

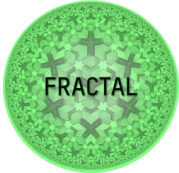
Deliverable

D7.1 Building Block Verification Plan into Verification Use Cases

Deliverable Id:	D7.1
Deliverable Name:	Building Block Verification Plan into Verification Use Cases
Status:	Submitted
Dissemination Level:	Public
Due date of deliverable:	28. February 2022
Actual submission date:	30. April 2022
Work Package:	WP7
Organization name of lead contractor for this deliverable:	ETH Zurich
Author(s):	Frank K. Gürkaynak, ETH Zürich
Partner(s) contributing:	Bernhard Peischi, AVL Bekim Chilku, Siemens Cristina Ganado Arteaga, Prointec Alfonso Gonzales Gil, ZYLK Michael Gautschi, ACP Pascal Muoka, University of Siegen Daniel Onwuchekwa, University of Siegen Inaki Paz Rey, LKS

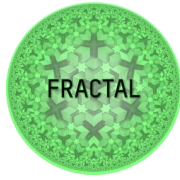
Abstract:

This report, identifies the FRACTAL features that will be demonstrated on use cases involved in WP7 (UC1-4), explains how FRACTAL features are expected to impact the use cases using KPIs, and describes how the impact will be verified for each use case and FRACTAL feature.

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

Contents

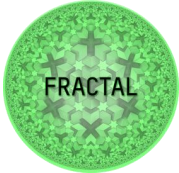
1	Summary	4
2	Introduction.....	5
2.1	Big Picture of FRACTAL	6
3	Use cases and objectives selected for WP7	8
3.1	UC1 – Engineering and Maintenance Works	8
3.2	UC2 – Automotive Airpath Control	9
3.3	UC3 – Smart Meter for everyone	10
3.4	UC4 –Low-latency Object Detection as a generic building block for perception in the edge for industrial application.....	11
4	FRACTAL features considered in WP7	13
4.1	Adaptability	14
4.2	Security	16
4.3	Cognitive awareness.....	17
4.4	Low Power.....	19
4.5	Connectivity to cloud / Communication	20
4.6	Openness	22
5	Verification plan for individual FRACTAL features	24
5.1	Plan for verifying Adaptability	24
5.2	Plan for verifying Security	26
5.3	Plan for verifying Cognitive Awareness.....	27
5.4	Plan for verifying Low Power.....	28
5.5	Plan for verifying Connectivity to Cloud / Communication	30
5.6	Plan for verifying Openness	32
6	Conclusions	35
7	List of Abbreviations	36
8	List of figures.....	38
9	List of tables.....	39



Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
Title	Feature Verification Plan into Verification Use Cases
Del. Code	D7.1

History

Version	Date	Modification reason	Modified by
v0.1	19.Jan.2022	Initial skeleton	Frank K. Gürkaynak
v0.2	30.Mar.2022	Preparation for final form	Frank K. Gürkaynak
v0.5	09.Apr. 2022	Copy for internal review	Frank K. Gürkaynak
v0.9	15.Apr.2022	Corrections from review	Frank K. Gürkaynak
v1.0	30.Apr.2022	Final submitted version	Frank K. Gürkaynak

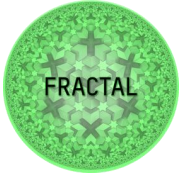
	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

1 Summary

The main goal of the FRACTAL project is to create a reliable computing platform node that can be used as a building block of scalable decentralized Internet of Things (IoT). Throughout the project several *features* were developed as part of the technical work packages (WP 4/5/6) that are expected to give FRACTAL based systems an advantage. Work package 7 is designed to unify the technical activities within FRACTAL and demonstrate the benefits of the improvements made throughout FRACTAL and verifies its efficiency on industrial level use cases.

The deliverable 7.1 is the first in a series of three deliverables and will describe what FRACTAL features will be demonstrated as part of work package 7, what benefits are expected from adoption of these FRACTAL features as part of each use case and how these benefits will be verified within the work package.

In particular, deliverable 7.1 will describe the use cases, their building blocks in section 2. The objectives that each use case targets through adaptation of FRACTAL *features* will be explained in section 3. The FRACTAL *features* that form the main contributions expected from FRACTAL for each use case will be explained in more detail in section 4. How each of these FRACTAL *features* will be verified and what KPIs have been identified will be covered in section 5, and finally section 6 will offer conclusions.

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

2 Introduction

The main objective of the FRACTAL project is to “*create a cognitive edge node enabling a FRACTAL edge that can be qualified to work under different safety-related domains*”. The first question that we have to answer is how does the FRACTAL project view *fractality*.

The shortest description for *fractal* is something that repeats itself at different scales. From a technical perspective, this translates to creating an abstraction over low-level functions/services so that nodes with different capabilities that make up a FRACTAL network can use these functions/services the same way. As an example, safety services (such as redundancy) developed for off-chip communication where multiple nodes are connected are used in the same way within individual nodes when providing the safety services that connect multiple cores/resources. The key point in FRACTAL is to leverage solutions and services at different levels (cloud, edge, mist).

Three work packages in FRACTAL provide the technical innovations that make up the FRACTAL system

- **WP4** Safety, Security & Low Power Techniques
- **WP5** AI & Safe Autonomous Decision
- **WP6** CPS Communication Framework

The enhanced capabilities for FRACTAL nodes will then be integrated into at least two hardware nodes:

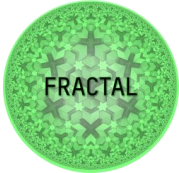
- **Commercial node** based around the Xilinx VERSAL ACAP (Adaptable Compute Acceleration Platform)
- **Customizable node** based around the open-source RISC-V based PULP platform

as part of WP3 “Node Architecture & Building Blocks”.

Finally, as part of WP7, these individual developments by all partners that compose FRACTAL architecture will be unified achieving data exchange, communication and complex tasks in simple and reliable ways, to provide high level of modularization and customization of the platform. This will be demonstrated on four selected use cases (UC):

- **UC1** – Engineering and Maintenance Works led by PROINTEC
- **UC2** – Automotive-Airpath-Control led by AVL
- **UC3** – SmartMeter led by ACP
- **UC4** – Low-latency Object Detection as a generic building block for perception in the edge for industrial application led by SIEMENS

The first task within WP7 is Task 7.1 “Coherent development and demonstration test methodology definition” with the objective to establish the verification/evaluation plans and evaluation metrics to be used for the assessment of the degree of

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

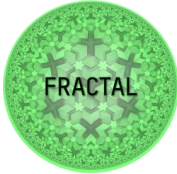
integration of the FRACTAL reference platform and the associated features in the reference verification use cases defined for WP7. The work of this task forms the basis of this deliverable D7.1 and identifies the FRACTAL features (Section 4) that will be demonstrated as part of WP7, provides the description of the use cases and how the FRACTAL *features* will improve the performance (Section 3). Finally, how FRACTAL partners will verify these features and presents associated KPIs (Section 5).

The implementation details for the verification described in Section 5 will be covered in D7.2 and the results of the evaluation will be given later in D7.3.

2.1 Big Picture of FRACTAL

As a large project with more than 20 partners, 8 use cases, at least 3 different hardware nodes and innovations at very different levels (cloud, edge, hardware, software) it is easy to lose the overview of how different contributions relate to each other.

Following the 2nd Technical Workshop of FRACTAL, the following *Big Picture* shown in Figure 1 has been developed to identify how the various FRACTAL building blocks are connected to build the FRACTAL node. However, for a given use case not all of these contributions are used simultaneously. For example, UC2 is more focused on FRACTAL capabilities that are deployed in the cloud, while UC3 concentrates on low-power applications on edge nodes. The result is that each use case has adopted a suitable subset of the FRACTAL developments. This has motivated the structure of this deliverable, which first discusses the use cases (in Section 3) and then moves to the selected FRACTAL *features* to be verified within these use case implementations (in Section 4).

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

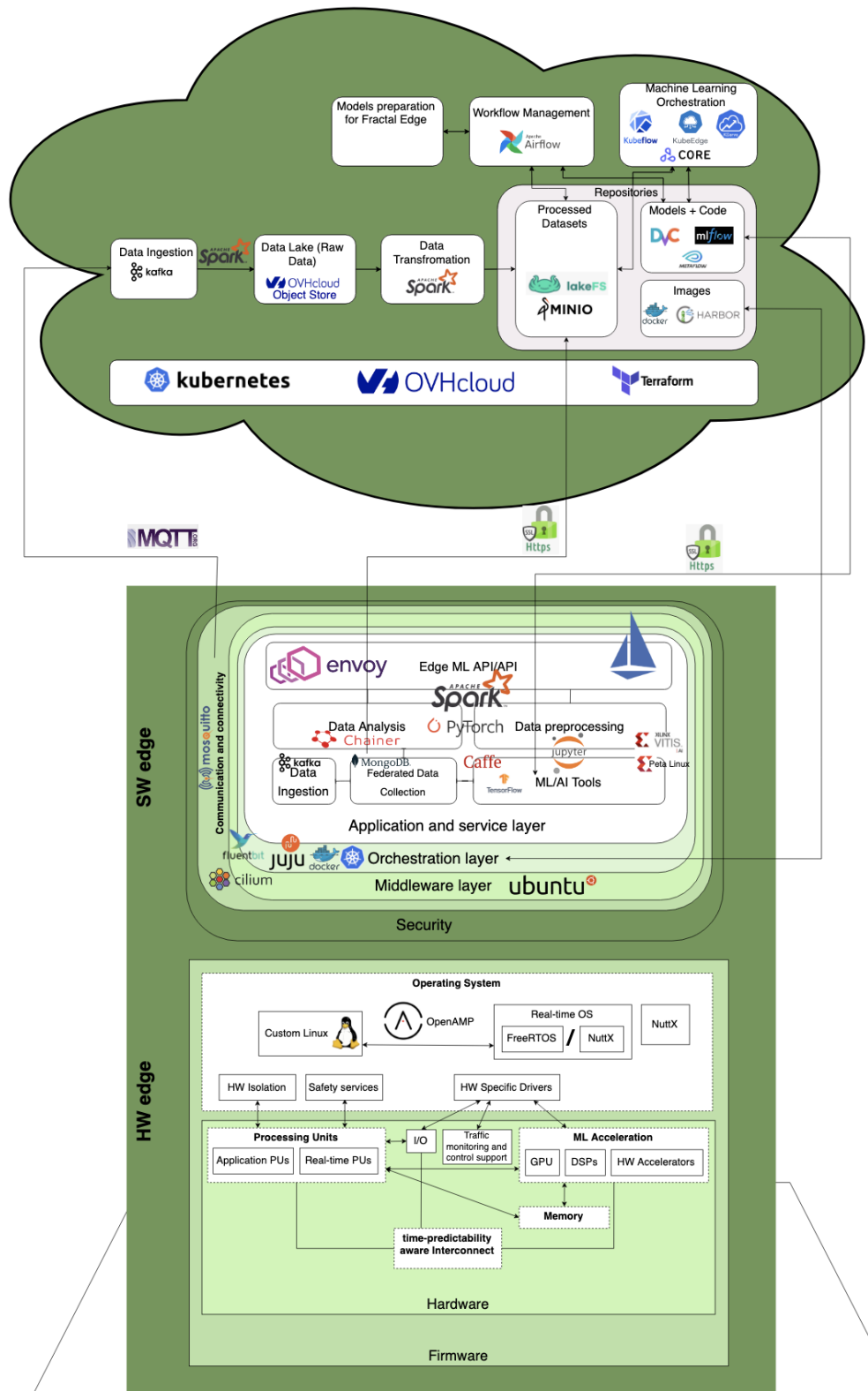
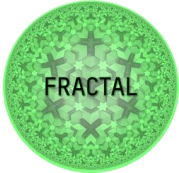


Figure 1 The Big Picture of FRACTAL (Apr 2022)

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

3 Use cases and objectives selected for WP7

The FRACTAL project has been set up around eight use cases that will be used to demonstrate the features developed in the project. WP7 specifically includes four use cases:

- **UC1** – Engineering and Maintenance Works led by PROINTEC
- **UC2** – Automotive-Airpath-Control led by AVL
- **UC3** – SmartMeter led by ACP
- **UC4** – Low-latency Object Detection as a generic building block for perception in the edge for industrial application led by SIEMENS

This section will describe these use cases and what FRACTAL objectives they are targeting.

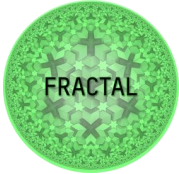
3.1 UC1 – Engineering and Maintenance Works

UC1 seeks to develop solutions supported by edge technologies for the improvement of safety on construction sites. This UC1 is divided into two demonstrators describing two different types of work within the construction sector.

Demonstrator 1 “*UAV supervision of critical structures*” is focused on the analysis of the surface of critical structures such as bridges or viaducts, where defects such as cracks are formed. Using drones, images of the affected areas susceptible to analysis are collected, thus reducing the exposure of workers to dangers posed by accessing the locations (which are usually high) during the visual identification of these defects. Through an AI algorithm, based on a deep learning model with convolutional neural networks (CNN), information is extracted from the images collected with the drone.

Indra (IFT) has been working in numerous image segmentation projects, including productive deployments all around the world. The instance segmentation developments have so far been mainly applied on earth-observation projects. The knowledge gained in these earlier projects has been directly applied in the FRACTAL demonstrator.

Demonstrator 2, “*Wireless Sensor Network (WSN) for safety at construction sites*”, focuses on the monitoring of workers and machinery within a construction site, in order to reduce conflicts that may pose a danger to the integrity of the workers. For this purpose, a WSN is deployed to generate information about the proximity between workers and machines as seen in Figure 2. This information will be managed through an IoT platform, registering possible dangers and alarms, in addition to establishing a protocol in case of emergency.

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

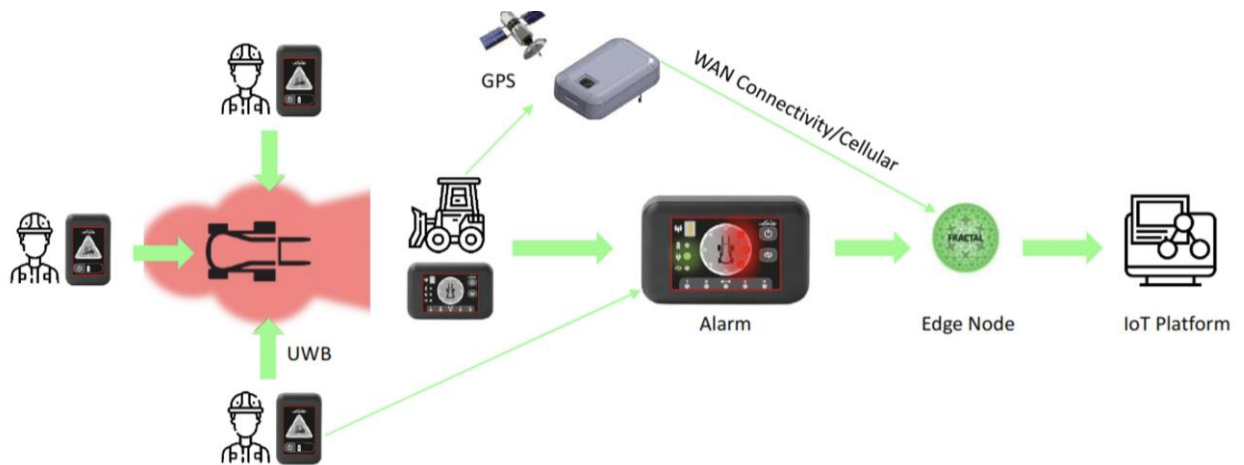


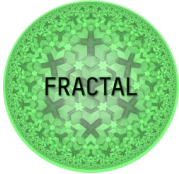
Figure 2 The wireless sensor network for safety at construction sites developed as part of UC1

3.2 UC2 – Automotive Airpath Control

In the famous essay “Why Software is Eating The World”, it was illustrated that we are in a dramatic technological and economic shift, in which the author expects many industries to be disrupted by software¹. In energy distribution, smart software agents are being actively developed where the agents autonomously decide on the distribution of electrical energy across the grid. In the logistics domain, the package delivery systems are accompanied with smart robots which are scaling up the delivery of goods efficiently and effectively.

Similarly, in the automotive industry the use of various software functions for digitalizing the vehicle sub-systems by the use of various sensors is a very active area of development and research. Recently, AI-enabled software functions are utilized where the vehicles are steered autonomously, while entertaining passengers. As software is exhibiting such **AI-enabled behavior**, autonomous control systems that may affect humans, or steer business-critical decisions are increasingly popular. **Validation and testing of these AI systems** is one of the new frontiers in software and systems engineering.

¹ Marc Andreessen, [Why Software Is Eating the World](#)

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

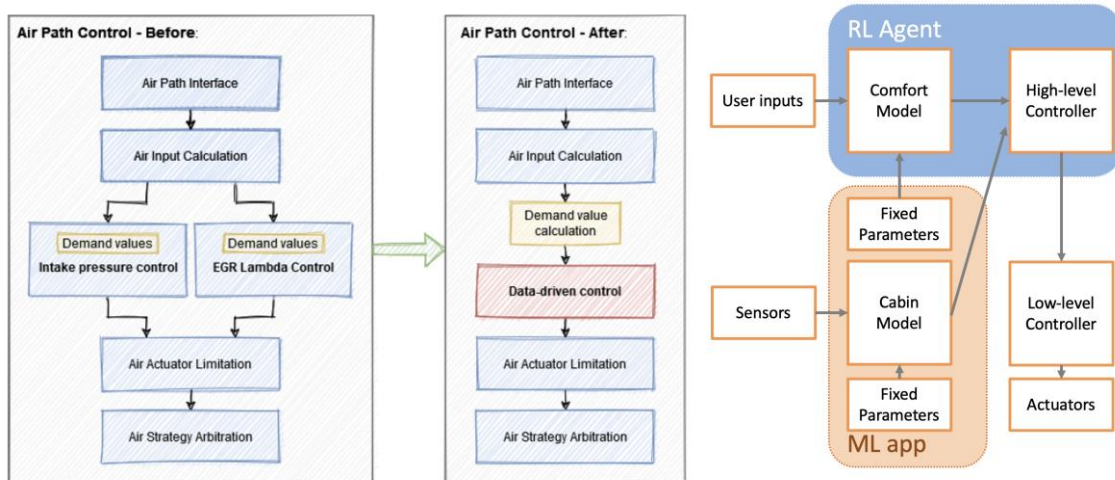


Figure 3 Intelligent multi-physics controls for the hybrid/electric powertrain: air-path and thermal management

In the FRACTAL project, AVL is designing, developing, and demonstrating the feasibility of **data-driven control strategies** for **multi-faceted systems** comprising components of various complexity levels. AVL will demonstrate how to substitute and/or augment a classical control strategy (i.e., rule-based implementation, hybrid automata) with a **data-driven control strategy** (i.e., using a ML approach such as for example, reinforcement learning). To take account the progress of the FRACTAL building blocks, AVL has divided its work into two main instantiations: control engineering for the air path (internal combustion engine) and thermal management. Whereas the first avenue is of interest for hybrid vehicles, the second addresses the needs for the highly integrated EV powertrain and enables several thermal management application scenarios such as predictive HV battery cooling, cabin heating mode selection, HV battery heating during driving and HV battery preconditioning in standstill.

3.3 UC3 – Smart Meter for everyone

UC3 targets a demonstrator of a secure IoT device that can read a meter by means of taking a picture of the mechanical meter display, analyzing it directly on the edge node, and transmitting the extracted information to the utility provider. The full pipeline is depicted in Figure 4 below:

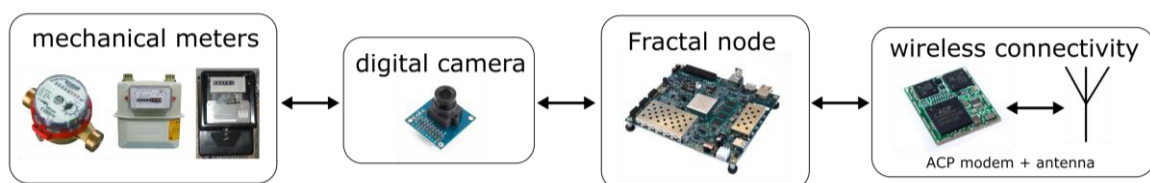
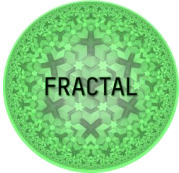


Figure 4 Smart meter pipeline used in UC3

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node	
	Title	Feature Verification Plan into Verification Use Cases	
	Del. Code	D7.1	

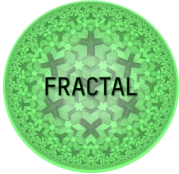
The main objectives of this use case are to achieve:

- **Low power consumption:** This is important because the device is going to be powered by a small sized battery. This restriction is mostly true for water, and gas meters, and less of a restriction for electricity meters.
- **Security:** Analyzing, and collecting data directly on the edge, namely on the device that is placed in the basements of a customer creates privacy concerns especially since such locations are typically not very well protected. The aim is to build a secure system that will handle the data with care and only store it encrypted.
- **Low price and size:** The proposed solution is a low-cost alternative to existing smart meters. It can only be successful if its price tag is small because the main benefit over existing solutions is that there is no need to replace existing meters with pricy smart meters, and there is also no need for extra electrification as the device is battery powered.
- **Connectivity to the cloud:** Meters are mostly located in basements where cellular reception quality is low. We target a cellular communication protocol, NB-IoT, that addresses this problem by supporting extended coverage.
- **Feature extraction with limited hardware resources:** Taking a picture of a meter and analyzing the image must be possible with limited amount of image resolution because of limited on-chip memory size.

Throughout the technical WPs of FRACTAL several features have been added to the platform such as security support through an open-source root-of-trust that will allow to implement a secure boot process, encryption and authentication services, or real-time aware caches and interrupt units that will allow to merge the FRACTAL node with the ACP modem and thus reduce the overall cost and power consumption of the whole system. Throughout WP7 the focus is on the integration and verification of those features.

3.4 UC4 – Low-latency Object Detection as a generic building block for perception in the edge for industrial application

Siemens objective with UC4 is to build a generic block that will be used in industrial application for detection and localization of objects. Deep neural networks (DNN) are nowadays widely used in different applications for computer vision because of the accuracy that they achieve, but they need high computational resources. In many applications, like automated defect detection on production line or autonomous robot arm movement, the DNN is also required to be run locally at the edge and many embedded devices have limitation with respect to computational resources.

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

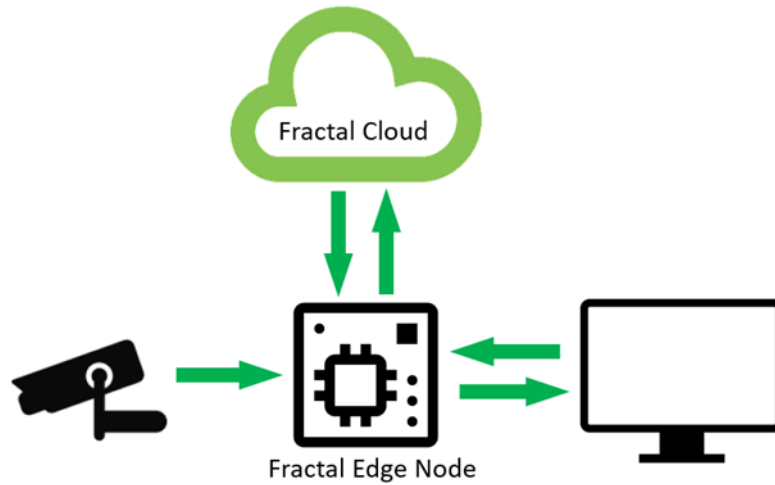
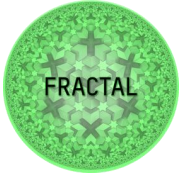


Figure 5 Object recognition on FRACTAL edge node

UC4 implements and evaluates the performances of YOLO, a DNN for object detection, on FRACTAL edge nodes. As shown in Figure 5, the building block receives the input from a connected camera and displays the output of recognized object on a monitor. When the DNN needs to be trained for a new object, it is pushed into FRACTAL cloud. To achieve the needed computation power on FRACTAL edge node, the UC uses the HW accelerator on the edge to run the inference of the DNNs convolutional layers faster. The presence of multiple Processing Elements (PE) on the HW accelerator gives to the FRACTAL edge node the capability to run in-parallel multiple operations that are part of the convolutional layer.

The first goal of the UC4 is to integrate the FRACTAL edge node, the Linux operating system, the system software for control of the HW accelerator and Yolo inference into a generic building block with attached camera for input generation and a display for output results.

The second goal is to evaluate the performance benefits of the HW accelerator on FRACTAL edge node for DNN execution. However, not all convolutional layers of DNN gain from the presence of a HW accelerator. For some convolutional layers the time cost to transfer the input data and network weights to the HW accelerator can be much higher than if the same remained in the CPU cache and was performed by a single CPU core. Therefore, a model is being devised to categorize a-priori the convolutional layers if they will be mapped on a HW accelerator or CPU.

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

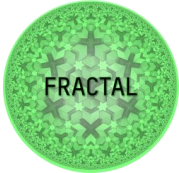
4 FRACTAL features considered in WP7

The FRACTAL project through its technical WPs has developed *features* that will enable systems that can be designed to operate with nodes at different levels (cloud, edge, mist). In this section, we identify six *features* out of FRACTAL developments from technical work packages that can be verified through the WP7 use cases.

Note that, every use case benefits from multiple FRACTAL *features*. However, some use cases have a better opportunity to show the benefits of the FRACTAL *features* than others. Table 1 shows a summary of the FRACTAL *features* we have considered in WP7 and the associated use cases that will be used to show the benefit of these *features* within the use case implementation. Most of these features are consistent with the upcoming D2.3. However, for this deliverable we have combined several sub-features described in D2.3 into a more generic '*Connectivity to Cloud, Communication*' feature for brevity purposes. For all but the *Openness feature*, we have identified one UC that will be the main driver for the verification. This has been marked with a ★ in the table below.

Table 1 Matrix showing the FRACTAL features considered in this WP and the UCs that lead the verification efforts of these features

Features considered	UC1		UC2	UC3	UC4
	<i>PROINTEC/ZYLK</i>		<i>AVL</i>	<i>ACP</i>	<i>SIEMENS</i>
	Dem1	Dem2			
Adaptability			★		
Security				★	
Cognitive Awareness	★	✓			★
Low Power				★	
Connectivity to Cloud Communication		✓	★	✓	✓
Openness			✓	✓	✓

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

4.1 Adaptability

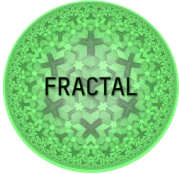
Main demonstration through UC2

UC2 addresses primarily the **FRACTAL project's feature of "adaptability"**. In a wider sense, the UC contributes to the development of **AI-enabled control strategies** and specifically needs to ensure that such **controlled systems continuously evolve** depending on perceived **context**. The context may have various factors, such as **geographical** (e.g. position), **physical** (e.g. emissions), **temporal** (e.g. assumption about the future) or **organizational** (e.g. regulatory compliance) affecting the control strategy.

Currently, control systems are designed with the help of human calibration, where rigorous efforts from calibration engineers are needed to make sure the controllers behave in an acceptable manner for various factors mentioned above. In order to reduce the effort of calibration and to improve the adaptability of a controller, AI-methodologies, like Reinforcement Learning (RL), are chosen. In the context of RL, there is an agent (in this case the controller) trying to do an action (in this case a control action like turning a valve, etc.). This is accomplished by observing the current state of the system and providing a suitable action. If the taken action is not sufficient, the system provides feedback in return which makes the agent adapt if necessary. This notion of adaptability can be applied irrespective of various factors affecting the environment.

In developing RL-based controllers, the key parameters to look at are the effort in training the model, model accuracy (comparing it to the human-calibrated controller) and the reduction in calibration effort. If the above parameters provide a satisfactory result, the transition to RL-based controllers will be widely accepted and will help in having an adaptable control strategy.

In UC2, the "adaptability" feature is realized with the help of dedicated software components at the application level. From an exhaustive simulation of the control strategy, we obtain a multitude of simulation traces representing the dynamic behavior of the control loop. A **machine learning pipeline** is in place to automate the workflow to produce a machine learning model. Our machine learning pipeline implements **reinforcement learning** consisting of multiple sequential steps that do everything from data extraction and preprocessing to model training and deployment. The initial model we obtain from this pipeline is deployed into the FRACTAL cloud alongside with an API, we can make use of simulation data to obtain the initial model and we retrain the model once we obtain new sensor data from a swarm of vehicles. Figure 6 illustrates the overall concept alongside with the used technology stack at the application level.

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

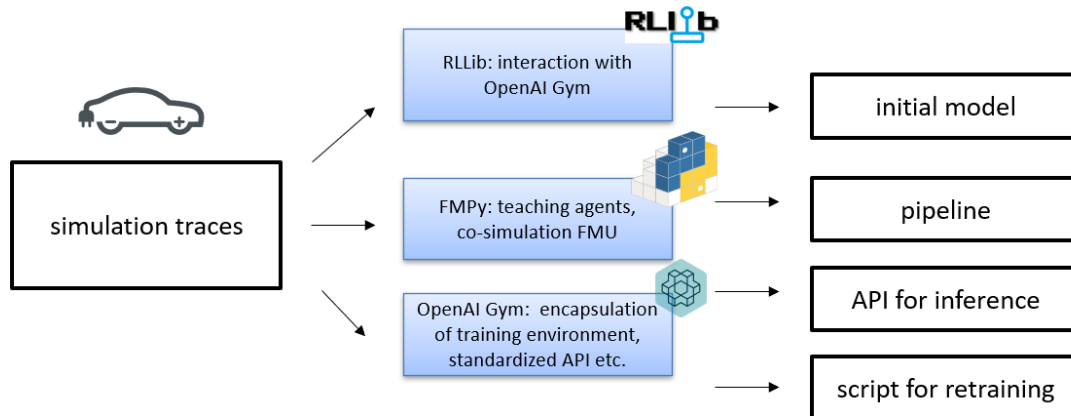


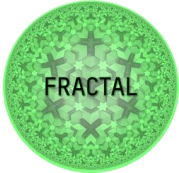
Figure 6 The technology stack alongside with the ML pipeline, initial model and the APIs for inference and retraining

In UC2, adaptability is enabled in terms of tailored use of the following SW-components:

- FMPy – simulation functional mockup units
- Open AI Gym - toolkit for developing and comparing reinforcement learning algorithms
- RL-Lib – universal API for distributed training of reinforcement learning algorithms

In addition to the contribution from AVL, adaptability is further provided through the work of **WP4**, which achieves adaptability at the node level using a hierarchical adaptive time-triggered multi-core architecture (HATMA). Energy efficiency in safety-critical systems, such as battery-powered devices, motivates adaptation at this level. Energy management strategies are used to control idle resources' energy consumption or manage the available energy to system resources efficiently. Energy efficiency can be used for economic or environmental reasons or as a safety justification to ensure that the battery capacity is sufficient for an application's duration.

In comparison to widely utilized high-cost approaches such as N-modular redundancy, adaptability at the node level minimizes the cost of fault-tolerance. Excessively duplicating system resources in cost-constrained systems with rigorous size requirements leads to a high price for fault tolerance. Adaptability in this context refers to the reconfiguration of node services to avoid the usage of failing resources. Computation of new time-triggered schedules specifying the temporal, spatial, and contextual distribution of system resources is involved in reconfiguration. In many circumstances, reconfiguration provides alternative computational and communications channels and sensory information from different sensors. When system resources are insufficient to support complete node services, the system switches to a degraded service mode.

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

At the node level, adaptability to changing environmental conditions requires guaranteeing specialized application-services based on the system's operational mode. Environmental changes in any scenario are context events that originate outside of the node. An example is the case of a heterogeneous multi-core architecture where performance cores are put in sleep-mode for low-demand applications, whereas applications requiring performance utilize all system resources. Dynamic service arbitration based on operational needs includes putting the system to sleep or making services and resources accessible for high-performance demands. Another example is the high-temperature level of passively-cooled electronics, which necessitates a reduction in computing load to avoid thermal damage. Similarly, certain battery levels inspire degradation notions, such as turning off comfort services in an electric car.

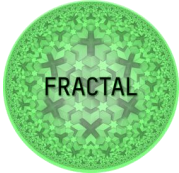
Using the adaptation logic of HATMA, developed in WP4, node resources are monitored for context events relevant to adaption. Context events are dynamic slack events that occur when jobs are completed earlier than expected on worst-case execution timescales. On the other hand, faulty or unavailable resources cause failure events. HATMA also monitors the entire system to ensure that system services adapt to current applications. To avoid system failure, the status of system resources must be consistent during an aligned schedule switch. The hierarchical interactive consistency protocol (HICP) of HATMA ensures that system resources have the same context information required for an aligned schedule change. An AI-based scheduler generates new schedules depending on the agreed-upon context information. Whenever the new schedule is generated, all system resources are switched to the new schedule.

4.2 Security

Main demonstration through UC3

Since the target node of UC3 is going to be placed in a non-secure environment which many people can access, and given the fact that sensitive data is going to be collected on the node itself, it is a must to provide a secure solution for the smart meter prototype. To guarantee that the system has not been altered before it is booting, the firmware must be verified before the system starts. This is typically achieved with a secure boot process, which verifies that the boot image has not been tampered with and only boots the system if it could successfully verify the image. Once the system has successfully booted, it is important to make sure that the user data (the gas, water, electricity meter data) is stored encrypted on a nonvolatile memory and that it remains encrypted until it reaches the utility provider.

In theory, a secure boot process and data encryption/decryption could be done in software. The disadvantage of performing those operations in software is that cryptographic functions are very compute intensive and would drain the battery a lot, which is why dedicated hardware blocks are preferred. In a system that wakes-up periodically, this will take place at least daily (if not hourly), which would account for a significant portion of the energy budget of the system. In addition, the full code

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

would need to be stored on an internal read-only memory, which does not come for free. The two main compute intensive functions for a secure boot process are:

- An algorithm to compute a hash of an image
- A signature verification algorithm

The open-source project OpenTitan² comes with all the required hardware blocks to implement a root-of-trust (RoT). The OpenTitan RoT not only offers a secure boot process, but also encryption, decryption, secure key generation and handling, and comes with numerous countermeasures against side channel attacks. Although the project is still under development, it is an ideal starting point to make the FRACTAL node secure thanks to the openness of OpenTitan.

Throughout the project, ACP will implement a secure boot process on OpenTitan by utilizing the hash accelerator that computes a sha256 and a signature verification algorithm that can run on the OpenTitan Big Number (OTBN) Accelerator. OpenTitan also comes with an aes256 block that will allow to encrypt user data using the Advanced Encryption Standard algorithm with 256-bit keys. Within UC3, this block can be used to encrypt the collected user data from the meters.

UC3 is going to be deployed in the field where support for a firmware upgrade over the air (FOTA) is a must. However, such upgrades can convey a big security risk. To support a secure FOTA, the new firmware needs to be signed, such that it can be verified on the edge device before it is activated. In addition, the firmware should be encrypted before it is sent to the edge device and decrypted on the node before being activated.

All this functionality will allow the low-end FRACTAL node to become secure, but it will also add a considerable amount of hardware, which can violate the low power consumption of the system. However, the desired functionality for UC3, a secure boot process, and encryption of user data can also be achieved with a subset of the functionality of OpenTitan. The optimal trade-off needs to be determined.

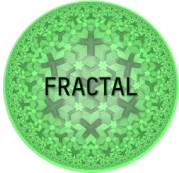
4.3 Cognitive awareness

Main demonstration through UC1 and UC4

One of the most representative features of the FRACTAL node is its capability to be cognitively aware of its environment, being able to take decisions depending on its internal state, the state of neighboring nodes and the scenario where it is operating. The range of applications that can benefit from the cognitive awareness feature (also known as context awareness within FRACTAL) is quite wide. For UC1 and UC4, this feature is the key component of their contribution.

In UC1, the application scenario features construction sites, which are ever-changing environments where the position and states of the elements in the edge scenario are of special relevance in terms of protection, both for the infrastructure and the working

² <https://opentitan.org> managed by lowRISC

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

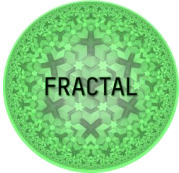
staff themselves. UC1 is focused on applying AI methods to improve safety in the overall construction site, and for this reason, Cognitive Awareness plays a leading role as one of the FRACTAL features to benefit from.

Demonstrator 1 of UC1 is being developed as a novel AI tool to segment cracks from images taken with an Unmanned Aerial Vehicle (UAV). The UAV will be controlled by an experienced operator, and the designed AI system will automatically process and analyze the images to gain knowledge of structural defects that may pose a future hazard to the construction. The designed system is based on a Convolutional Neural Network (CNN) with U-Net architecture and residual connections. In addition, the system has been improved by using image augmentation techniques (translations, rotations, superposition of different textures, etc.) and a manually labeled dataset. The training of the model has been done offline and several variations of the proposed architecture have been tested to better adapt the system to the use case. As well, the inference will run online and in real-time on the edge. In addition, the images taken by the UAV and the inference made by our model will be uploaded to the cloud, so they can be used by a structure's expert.

The system will be considered successful if it helps the technicians to better detect cracks. This can be measured in several ways. First, the quality of the Deep Learning Model will be evaluated with metrics such as IoU score and Categorical Cross Entropy. As well, several KPIs can be measured to assess the quality of the system. This model should reduce the costs and accidents inherent to "traditional" inspection methods that make use of special machinery. Moreover, it must help the technicians to better detect the number of cracks. Thus, it must result in a cognitive node that is aware of the defects of the structures (cracks) and introduces an improvement over a "traditional" method. This will be verified by technical metrics and, in a later stage, by the technicians who use this tool.

For **Demonstrator 2** of UC1, the main *objects* to be cognitively aware of are construction workers who can potentially be at risk while approaching dangerous or heavy machinery, and the sensitive machinery itself. A monitoring of the alarms produced when a staff member approaches a sensitive machine at a distance lower than a previously defined *safety-threshold* can be used to train AI models that predict when these alarms are likely to happen, how to manage them in the most efficient way (stopping the machinery on time, notifying in advance the workers that an alarm could happen soon...) and classify these alarms so the site manager is able to keep an overall safer working environment.

This cognitive awareness capabilities will be enabled by a sensor network deployed over the machinery and wearable devices. These sensors will be collecting data about the interactions between the construction workers and the machinery, whenever an approach that can be considered hazardous happens between them. Notice that these sensors never collect personal information from the workers nor interact with them in any way (wearable sensors do not trace the positions of the workers out of alarm areas, do not collect information on how much time the worker has been on each location), so no GDPR rules are broken. This is a relevant aspect

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

as data obtained from this sensor network is anonymized by default (the sensor ID is the only information present in the data, but not the worker's personal information), so in case a cyber-security leak happens, no personal data will be exposed.

The information given by the sensors will then be pre-processed in the edge and used as input for Machine Learning models, which will use the correlations between the time-series alarms and relative positions of the workers with the machinery to be aware of how the elements in the scenario interact. The given response based on Deep Learning algorithms demonstrates the cognitive awareness of the platform.

UC4 utilizes the cognitive awareness feature for industrial applications. Computer vision as part of cognitive awareness is a crucial component to extract meaningful information from the input and use this information to trigger different actions. The vision-based AI inference for object detection that runs on the *generic building block* makes the FRACTAL edge node aware of shapes and objects in the environment for which the inference is trained. The inference also has the capability to find the position of the detected objects within the observed environment. By generating these types of outputs, the UC4 generic building block can be applied as part of:

- production line to identify defective products,
- a warehouse monitoring system to update the inventory record by registering each component that goes in and out of the warehouse,
- a robotic arm to identify and locate the products so the arm knows exactly where it should intervene, or
- a safety system in an industrial environment by identifying workers when they are in area where they might be injured by the production line.

4.4 Low Power

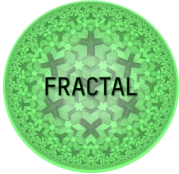
Main demonstration through UC3

One of the key features in the FRACTAL node is low power consumption. The smart meter of UC3 is an edge application where low power consumption is one of the key features to achieve a long battery, and thus product lifetime. Therefore, UC3 is perfectly suited to demonstrate the low power consumption of the FRACTAL node.

Power consumption consists of two components:

- Static power consumption (leakage)
- Dynamic power consumption

While functionality can be demonstrated on an FPGA based prototype, it makes little sense to optimize for power consumption on an FPGA. Since UC3 targets an ASIC implementation of the FRACTAL node, power consumption is optimized for an ASIC target rather than for an FPGA. To achieve a long battery lifetime, both static and dynamic power need to be optimized individually. Dynamic power scales with the clock frequency: The higher the clock frequency, the more switching activity in a

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

chip, and hence, the more power is dissipated. Dynamic power consumption can be optimized by means of clock gating, by using dedicated hardware accelerators to compute time-consuming, resource-intensive tasks faster, or by simply decreasing the clock frequency to a minimum that is acceptable to perform the desired workload. Static power consumption optimizations include power gating inactive parts of a chip, the use of low leakage transistors, and reducing the overall transistor count, or simply reducing the chip area. Further, dynamic and leakage power also depend on the supply voltage. Running a chip at a lower supply voltage will make the transistors slower, but at the same time also reduce the power consumption significantly.

The focus of UC3 lies mainly on optimizations of dynamic power consumption through clock gating and leakage power optimizations through the introduction of several power domains that can be individually turned on and off. Different sleep modes can be used to switch from one state to another. Since the node developed as part of UC3 is going to be inactive for the largest part of a day, the SoC needs a very efficient deep sleep mode in which only a few μA are consumed.

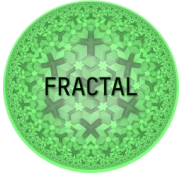
- Clock gating (dynamic power):
 - Automatic clock gating to gate registers when they are not used. Clock gating insertion is done by a synthesis tool.
 - In addition to automatically inserted clock gates, it is important to strategically place manual clock gates higher up in a clock tree to enable/disable full modules rather than only registers.
- Power domains and sleep modes (leakage power):
 - Introducing different sleep modes by splitting the chip into multiple power domains that can be turned on or off.
 - Software scheduling optimizations to minimize active times and enter deep sleep states as fast and often as possible.
- Area optimizations (dynamic + leakage power):
 - Merging the capabilities of ACP's modem with the functionality of the FRACTAL node will allow to significantly reduce the area with respect to a two-chip solution. Sharing resources such as memory, processors, and peripherals will allow for a smaller chip, consisting of fewer transistors, and hence consuming less power.

These optimizations will allow UC3 to perform successfully in the field.

4.5 Connectivity to cloud / Communication

Main demonstration through UC2, additional demonstration UC1 and UC4

Thinking about **data as streams** is a popular approach nowadays. In many cases, it allows for the building of data platform in a more efficient way than when thinking about data as a batch. There may be a need for dynamic routing of data to the proper data-processing node (e.g. as part of a dynamic data-routing function). The building block for data routing and preprocessing is usually a data middleware platform (or a combination of two or more such platforms). These address different technical

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

requirements, depending on the nature of the data that is collected and routed (e.g., streaming data, data at rest). For instance, stream-processing middleware platforms can be used to support the **routing and preprocessing of streaming data**. Data routing needs to consider technical aspects, like horizontal³ and vertical⁴ scalability, but also aspects resulting from data usage policies, like jurisdiction for data processing or combination with other data.

In UC2, we use Apache Kafka to control when data is processed. For example, as soon as enough new data becomes available, training (pre-training of models, re-training of models etc.) is meaningful. In this sense, we aim at constantly analyzing streams of data and associated metrics **and trigger actions at the cloud side** immediately after deviations are detected.

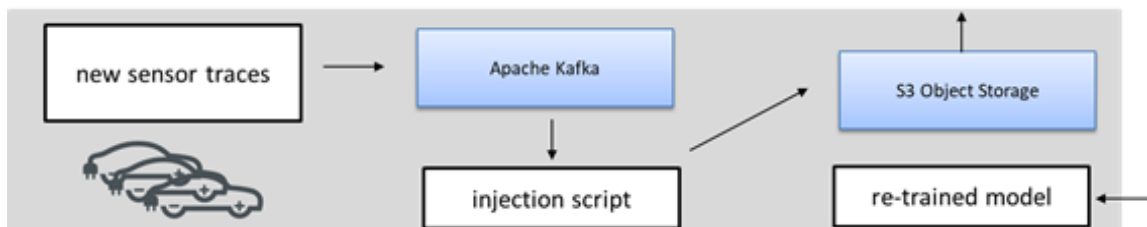


Figure 7 Injection of sensor data to perform the ML learning using Apache Kafka

In addition, the following use cases also expect to achieve objectives through the connectivity to cloud / communication feature of FRACAL:

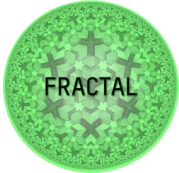
Demonstrator 2 of UC1 is an application that makes use of an edge communication framework. Edge computing has a clear trade-off with the cloud in terms of time invested to get the processes done and communication overhead, at the price of giving up high processing power capabilities. In this sense, it is clear that real-time communications should be a key target for edge architectures, thus keeping the communication overhead low and the networking as simple and fast as possible.

These real time responses can be achieved by using lightweight and IoT-oriented protocols like MQTT and highly specialized edge devices that can run ML models and inferences faster than their cloud counterparts.

Data can be processed and stored in the edge devices in such a way that the ML models provide a real-time response. These "real-time" capabilities clearly depend on the specific requirements of each use case, going from milliseconds for autonomous driving vehicles, which should provide as-quick-as-possible responses, to a few seconds in the case of very complex scenarios to be analyzed (video analysis, image processing...). It is then clear that a response would take much longer if the input data (usually large files) had to be sent to an external cloud, processed, downloaded back, and then responded to. This issue is aggravated when a large

³ adding additional computing nodes

⁴ adding more computing power to an existing node

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

number of sensors and IoT devices are generating data and creating networking bottlenecks.

In the case of UC1 – Demonstrator 2, the ML models must provide a quick response, and having real-time inferences is a must, because the safety of workers will be improved if a fast response is provided to forecasted hazardous events. These real-time capabilities are achieved by having a sensor network deployed at the edge, which collects data from the scenario, positions of workers and machinery, and sends alarms to the edge controller whenever the workers approach a machine from a dangerous area being monitored. These data can then be processed directly at the edge without a strong dependency to the cloud, which results in faster responses.

The prototype of **UC3** is going to be attached to a meter, which is more of than not located in the basement of a building with poor cellular reception. A communication standard that supports extended coverage is required to send data to the cloud. Within UC3, the low-end FRACTAL node will be connected to the cloud through a newer cellular IoT standard, NB-IoT.

In **UC4**, Siemens runs the inference on the FRACTAL node. When the AI algorithm needs to be retrained for the detection of new objects, the algorithm is pushed to the FRACTAL cloud. Once the training is performed, the new inference is brought back into the edge. The push and pull of the AI algorithm from the edge node to the cloud is done by utilizing the FRACTAL connectivity *feature*.

4.6 Openness

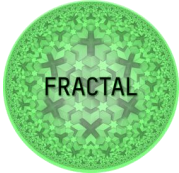
One of the distinguishing features of the FRACTAL project is trying to use open-source principles as much as possible. At the same time, from a business perspective, it is normal that some parts of the use case implementations need to contain an innovative component that sets them apart and allows the company that has invested in the implementation to generate revenue from their products.

We have identified several different aspects where openness is reflected in the use cases:

- Open standards
- Openly available datasets for verification
- Open-source libraries / software
- Open-source hardware

A second dimension of the openness evaluation is how openness manifests itself in the use cases

- Rely on existing openly available components for parts of the implementation
- Actively contribute to existing open-source projects as part of FRACTAL activity
- Develop new open-source projects as part of the developments within FRACTAL.

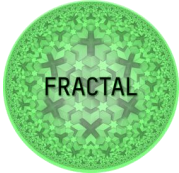
	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

Finally, a third dimension is the type of license(s) used for the open parts within the use cases. Although everyone is free to dictate the terms and conditions on how their contribution may be used openly, most would use a well-understood license that encapsulates how the open contributions can be used. A detailed discussion of available open-source licenses is beyond the scope of this deliverable⁵ but there are two main classes of open-source licenses:

- **Permissive** licenses do not have restrictions on where the open artefacts can be used. Apache/Solderpad, BSD, MIT licenses are all commonly used permissive licenses.
- **Reciprocal** licenses require that projects that embed the openly available artifacts be released with a license that is compatible with the open license. GNU Public License (GPL) is probably the best-known example.

Practically all use cases in WP7 make use of openness in one form or the other, and this deliverable will capture the extent of openness in each use case and identify open components, how they are used and what licenses are used for each component.

⁵ interested readers can find more information under <https://opensource.org/licenses>

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node	
	Title	Feature Verification Plan into Verification Use Cases	
	Del. Code	D7.1	

5 Verification plan for individual FRACTAL features

In this section, we will present a plan to determine, for each of the FRACTAL *features* identified in Section 4, how they improve the performance of the use cases and determine the KPIs associated with this improvement. The subsequent deliverable, D7.2, will explain the implementation details of how these measurements will be made, and finally D7.3 will present results on actual implementations.

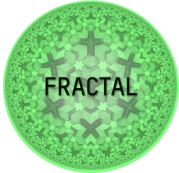
We present two levels of KPIs for each use case:

- **Technical KPIs:** refer to indicators that directly relate to measurable quantities obtained from the performance.
- **Business KPIs:** refer to indicators that bring an advantage in terms of the business, such as improvement in productivity.

5.1 Plan for verifying Adaptability

Because of the development of electric/hybrid and autonomous vehicles, software-enabled systems are becoming complex in the automotive industry, mainly in calibrated control systems. The effort to calibrate such a control system is growing exponentially, this triggered research in AI-enabled control systems which can continuously calibrate and self-adapt based on the perceived context. As described in section 4.1, UC2 implements the "Adaptability" FRACTAL *feature*. In UC2, an AI-enabled controller is developed using Reinforcement Learning algorithms. In these AI-enabled controllers, self-adaptive capabilities are provided by a Reinforcement Learning **runtime manager which executes adaptation policies** and algorithms based on the information sensed from the environment.

Consequently, because of the self-adaptation, stringent quality requirements like **performance and optimization** are expected to be implemented in the architecture and software of such systems. From the perspective of the **adaptability FRACTAL feature**, this demands a **continuous evolution of the adaptation capabilities** that aims to ensure the quality of the **performed runtime (re)-calibrated tasks**. Therefore, there is a clear need to monitor and evaluate the **quality of the adaptations and modified behavior of AI-enabled controllers**.

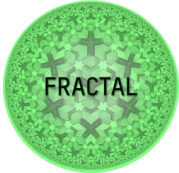
	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

UC2 will verify the following technical and business KPIs:

FRACTAL feature Adaptability	
UC2 contributes to the development of AI-enabled control strategies and enables a controlled system to continuously evolve depending on perceived context.	
Technical KPIs	<ul style="list-style-type: none"> The degree to which the control reaches its goals, deviation measured from Dynamic Programming solution should be on the scale of 5% or less for the reinforcement learning based solution. The data-driven control strategy should be better on average when compared to the reference strategy (i.e., the model/rule-based design of the control) for unseen drive cycle, measured by lower energy consumption on the cycle. <p>Note: Time / amount of data to trigger retraining is not an obstacle. Retraining can be done whenever the deviation of other measurements from KPI are noticed.</p>
Business KPIs	<ul style="list-style-type: none"> Reduction of calibration effort – at least 25% of PM. Responsiveness for unseen scenarios (unseen scenarios handled).

The University of Siegen for its HATMA solution foresees the following KPIs.

FRACTAL feature Adaptability	
WP4 contributes to the development of AI-based scheduling services and adaptability to context events through aligned schedule changes.	
Technical KPIs	<ul style="list-style-type: none"> Adaptation time to context events should be <10% of the available adaptation window for dynamic slack events. A context monitor is able to observe multiple attributes of a single resource.
Business KPIs	<ul style="list-style-type: none"> Lower cost for fault-tolerance through fault recovery and reduced redundancy degrees. Reliability under changing environmental conditions.

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

5.2 Plan for verifying Security

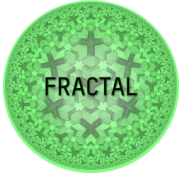
ACP plans to implement a secure boot process with OpenTitan. A first step is to map the system to an FPGA on which the security features can be implemented and verified. The secure boot process can then be implemented in software by utilizing the hardware blocks. This includes writing code for the boot ROM and the first flash section, the boot loader. The secure boot process itself is not timing critical but must also not take long as it will impact energy consumption. Verifying an image of a few megabytes within less than a second is a reasonable goal and will not keep the system active for too long.

To support a secure FOTA, the firmware needs to be signed and encrypted before being sent to the edge device. The whole process is not a very frequent operation and does therefore not need to be fast. But on the other hand, the FOTA should also not take many minutes, because it would drain the battery of UC3. Decrypting a 4MB firmware in one minute is sufficient.

During operation of the smart meter, the data to be encrypted is in the range of a few bytes to kilo bytes. Hence, the required throughput is low. The AES block could, however, also be used for other applications such as the modem, which runs real-time critical code and can require throughputs of up to 128 kB/s.

Data encryption and decryption are best verified on the FPGA platform. The execution times of these functions directly depend on the operational frequency and can be measured with a cycle counter.

FRACTAL feature Security	
UC3 contributes to the implementation of a secure boot process with the goal of making the FRACTAL node secure and to be able to encrypt user data directly on the edge in an energy-efficient way.	
Technical KPIs	<ul style="list-style-type: none"> • Possibility to verify a firmware during the boot process (firmware verified during boot). • The verification of a 4 MB firmware should take less than 1s. • Being able to encrypt/decrypt data on the edge device (en/decryption capability) • AES256 encryption/decryption of a 16 kB block in < 100ms • AES256 encryption/decryption of a 4 MB firmware in less than 1 minute.
Business KPIs	<ul style="list-style-type: none"> • Having a secure boot process is a big selling point for an SoC

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

5.3 Plan for verifying Cognitive Awareness

The objective of **UC1 Demonstrator 1** is the detection of cracks in concrete structures. This project will replace the current visual inspection methods that require costly auxiliary machinery, traffic interruptions and can be a source of accidents.

In order to demonstrate the cognitive awareness component, the crack perception ability of a technical expert will be compared with the performance of the AI algorithm. The KPI "number of cracks detected" will be used to evaluate this component.

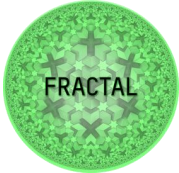
Another capability that will be measured is the algorithm's ability to differentiate cracks from other imperfections such as graffiti, paint chipping, aggregate nests, etc. If the algorithm is able to detect cracks with high accuracy on different surfaces and structures, it will be considered a successful system. This "imitation" of human cognitive function will be achieved by retraining the model and by using our image augmentation module, which overlaps different textures with the original images. One way to measure this component is by using the KPI "average performance difference (number of cracks detected) between the technical expert and the algorithm by inspection". Notice that the mean performance should increase with our system, and there must be a minimum accepted performance (compared with the performance of a technical expert).

Finally, the main objective of the system is to reduce the number of accidents inherent to "traditional" inspection methods. Thus, the KPI used to measure this component will be the "reduction of the exposure time of the site worker" during the work activity on field.

The main objective of the **UC1 Demonstrator 2** is to improve occupational safety and health in civil works environment. Cognitive awareness is a novel feature in these kind of construction scenarios. Although Machine Learning models were available and have been applied before to construction sites, their applicability was limited to predictive maintenance works, anomalous functioning of machines, or other kind of machine-related aspects. Being able to cognitively adapt to the environment and the present elements in the scenario opens a gate to new models being deployed.

The KPI for measuring the impact of cognitive awareness capabilities in the construction scenario would be the monitoring of the IoT platform data model's performance in terms of accuracy and reliability.

An alarm prediction model that is trained by just knowing the overall number of people and machinery on the working site should be outperformed by a model that actively knows how many people and machines are involved in each alarm event. The model should work better if the number of people on site can be detected automatically. A comparison between the model's performance before and after enabling cognitive awareness would be enough to determine whether cognitive awareness is a relevant aspect in the model's accuracy.

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

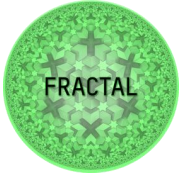
FRACTAL feature Cognitive Awareness	
UC1 contributes to the development of AI-based cognitive awareness feature to make the FRACTAL edge node aware of its environment by identifying the surrounding.	
Technical KPIs	<ul style="list-style-type: none"> (UC1 Demonstrator 1) Percentage of the number of cracks detected by the algorithm with respect to those detected by an expert > 95% (UC1 Demonstrator 2) The model works better when the number of people in the scenario can be detected automatically (Improved quality with people detection)
Business KPIs	<ul style="list-style-type: none"> (UC1 Demonstrator 1) Reduction of the exposure time of the site worker >50% (UC1 Demonstrator 2) Make the system capable of determining how many people there are on stage at any given moment and configure itself based on this (People detection capability)

For **UC4** The goal is to utilize HW developed in WP3 for performance improvement on object recognition and localization when DNN inference is run on the FRACTAL edge node. The KPI for UC4 will be the evaluation of the impact that the HW accelerator has on performance of the FRACTAL edge node when images for object recognition are processed. The higher the rate of processed frames, the better the performances are.

FRACTAL feature Cognitive Awareness	
In UC4, the HW developed in WP3 will be used to detect and locate objects.	
Technical KPIs	<ul style="list-style-type: none"> The frame rate at which the input images will be processed (> 5fps).

5.4 Plan for verifying Low Power

Since we target ultra-low power consumption, the main implementation target is an ASIC, the FPGA evaluation platform is used to verify the functional correctness of the implementation, and to characterize the active and idle times, which are used to compute the total energy. When aiming for an ASIC, the power consumption can only be analyzed with a specific target in mind because it will depend heavily on the chosen technology. However, the relative improvements are, to some degree, technology

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

independent and can be ported to a different technology. In UC3, we target a mature 110nm CMOS technology in which the power consumption is analyzed.

To accurately characterize the power consumption of the system, it needs to be fully implemented in RTL, synthesized, mapped to a technology, placed and routed. The most significant modes then need to be simulated with a back-annotated post-layout simulation, and the switching activity of each mode can then be exported and analyzed in the place and route tool in typical case conditions. To analyze different implementation flavors, the whole process can be repeated with different options. E.g., with clock gating optimizations, with power domains and with area optimizations.

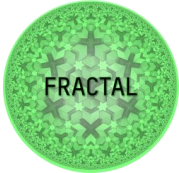
In UC3, the target battery lifetime with a 2200mAh battery should exceed 5 years assuming that the meter is read once per day and data is regularly transmitted to the cloud. If the active time remains short, the overall energy will be heavily dominated by the deep sleep power consumption. Clock gating aims to improve the active power consumption while leakage power optimizations are mainly targeting the deep sleep power consumption. To achieve a long-lasting battery lifetime, the deep sleep power consumption should be below 10 μ A.

The energy can be computed as:

$$\text{Total Energy} = \text{power}_{\text{sleep}} \times \text{time}_{\text{sleep}} + \text{power}_{\text{active}} \times \text{time}_{\text{active}}$$

A total energy of 2200 mAh and a 5-year lifetime target leaves only 1.2 mAh per day of which 0.25 mAh (assuming 10 μ A deep sleep power) are already used up by the deep sleep power consumption. Hence, it is not only enough to reduce the overall power consumption, but also important to minimize the active times. In the case of UC3, the active time is dominated by the time that is required to take a picture, analyze it and transmit the relevant information to the cloud.

FRACTAL feature Low Power	
As part of UC3, the power consumption of the FRACTAL node will be optimized such that the low-end node based on Pulpissimo can achieve a multiyear battery lifetime. The key to a long battery lifetime is an ultra-low power deep sleep state.	
Technical KPIs	<ul style="list-style-type: none"> • Deep sleep current consumption < 10 μA • Idle power reduction of > 10 % thanks to clock gating • Battery lifetime with a 2200mAh in the order of > 5 years
Business KPIs	<ul style="list-style-type: none"> • Reducing the area of an IC not only reduces its power consumption, but also its price (>20% area reduction)

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

5.5 Plan for verifying Connectivity to Cloud / Communication

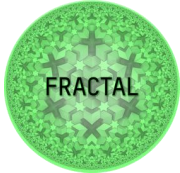
With the FRACTAL feature “Connectivity” to the Cloud we address a challenge that arises when **data providers** (data sources) and **data consumers** (data receivers) are connected directly. For example, when the underlying systems are directly connected, the capability to cache the data might be limited, e.g., when the receiver/or provider is not available. In addition, a data provider can overload the data consumer, when it sends the data faster than the data consumer can receive or process it. Connectivity to the cloud is implemented using Apache Kafka which acts as a **messaging system between the data provider and the data consumer**. Unlike a simple message queue, connectivity on the FRACTAL platform operates in a **fault-tolerant** manner and is **extremely scalable**.

In UC2, we use Apache Kafka to enable the storage of streaming data and to provide data with high availability. In this way, we assure that the data can be collected and **aggregated in real time**. Once enough data is available, we can **trigger the reinforcement learning process** and data can be leveraged to alter the ML model. Regarding UC2, the following KPIs reflect Cloud Communication:

FRACTAL feature Cloud Communication	
UC2 contributes to the FRACTAL feature Cloud Communication as it uses Apache Kafka for data routing when it comes to pre-training of the ML model.	
Technical KPIs	<ul style="list-style-type: none"> • Number of pushed messages for pre-training / frequency of messages: up to 100 messages per day per IoT device. • Size of batches – > 1MB • Number of pulled messages for pre-training / frequency of messages: up to 100 messages per day per IoT device.
Business KPIs	<ul style="list-style-type: none"> • > 99% of information sent to Kafka is retained and consumed.

From UC1 Demonstrator 2 perspective, the key component is not the communication with the cloud itself, but the data offloading into the cloud after the data have been processed and value obtained from them, to keep a historical data tracking in a centralized environment in case a retrain of the models is necessary.

The KPI would be to evaluate if the platform is able to provide full ML functionality (data ingestion, processing and inference), while being totally independent from the cloud, and then offloading the unused or no longer necessary data into the cloud.

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

Cloud independency is a clear KPI here in terms of processing capabilities and ML functionalities in the edge.

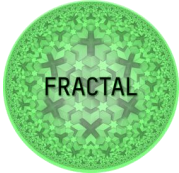
FRACTAL feature Connectivity: Communications	
UC1 Demonstrator 2 requires real-time responses for alarms and early detection. This feature is not present in similar solutions.	
Technical KPIs	<ul style="list-style-type: none"> UC1 Demonstrator 2: Real-time response capabilities (Yes/No)
Business KPIs	<ul style="list-style-type: none"> Reduction of latency and actions taken (>20%). Data have no longer to be evaluated from day to day and predicted alarms can be handled in real time, overall reducing risks and accidents (real-time capability).

For UC3, the contribution will be to extend the coverage through the NB-IoT modem.

FRACTAL feature Connectivity to Cloud / extended coverage	
Within UC3 the FRACTAL node on an FPGA will be connected to the NB-IoT modem of ACP and controlled with AT-commands.	
Technical KPIs	<ul style="list-style-type: none"> Establish a connection to the cloud through NB-IoT in poor SNR conditions (-10dB). Send small packets containing meter data to a webserver (Can send packets to server)

Finally, for UC4 the connectivity will allow new models to be transferred between the cloud and the edge.

FRACTAL feature Connectivity to Cloud Connectivity	
UC4 performs the retraining of the AI algorithm in the FRACTAL cloud. The communication between edge and cloud is utilized to transfer the algorithm for retraining.	
Technical KPIs	<ul style="list-style-type: none"> Successful transfer of AI-algorithm between FRACTAL edge and FRACTAL cloud (Transfer between edge and cloud successful).

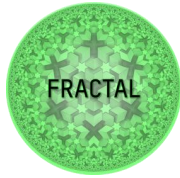
	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

5.6 Plan for verifying Openness

To verify openness in the actual implementations within use cases we have chosen to provide a table that captures the open components within each use case implementation as described in subsection 4.6. This table will be populated as WP7 progresses and will be completed within D7.3. The goal of this table is not to grade the use case implementations, which will have varying amounts of proprietary components, but to capture the openness of each use case properly. Verification will be established by the availability of the stated components by the provided links.

UC1: Engineering and Maintenance Works (Demonstrator 1)		PROIN/ZYLK
	Open Standards	License
	Tbd	
	Open Datasets	
	Tbd	
	Open-Source Software / Libraries	
	Tbd	
	Open-Source Hardware	
	Tbd	

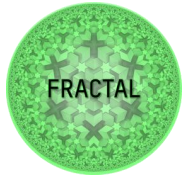
UC1: Engineering and Maintenance Works (Demonstrator 2)		PROIN/ZYLK
	Open Standards	License
	Open Datasets	
	No base open datasets to be used. However historical data will be opened once collected.	
	Open-Source Software / Libraries	
	Python Flask API	MIT
	Tensorflow (https://github.com/tensorflow/tensorflow)	Apache License 2.0
	MLBuffet (https://github.com/zyklab/mlbuffet)	GNU Affero v3
	Linux OS (Debian distributions - https://debian.org)	GNU General Public License v2
	Open-Source Hardware	



Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node	
Title	Feature Verification Plan into Verification Use Cases	
Del. Code	D7.1	

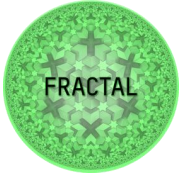
UC2: Automotive-Airpath-Control		AVL
	Open Standards	License
	Tbd	
	Open Datasets	
	Tbd	
	Open-Source Software / Libraries	
	RLlib – a scalable reinforcement library (https://github.com/ray-project/ray)	Apache License 2.0
	OpenAI Gym – a toolkit for developing and comparing reinforcement learning algorithms (https://github.com/openai/gym)	The MIT License
	Open-Source Hardware	
	Tbd	

UC3: SmartMeter		ACP
	Open Standards	License
	Tbd	
	Open Datasets	
	Tbd	
	Open-Source Software / Libraries	
	Tbd	
	Open-Source Hardware	
	OpenTitan (https://github.com/lowRISC/opentitan)	Apache v2.0 (permissive)
	PULPissimo (https://github.com/pulp-platform/pulpissimo)	Solderpad v0.51 (permissive)



Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node	
Title	Feature Verification Plan into Verification Use Cases	
Del. Code	D7.1	

UC4: Low-latency Object Detection as a generic building block for perception in the edge for industrial application		SIEM
	Open Standards	License
	Tbd	
	Open Datasets	
	Tbd	
	Open-Source Software / Libraries	
	Darknet (https://pjreddie.com/darknet/yolo/)	
	Open-Source Hardware	
	CVA6 / Ariane (https://github.com/openhwgroup/cva6)	Solderpad v0.51 (permissive)

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

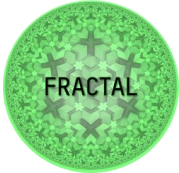
6 Conclusions

This deliverable has identified six key FRACTAL *features*

- Adaptability
- Security
- Cognitive Awareness
- Low power
- Connectivity to Cloud / Communications
- Openness

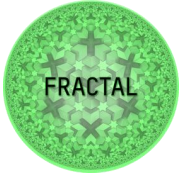
And has described how these FRACTAL *features* are expected to contribute to use cases that are part of WP7. Then the deliverable has described how FRACTAL partners plan to validate the efficacy of these FRACTAL features and has provided KPIs that will be used to measure their success.

Implementation details of each use case will be described in the subsequent D7.2, and finally the validation results will be presented in D7.3.

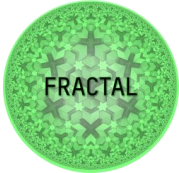
	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

7 List of Abbreviations

ACAP	Adaptable Compute Acceleration Platform
AE	Adaptable Engines
AES	Advanced Encryption Standard
AES256	Advanced Encryption Standard with 256 bit keys
AHB	Advanced High Performance
AI	Artificial Intelligence
AIE	Artificial Intelligence Engines
AMBA	Advanced Microcontroller Bus Architecture
API	Application Programming Interface
ASIC	Application Specific Integrated Circuit
AXI	Advanced Extensible Interface
CCIX	Cache Coherent Interface for Accelerators
CNN	Convolutional Neural Networks
COTS	Components of the Shelf
DDR	Double Data Rate
DMA	Direct Memory Access
DNN	Deep Neural Network
DRAM	Dynamic Random-Access Memory
DSP	Digital Signal Processing
EV	Electric Vehicle
FOTA	Firmware update Over The Air
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
GDPR	General Data Protection Regulations
GPU	Graphical Processing Unit
HATMA	Hierarchical Adaptive Time-triggered Multi-core Architecture
HICP	Hierarchical Interactive Consistency Protocol
HeSoC	Heterogenous System on Chip
HV	High Voltage (battery)
HW	Hardware
ID	Identifier
I/O	Input / Output
IoT	Internet of Things
IoU	Intersection of Union (score)
ISA	Instruction Set Architecture
KPI	Key Performance Indicator
ML	Machine Learning
MPSoC	Multi-Processor, System on Chip
MQTT	Message Queuing Telemetry Transport
NB-IoT	Narrow Band Internet of Things (communication standard)
NoC	Network on Chip
OS	Operating System
OTBN	Open Titan Big Number (accelerator)
PCIe	Peripheral Component Interconnect Express
PE	Processing Element
PL	Programmable Logic
PMU	Performance Monitoring Unit
PPA	Power Performance Area
PS	Processing System
PULP	Parallel Ultra Low Power

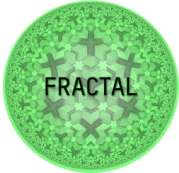
	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

PVT	Process Voltage Temperature
RL	Reinforcement Learning
RoT	Root of Trust
RTL	Register Transfer Level
SDK	Software Development Kit
SIMD	Single Instruction Multiple Data
SW	Software
TLB	Translation Lookaside Buffer
UAV	Unmanned Aerial Vehicle
UC	Use Case
VLIW	Very Large Instruction Word
WP	Work Package
WSN	Wireless Sensor Network
XRT	Xilinx Runtime

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

8 List of figures

Figure 1 The Big Picture of FRACTAL (Apr 2022)	7
Figure 2 The wireless sensor network for safety at construction sites developed as part of UC1	9
Figure 3 Intelligent multi-physics controls for the hybrid/electric powertrain: air-path and thermal management.....	10
Figure 4 Smart meter pipeline used in UC3	10
Figure 5 Object recognition on FRACTAL edge node	12
Figure 6 The technology stack alongside with the ML pipeline, initial model and the APIs for inference and retraining	15
Figure 7 Injection of sensor data to perform the ML learning using Apache Kafka ..	21

	Project	FRACTAL: Cognitive Fractal and Secure Edge Based on Unique Open-Safe-Reliable-Low Power Hardware Platform Node
	Title	Feature Verification Plan into Verification Use Cases
	Del. Code	D7.1

9 List of tables

Table 1 Matrix showing the FRACTAL features considered in this WP and the UCs that lead the verification efforts of these features.....13