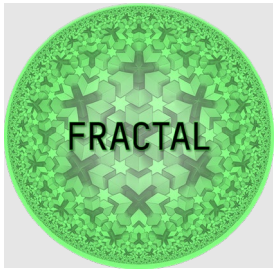

D2.3 Platform Specification (b)

Deliverable Id:	**D2.3**
Deliverable name:	**Platform Specification (b)**
Status:	**Final**
Dissemination level:	**Public**
Due date of deliverable:	**2022-06-30 (M22)**
Actual submission date:	**2022-07-08**
Work package:	**WP2 "Specifications & Methodology"**
Organization name of lead contractor for this deliverable:	Thales Research & Technology with the support of LKS who coordinated the appendices.
Authors:	Ikerlan: Carles Estrada BSC: Jaume Abella, Ramon Canal UPV: Carles Hernández, Ilya Tuzov Solver IA: Leticia Pascual ZYLK: Alfonso González, Ester Sola LKS Next: Iñaki Paz Rulex Innovation Labs: Enrico Ferrari Aitek: Stefano Delucchi, Jacopo Motta Università degli Studi dell'Aquila: Tania Di Mascio, Giacomo Valente, Luigi Pomante Modis Consulting: Pietro Abbatangelo, Marco Cappella, Luca Visconti Università degli Studi di Modena e Reggio Emilia: Gianluca Brilli Ro Technology: Nadia Caterina Zullo Lasala, Amal Alrish, Damiano Vallocchia AVL: Boris Bulatovic, Tomislav Bukić, Harisyam Manda, Bernhard Peischl Siemens: Bekim Chilku Virtual Vehicle Research: Markus Postl University of Siegen: Daniel Onwuchekwa BEEWEN Automation: Artur Kaufmann Thales: Jérôme Quévremont ACP: Michael Gautschi

Project FRACTAL

Title Platform Specification (b)

Del. Code D2.3

	University of Oulu: Tuomas Paso, Susanna Pirttikangas, Mickaël Bettinelli, Vahid Mohsseni, Nanna Setämaa, Konstantin Mikhaylov, Jukka Riekki, Lauri Lovén Haltian: Matti Vakkuri PLC2: Alexander Flick
Reviewers:	Iñaki Paz Rey, LKS Antti Takaluoma, OFFC

Abstract:

D2.3 "Platform Specification (b)" provides a holistic approach of the FRACTAL platforms by describing them as a collection of use case requirements, FRACTAL features and components.

These three notions are combined through some traceability artefacts.

The assembly of components building the FRACTAL node provides a "big picture" of the project.

Contents

Acknowledgement

ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 877056. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Italy, Austria, Germany, Finland and Switzerland.

1 History

Version	Date	Modification reason	Modified by
0.1	2022-05-12	Draft version, empty appendices	Thales
0.2	2022-05-30	Draft version, empty appendices, reviewed	Reviewers
0.3	2022-06-27	Draft version, empty appendices, review remarks addressed	Thales
0.4	2022-07-04	Completed appendices from Excel sheets	Thales, LKS
0.5	2022-07-08	Complete version, reviewed	Reviewers
1.0	2022-07-08	Final cleanup, delivered version	Thales

Table 1 – Document history

This document has been edited online. The Microsoft Sharepoint solution has been selected to keep information under EU legislation. This solution offers a reduced feature set compared to a "regular" Word editor. For instance, we have not been able to build a table of references and have instead used footnotes.

The document was prepared with two streams. The main part is a Word document coordinated by Thales. The appendices containing the use case requirements, FRACTAL features and components are a snapshot of the Excel workbook maintained by FRACTAL partners under LKS's coordination.

2 Summary

This D2.3 "Platform Specification (b)" deliverable provides a holistic approach of the FRACTAL platforms by describing them as a collection of use case requirements, FRACTAL features and components.

These three notions are combined through some traceability artefacts, identifying the features used by the various use cases on one hand and the features fulfilled by the components on the other hand.

Finally, the assembly of components to build the FRACTAL node is provided as a "big picture" of the project.

D2.3 is not an update of D2.1 "Platform specification (b)" but a complement which helps to strengthen the project holistic view, following recommendations of the first period review.

3 Introduction

Artificial Intelligence is key for the IoT to enhance existing services and to operate in a more efficient manner. If AI is not implemented in the IoT, its scope is very much limited. Cognitivity, provided by Artificial Intelligence, will support the IoT to adapt to surrounding world changes, learning in real-time to improve its performance.

The goal of the FRACTAL project is to create a basic platform called the FRACTAL node. It is a reliable computing platform node able to build a Cognitive Edge (a network that makes predictions and diagnoses) under industry standards. The FRACTAL node will be the building block of scalable decentralized Internet of Things (ranging from Smart Low-Energy Computing Systems to High-Performance Computing Edge Nodes).

FRACTAL is driven by its eight use cases that represent different classes of applications (Transport & Mobility, Digital Life, Digital Industry and Energy). From these eight use cases, the characteristics of building blocks of the FRACTAL node were identified in D2.1 "Platform Specification (a)" in a top-down fashion.

This D2.3 "Platform Specification (b)" deliverable adopts a more consistent view by describing FRACTAL as

- Requirements needed by use cases
- Features offered by the FRACTAL platforms
- Components included in FRACTAL platforms to deliver these features

and providing some traceability between them:

- The features used by the various use cases
- The features fulfilled by the components.

Finally, the assembly of components to build the FRACTAL node is provided as a "big picture" of the project.

As the project has progressed, D2.3 was not built as an update of D2.1, but as a complement to strengthen FRACTAL holistic view and bridge some gaps identified during the first period review.

In the following sections, the notions of *use case requirements*, *FRACTAL features* and *components* are defined. Their traceability is then explained. Finally, the document lists the requirements, features and components in a tabular fashion.

4 Requirements, features and components

4.1 Definitions

4.1.1 Use case requirements

Use case requirements are the functional and non-functional needs as they have been captured for FRACTAL's eight use cases:

- VER-UC1: Improving the quality of engineering and maintenance works through drones
- VER-UC2: Improving the quality of automotive air control
- VER-UC3: Smart meters for everyone
- VER-UC4: Low-latency Object Detection as a generic building block for perception in the edge for Industry 4.0 applications
- VAL-UC5: Increasing the safety of an autonomous train through AI techniques
- VAL-UC6: Elaborate data collected using heterogeneous technologies (intelligent totem)
- VAL-UC7: Autonomous robot for implementing safe movements
- VAL-UC8: Improve the performance of autonomous warehouse shuttles for moving goods in a warehouse

All requirements are numbered; the numbering includes the use case number. If several use cases have the same need, a similar requirement is created for each use case.

The list of requirements is in appendix A. All requirements relate to a use case; there are no requirements that do not relate to a use case.

4.1.2 FRACTAL features

The *FRACTAL feature* notion was introduced in section "7.1 Operation Integration" of deliverable D2.2 "Methodological Framework (a)".

There exist many definitions of *features* in the community[1]. In this sense, a FRACTAL feature is a distinguishing characteristic of FRACTAL, visible to users that will configure Fractal for their use cases. The concept of "feature" allows a consistent abstraction to be employed when making choices from a whole FRACTAL product configuration all the way down to the deployment of components within a low-level subsystem. It also enables to address the variability introduced by the UCs to build a product family.

Features are normally described in a hierarchical characteristic and/or tree, normally called Feature Diagram or Feature Model. In this sense, the following diagram presents the top-level features that describe FRACTAL as presented in previous documents. The figures in appendix B gives the complete hierarchy of FRACTAL features.

[1] Kang, Kyo & Cohen, Sholom & Hess, James & Novak, William & Peterson, A. (1990). Feature-Oriented Domain Analysis (FODA) feasibility study.

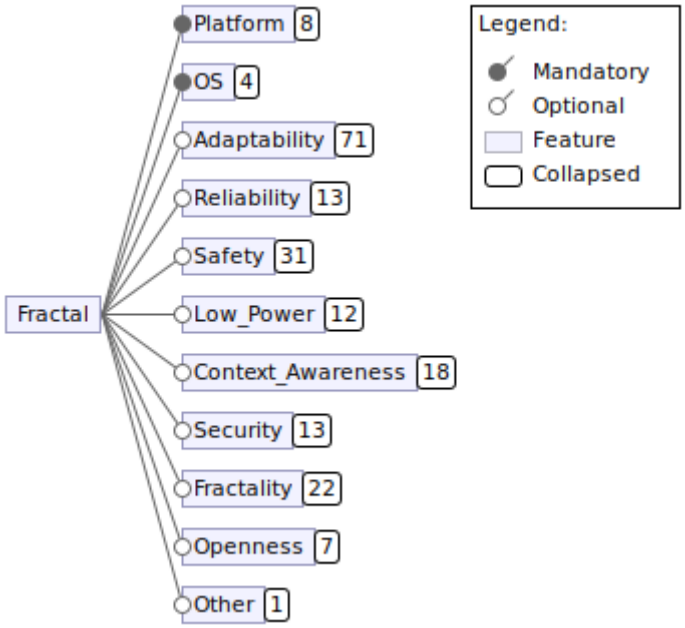

Figure 1 – FRACTAL top-level features

In the context of the UCs, some features are defined as optional since they are not addressed in all the use cases. Features may also present restrictions:

- In certain cases, only a child feature might be selected (e.g., a use case will only use one platform: PULP, CVA6, VERSAL…)
- Certain features might restrict variability on other features (e.g., the selection of an AI learning paradigm will restrict the possible AI libraries that can be selected). The specification of this type of restriction is still a work in progress.

The actual list of features implemented in each use case is called a "bill-of-features".

The list of features can be seen as the datasheet of FRACTAL platforms. It is a technical notion that can be used to promote FRACTAL platforms in the dissemination and exploitation activities.

With a view on the future exploitation of FRACTAL building blocks and platforms, their providers can decide to include features not required by a use case (e.g., a higher processing power, an additional interface). These extra features are captured in the feature list, not as requirements (unlike in D2.1).

4.1.3 Components

The components are the building blocks of the FRACTAL platforms needed to fulfill FRACTAL features. They can be made of software or hardware and are developed in the "technical" work packages (WP3, 4, 5 and 6).

4.2 Traceability

In the first period ECSEL review, the experts have recommended to "better analyze and demonstrate the requirements derivation and allocation process across all FRACTAL layers" and recommended to keep a holistic approach.

Two traceability levels have therefore been built.

4.2.1 Use case requirements / FRACTAL features traceability

For each use case, its list of actual FRACTAL features it uses has been identified in appendix C. These use case specific lists of features are called "bill-of-features".

4.2.2 Components / FRACTAL features traceability

For each component, the FRACTAL feature(s) it fulfills is(are) identified in a specific column in the list of components in appendix D.

4.3 Lists of Requirements, FRACTAL Features and Components

For legibility reasons, the lists of requirements, FRACTAL features and components are in the appendices below.

These lists are snapshots of the online Excel sheets used by the project partners to ensure the consistency and relevance of their developments. These lists include the aforementioned traceability.

The snapshot was taken on 2022-07-04, after an internal review for consistency and a face-to-face consortium meeting.

4.4 What happened to D2.1?

The *FRACTAL feature* notion was introduced in section "7.1 Operation Integration" of deliverable D2.2 "Methodological Framework (a)". This notion was not yet used when D2.1 "Platform Specification (a)" was prepared.

Because of the methodology streamlining introduced by the *FRACTAL feature* notion, D2.3 was not written as an update of D2.1 but as a complement.

Therefore, the interested reader may wonder if anything from D2.1 was lost. The table below summarizes where the D2.1 information is now.

In D2.1	What happened to them?
Use case descriptions	The description of use cases was not copied to D2.3. It can be found in "Description of Action (part B)", D2.1, D7.1 and D8.1.

In D2.1	What happened to them?
Use case requirements	The requirements of this report are an updated version of those captured in the use case sections of D2.1.
Chapters related to "technical" work packages: "5 Safety, Security & Low Power Techniques" (WP4) "6 AI and safe autonomous decision" (WP5) "7 Mutable and fractal communications" (WP6) "8 Node architecture and building blocks" (WP3)	Up to date technical descriptions can be found in the related WP deliverables: D3.1 to D3.6, D4.1, D4.2, D5.1, D6.1. The capabilities, offered by WP3 to 6, were often summarized as requirements in D2.1 sections 5 to 8. This notion of requirement is not valid anymore for them and most have been translated to FRACTAL features or components.
KPIs	KPIs related to use cases are tracked in WP7 and 8 deliverables.

Table 2 – What happened to D2.1?

In a nutshell, D2.1 was a great place for FRACTAL partners to gather their contributions and better articulate them at the beginning of the project. The structured capture of requirements, FRACTAL features and components performed across WP3 to 8 (and captured in this document) allowed to build the holistic approach that was missing in D2.1.

5 FRACTAL "Big Picture"

During the February 2022 technical workshop, FRACTAL partners have prepared a "big picture" to illustrate the assembly of the components to build the FRACTAL node. The picture below shows the "main" big picture, with all possible components. The various platforms (Versal, Pulpissimo, Ariane/CVA6, Noel-V) that implement the FRACTAL node will actually drop some components that are not used by the use cases they serve.

One can read the picture in a bottom-up way, starting with the hardware part of the platforms, and the low-level software layers (OS, services, drivers...). On top of them, the various layers of the edge application software are integrated. Finally, this node communicates with its cloud counterpart that includes for instance learning and orchestration.

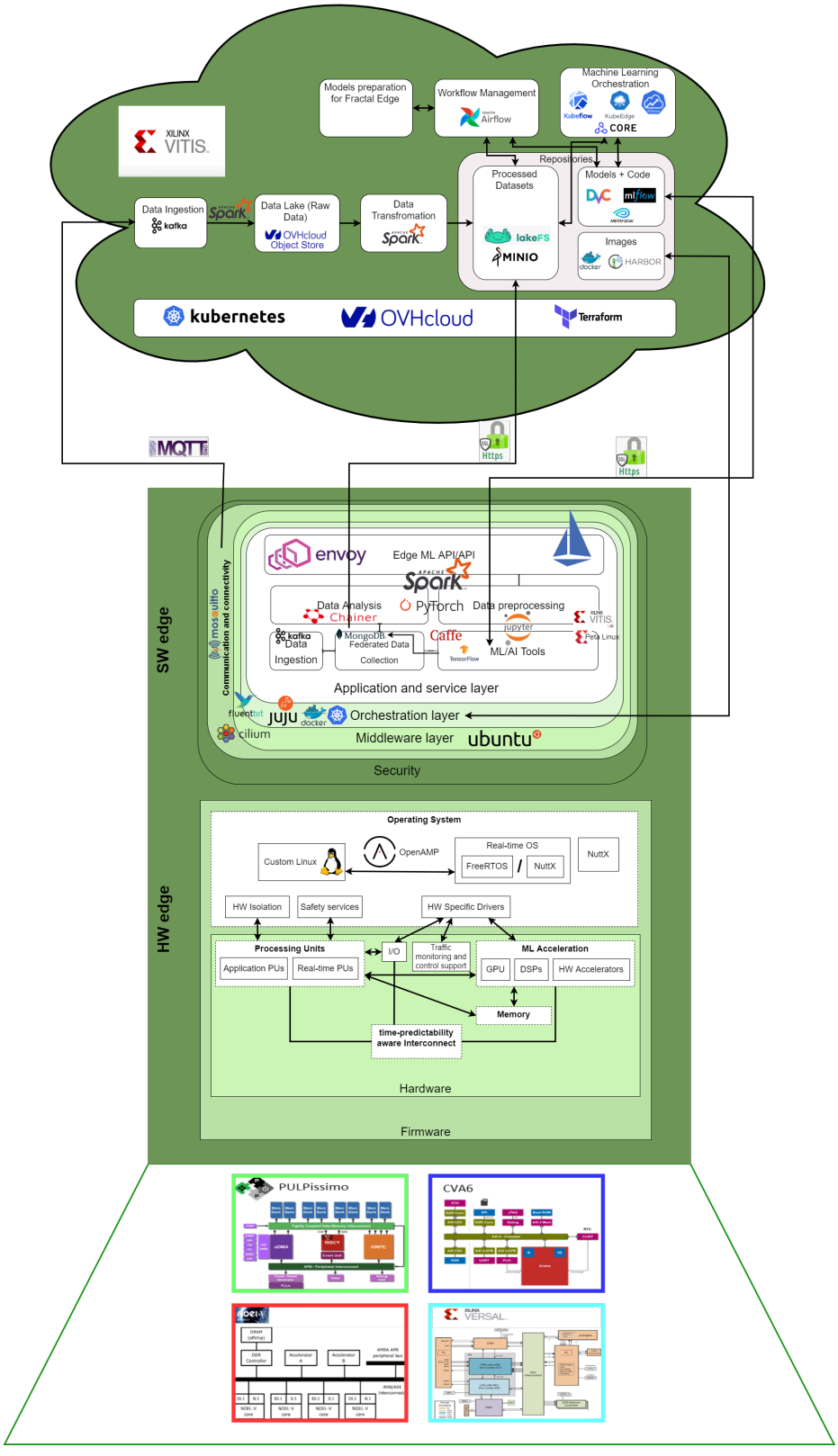

Figure 2 – FRACTAL "big picture"

From this "big picture", an integration terminology can be introduced to distinguish between various scopes in the FRACTAL project:

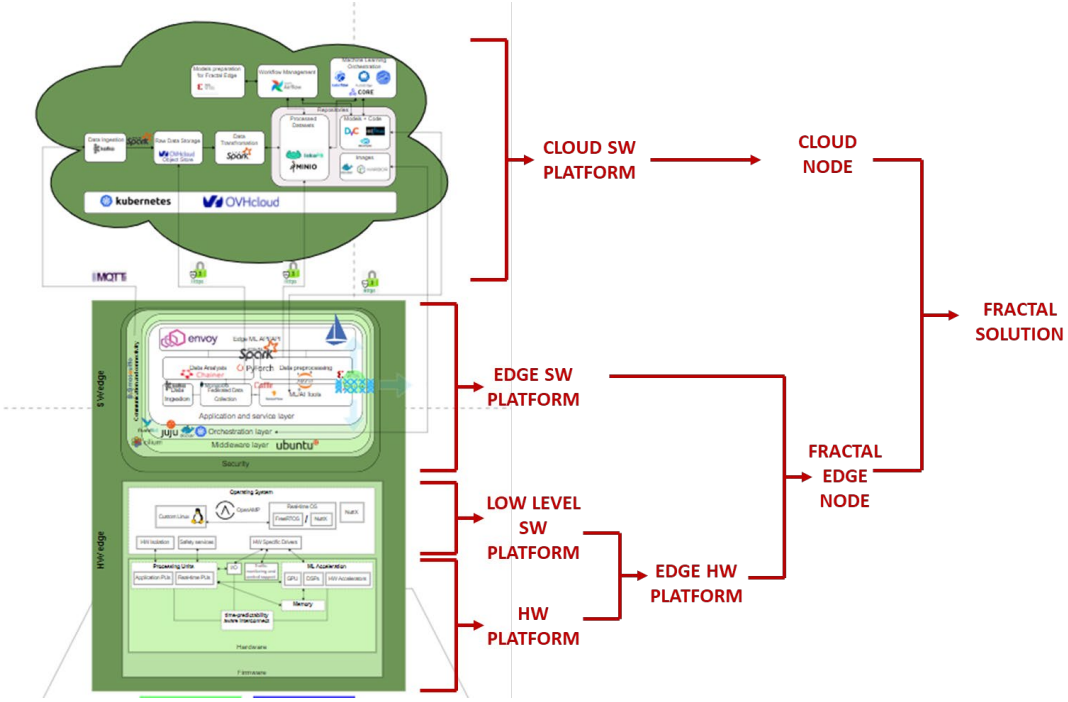

Figure 3 – FRACTAL integration terminology

6 Conclusions

This deliverable identifies the use case requirements, FRACTAL features and components and articulates together in a holistic way.

Compared to D2.1 "Platform Specification (a)", it adds the notion of FRACTAL features and a structured traceability between requirements, features and components to build the FRACTAL platform.

This activity was positively influenced by the feedback provided by ECSEL experts at the first period review.

7 List of Figures

8 List of Tables

9 List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
APU	Application Processor Unit
ARM	Advanced RISC Machines
ASIL	Automotive Safety Integrity Level
AXI	Advanced eXtensible Interface
CAFFE	Convolutional Architecture for Fast Feature Embedding
CAN	Controller Area Network
CD	Continuous Delivery
CENELEC	Comité Européen de Normalisation Électrotechnique (European Committee for Electrotechnical Standardization)
CI	Continuous Integration
CLAHE	Contrast Limited Adaptative Histogram Equalization
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DDR	Double Data Rate memory
DIN	Deutsche Industrie Normen
DMA	Direct Memory Access
DNN	Deep Neural Network
DPU	Deep-learning Processing Unit
DVC	Dataset Version Control
EN	European Norm
ETH	Ethernet
EU	European Union
FPGA	Field-Programmable Gate Array
GA	Grant Agreement
GB	Gigabyte
GDPR	General Data Protection Regulation
GFLOPS	Giga Floating Operations per Second
GNN	Genetic Neural Network
GPU	Graphics Processing Unit
HARA	Hazard and Risk Analysis
HATMA	Hierarchical Adaptive Time-triggered Multi-core Architecture
HD	High Definition
HW	Hardware
Hz	Hertz
ID	Identification
IO	Input/Output
IP	Internet Protocol

ISO	International Organization for Standardization
JBDC	Java Database Connectivity
JSON	JavaScript Object Notation
JU	Joint Undertaking
LEDEL	Low Energy DEep Learning Library (LEDEL)
LIDAR	Light Detection And Ranging
LZW	Lempel-Ziv-Welch
MB	Megabyte
ML	Machine Learning
MMU	Memory Management Unit
MOP	Million of Operations
MQTT	Message Queuing Telemetry Transport
NB-IoT	Narrow Band IoT
NN	Neural Network
NumPy	Numeric Python
OBDC	Open Database Connectivity
OLM	Operator Lifecycle Manager
ONNX	Open Neural Network Exchange
OpenCV	Open-Source Computer Vision Library
OS	Operating system
OTA	Over The Air
PULP	Parallel Ultra Low Power
RAM	Random Access Memory
REST	Representational State Transfer
RISC	Reduced Instruction Set Computer
RNN	Recurrent Neural Network
RPU	Real-Time Processing Unit
RTOS	Real Time Operating System
SAE	Society of Automotive Engineers
SQL	Structured Query Language
SSL	Secure Socket Layer
SW	Software
TARA	Threat Analysis and Risk Assessment
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TSN	Time Sensitive Networking
UAV	Unmanned Aerial Vehicle
UC	Use case
UDP	User Datagram Protocol
UNE	Asociación Española de Normalización
USB	Universal Serial Bus
VAL	Validation
VER	Verification

WiFi	Wireless Fidelity
WP	Work Package
WS	Web Service
XAI	Explainable Artificial Intelligence
YOLO	You Only Look Once
ZCU	Zonal Control Unit

The short names of FRACTAL partners are not considered as abbreviations: ACP, AITEK, AVL, BEE, BSC, CAF, ETH, HALTIAN, IKER, LKS, MODIS, OFFC, PLC2, PROINTEC, QUA, ROT, RULEX, SIEG, SIEM, SML, THA, UNIGE, UNIMORE, UNIVAQ, UOULU, UPV, VIF, ZYLK.

Appendix A: Use case requirements

Requirement ID	Description	Priority
	UC1: Engineering and maintenance works (2 demonstrators)	
	Demonstrator 1: UAV supervision of critical structures	
REQ_UC1_DEM1_01	The edge node shall be able to collect and pre-process images at 2 fps rate.	High
REQ_UC1_DEM1_02	The edge node models shall perform real-time inference on data (2 fps rate).	High
REQ_UC1_DEM1_03	The edge node shall be able to collect and analyze a video stream.	High
REQ_UC1_DEM1_04	The edge node will run a single instance of the model and will perform inference on single instances of the images.	Medium
REQ_UC1_DEM1_05	The edge node shall work over 64bit Linux OS.	Medium
REQ_UC1_DEM1_06	The edge node shall be able to run Docker containers.	Low
REQ_UC1_DEM1_07	The edge node shall be able to run Python code (TensorFlow, OpenCV and Numpy).	High
REQ_UC1_DEM1_08	The edge node shall have some communication capabilities with the cloud.	Low
	Demonstrator 2: Wireless Sensor Network for safety at construction sites	
REQ_UC1_DEM2_01	The edge node models shall perform real-time inference on data.	High
REQ_UC1_DEM2_02	The edge node shall gather and pre-process data for real-time inference.	High
REQ_UC1_DEM2_03	The edge node shall be able to act in case of Alarm detection/prediction.	High
REQ_UC1_DEM2_04	The edge node should be able to reallocate tasks distributedly.	Medium
REQ_UC1_DEM2_05	The edge node might run multiple ML models simultaneously.	Low
REQ_UC1_DEM2_06	The edge node shall work over 64bit Linux OS.	High
REQ_UC1_DEM2_07	The edge node shall be able to run Docker containers.	High
REQ_UC1_DEM2_08	The edge node shall allow communication with the sensor network (Bluetooth, WiFi, others...) and with different data sources, directly or indirectly (WS, API, REST, JBDC, OBDC).	High
REQ_UC1_DEM2_09	The edge node shall be able to run Python3 code.	High
REQ_UC1_DEM2_10	The edge node should support training (via FPGA/GPU).	Medium
REQ_UC1_DEM2_11	The edge node should support Tensorflow/Keras/TensorRT/TFLite SW Stack.	Low
REQ_UC1_DEM2_12	The edge node should be able to form an inter-nodal network (and have network capabilities).	Low
REQ_UC1_DEM2_13	The edge node might have a communication latency lower than 2 seconds.	Low
	UC2: Improving the quality of automotive air control	
REQ_UC2_01	The edge node shall be able to execute TensorFlow framework models.	High
REQ_UC2_02	The edge node shall be capable of communicating with cloud services for federated learning as well as diagnosis and adaptation.	High
REQ_UC2_03	The edge node shall be capable of receiving over the air updates (OTA) for adaptation of the models after the development phase.	High
REQ_UC2_04	The edge node shall have CAN communication ports, to be able to communicate with the Engine Control Unit and other nodes within the vehicle.	High
REQ_UC2_05	The edge node shall be capable of communicating via Ethernet, to be able to communicate with additional sensors (e.g. information from the multimedia systems, etc.) and other nodes within the vehicle.	High
REQ_UC2_06	The edge node shall have a non-volatile memory of 20 MB or more, to store information also after the engine is switched off (e.g. storage of adaption values, ageing parameters, etc.).	High
REQ_UC2_07	The edge node shall have a storage media (e.g. solid state disc) of 200B or more, for metadata or offline communication (e.g. driving in a tunnel).	High
REQ_UC2_08	The inputs to the edge node will be very transient, therefore, the node shall be able to compute new actuator positions every 10 ms in real time.	Low

Requirement ID	Description	Priority
REQ_UC2_09	Good performant math libraries shall be available (e.g., possibility to apply filters, or perform simple aggregations like moving-averages, etc.), with a deviation from actual values be up to 0.001%.	Medium
REQ_UC2_10	The edge node shall have the possibility of parallel processing, with at least 4 cores.	Low
REQ_UC2_11	The edge node shall have at least 16GB RAM.	Medium
REQ_UC2_12	The edge node shall have an uninterruptible power supply.	Medium
REQ_UC2_13	The edge node shall have an internal voltage transformer.	Low
REQ_UC2_14	The edge node shall have Linux OS.	Medium
REQ_UC2_15	A C++ compiler shall be available to compile code for the edge node.	Medium
REQ_UC2_16	The edge node shall be robust to operate up to 100°C.	High
REQ_UC2_17	The edge node shall have a housing that protects it against dust and spray water.	High
REQ_UC2_18	The edge node shall be robust against vibration.	High
REQ_UC2_19	The edge node shall have the capacity to process time-series input-data at a 10 Hz update rate.	Medium
	UC3: Smart meters for everyone	
REQ_UC3_01	The edge node shall have a low static power consumption (<10uW) to enable a long battery lifetime.	High
REQ_UC3_02	The edge node shall be able to encrypt data.	High
REQ_UC3_03	The edge node shall support a secure boot process.	High
REQ_UC3_04	The edge node shall be able to establish a connection to the cloud through cellular services.	High
REQ_UC3_05	The edge node shall have the following minimum hardware characteristics: - non-volatile storage - ~512kB on chip memory - ~100 MOP/s.	High
REQ_UC3_06	The edge node shall be able to extract a number from images.	High
REQ_UC3_07	When integrated in a microchip, the edge node shall be smaller than 5cm^2.	High
REQ_UC3_08	The edge node shall be able to interface low power cameras through common interfaces such as SPI, I2C.	High
	UC4: Low-latency Object Detection (Industry 4.0)	
REQ_UC4_01	The edge node shall be able to detect 5 objects/cell in the input video stream.	High
REQ_UC4_02	The edge node shall be able to classify the objects in the input video stream.	High
REQ_UC4_03	The edge node shall be able to localize the objects in the input video stream.	High
REQ_UC4_04	The edge node shall be able to run the Darknet framework.	High
REQ_UC4_05	The edge node hardware shall consist of RISC-V CPU and specialized HW accelerator.	High
REQ_UC4_06	The edge node shall provide dedicated HW accelerator to process the CNN layers (Yolo) of AI inference.	High
REQ_UC4_07	The edge node shall provide special SW to control (driver) the operation of HW accelerator.	High
REQ_UC4_08	The edge node shall be able to control the data movement between main memory and HW accelerator through the DMA.	High
REQ_UC4_09	The edge node shall provide network service for communication between edge node and Fractal cloud.	Medium
REQ_UC4_10	The Fractal cloud shall provide support for training of the vision-based Neural Network.	Medium
REQ_UC4_11	The edge node shall perform all object detection operations on a single edge node.	High
REQ_UC4_12	The edge node shall be able to process the input video stream at rate > 5fps.	Low
REQ_UC4_13	The edge node processor shall run 64-bit Linux operating system.	High

Requirement ID	Description	Priority
REQ_UC4_14	The edge node shall support compiler for C/C++ source code.	High
REQ_UC4_15	The edge node should support compiler for Python source code.	Low
	UC5: Increasing the safety of an autonomous train	
REQ_UC5_01	The edge node shall support OpenCV library.	High
REQ_UC5_02	The edge node shall have ONNX Interpreter.	High
REQ_UC5_03	The HW accelerators might be compatible with Tensorflow.	Low
REQ_UC5_04	The edge node shall provide multi-core technology with at least 4 CPU cores.	Medium
REQ_UC5_05	The edge node shall handle multi-threading applications.	Medium
REQ_UC5_06	The edge node shall have at least 60 GFLOPS.	Medium
REQ_UC5_07	The edge node should provide at least 16GB DDR RAM.	Low
REQ_UC5_08	The edge node shall incorporate HW Acceleration.	High
REQ_UC5_09	The edge node shall incorporate IO interfaces: 2xGbit Eth, 2xUSB 3.0, 1xHDMI (and their Linux Drivers).	Low
REQ_UC5_10	The edge node shall have Linux OS.	High
REQ_UC5_11	The edge node release shall have C++ compiler/ cross-compiler toolchain.	High
REQ_UC5_12	The edge node should be compliant with non-functional railway equipment requirements described in EN50155, EN50125, EN45545, EN50121, UNE EN 61373: temperature, Humidity, Vibration, Electromagnetic Compatibility, Fire and Electrical Protection.	Low
REQ_UC5_13	The edge node shall support real-time performance for UC5 functionality providing processing capability of 100ms/10fps.	High
	UC6: Smart totem	
REQ_UC6_01	The edge node shall be able to execute TensorFlow-Keras framework models and/or ONNX (https://onnx.ai/).	High
REQ_UC6_02	The edge node shall OpenCV Library.	High
REQ_UC6_03	The edge node shall detect user face.	High
REQ_UC6_04	The edge node shall detect user age.	High
REQ_UC6_05	The edge node shall detect user gender.	High
REQ_UC6_06	The edge node shall detect people at totem proximity.	High
REQ_UC6_07	The edge node shall count people, or track people density, in totem proximity.	Medium
REQ_UC6_08	The edge node should detect crowd intensity and variation.	Low
REQ_UC6_09	The edge node shall detect speaker Language.	High
REQ_UC6_10	The edge node shall select content/info to be provided.	High
REQ_UC6_11	The edge node shall select the output channel among those available (e.g., video, audio, etc.).	Medium
REQ_UC6_12	The edge node shall support TCP/IP protocol; the ideal network protocol to transport messages among the devices shall be MQTT or any other publish/subscribe communication protocol.	High
REQ_UC6_13	The edge node shall expose a set of APIs which shall allow HTTPs REST calls to and from other nodes, central application and user devices.	High
REQ_UC6_14	The edge node shall acquire images from at least one HD camera.	High
REQ_UC6_15	The edge node shall acquire audio signal from at least one microphone.	High
REQ_UC6_16	The edge node shall support programmable accelerator engines, such as for instance FPGA/programmable logics, or AI engines (e.g., VERSAL's), or in case, GPGPUs.	High
REQ_UC6_17	The edge node shall have Linux OS such as Ubuntu or Petalinux.	High
REQ_UC6_18	The edge node shall have a C++ compiler and related standard libraries.	High

Requirement ID	Description	Priority
REQ_UC6_19	The edge node shall support wired connectivity (e.g., Ethernet) in order to ensure network stability, it shall have at least 1Gbit Ethernet connection.	High
REQ_UC6_20	The edge node shall support wireless connectivity (e.g., Wi-Fi).	Medium
REQ_UC6_21	The edge node shall have a hardware computing node that allows accelerating convolutional neural networks applications.	High
REQ_UC6_22	The edge node should control an interactive touchscreen display.	Medium
REQ_UC6_23	The edge node should control an audio speaker.	Medium
REQ_UC6_24	The edge node shall have a monitoring system able to measure response time of tasks, both implemented on microprocessors and accelerators.	High
REQ_UC6_25	The edge node shall be able to share the computational workload with other nodes when necessary.	High
REQ_UC6_26	The edge node shall guarantee that the user’s thoughts to remain uninterrupted is not more then 1.0 second	High
REQ_UC6_27	The edge node should have a modular and scalable architecture to allow an easy and quick integration of new data sources without changing the architecture.	Medium
REQ_UC6_28	The edge node shall store data locally in a secure manner.	Medium
	UC7: Autonomous robot	
REQ_UC7_01	The edge node shall calculate nearest object distance from a cost map.	Medium
REQ_UC7_02	The edge node shall initiate emergency brake if detected object in inside defined area of the costmap.	Medium
REQ_UC7_03	The edge node shall calculate optimal trajectory based on the planned path and obstacles.	Medium
REQ_UC7_04	The edge node shall monitor metrics to initiate an emergency brake at connection loss or timeout.	Medium
REQ_UC7_05	The edge node shall be able to process real-time cost map data of UC7 at a rate up to 50 Hz.	High
REQ_UC7_06	The edge node shall provide an Ethernet connection with TCP/UDP stack.	High
REQ_UC7_07	The edge node shall provide libraries for ROS2 support.	High
REQ_UC7_08	The edge node shall provide a node for diverse redundancy to avoid common cause faults.	Low
REQ_UC7_09	The edge node shall support LEDEL library.	Medium
REQ_UC7_10	The edge node should support calculation of maximum timeframe for 10 Hz processing loop-rates.	Low
REQ_UC7_11	The edge node shall provide a hardware accelerator supporting NN of UC7.	Low
	UC8: Warehouse shuttles	
REQ_UC8_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.	Medium
REQ_UC8_02	These communication protocols shall be used from Linux OS: - MQTT over WiFi mesh network for communication between nodes - CAN Bus for internal communication.	High
REQ_UC8_03	The edge node shall provide enough interfaces for two cameras.	High
REQ_UC8_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.	High
REQ_UC8_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.	High
REQ_UC8_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.	High
REQ_UC8_07	The node shall feature Linux operating system with real time capability (e.g. time-triggered communication capabilities).	High
REQ_UC8_08	Safety wireless communication should be over a black channel (ASIL 3, ISO 26262) between nodes.	Low
REQ_UC8_09	For the edge nodes a cross compiler shall be available to port control software.	High

Requirement ID	Description	Priority
REQ_UC8_10	The edge node shall support libraries, like Tensorflow/ Keras.	High

Table 3 – List of requirements

Appendix B: FRACTAL Features

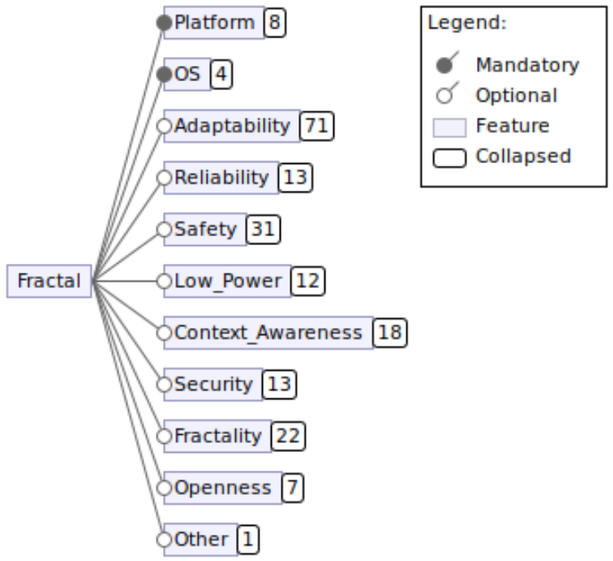

Figure 4 – FRACTAL top-level features

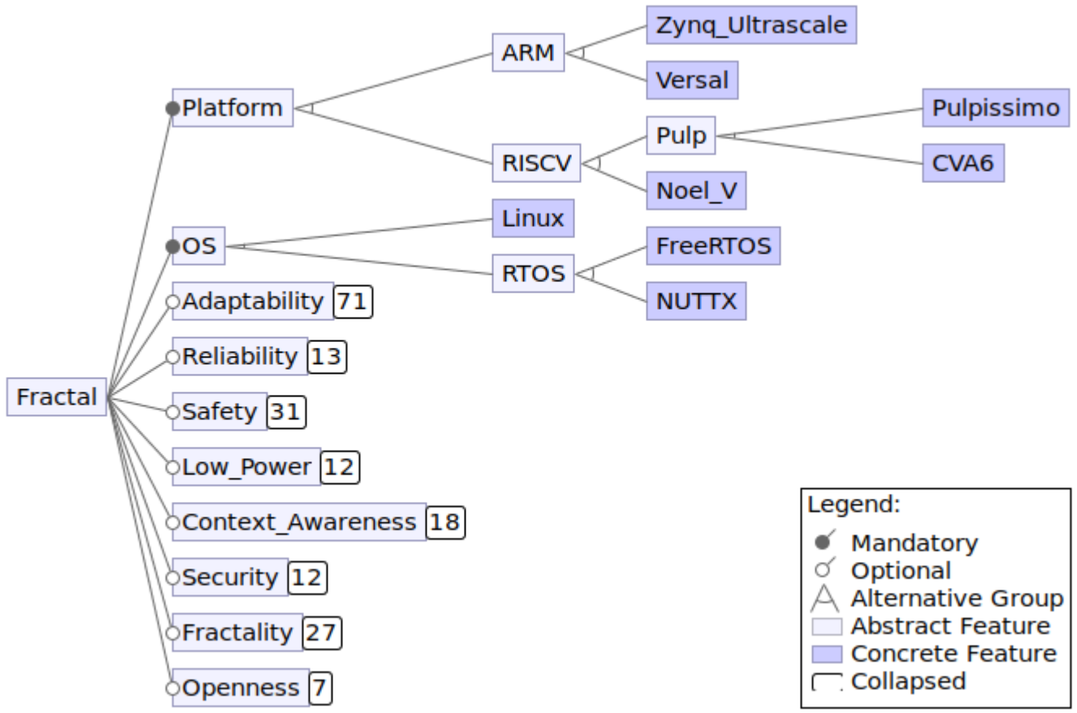

Figure 5 – Platform and OS features

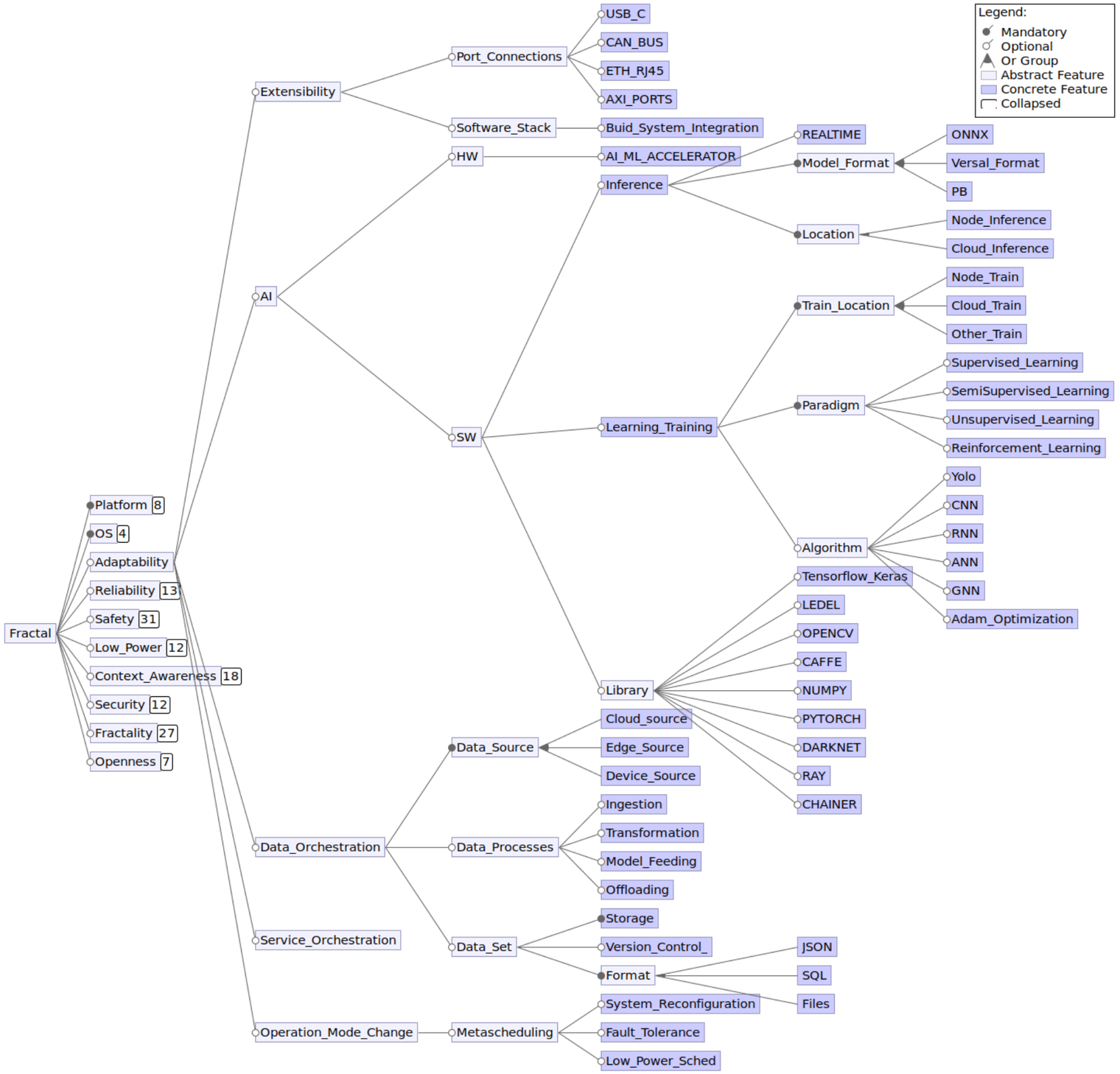

Figure 6 – Adaptability features

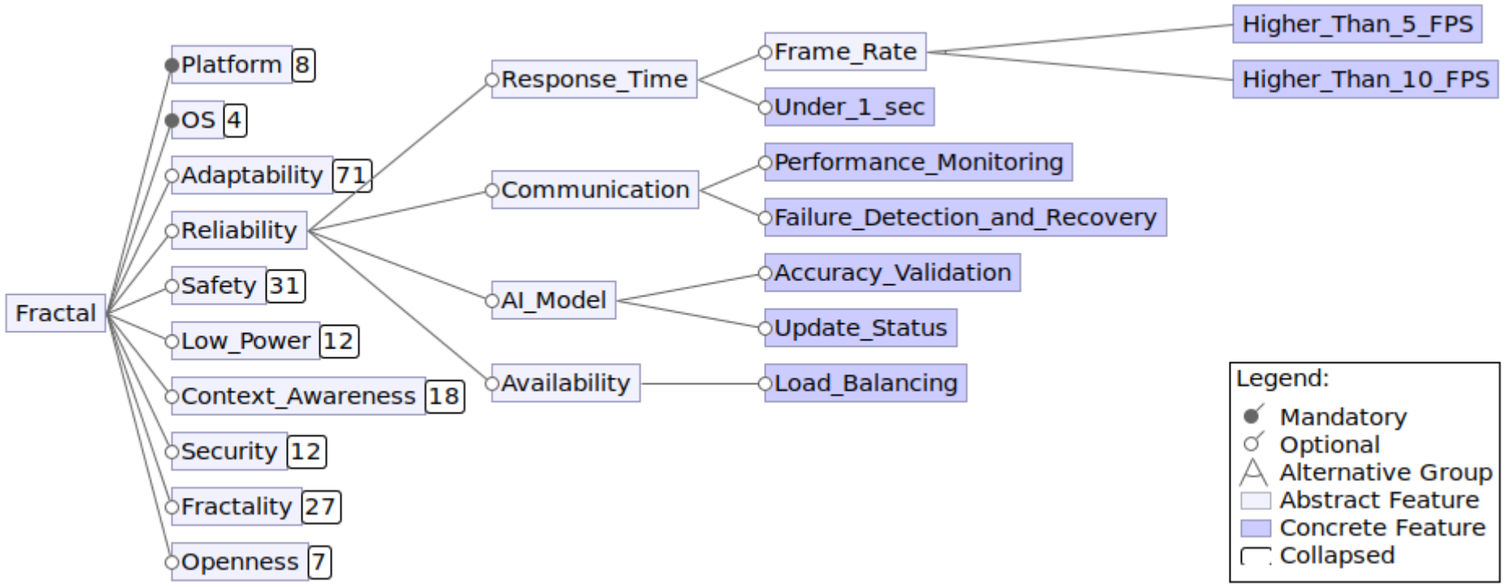

Figure 7 – Reliability features

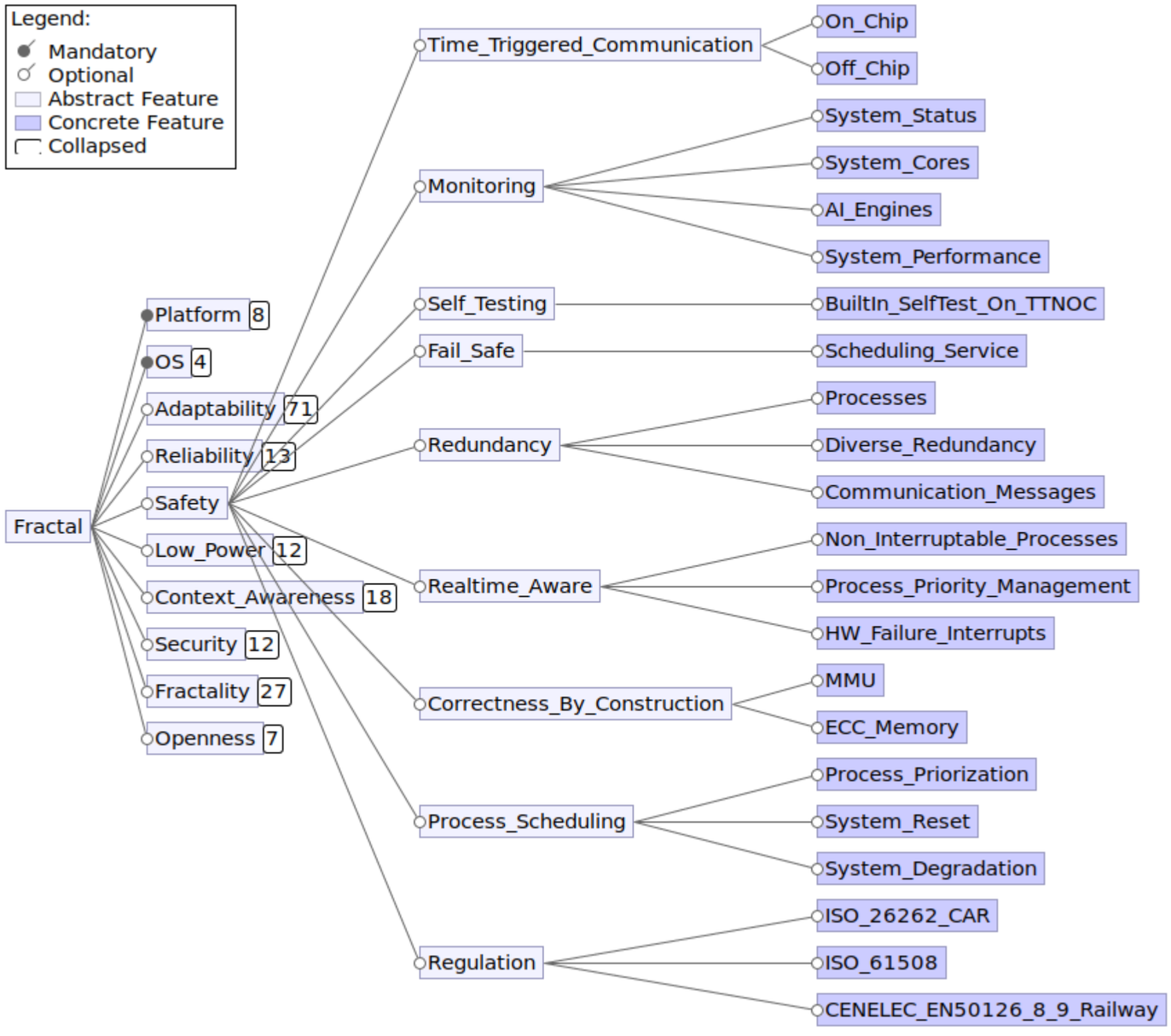

Figure 8 – Safety features

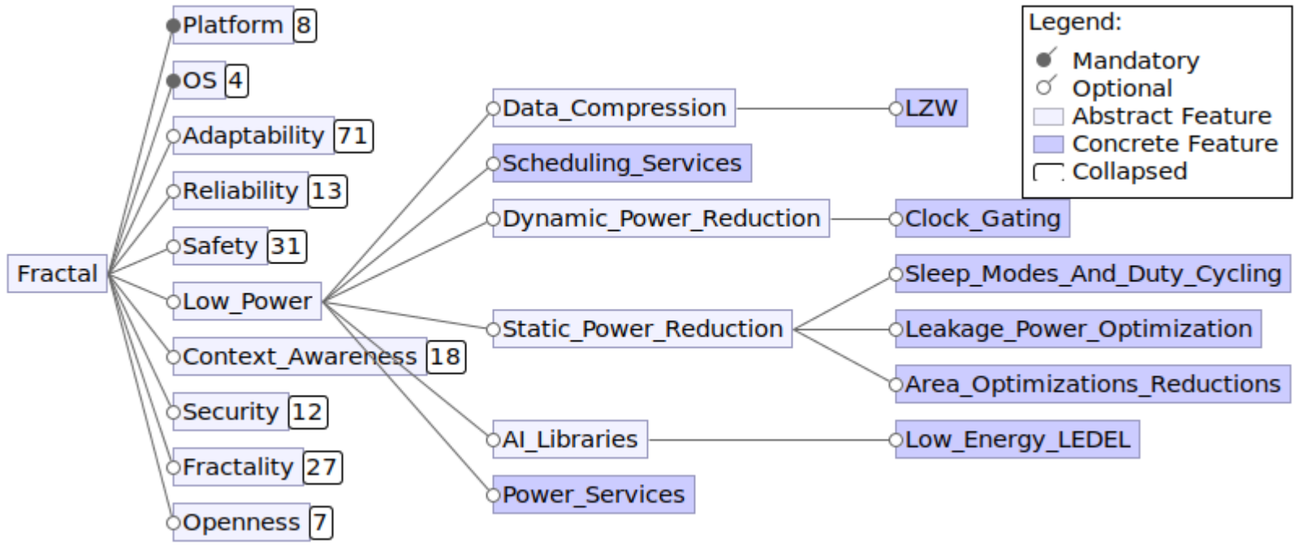

Figure 9 – Low power features

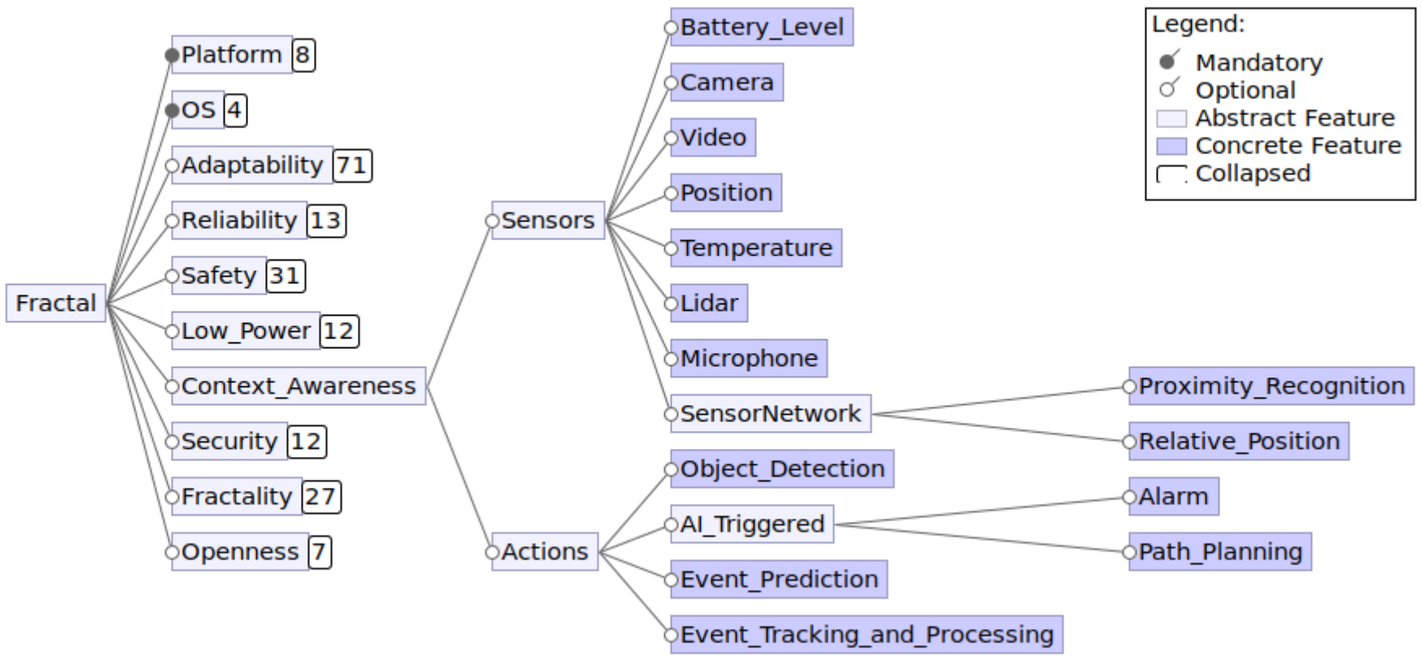

Figure 10 – Context awareness features

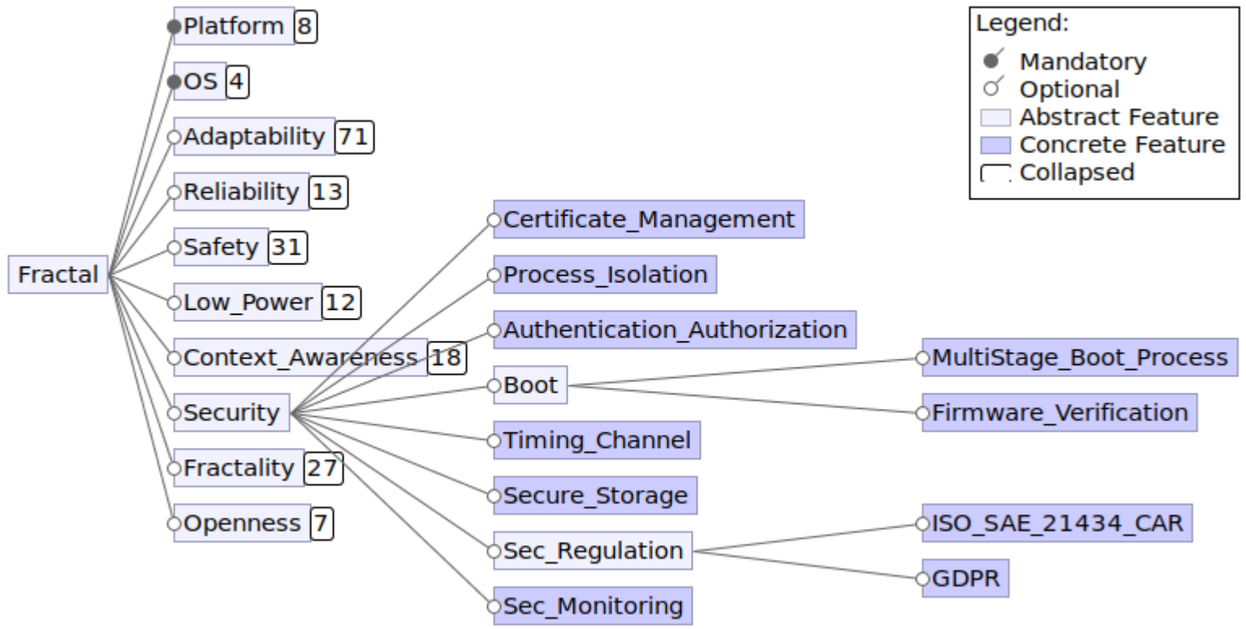

Figure 11 – Security features

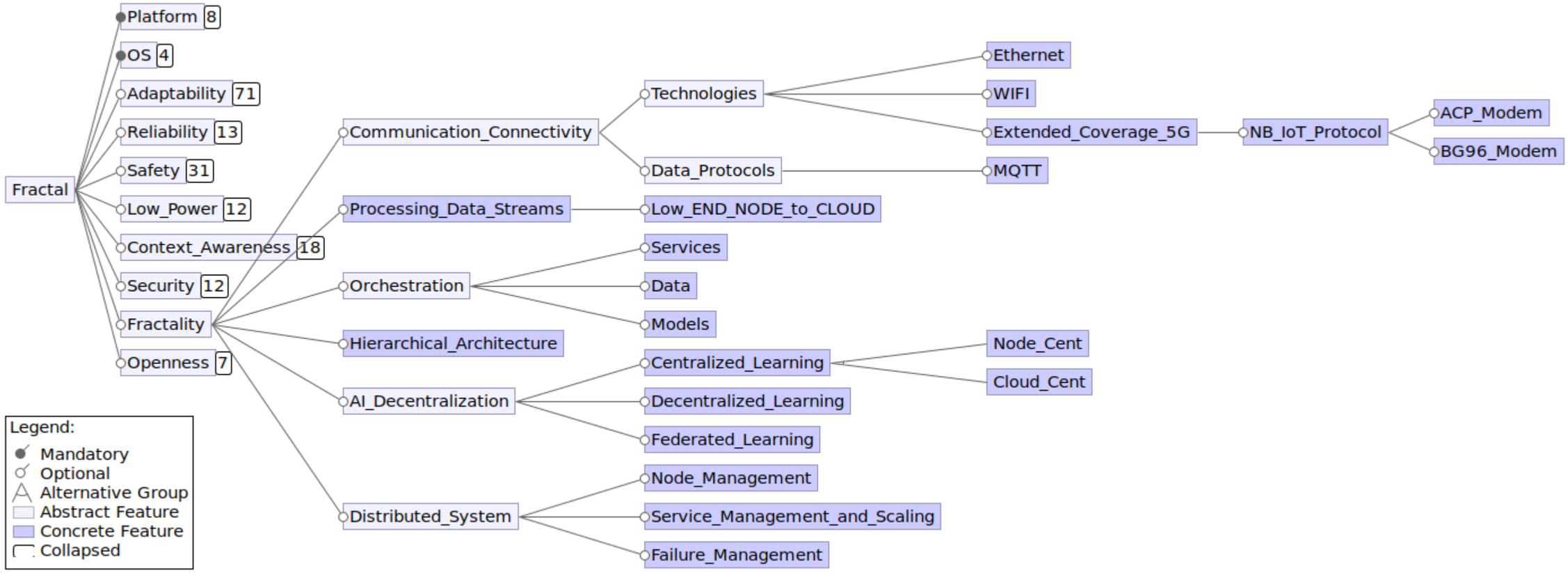

Figure 12 – Fractality features

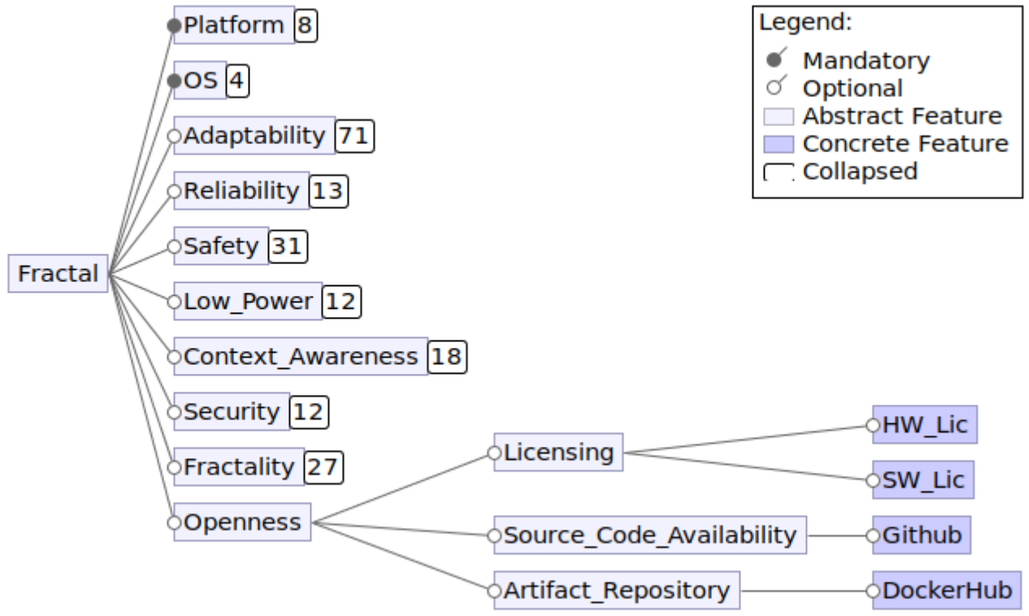

Figure 13 – Openness features

Appendix C: FRACTAL Features / Use Cases mapping

Features breakdown							UC1-1	UC1-2	UC2	UC3	UC4	UC5	UC6	UC7	UC8
ADAPTABILITY															
	EXTENSIBILITY														
		PORT CONNECTIONS													
			USB C					X				X			
			CAN BUS												X
			ETH RJ45					X				X			
			AXI PORTS								X				
		SOFTWARE STACK													
			BUILD - SYSTEM INTEGRATION									X			
	AI														
		HW													
			AI / ML ACCELERATOR				X				X	X	X		X
		SW													
			INFERENCE								X		X		
				REALTIME					X		X	X	X		X
				MODEL											
					FORMAT										
						ONNX		X		X		X	X	X	
						VERSAL		X	X			X	X		X
						PB	X	X							
				LOCATION											
					NODE		X		X	X	X	X	X	X	X
					CLOUD										
			LEARNING / TRAINING												
				LOCATION											
					NODE			X							
					CLOUD		X		X		X				X
					OTHER					X		X	X		X
				PARADIGM											
					SUPERVISED LEARNING		X	X		X	X	X	X		X
					UNSUPERVISED LEARNING			X							
					SEMI-SUPERVISED LEARNING										
					REINFORCEMENT LEARNING				X					X	
				ALGORITHMS											
					YOLO						X	X			X
					CNN		X			X	X		X	X	X

Features breakdown						UC1-1	UC1-2	UC2	UC3	UC4	UC5	UC6	UC7	UC8
					RNN		X							
					ANN			X						X
					GNN									X
					ADAM OPTIMIZATION	X		X						
			LIBRARY											
				TENSORFLOW - KERAS		X	X					X		X
				LEDEL									X	
				OPENCV		X					X			X
				CAFFE										
				NUMPY			X	X						X
				PYTORCH			X	X						X
				DARK-NET						X				
				RAY				X						
	DATA ORCHESTRATION													
		DATA SOURCE												
			CLOUD					X						
			EDGE			X	X	X		X				X
			DEVICE				X			X				
		PROCESSES												
			INGESTION				X	X						
			TRANSFORMATION / DATA ADAPTATION				X	X						
			MODEL FEEDING				X			X				
			OFFLOADING											
		DATA SET												
			STORAGE			X	X	X		X	X		X	
			VERSION CONTROL					X			X			
			FORMAT											
				JSON			X	X						
				SQL										
				FILES				X		X	X			
	SERVICES ORCHESTRATION													
	OPERATION MODE CHANGE													
		METASCHEDULING												X
			SYSTEM RECONFIGURATION											X
			FAULT TOLERANCE											X
			LOW POWER											X
RELIABILITY														
	RESPONSE TIME													

Features breakdown			UC1-1	UC1-2	UC2	UC3	UC4	UC5	UC6	UC7	UC8
		FRAME RATE	X				X	X		X	
		UNDER 1 SEC									
	COMMUNICATION										
		PERFORMANCE MONITORING			X						
		FAILURE DETECTION AND RECOVERY									
	AI MODEL										
		ACCURACY / VALIDATION	X	X	X		X	X			X
		UPDATE STATUS		X	X						
	AVAILABILITY										
		LOAD BALANCING							X		
SAFETY											
	TIME TRIGGERED COMMUNICATION										X
		OFF CHIP									X
		ON CHIP						X			X
	MONITORING										X
		STATUS									
		CORES									X
		AI ENGINES								X	
		PERFORMANCE	X						X	X	
	SELF TESTING										
		BUILT-IN SELF TEST ON ADAPTIVE TIME TRIGGERED NOC									X
	FAIL-SAFE										
		SCHEDULING SERVICES									X
	REDUNDANCY										
		PROCESSES								X	
		DIVERSE REDUNDANCY								X	
		COMMUNICATION MESSAGES								X	X
	REALTIME AWARE										
		NON-INTERRUPTABLE PROCESSES				X					
		PROCESS PRIORITY MANAGEMENT				X					
		HW FAILURE INTERRUPTS									
	CORRECTNESS BY CONSTRUCTION										
		MMU					X				
	PROCESS SCHEDULING										
		PROCESS PRIORIZATION									
		SYSTEM RESET									
		SYSTEM DEGRADATION									
	REGULATION										

Features breakdown				UC1-1	UC1-2	UC2	UC3	UC4	UC5	UC6	UC7	UC8
		ISO 26262 - CAR - VARIATION OF 61508				X			X		X	
		ISO 61508 - Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems						X	X			X
		CENELEC EN50126/8/9: Railway Industry							X			
LOW POWER												
	DATA COMPRESSION											
		LZW										
	SCHEDULING SERVICES											X
	DYNAMIC POWER REDUCTION						X					
		CLOCK GATING					X					
	STATIC POWER REDUCTION						X					
		LEAKAGE POWER OPTIMIZATIONS					X					
		AREA OPTIMIZATIONS					X					
	AI LIBRARIES										X	
		LEDEL									X	
	POWER SERVICES						X					
CONTEXT-AWARENESS												
	SENSORS											
		BATTERY LEVEL				X						X
		CAMERA		X			X	X		X		X
		VIDEO		X				X	X			
		POSITION		X	X							X
		TEMPERATURE				X						
		LIDAR									X	
		MICROPHONE								X		
		SENSOR NETWORK										
			PROXIMITY RECOGNITION		X							
			RELATIVE POSITION	X	X							X
	ACTIONS											
		OBJECT DETECTION		X				X	X			X
		AI TRIGGERED								X		X
			ALARM		X					X		X
			PATH PLANNING								X	
		EVENT PREDICTION			X							
		EVENT TRACKING AND PROCESSING			X							
SECURITY												
	CERTIFICATE MANAGEMENT											
	PROCESS ISOLATION				X				X			
	AUTHENTICATION - AUTHORIZATION								X			

Features breakdown						UC1-1	UC1-2	UC2	UC3	UC4	UC5	UC6	UC7	UC8
	BOOT								X					
		MULTI-STAGE BOOT PROCESS							X					
		FIRMWARE VERIFICATION							X					X
	TIMING CHANNEL													
	SECURE STORAGE						X				X			
	REGULATION													
		ISO SAE 21434 - CAR											X	
		GDPR												
	SEC. MONITORING													
FRACTALITY														
	COMMUNICATION / CONNECTIVITY													
		TECHNOLOGIES												
			ETHERNET							X	X	X	X	X
			WIFI				X					X		X
			RELIABLE LINK WITH EXTENDED COVERAGE						X					
				NB-IoT PROTOCOL					X					
					ACP MODEM				X					
					BG96 MODEM									
		DATA PROTOCOLS												
			MQTT				X	X				X		X
	PROCESSING OF DATA STREAMS							X						
		LOW END NODE TO CLOUD												
	ORCHESTRATION													
		SERVICES						X			X			
		DATA						X						
		MODEL						X		X	X			
	HIERARCHICAL ARCHITECTURE													X
	AI DECENTRALIZATION													
		CENTRALIZED LEARNING												
			CLOUD					X		X				
			NODE											
		DECENTRALIZED LEARNING												
		FEDERATED LEARNING												
	DISTRIBUTED SYSTEM													
		NODE MANAGEMENT												
		SERVICE MANAGEMENT AND SCALING												
		FAILURE MANAGEMENT												

Features breakdown				UC1-1	UC1-2	UC2	UC3	UC4	UC5	UC6	UC7	UC8
OPENNESS												
	LICENSING											
		HW										
		SW				X						
	SOURCE CODE AVAILABILITY											
		GITHUB				X						
	ARTIFACT REPOSITORY											
		DOCKERHUB				X						
OTHER: NON-FUNCTIONAL												
	PLATFORM											
		VERSAL - ARM		X		X			X	X		X
		PULP					X					
		NOEL-V - RISC-V									X	
		CVA6 - RISC-V						X				
		ZYNQ ULTRASCALE+ (VERSAL ALTERNATIVE)						X		X		X
	OS											
		LINUX		X	X	X		X	X	X	X	X
		RTOS					X					
			NUTTX				X					
			FREERTOS				X					
	INFRASTRUCTURE AS CODE											

Table 4 – Mapping between features and use cases

Appendix D: Components

Component ID	Component Name	Description	Related Fractal Features addressed with this component
WP3			
WP3-AI	**AI accelerator (hardware and software support)**		
WP3T32-01	HW accelerator (SIEDLA+)	HW accelerator for vision-based AI inference	ADAPTABILITY → AI → HW → ML ACCELERATOR
WP3T32-05	ML inference demo PULPissimo	Example application that uses ML acceleration on PULPissimo	ADAPTABILITY → AI → SW → INFERENCE ADAPTABILITY → AI → HW → ML ACCELERATOR ADAPTABILITY → AI → SW → INFERENCE → LOCATION → NODE OTHER: NON-FUNCTIONAL → PLATFORM → PULP
WP3T32-07	Age and Gender identifier at the edge	CNN development and integration on Zynq Ultrascale+ and VERSAL for UC6	ADAPTABILITY → AI → HW → AI / ML ACCELERATOR ADAPTABILITY → AI → SW → INFERENCE → REAL-TIME ADAPTABILITY → AI → SW → LEARNING / TRAINING → ALGORITHMS → CNN ADAPTABILITY → AI → LIBRARY → TENSORFLOW - KERAS CONTEXT AWARENESS → ACTIONS → AI TRIGGERED → ALARM OTHER: NON-FUNCTIONAL → PLATFORM → VERSAL OTHER: NON-FUNCTIONAL → PLATFORM → ZYNQ ULTRASCALE+ (VERSAL ALTERNATIVE) OTHER: NON-FUNCTIONAL → PLATFORM → OS → LINUX
WP3T32-10	VERSAL accelerator building-blocks	Development of building-blocks for accelerators for VERSAL	ADAPTABILITY → EXTENSIBILITY → SOFTWARE STACK → BUILD SYSTEM INTEGRATION ADAPTABILITY → AI → HW → AI /ML ACCELERATOR ADAPTABILITY → AI → SW → LIBRARY → TENSORFLOW - KERAS ADAPTABILITY → AI → SW → LIBRARY → OPENCV ADAPTABILITY → AI → SW → LIBRARY → CAFFE ADAPTABILITY → AI → SW → LIBRARY → NUMPY ADAPTABILITY → AI → SW → LIBRARY → PYTORCH OPENNESS → SOURCE CODE AVAILABILITY → GITHUB OTHER: NON-FUNCTIONAL → PLATFORM → VERSAL - ARM OTHER: NON-FUNCTIONAL → OS → LINUX OTHER: NON-FUNCTIONAL → OS → RTOS → FREERTOS
WP3T33-01	Ariane RISC-V for ZCU102	Porting Ariane 64bit to Xilinx ZCU 102 board	ADAPTABILITY → AI → HW → ML ACCELERATOR OTHER: NON-FUNCTIONAL → PLATFORM → CVA6 - RISC-V
WP3T34-03	Versal Model deployment layer	Model deployment on the Versal APU + DPU control from model repository images	ADAPTABILITY → AI → SW → INFERENCE → MODEL FORMAT → VERSAL ADAPTABILITY → AI → HW → AI /ML ACCELERATOR FRACTALITY → ORCHESTRATION → MODEL
WP3T35-01	SW driver for HW accelerator	SW driver for managing HW accelerator and data movements	ADAPTABILITY → AI → HW → ML ACCELERATOR
WP3T35-02	Accelerator Adaptation to AI library	Implementing support for missing functionalities/layers and data formats	ADAPTABILITY → AI → HW → ML ACCELERATOR OTHER: NON-FUNCTIONAL → PLATFORM → NOEL-V RISCV
WP3T35-03	LEDEL (Low Energy EDDL)	EDDL integration on NOEL-V	ADAPTABILITY → AI → SW → LIBRARY → LEDEL ADAPTABILITY → AI → SW → INFERENCE → MODEL → FORMAT → ONNX LOW POWER → AI LIBRARIES → LEDEL

Component ID	Component Name	Description	Related Fractal Features addressed with this component
WP3T35-04	Deep learning based automatic iris diagnosis	This component is devoted to the automatic diagnosis of iris pathologies from high-definition ocular fundus images. The component will be based on a deep learning architecture for classification potentially enhanced with advanced pre-processing techniques such as CLAHE for contrast augmentation. Moreover, we aim at equipping this component with an explainer, in order to show the potential of XAI methods in providing high-accuracy model with interpretable predictions.	ADAPTABILITY → AI → SW → INFERENCE RELIABILITY → AI MODEL → ACCURACY / VALIDATION
WP3T35-05	Idiom Recognition	Idiom/Language recognition system based on speech signal registration through Speech-to-Text ML solutions.	ADAPTABILITY → AI → SW → INFERENCE RELIABILITY → AI MODEL → ACCURACY / VALIDATION CONTEXT AWARENESS → ACTIONS → AI TRIGGERED → ALARM OTHER: NON-FUNCTIONAL → PLATFORM → VERSAL OTHER: NON-FUNCTIONAL → OS → LINUX
WP3-CPU/OS	**CPU and OS support**		
WP3T32-02	PULPissimo platform for IoT applications	Base RISC-V platform that can be used to implement FRACTAL specific components	OPENNESS → LICENSING → HW OTHER: NON-FUNCTIONAL → OS → RTOS → FREERTOS OPENNESS → SOURCE CODE AVAILABILITY → GITHUB OTHER: NON-FUNCTIONAL → PLATFORM → PULP
WP3T32-08	Real-time aware caches	Real time aware cache prevents the user application to flush real-time code from the caches	SAFETY → REALTIME AWARE → NON-INTERRUPTABLE PROCESSES LOW POWER → STATIC POWER REDUCTION → AREA OPTIMIZATION OTHER: NON-FUNCTIONAL → PLATFORM → PULP
WP3T32-09	Smart Interrupt distribution system	The smart interrupt distribution system prevents low priority interrupts to block real time critical threads	SAFETY → REALTIME AWARE → NON-INTERRUPTABLE PROCESSES SAFETY → REALTIME AWARE → PROCESS PRIORITY MANAGEMENT LOW POWER → STATIC POWER REDUCTION → AREA OPTIMIZATION OTHER: NON-FUNCTIONAL → PLATFORM → PULP
WP3T32-10	Security services - TL2AXI adapter	Bus protocol adapter allows to interface openTitan which hosts all secure services with pulpissimo	SECURITY → BOOT → FIRMWARE VERIFICATION OTHER: NON-FUNCTIONAL → PLATFORM → PULP
WP3T32-02b	Ariane for Linux capable RISC-V platform	Base RISC-V platform that can be used to implement FRACTAL specific components (note: "platform" beyond the "core")	OPENNESS → LICENSING → HW OPENNESS → SOURCE CODE AVAILABILITY → GITHUB OTHER: NON-FUNCTIONAL → OS → LINUX OTHER: NON-FUNCTIONAL → PLATFORM → CVA6 - RISC-V OTHER: NON-FUNCTIONAL → PLATFORM → PULP
WP3T32-03	PULP trainings	Training videos for using PULP	OPENNESS → LICENSING → SW OTHER: NON-FUNCTIONAL → OS → RTOS → FREERTOS OPENNESS → SOURCE CODE AVAILABILITY → GITHUB OTHER: NON-FUNCTIONAL → PLATFORM → PULP
WP3T32-04	FreeRTOS port to PULP	Port of the FreeRTOS to PULP	OPENNESS → LICENSING → SW OTHER: NON-FUNCTIONAL → OS → RTOS → FREERTOS
WP3T33-03	CVA6 (former Ariane) RISC-V core	CVA6 (former ARIANE) application core enhanced with 32b flavor	OTHER: NON-FUNCTIONAL → PLATFORM → CVA6 - RISC-V OPENNESS → SOURCE CODE AVAILABILITY → GITHUB

Component ID	Component Name	Description	Related Fractal Features addressed with this component
WP3T36-01	Linux for CVA6 (former Ariane)	Linux support for CVA6 (both 32b and 64b): UBoot (boot loader) + OpenSBI (firmware) + Yocto (embedded Linux generation)	OTHER: NON-FUNCTIONAL → OS → LINUX OTHER: NON-FUNCTIONAL → PLATFORM → CVA6 ADAPTABILITY → EXTENSIBILITY → SOFTWARE STACK → BUILD SYSTEM INTEGRATION OPENNESS → SOURCE CODE AVAILABILITY → GITHUB SAFETY → CORRECTNESS BY CONSTRUCTION → MMU
WP3T36-02	Load Balancing Module	Software module designed to collect computational loads from nodes and, in case of overload, able to distribute computational load.	RELIABILITY → AVAILABILITY → LOAD BALANCING
WP3T36-03	Nuttx on PULP	Porting Nuttx to PULP	OTHER: NON-FUNCTIONAL → OS → RTOS → NUTTX OTHER: NON-FUNCTIONAL → PLATFORM → PULP
WP3-Safety	**Safety and security features for CPU**		
WP3T31-01	Edge-oriented monitoring unit	AXI-compliant statistics unit to support safety measures and validation in the context of edge systems	SAFETY → MONITORING → PERFORMANCE ADAPTABILITY → EXTENSIBILITY → PORT CONNECTIONS → AXI PORTS
WP3T31-02	Interconnect to support Accelerators integration	Interconnect: AXI pulp library Integration	OTHER: NON-FUNCTIONAL → PLATFORM → NOEL-V RISCV ADAPTABILITY → AI → HW → HW ACCELERATOR ADAPTABILITY → EXTENSIBILITY → PORT CONNECTIONS → AXI PORTS
WP3T31-03	Safety and security hardware support	Extensions to the interconnect and other NOEL-V components for Security and Safety	SAFETY → REDUNDANCY → COMMUNICATION MESSAGES SAFETY → MONITORING → AI ENGINES
WP3T32-06	Redundant Acceleration Scheme	Integration of a redundant AI inference accelerator in the platform	SAFETY → REDUNDANCY → PROCESSES ADAPTABILITY → AI → HW → ML ACCELERATOR
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.	SAFETY → MONITORING → PERFORMANCE
WP3T34-01	Driver for the edge-oriented monitoring unit	Driver for the statistics unit supporting safety measures and validation in edge systems	SAFETY → MONITORING → PERFORMANCE
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	SAFETY → REDUNDANCY → PROCESSES SAFETY → REDUNDANCY → DIVERSE REDUNDANCY
WP4			
WP4T41	**Low Power Services**		
WP4T41-01	Data Compression for Low-Power Services	Data compression technique for low-power devices, to be applied at system level.	LOW POWER → DATA COMPRESSION → LZW
WP4T41-02	HATMA	Hierarchical Adaptive Time-triggered Multi-core Architecture to facilitate services at the different hierarchies	ADAPTABILITY → OPERATION MODE CHANGE → METASCHEDULING → LOW POWER
WP4T41-03	Low Power services for PULP systems	Enhancements to existing platforms to support low power FRACTAL services	LOW POWER → POWER SERVICES
WP4T41-04	Versal RPU access for Power Services	Access Dynamic Power, Frequency Scaling features on Versal [VCK190]	OTHER: NON-FUNCTIONAL → PLATFORM → VERSAL - ARM LOW POWER → POWER SERVICES SAFETY → MONITORING → CORES

Component ID	Component Name	Description	Related Fractal Features addressed with this component
WP4T41-05	Agreement protocol for Low-Power Services	Implementation of the agreement protocol on a wireless network on low-power devices	LOW POWER → STATIC POWER REDUCTION → SLEEP MODES & DUTY CYCLING
WP4T41-06	Versal Isolation Design - Functional Safety	Enhance the common Versal platform to strictly separate functional accesses, services from underlying HW access	OTHER: NON-FUNCTIONAL → PLATFORM → VERSAL - ARM SAFETY → REGULATION → ISO 61508 LOW POWER → POWER SERVICES SECURITY → PROCESS ISOLATION
WP4T42	**AI-Based Scheduling**		
WP4T42-01	Capabilities for AI supported adaptability in PULP	Enhancements to existing platforms to support resource allocation	ADAPTABILITY → AI → SW → INFERENCE ADAPTABILITY → AI → HW → ML ACCELERATOR
WP4T42-02	Versal RPU access to AI acceleration	Enhance RPU libraries to (1) access APU based AI as a service, (2) enable local AI [acceleration] deployment from RPU	ADAPTABILITY → AI → SW → INFERENCE → LOCATION → NODE ADAPTABILITY → AI → HW → ML ACCELERATOR FRACTALITY → HIERARCHICAL ARCHITECTURE
WP4T42-03	Scenario Generator	Scenario Generator that provides the inputs for the machine learning algorithm.	OTHER: NON-FUNCTIONAL → INFRASTRUCTURE AS CODE
WP4T42-04	GA-Scheduler	Scheduler (Genetic Algorithm) that provides the solutions of the scheduling problems given by the Scenario generator component.	OTHER: NON-FUNCTIONAL → INFRASTRUCTURE AS CODE SAFETY → PROCESS SCHEDULING → PROCESS PRIORIZATION
WP4T42-05	AI-Scheduler Model	Machine learning model used to predict schedules.	ADAPTABILITY → AI -- > SW → INFERENCE ---> REALTIME RELIABILITY → AI MODEL → ACCURACY/VALIDATION ADAPTABILITY → AI → SW → LEARNING/TRAINING → PARADIGM → SUPERVISED LEARNING ADAPTABILITY → AI → SW → LEARNING/TRAINING → ALGORITHM → ANN/GNN ADAPTABILITY → AI → SW → LIBRARY → PYTORCH
WP4T42-06	Schedule Verifier	Schedule verifier/reconstructor that takes the predictions of the machine learning model and convert them into a schedule.	RELIABILITY → AI MODEL → ACCURACY/VALIDATION SAFETY → PROCESS SCHEDULING → PROCESS PRIORIZATION
WP4T42-07	Hierarchical Metascheduler	Offline tool to compute time-triggered schedules by considering context events such as dynamic slack, failure scenarios.	ADAPTABILITY → OPERATION MODE CHANGE → METASCHEDULING → FAULT TOLERANCE
WP4T43	**Safety Services**		
WP4T43-01	Performance monitoring services	Services to configure the multicore-aware monitoring unit and retrieve information on the multicore interference observed	SAFETY → MONITORING → PERFORMANCE
WP4T43-02	Safety services for PULP systems	Enhancements to existing PULP systems to support safety services	SAFETY → REDUNDANCY → PROCESSES
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	SAFETY → REDUNDANCY → PROCESSES SAFETY → REDUNDANCY → DIVERSE REDUNDANCY
WP4T43-04	ATTNoC	Adaptive TTNoC provides time triggered communication for NoC and allow the systems to switch schedules in case of any failures occurs in the NoC.	SAFETY → TIME-TRIGGERED COMMUNICATION → ON-CHIP
WP4T43-05	Redundant Acceleration Scheme Safety Analysis	Safety analysis after inspection and fault-injection of a redundant acceleration scheme	SAFETY → REDUNDANCY → PROCESSES

Component ID	Component Name	Description	Related Fractal Features addressed with this component
WP4T43-06	FPGA Fault-injector	A tool to inject faults in the NOEL-V multicore. It is actually suitable for any Ultrascale+ FPGA	SAFETY → REDUNDANCY → PROCESSES
WP4T43-07	Safety Case	Safety Case ensuring that any hazard in the control of automotive components / system cannot lead to potential life treating danger	SAFETY → REGULATION → ISO 26262 - CAR - VARIATION OF 61508
WP4T43-08	Seamless redundancy for ATTNoC	Seamless redundancy 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 mask	SAFETY → REDUNDANCY → COMMUNICATION MESSAGES
WP4T43-09	Safety functions consideration during ML	integration of safety functions during preprocessing, to avoid triggering safety reaction in the field	SAFETY → REGULATION → ISO 26262 - CAR - VARIATION OF 61508
WP4T43-10	Safety functions consideration during Reinforcement learning	integration of safety functions in learning procedure for reinforcement learning to avoid triggering safety reaction in the field	SAFETY → REGULATION → ISO 26262 - CAR - VARIATION OF 61508
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.	SAFETY → TIME-TRIGGERED COMMUNICATION → ON CHIP
WP4T43-12	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.	SAFETY → REGULATION → ISO 26262 - CAR - VARIATION OF 61508
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.	SAFETY → REGULATION → ISO 61508 - Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems
WP4T44	**Security Services**		
WP4T44-01	Security services for PULP systems	Enhancements to existing PULP systems to support security services / components/ technologies for secure operation	SECURITY → TIMING CHANNEL
WP4T44-02	OS Security Layer	Implementation of security countermeasures in a transversal security layer	SECURITY → AUTHENTICATION - AUTHORIZATION
WP4T44-03	Security services at application and network layers	Security services are being implemented at app and network level, making applications running on the platform secure and also the communications between nodes or microservices exchanging data. E.g., HTTPs, TLS, user auth.	SECURITY → AUTHENTICATION - AUTHORIZATION SECURITY → AUTHENTICATION - CERTIFICATE MANAGEMENT
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 VAL_UC7.	SECURITY → REGULATION → ISO SAE 21434 - CAR
WP4T44-05	IoT Gateway	IoT network Gateway for external communication monitoring	FRACTALITY → COMMUNICATION → DATA PROTOCOL → MQTT
WP4T44-06	GDPR Compliance	Data Protection Impact Analysis	SECURITY → REGULATION → GDPR
WP4T44-07	Node monitoring and system status	Metrics collector for security related aspects and issue addressing	SECURITY → MONITORING SECURITY → AUTHENTICATION, AUTHORIZATION
WP4T44-08	TLS Implementation on containers	Inter-nodal secure communications	SECURITY → CERTIFICATE MANAGEMENT

Component ID	Component Name	Description	Related Fractal Features addressed with this component
WP4T44-09	Runtime security	Process isolation through containerization and user control	FRACTALITY → ORCHESTRATION → SERVICES ADAPTABILITY → SERVICES ORCHESTRATION SECURITY → PROCESS ISOLATION
WP4T44-10	LEDEL validation	Validation process for the LEDEL library	LOW POWER → AI LIBRARIES → LEDEL ADAPTABILITY → AI → SW → INFERENCE RELIABILITY → AI MODEL → ACCURACY / VALIDATION
WP5			
WP5T52-01	**Cloud Platform Data Ingestion**	The platform allows to ingest data in two different approaches: on the first approach, streaming data ingestion, and bulk data ingestion. Streaming data ingestion will be addressed by the data ingestion service (i.e., Kafka), whereas bulk data ingestion will be supported by the raw data storage component.	
WP5T52-01-01	Data Ingestion Service	The scope of work and the services for Kafka will be developed and tested locally first and then helm chart will be handed over for the deployment in OVH cloud	ADAPTABILITY → DATA ORCHESTRATION → PROCESSES → INGESTION RELIABILITY → COMMUNICATION → FAILURE DETECTION AND RECOVERY RELIABILITY → AVAILABILTIY → LOAD BALANCING
WP5T52-02	**Cloud Platform Raw Data Storage**	Cloud service to store the raw data for training /retraining models	
WP5T52-02-01	Raw data Object storage service	OVH S3 service is provided for Data Storage. If Special Databases are required to store data (e.g., time series database, relational database, etc.) these should be also deployed together with the required scripts in the data transformation step.	ADAPTABILITY → DATA ORCHESTRATION → DATA SET → STORAGE FRACTALITY → ORCHESTRATION → DATA
WP5T52-03	**Cloud Platform Data Transformation**		
WP5T52-03-01	Data transformation	This component retrieves the raw data stored in the data storage component and applies to them different data pre-processing techniques in order to transform the raw data into a dataset that is prepared and optimized to be exploited by using advanced data analytic techniques Deploy a Service for Data processing (data pre-processing, cleaning, integration, etc.)	ADAPTABILITY → DATA ORCHESTRATION → PROCESSES → TRANSFORMATION FRACTALITY → ORCHESTRATION → DATA FRACTALITY → PROCESSING OF DATA STREAMS
WP5T52-04	**Cloud Platform Repositories**		
WP5T52-04-01	Model version control	Datasets version control, Model version control	ADAPTABILITY → DATA ORCHESTRATION → DATASET → VERSION CONTROL FRACTALITY → ORCHESTRATION → DATA
WP5T52-04-03	S3 compatible data storage	Data Lake. High-performance, S3 compatible object (images, video) storage.	ADAPTABILITY → DATA ORCHESTRATION → DATASET → STORAGE FRACTALITY → ORCHESTRATION → DATA
WP5T52-04-05	Datasets version control	Dataset version control repository	ADAPTABILITY → DATA ORCHESTRATION → DATASET → VERSION CONTROL FRACTALITY → ORCHESTRATION → DATA
WP5T52-04-06	Feature storage	Stores trained features. Serves the features for ML models inference or training.	ADAPTABILITY → DATA ORCHESTRATION → DATASET → STORAGE, VERSION CONTROL FRACTALITY → ORCHESTRATION → DATA
WP5T52-04-07	Images repository	Container Registry for Docker Images	FRACTALITY → ORCHESTRATION → SERVICES ADAPTABILITY → SERVICES ORCHESTRATION

Component ID	Component Name	Description	Related Fractal Features addressed with this component
WP5T52-04-08	Model repository	Metadata version control and Models version control framework	FRACTALITY → ORCHESTRATION → MODEL
WP5T52-04-09	Machine learning pipeline	Metadata version control and Models version control framework	FRACTALITY → ORCHESTRATION → MODEL
WP5T52-04-10	MLBuffet as a Cloud Model Storage	Light-weight Model repository	FRACTALITY → ORCHESTRATION → MODEL
WP5T52-05	**Cloud Platform Orchestration**	Workflow Management, Machine Learning workflow	
WP5T52-05-01	MLBuffet as a Cloud Model Orchestrator	FRACTAL models orchestration (Manage and run models on cloud). It can be used as a validation for the models to be deployed on the Edge to be running properly.	FRACTALITY → ORCHESTRATION → MODEL
WP5T52-05-02	Data pipelines and workflows orchestrator	Deploy Airflow and configure its integration with other services for their orchestration.	FRACTALITY → ORCHESTRATION → DATA FRACTALITY → ORCHESTRATION → SERVICES ADAPTABILITY → DATA ORCHESTRATION → PROCESSES → TRANSFORMATION / DATA ADAPTATION
WP5T52-06	**Cloud Platform Models Serving**		
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	ADAPTABILITY → AI → SW → INFERENCE → MODEL FORMAT → VERSAL FRACTALITY → ORCHESTRATION → MODEL
WP5T52-06-02	ML pipeline connection to model repository	Extension of the cloud platform towards the edge-cloud computing continuum.	FRACTALITY → ORCHESTRATION → SERVICES FRACTALITY → ORCHESTRATION → DATA
WP5T52-07	**Cloud Platform Infrastructure**		
WP5T52-07-01	Kubernetes-based cloud platform container orchestrator	Managed Kubernetes Service for containerized services orchestration.	ADAPTABILITY → SERVICES ORCHESTRATION FRACTALITY → ORCHESTRATION → SERVICES RELIABILITY → AVAILABILITY → LOAD BALANCING
WP5T52-07-02	Cloud container orchestrator services access	Allows to expose Kubernetes services and routing requests to different endpoints.	ADAPTABILITY → SERVICES ORCHESTRATION FRACTALITY → ORCHESTRATION → SERVICES RELIABILITY → AVAILABILITY → LOAD BALANCING
WP5T54-01-01	MLBuffet	Machine Learning tool for model serving and management. Deployable in containers with Swarm and Kubernetes. API for managing models and sending input/Outputs Training module to be developed during 2022	FRACTALITY → ORCHESTRATION → MODEL ADAPTABILITY → DATA ORCHESTRATION → PROCESSES → MODEL FEEDING
WP5T54-01-02	Training module for MLBuffet	Training module for supporting model training being framework agnostic.	ADAPTABILITY → AI → SW → TRAINING ADAPTABILITY → AI → SW → TRAINING → LOCATION → CLOUD
WP5T54-02	**Orchestrators**		
WP5T54-02-01	Docker Swarm	Open-Source orchestrator integrated in Docker Engine.	ADAPTABILITY → SERVICES ORCHESTRATION FRACTALITY → ORCHESTRATION → SERVICES RELIABILITY → AVAILABILITY → LOAD BALANCING
WP5T54-02-02	Kubernetes-based Container Orchestrators for the Edge	Open-Source orchestrator for cluster management and container orchestration.	ADAPTABILITY → SERVICES ORCHESTRATION FRACTALITY → ORCHESTRATION → SERVICES RELIABILITY → AVAILABILITY → LOAD BALANCING

Component ID	Component Name	Description	Related Fractal Features addressed with this component
WP5T54-03	MLOps Toolchain	Continuous Integration and Continuous Development tools have been identified and research has been done on tool integration for a full CI/CD toolchain that covers all the steps in ML processes, from model design, training and version controlling, to model serving into production.	FRACTALITY → ORCHESTRATION → MODEL ADAPTABILITY → AI → SW → TRAINING ADAPTABILITY → AI → SW → TRAINING → LOCATION → CLOUD
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.	ADAPTABILITY → AI → SW → LEARNING / TRAINING → ALGORITHMS → CNN ADAPTABILITY → AI → SW → TRAINING → LOCATION → CLOUD
WP6			
WP6T61-02	**Orchestration layer**		
WP6T61-02-01	Docker	platform as a service product that uses OS-level virtualization to deliver software in packages called containers.	ADAPTABILITY → SERVICES ORCHESTRATION FRACTALITY → ORCHESTRATION → SERVICES RELIABILITY → AVAILABILITY → LOAD BALANCING
WP6T61-02-02	Mosquitto	open source (EPL/EDL licensed) message broker that implements the MQTT protocol	FRACTALITY → COMMUNICATION → DATA PROTOCOLS → MQTT
WP6T61-02-03	Microk8S	low-ops, minimal production Kubernetes and it is an open-source system for automating deployment, scaling, and management of containerized applications	ADAPTABILITY → SERVICES ORCHESTRATION FRACTALITY → ORCHESTRATION → SERVICES RELIABILITY → AVAILABILITY → LOAD BALANCING
WP6T61-02-04	K3S	low-ops, minimal production Kubernetes and it is an open-source system for automating deployment, scaling, and management of containerized applications	ADAPTABILITY → SERVICES ORCHESTRATION FRACTALITY → ORCHESTRATION → SERVICES RELIABILITY → AVAILABILITY → LOAD BALANCING
WP6T61-02-05	Ingress	API object that manages external access to the services in a cluster. Provides provide load balancing, SSL termination and name-based virtual hosting	FRACTALITY → ORCHESTRATION → SERVICES RELIABILITY → AVAILABILITY → LOAD BALANCING
WP6T61-02-06	Juju	model-driven Operator Lifecycle Manager (OLM)	FRACTALITY → ORCHESTRATION → SERVICES
WP6T61-02-07	Cilium	software for providing, securing, observing network connectivity between container workloads and load balancing between application workloads	FRACTALITY → ORCHESTRATION → SERVICES RELIABILITY → AVAILABILITY → LOAD BALANCING
WP6T61-02-08	Prometheus	systems and service monitoring system	RELIABILITY → COMMUNICATION → PERFORMANCE MONITORING SAFETY → MONITORING → STATUS & PERFORMANCE FRACTALITY → ORCHESTRATION → SERVICES
WP6T61-02-09	Velero	open-source tool for backup and restore	ADAPTABILITY → AI → DATA ORCHESTRATION → DATA SET → STORAGE FRACTALITY → ORCHESTRATION → SERVICES
WP6T61-02-10	Istio	traffic management, telemetry, and security extension to Kubernetes family	RELIABILITY → COMMUNICATION → PERFORMANCE MONITORING SAFETY → MONITORING → STATUS & PERFORMANCE FRACTALITY → ORCHESTRATION → SERVICES
WP6T61-02-11	Envoy	open-source edge and service proxy	FRACTALITY → ORCHESTRATION → SERVICES RELIABILITY → AVAILABILITY → LOAD BALANCING

Component ID	Component Name	Description	Related Fractal Features addressed with this component
WP6T61-02-12	Fluent Bit	open-source Log Processor and Forwarder	FRACTALITY → ORCHESTRATION → SERVICES SAFETY → MONITORING → STATUS
WP6T61-02-13	Virtual Kubelet	Kubernetes kubelet implementation that masquerades as a kubelet for the purposes of connecting Kubernetes to other API	ADAPTABILITY → SERVICES ORCHESTRATION → SCHEDULING, REPLICATION, SCALABILITY, MIGRATION FRACTALITY → ORCHESTRATION → SERVICES RELIABILITY → AVAILABILITY → LOAD BALANCING
WP6T62-03	Run time Manager	Management of component interaction, task scheduling and node-to-node communication	FRACTALITY → ORCHESTRATION → SERVICES
WP6T61-03	**Application Layer**		
WP6T61-03-01	Machine learning toolkit for Kubernetes	machine learning toolkit for Kubernetes making deployments of machine learning (ML) workflows on Kubernetes (Kubeflow)	FRACTALITY → ORCHESTRATION → MODEL
WP6T61-03-02	End-to-end machine learning toolkit	end-to-end open-source platform for machine learning (Tensorflow)	AI → SW → LEARNING/TRAINING → LOCATION → NODE ADAPTABILITY → AI → SW → LIBRARY → TENSORFLOW/KERAS
WP6T61-03-03	Deep learning toolkit	deep learning framework (Caffe)	AI → SW → LEARNING/TRAINING → LOCATION → NODE ADAPTABILITY → AI → SW → LIBRARY → CAFFE
WP6T61-03-04	Jupyter	Jupyter Notebook is a web-based interactive computing platform	AI → SW → LEARNING/TRAINING → LOCATION → NODE
WP6T61-03-05	Optimized toolkit for tensor machine learning	open-source machine learning framework and an optimized tensor library (PyTorch)	AI → SW → LEARNING/TRAINING → LOCATION → NODE ADAPTABILITY → AI → SW → LIBRARY → PYTORCH
WP6T61-03-06	Computer vision toolkit	open-source computer vision and machine learning software library (OpenCV)	AI → SW → LEARNING/TRAINING → LOCATION → NODE ADAPTABILITY → AI → SW → LIBRARY → OPENCV
WP6T61-03-07	Kafka	open-source distributed event streaming platform	ADAPTABILITY → DATA ORCHESTRATION → PROCESSES → INGESTION FRACTALITY → ORCHESTRATION → DATA
WP6T61-03-08	MongoDB	cross-platform document-oriented database	ADAPTABILITY → DATA ORCHESTRATION → DATA SET → STORAGE ADAPTABILITY → DATA ORCHESTRATION → DATA SET → FORMAT → JSON
WP6T61-03-09	PostgreSQL	open source object-relational database	ADAPTABILITY → DATA ORCHESTRATION → DATA SET → STORAGE ADAPTABILITY → DATA ORCHESTRATION → DATA SET → FORMAT → SQL
WP6T61-03-10	Spark	open-source unified analytics engine for large-scale data processing	ADAPTABILITY → DATA ORCHESTRATION → PROCESSES → INGESTION FRACTALITY → ORCHESTRATION → DATA
WP6T61-03-11	Machine learning training accelerator	cross-platform inference and training machine-learning accelerator (ONNX)	ADAPTABILITY → AI → SW → INFERENCE → MODEL → FORMAT → ONNX
WP6T61-12	Neural network toolkit	open-source neural network framework (Darknet)	AI → SW → LEARNING/TRAINING → LOCATION → NODE ADAPTABILITY → AI → SW → LIBRARY → DARKNET
WP6T61-13	Deep learning toolkit	deep learning framework (Keras)	AI → SW → LEARNING/TRAINING → LOCATION → NODE ADAPTABILITY → AI → SW → LIBRARY → TENSORFLOW/KERAS
WP6T61-14	Python-based deep learning toolkit	Python-based deep learning framework (Chainer)	AI → SW → LEARNING/TRAINING → LOCATION → NODE ADAPTABILITY → AI → SW → LIBRARY
WP6T62-01	TSN to Interconnect Multiple Nodes	TSN Network to facilitate the communication between various nodes for off-chip communication	SAFETY → TIME-TRIGGERED COMMUNICATION → OFF-CHIP

Component ID	Component Name	Description	Related Fractal Features addressed with this component
WP6T62-02	Hierarchical Interactive Consistency Protocol (HICP)	Broadcast protocol to facilitate adaptation through agreement of system state at various hierarchies	ADAPTABILITY → OPERATIONAL MODE CHANGE → METASCHEDULING → SYSTEM RECONFIGURATION

Table 5 – Components