be divided into two groups: *specific* components produced by the Use Case, and general *common* FRACTAL Components used by the Use Case.

The validation of Use Case Specific components is done through the corresponding KPIs. However, FRACTAL common components cannot be validated by just one UC, therefore, validation through this Use Case can be considered only as just a partial validation of the component.

7.7.3.1 Case Specific Components

7.7.3.1.1 UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)

Hardware preparation of FPGAs regarding hardware interfaces for successful migration of control services from SBC to the edge nodes. Validation is done by successfully executing the tests of the following KPIs (see the corresponding KPI for details of the test):

- UC8_KPI_IP_02 - Duty cycle of control software (Versal) with target design - < 20 ms
- UC8_KPI_IP_06 - Duty cycle of control software (Kria) with target design - < 20 ms
- UC8_KPI_FO_01 - Cycle time of services on edge node with accelerated orchestrator implemented and running. (Versal) - < 20 ms
- UC8_KPI_FO_02 - Cycle time of services on edge node with accurate cognitive AI application implemented and running. (KRIA) - < 20 ms
- UC8_KPI_FO_06 - Safe wireless communication between nodes. - % telegram losses

7.7.3.1.2 UC8_CMP_02 - Evaluation of object detection

Safety critical zone evaluation in combination with object detection based on a neural network for Kria platform. The connection to existing safety PLCs is crucial for successful implementation. Validation is done by successfully executing the tests of the following KPIs (see the corresponding KPI for details of the test):

- UC8_KPI_IP_07 - Build OpenCV for target (Kria - ARM) success - True/ False
- UC8_KPI_IP_09 - Build object detection application success - True/ False
- UC8_KPI_IP_10 - Model accuracy of the object detection - > 95 %
- UC8_KPI_IP_11 - Build zone evaluation logic application success - True/ False
- UC8_KPI_IP_12 - Inference time of object detection - < 100 ms
- UC8_KPI_IP_13 - Failure rate of connection between FPGA and safety plc - %

- UC8_KPI_FO_03 - Self-sufficient decisions for each shuttle in respect to functional safety and additional degradation steps. High accuracy in detection is required. - > 95 %
- UC8_KPI_FO_05 - Real-time inference for object detection on edge node with all services and accelerators implemented. - 10 fps

7.7.3.1.3 UC8_CMP_03 - AI accelerated orchestrator/ scheduler

Orchestrator for material handling equipment. In case of the test setup, this means two shuttles and two lifts, which are provided with tasks. Optimization will be done for three different storage scenarios: Weight flow optimization, priority flag sorting and obstacle avoidance. Pathfinding will be realized by ant colony optimization. Validation is done by successfully executing the tests of the following KPIs (see the corresponding KPI for details of the test):

- UC8_KPI_IP_03 - Build AA - shuttle orchestrator for target (Versal - ARM) - True/ False
- UC8_KPI_IP_04 - Inference time of predictions - shuttle orchestrator (Versal - ARM) - < 2 s
- UC8_KPI_FO_04 - Real-time inference for meta scheduler, which can react on various pre-defined events and make safe decisions for pathfinding and storage strategies for different goods. - < 2 s

7.7.3.1.4 UC8_CMP_04 - Cloud service orchestration

Implementation of fleet management system in the test setup to manage control services, AI models and data. Validation is done by successfully executing the tests of the following KPIs (see the corresponding KPI for details of the test):

- UC8_KPI_IP_14 - Setup cloud service orchestrator success - True/ False
- UC8_KPI_FO_06 - Safe wireless communication between nodes. - % telegram losses

7.7.3.1.5 UC8_CMP_05 - Integrated demonstration software on target

Merging and preparation for test setup integration of elaborated components. Validation is done by successfully executing the tests of the following KPIs (see the corresponding KPI for details of the test):

- UC8_KPI_IP_08 - Build demonstration software on target success - True/ False

Consists of:

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)
- UC8_CMP_02 - Evaluation of object detection
- UC8_CMP_03 - AI accelerated orchestrator/ scheduler
- UC8_CMP_04 - Cloud service orchestration

7.7.3.2 FRACTAL Common Components

7.7.3.2.1 WP3T32-10 - Versal accelerator building-blocks

This component consists of the development of building-blocks for accelerators for Versal. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)

7.7.3.2.2 WP3T34-03 - Versal Model deployment layer

This component consists of the model deployment on the Versal APU + DPU control from model repository images. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)
- UC8_CMP_02 - Evaluation of object detection
- UC8_CMP_03 - AI accelerated orchestrator/ scheduler

7.7.3.2.3 WP4T41-02 – HATMA

This component consists of the Hierarchical Adaptive Time-triggered Multi-core Architecture to facilitate services at the different hierarchies. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)

7.7.3.2.4 WP4T41-04 - Versal RPU access for Power Services

This component consists of a component to access dynamic power, frequency scaling features on Versal [VCK190]. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)
- UC8_CMP_03 - AI accelerated orchestrator/ scheduler

7.7.3.2.5 WP4T41-05 - Agreement protocol for Low-Power Services

This component consists of the implementation for the agreement protocol on a wireless network on low-power devices. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)

7.7.3.2.6 WP4T41-06 - Versal Isolation Design - Functional Safety

This component consists of enhancing the common Versal platform to strictly separate functional accesses, services from underlying HW access. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)

7.7.3.2.7 WP4T42-02 - Versal RPU access to AI acceleration

This component consists of the enhance RPU libraries to (1) access APU based AI as a service, (2) enable local AI [acceleration] deployment from RPU. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific*

components that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)
- UC8_CMP_02 - Evaluation of object detection
- UC8_CMP_03 - AI accelerated orchestrator/ scheduler

7.7.3.2.8 WP4T42-03 - Scenario Generator

This component consists of the Scenario Generator that provides the inputs for the machine learning algorithm. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)
- UC8_CMP_03 - AI accelerated orchestrator/ scheduler

7.7.3.2.9 WP4T42-04 - GA-Scheduler

This component consists of the scheduler (Genetic Algorithm) that provides the solutions of the scheduling problems given by the Scenario Generator component. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)
- UC8_CMP_03 - AI accelerated orchestrator/ scheduler

7.7.3.2.10 WP4T42-05 - AI-Scheduler Model

This component consists of the machine learning model used to predict schedules. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)
- UC8_CMP_03 - AI accelerated orchestrator/ scheduler

7.7.3.2.11 WP4T42-06 - Schedule Verifier

This component consists of the Schedule verifier/ reconstructor that takes the predictions of the machine learning model and convert them into a schedule. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)
- UC8_CMP_03 - AI accelerated orchestrator/ scheduler

7.7.3.2.12 WP4T42-07 - Hierarchical Meta scheduler

This component consists of the offline tool to compute time-triggered schedules by considering context events such as dynamic slack, failure scenarios. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific*

components that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)
- UC8_CMP_03 - AI accelerated orchestrator/ scheduler

7.7.3.2.13 WP4T43-04 – ATTNoC

This component consists of the Adaptive TTNoC, which provides time triggered communication for NoC and allow the systems to switch schedules in case of any failures occurs in the NoC. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)
- UC8_CMP_02 - Evaluation of object detection
- UC8_CMP_03 - AI accelerated orchestrator/ scheduler

7.7.3.2.14 WP4T43-06 - FPGA Fault-injector

This component consists of a tool to inject faults in the NOEL-V multicore. It is suitable for any Ultrascale+ FPGA. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (KRIA)

7.7.3.2.15 WP4T43-08 - Seamless redundancy for ATTNoC

This component consists of the seamless redundancy, which provides fault tolerance on the NoC by sending two set of seamless data with seamless path at the same time, so failures in one path can be masked. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)
- UC8_CMP_02 - Evaluation of object detection
- UC8_CMP_03 - AI accelerated orchestrator/ scheduler

7.7.3.2.16 WP4T43-11 - Time-Triggered Extension Layer for Versal NoC

This component consists of a time-triggered extension layer that is an extension layer developed for Versal NoC that allow the Versal NoC to transfer messages using time triggered traffic. Partial validation is done by successfully executing the tests of the KPIs of the Use Case specific components that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)
- UC8_CMP_02 - Evaluation of object detection
- UC8_CMP_03 - AI accelerated orchestrator/ scheduler

7.7.3.2.17 WP4T43-13 - Safety Analysis

Safety concept by performing a risk analysis within the scope of the concept phase of ISO 61508 by application of DIN EN ISO 3691-4 (item definition, risk assessment and functional safety concept) on the system, in context of VAL_UC8. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)
- UC8_CMP_02 - Evaluation of object detection

7.7.3.2.18 WP4T44-02 - OS Security Layer

This component consists of an Implementation of security countermeasures in a transversal security layer. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_01 - Hardware design with CAN Bus connectivity (Versal and KRIA)
- UC8_CMP_05 - Integrated demonstration software on target

7.7.3.2.19 WP5T52-04-05 - LakeFS deployment and configuration

This component consists of the dataset version control repository. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_04 - Cloud service orchestration

7.7.3.2.20 WP5T52-04-07 - Images repository

This component consists of the container Registry for Docker Images. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_04 - Cloud service orchestration

7.7.3.2.21 WP5T52-05-02 - Airflow deployment and configuration

This component consists in cloud deployment of Airflow and configure its integration with other services for their orchestration. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_04 - Cloud service orchestration

7.7.3.2.22 WP5T52-06-01 - Model preparation for FRACTAL Edge (Versal Xilinx Vitis AI)

This component consists of the Workflows to compile models for Versal with Xilinx Vitis AI, add containerized toolchain to the cloud. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_04 - Cloud service orchestration

7.7.3.2.23 WP5T54-01-01 - MLBuffet

This component consists of the Machine Learning tool for model serving and management. Deployable in containers with Swarm and Kubernetes. API for managing models and sending input/Outputs. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_04 - Cloud service orchestration

7.7.3.2.24 WP5T54-02-02 – Kubernetes

This component consists of an Open-Source orchestrator for cluster management and container orchestration. *Partial* validation is done by successfully executing the tests of the KPIs of the *Use Case specific components* that use this common component (see the corresponding KPI for details of the test)

- UC8_CMP_04 - Cloud service orchestration

8 Conclusions

This document has presented the four Use Cases for the Industrial Validation of FRACTAL, guiding the WP8 tasks that will be started next: T8.3 "Case Study Implementation" and T8.4 "Case Study Justification File".

In this sense, each Use Case has been presented and described in the context of FRACTAL, along with an exploration of the state of the art in order to understand how to compare FRACTAL UC implementation with existing solutions. In D8.2 "System Requirement" a benchmark will be defined for that scope.

The Use Cases have also explored the implementation plan, defining implementation steps and how to evaluate implementation progress. Also, a set of KPIs has been defined related to implementation tasks, FRACTAL objectives and Use Case specific characteristics.

Finally, a justification plan has been presented that defines how the Use Case will be evaluated in the following terms:

- Implementation progress;
- Use Case Requirements;
- FRACTAL Objectives;
- FRACTAL Features;
- FRACTAL Components.

The results from the execution of the justification plan for the four Use Cases will be collected in D8.3 "Evaluation Result", which will also expand the evaluation methods that have been shortly described in this document.

9 Bibliography

[1] *https://shift2rail.org/.*

[2] *https://projects.shift2rail.org/s2r_ipx_n.aspx?p=tauro.*

[3] *https://www.europeanfiles.eu/climate/shifting-to-europes-rail.*

[4] *Ritika, S., Mittal, S., and Rao, D., "Railway track specific traffic signal selection using deep learning." arXiv preprint arXiv:1712.06107 (2017).*

[5] *Choodowicz, E., Lisiecki, P., and Lech, P., "Hybrid Algorithm for the Detection and Recognition of Railway Signs." International Conference on Computer Recognition Systems. Springer, Cham, 2019..*

[6] *Karagiannis, G., Olsen, S., and Pedersen, K., "Deep Learning for Detection of Railway Signs and Signals." Science and Information Conference. Springer, Cham, 2019..*

[7] *Haseeb, M. A., Guan, J., Ristic-Durrant, D., and Gräser, A., "DisNet: ¨ a novel method for distance estimation from monocular camera." 10th Planning, Perception and Navigation for Intelligent Vehicles (PPNIV18), IROS (2018)..*

[8] *Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012..*

[9] *Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... and Murphy, K., "Speed/accuracy trade-offs for modern convolutional object detectors." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017..*

[10] *Zeiler, M. D., and Fergus, R., "Visualizing and understanding convolutional networks." European conference on computer vision. Springer, Cham, 2014..*

[11] *Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... and Rabinovich, A., "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015..*

[12] *Simonyan, K., and Zisserman, A., "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014)..*

[13] *He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016..*

[14] *Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., and Sun, J., "Detnet: A backbone network for object detection." arXiv preprint arXiv:1804.06215 (2018)..*

[15] *R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014..*

[16] *Girshick, R., "Fast r-cnn." Proceedings of the IEEE international conference on computer vision. 2015..*

[17] *] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016..*

[18] *Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., and Berg, A. C. "SSD: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016..*

[19] *Lin, T. Y., Goyal, P., Girshick, R., He, K., and Dollar, P., "Focal ´ loss for dense object detection." Proceedings of the IEEE international conference on computer vision. 2017..*

[20] *J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 6517–6525. IEEE, 2017..*

[21] *Redmon, J., and Farhadi, A., "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767. 2018..*

[22] *Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection". Cornell University. https://doi.org/10.48550/arXiv.2004.10934.*

[23] *Sen Wang et al. "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks". In: 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE. 2017, pp. 2043–2050..*

[24] *Raul Mur-Artal and Juan D Tardós. "Orb-slam2: An opensource slam system for monocular, stereo, and rgb-d cameras". In: IEEE Transactions on Robotics 33.5 (2017), pp. 1255–1262..*

[25] *Muhammad Abdul Haseeb et al. "DisNet: A novel method for distance estimation from monocular camera". In: 10th Planning, Perception and Navigation for Intelligent Vehicles (PPNIV18), IROS (2018)..*

[26] *FPGA vs. GPU vs. CPU – hardware options for AI applications. https://www.avnet.com/wps/portal/silica/resources/article/fpga-vs-gpu-vs-cpu-hardware-options-for-ai-applications/.*

[27] *] Farhad Fallahlalehzari. FPGA vs GPU for Machine Learning Applications: Which one is better? https://www.aldec.com/en/company/blog/167--fpgas-vs-gpus-for-machine-learning-applications-which-one-is-better.*

[28] *NVIDIA Xavier Achieves Industry First with Expert Safety Assessment - https://blogs.nvidia.com/blog/2020/05/20/xavier-achieves-industry-first-safety-assessment/.*

[29] *Object Detection Metrics. https://towardsdatascience.com/on-object-detection-metrics-with-worked-example-216f173ed31e.*

[30] *Dehaene, Michiel. 2008. Heterotopia and the City: Public Space in a Postcivil Society. 1st ed. Routledge. https:// doi. org/10.4324/9780203089415..*

[31] *Andrade, L., Quintero, J., Gamess, E., & Russoniello, A. (2018). A Proposal for a Technological Solution to Improve user Experience in a Shopping Center based on Indoor Geolocation Services.,* INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS, 9(6), 389-401.

[32] *https://trendwatching.com/quarterly/2019-02/future-experiences-asia/#sentient-spaces.*

[33] *https://www.businessmodelsinc.com/sentient-spaces-are-shaping-the-future-of-retail/.*

[34] *Hore, Bijit, et al. "Design and implementation of a middleware for sentient spaces." 2007 IEEE Intelligence and Security Informatics. IEEE, 2007.*

[35] *https://www.smartmediaworld.net/products/interactive-kiosks.html.*

[36] *https://www.sita.aero/innovation/sita-lab/intelligent-machines/kate-sitas-autonomous-robot.*

[37] *https://www.intel.it/content/www/it/it/internet-of-things/iot-solutions/kiosk/overview.html.*

[38] *https://www.zebra.com/it/it/products/interactive-kiosks.html.*

[39] *Gales, M.J.F., Knill, K.M., Ragni, A. et al., "Speech recognition and keyword spotting for low-resource languages : Babel project research at CUED.", Fourth International Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU-2014).,* 14-16 May 2014, St. Petersburg, Russia, pp. 16-23.

[40] *Y. Permanasari, E.H. Harahap, and E. Prayoga Ali, "Speech recognition using Dynamic Time Warping (DTW)", 2nd International Conference on Applied & Industrial Mathematics and Statistics 23–25 July 2019, Kuantan, Pahang, Malaysia.*

[41] *E. Singer, P. A. Torres-Carrasquillo, T. P. Gleason, W. M. Campbell and D. A. Reynolds, "Acoustic phonetic and discriminative approaches to automatic language recognition", Proc. Eurospeech, pp. 1345-1348, 2003.*

[42] *M. A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech", IEEE Trans. Speech Audio Process., vol. 4, no. 1, pp. 31-44, Jan. 1996.*

[43] *A. Gupta and A. Joshi, "Speech Recognition Using Artificial Neural Network," 2018 International Conference on Communication and Signal Processing (ICCSP), 2018, pp. 0068-0071, doi: 10.1109/ICCSP.2018.8524333.*

[44] *Sadanandam, M. & Prasad, V. & Janaki, V. & Nagesh, Akkaladevi. (2012). Text independent language recognition system using DHMM with new features. 511-514. 10.1109/ICoSP.2012.6491537.*

[45] *P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds and J. R. Deller, "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features", Proc. ICSLP, pp. 89-92, 2002.*

[46] *W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition", Comput. Speech Lang., vol. 20, pp. 210-229, 2006.*

[47] *B. Ma, H. Li and R. Tong, "Spoken Language Recognition Using Ensemble Classifiers," in IEEE Transactions on Audio, Speech, and Language Processing, vol. 15, no. 7, pp. 2053-2062, Sept. 2007, doi: 10.1109/TASL.2007.902861.*

[48] *https://www.v2c2.at/spider/.*

[49] *Quigley, Morgan, et al. "ROS: an open-source Robot Operating System." ICRA workshop on open source software. Vol. 3. No. 3.2. 2009.*

[50] *https://www.ros.org/.*

[51] *Rusu, Radu Bogdan, and Steve Cousins. "3d is here: Point cloud library (pcl)." 2011 IEEE international conference on robotics and automation. IEEE, 2011.*

[52] *Hoffmann, G. M., Tomlin, C. J., Montemerlo, M., & Thrun, S. (2007, July). Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. In 2007 American control conference (pp. 2296-2301). IEEE.*

[53] *Gutiérrez, R., López-Guillén, E., Bergasa, L. M., Barea, R., Pérez, Ó., Gómez-Huélamo, C., ... & López-Fernández, J. (2020). A waypoint tracking controller for autonomous road vehicles using ROS framework. Sensors, 20(14), 4062.*

[54] *Perez-Leon, H., Acevedo, J. J., Millan-Romera, J. A., Castillejo-Calle, A., Maza, I., & Ollero, A. (2019, November). An aerial robot path follower based on the 'carrot chasing'algorithm. In Iberian Robotics conference (pp. 37-47). Springer, Cham.*

[55] *Samuel, M., Hussein, M., & Mohamad, M. B. (2016). A review of some pure-pursuit based path tracking techniques for control of autonomous vehicle. International Journal of Computer Applications, 135(1), 35-38.*

[56] *Hoffmann, G. M., Tomlin, C. J., Montemerlo, M., & Thrun, S. (2007, July). Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. In 2007 American control conference (pp. 2296-2301). IEEE.*

[57] *Rostami, S. M. H., Sangaiah, A. K., Wang, J., & Liu, X. (2019). Obstacle avoidance of mobile robots using modified artificial potential field algorithm. EURASIP Journal on Wireless Communications and Networking, 2019(1), 1-19.*

[58] *Wiig, M. S., Pettersen, K. Y., & Krogstad, T. R. (2018, August). A 3D reactive collision avoidance algorithm for nonholonomic vehicles. In 2018 IEEE Conference on Control Technology and Applications (CCTA) (pp. 67-74). IEEE.*

[59] *Leca, D., Cadenat, V., & Sentenac, T. (2019, September). Sensor-based algorithm for collision-free avoidance of mobile robots in complex dynamic environments. In 2019 European Conference on Mobile Robots (ECMR) (pp. 1-6). IEEE.*

[60] *https://keras.io/.*

[61] *https://www.tensorflow.org/.*

[62] *Schwarzl, Christian, Gerhard Greiner, and Paolo Pretto. "Testing of Safety-critical ADAS Functions with Mobile HiL Platforms." ATZ worldwide 122.10 (2020): 36-41.*

[63] *IEC 61508: 2010 - "Functional safety of electrical/electronic/programmable electronic (E/E/PE) safety-related systems" (7 parts),.*

[64] *ISO 26262:2018 – "Road vehicles — Functional safety" (12 Parts).*

[65] *ISO/FDIS 21448:2022 – "Road vehicles — Safety of the intended functionality".*

[66] *ISO/TR 4804:2020 – "Road vehicles — Safety and cybersecurity for automated driving systems — Design, verification and validation",.*

[67] *ISO/SAE PAS 22736:2021 – "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles".*

[68] *ISO/AWI TS 5083 (not released) – "Road vehicles — Safety for automated driving systems — Design, verification and validation",.*

[69] *Koscher, Karl, et al. "Experimental security analysis of a modern automobile." 2010 IEEE symposium on security and privacy. IEEE, 2010.*

[70] *ISO/SAE. Road Vehicles – Cybersecurity engineering. Standard ISO/SAE 21434:2021. Geneva, CH, 2021. url: https://www.iso.org/standard/70918.html.*

[71] *Russell, S., & Norvig, P. (2002). Artificial intelligence: a modern approach.*

[72] *Nilsson, N. J. (2009). The quest for artificial intelligence. Cambridge University Press.*

[73] *Tabernik, D., & Skočaj, D. (2019). Deep learning for large-scale traffic-sign detection and recognition. IEEE transactions on intelligent transportation systems, 21(4), 1427-1440.*

[74] *Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255-260.*

[75] *Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., & Pérez, P. (2021). Deep reinforcement learning for autonomous driving: A survey. IEEE Transactions on Intelligent Transportation Systems.*

[76] *Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., ... & Zieba, K. (2016). End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316.*

[77] *Taylor, J., Yudkowsky, E., LaVictoire, P., & Critch, A. (2016). Alignment for advanced machine learning systems. Ethics of Artificial Intelligence, 342-382.*

[78] *Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. arXiv preprint arXiv:1606.06565.*

[79] *https://www.selene-project.eu/.*

[80] *https://github.com/siemens/isar-riscv.*

[81] *https://github.com/onnx/keras-onnx.*

[82] *https://gazebosim.org/home.*

[83] *https://index.ros.org/p/demo_nodes_cpp/.*

[84] *https://docs.ros.org/en/rolling/Concepts/About-Quality-of-Service-Settings.html.*

[85] *https://gcc.gnu.org/onlinedocs/gcc/Gcov.html.*

[86] *https://docs.ros.org/en/rolling/Concepts/About-Quality-of-Service-Settings.html.*

[87] *https://github.com/ros2/rosbag2.*

[88] *https://www.digitalocean.com/community/tutorials/how-to-use-top-netstat-du-other-tools-to-monitor-server-resources.*

[89] *https://docs.ros.org/en/rolling/Tutorials.html.*

[90] *https://www.it-production.com/automation-und-robotik/schwarmintelligenz-knapp-ag/, Accessed: 22.07.2022.*

[91] M. De Ryck, M. Versteyhe and F. Debrouwere, "Automated Guided Vehicle Systems, State-Of-The-Art Control Algorithms and Techniques," *Journal of Manufacturing Systems,* vol. 54, pp. 152-173, 2020.

[92] D. Yingying, H. Yan e J. Jingping, «Multi-Robot Cooperation Method Based On The Ant Algorithm,» *Proceedings of the 2003 IEEE Swarm Intelligence Symposium,* pp. 14-18, 2003.

10 List of figures

11 List of tables

12 List of Abbreviations

AI	Artificial Intelligent
AGV	Automated Guided Vehicle
AP	Average Precision
API	Application Programming Interface
APU	Application Processing Unit
AS/RS	Automated Storage and Retrieval System
ATO	Automatic Train Operation
ATP	Automatic Train Protection
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CV	Computer Vision
DDR RAM	Double Date Rate Random Access Memory
DeepVO	Deep Visual Odometry
DoA	Document of Agreement
DPU	Data Processing Unit
DTO	Driverless Train Operation
EDDL	European Distributed Deep Learning
ERP	Enterprise Resource Planning
FDIS	Final Draft International Standard
FPGA	Field Programmable Gate Array
FPS	Frames Per Second
GoA	Grade of Automation
GPU	Graphics Processing Unit
HW	Hardware
IEC	International Electrotechnical Commission
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
IoU	Intersection over Union
ISO	International Organization for Standardization
KPI	Key Performance Indicator
mAP	Mean Average Precision
MFC	Material Flow Controller
ML	Machine Learning
MPSoC	MultiProcessor System on a Chip
MQTT	Message Queuing Telemetry Transport
NFS	Network File System
NoC	Network on Chip
ONNX	Open Neural Network Exchange
OpenCV	Open Computer Vision
ORB-SLAM2	Open-Source Simultaneous Localization And Mapping System for Monocular, Stereo and RGB-Cameras
OS	Operating System
PER	Perception

RGB	Red Green Black
RISC	Reduced Instruction Set Computer
ROS	Robot Operating System
RTSP	Real Time Streaming Protocol
SBC	Single Board Computer
SGBM	Semi Global Block Matching
SW	Software
TBD	To Be Defined
TT	Time Triggered
UC	Use Case
UDP	User Data Protocol
WCS	Warehouse Control System
WMS	Warehouse Management System
WP	Work Package
Yolo	You Only Look Once