# D5.3 Theoretical study of Fractal AI

| | |
|---|---|
| Deliverable Id: | **D5.3** |
| Deliverable Name: | **Theoretical study of Fractal AI** |
| Status: | **Final** |
| Dissemination Level: | **PU** |
| Due date of deliverable: | **M24** |
| Actual submission date: | **M24** |
| Work Package: | **WP5** |
| Organization name of lead contractor for this deliverable: | **UOULU** |
| Author(s): | **Susanna Pirttikangas**<br>**Mickaël Bettinelli, Henna Kokkonen, Alfonso Gonzalez Gil** |
| Reviewers: | **AITEK, VIF** |
| Partner(s) contributing: | **ZYLK, RULEX, AITEK, HALTIAN** |

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

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

**Abstract:** Task 5.1 studies the theoretical aspects of Fractal system. This deliverable synthesizes the outputs of Fractal theoretical studies that have been published during M1-M20. The content of the scientific articles published in this project represent several different aspects related to intelligence and efficiency of fractal-like systems. In this deliverable, Fractal concepts and definitions for Fractal AI are defined. Second, project scientific publications are listed and summarized. Third, use case related AI studies are identified and summarized. Finally, publications related to safety, sustainability and energy-efficiency are shown. In the final section, this deliverable links the theoretical studies to the implementation plans and activities of integration towards a fully operational intelligent Fractal system.

# Contents

1 History ... 4

2 Summary ... 5

3 Introduction ... 6

3.1 Objectives and Approaches ... 6

4 Concepts and definitions ... 7

4.1 Multiagent paradigm & Agent autonomy ... 8

4.2 Orchestration paradigms ... 8

4.3 AI methods ... 11

4.3.1 Distributed learning techniques ... 12

4.3.2 Decision-making ... 13

5 Theoretical studies ... 15

5.1 AI for edge ... 15

5.1.1 Vision ... 15

5.1.2 Analysis ... 17

5.1.3 Method ... 18

5.1.4 Synthesis ... 19

5.2 AI on edge ... 20

*5.2.1* Fractal features: Safety ... 23

5.2.2 Fractal features: sustainability, energy-efficiency and low power ... 24

6 Links to other deliverables and implementation ... 26

6.1.1 T5.2 Fractal AI Platform ... 26

6.1.2 T5.3 Applied Fractal AI ... 27

6.1.3 T6.1 Edge node design and implementation ... 28

7 Conclusions ... 29

8 Bibliography ... 30

9 List of figures ... 34

10 List of tables ... 35

11 List of Abbreviations ... 36

# 1 History

| Version | Date | Modification reason | Modified by |
|---|---|---|---|
| 0.1 | 2022-04-11 | Initial draft | Susanna Pirttikangas / UOULU |
| 0.2 | 2022-07-01 | Polished by chapter review | Susanna Pirttikangas / UOULU |
| 0.3 | 2022-08-29 | Polished by partner review | Henna Kokkonen, Susanna Pirttikangas / UOULU |

# 2 Summary

This deliverable belongs to Task 5.1 studying the theoretical aspects of Fractal system. In this deliverable, we synthesize the outputs of Fractal theoretical studies that have been published during M1-M20. The main features of Fractal have been defined in WP2, and this deliverable synthesizes the theoretical studies based on this classification. However, the scientific articles published in this project represent a larger umbrella of different aspects related to intelligence and efficiency of fractal-like systems.

First, this deliverable defines concepts and definitions for Fractal AI. Second, we list and synthesize all the published papers on *AI for edge*, summarizing the visions and methods presented in the papers. Third, use case related AI studies are identified and summarized. Finally, publications related to safety, sustainability and energy-efficiency are shown. In the final section, this deliverable links the theoretical studies to the implementation plans and activities of integration towards a fully operational intelligent Fractal system.

# 3 Introduction

## 3.1 Objectives and Approaches

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).

In task 5.1, the theoretical aspects, as well as the reflections of theoretical studies towards possible implementation paths of the recursive Fractal system are studied. As each node is expected to have autonomy to a certain degree, **the focus of the task is the interplay between subsystems and nodes, comprising a novel, distributed learning, decision-making and data analytics architecture.**

In this deliverable, we synthesize the outputs of Fractal theoretical studies that have been published during M1-M20. The main research questions related to the studies are listed in Table 1. The research method used for FRACTAL T5.1 is constructive.

| | |
|---|---|
| RQ1: What is the state of the art in distributed learning and control? | What is the current knowledge on distributed control, learning and data that is relevant to the Fractal system? |
| RQ2: What kind of methodology and intelligent capabilities are required 1) AI for edge and 2) AI on edge. | What kind of novel decision-making, learning and data architectures are suitable for Fractal subsystem interoperability and enhanced decision-making? How to mitigate varying levels of co-operation and local decision-making between Fractal subsystems? How can we decide the level of autonomy of the individual nodes and subsystems? |
| RQ3: What are the proposed strategies for implementation? | What are the practical implementation paths for Fractal AI? This RQ is mainly approached through the links to other deliverables of the Fractal project. |

Table 1. D5.3 research questions

# 4 Concepts and definitions

In this section, we explain the main definitions and concepts behind the Fractal system theoretical studies. The Fractal features have been defined in Fractal WP2 and the upmost level of features is shown in Figure 1. (The figure is updated based on the project advancements. Therefore, the final feature listing can be different from the figure below.)

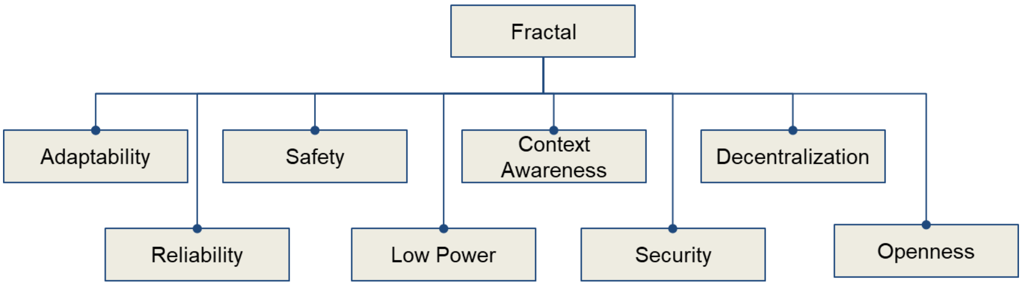


Figure 1. Fractal features (modified from deliverable D2.3).

In this deliverable, we focus on the features that that can be addressed by algorithmic solutions and are linked to the high-level behaviours of the Fractal node. For this reason, we will focus on the *adaptability*, *context-awareness* and *decentralization features* of the Fractal overall architecture. Other features such as *reliability*, *low-power*, *safety*, *security*, and *openness*, which are more related to the hardware architecture of the node or to low-level mechanisms, are outside the scope of this document. However, we list the scientific publications related to some of the lower-level features in section 4.2.

Table 2 associates the relevant concepts from AI literature to the Fractal features. The first one, *decentralization*, refers to the distribution of the system operation. Distribution of the decision-making can be achieved with, e.g., a multiagent paradigm (Weiss, 2013) where each node of the system is seen as an autonomous agent. Autonomy refers to an agent's ability to achieve its goal without any external control, by, e.g., administrators or other agents. Modelling nodes as autonomous agents supports loose coupling of the system components and brings more adaptability into the nodes' behavior (Mämmelä & Riekki, New network architectures will be weakly coupled, 2022). Decentralized behaviors require distributed mechanisms to operate, such as distributed learning which allows agents to learn, infer and adapt to uncertain and evolving environments.

*Context-awareness* is a feature referring to the ability of the node to perceive and leverage available information in the environment of the node. It can be any information that characterizes the situation of the node such as the time, the available data from neighboring nodes, etc. *Adaptability* represents the capacity of a node to change its behavior depending on its objective, its knowledge or even its context. To enable context awareness and adaptation, we need **mechanisms** to allow the system to decide how to allocate tasks among the nodes or how to distribute resources of the system. These mechanisms include **orchestration paradigms** (Kokkonen, ym., 2022)**,** which has been one large topic in Fractal theoretical studies. To enable decision-making, learning techniques that can provide new ways to make agents aware of their situation and adaptable to uncertain and variable environments are required. The following sections will define these concepts in more detail.

| **Fractal features** | **Related AI concepts from the literature** |
|---|---|
| Decentralization | Multiagent paradigm, orchestration paradigms, distributed learning |
| Context-awareness & Adaptability | Orchestration paradigms, decision-making |

Table 2. Association of AI concepts to Fractal features

## 4.1 Multiagent paradigm & Agent autonomy

Distributed artificial intelligence is required when individual components of a learning system do not have enough information or resources to achieve their objectives. This is a common situation in the edge-cloud continuum. Here, multi-agent systems (MAS) can be used to provide cooperative and collaborative capabilities as the agents can communicate their understanding of the environment, and their progress towards their objectives. Agents here refer to a computational abstraction which a) has externally set objectives and 2) can affect its environment with actions that bring it closer to achieving those objectives. Agents may possess various degrees of intelligence, defined by their reactivity, sociality, proactivity, and learning capability (Weiss, 2013). Reactivity relates to an agent's alertness towards environmental changes, sociality to its interaction with other agents and proactivity to its capability to predict and make changes to its own behavior.

Autonomy refers to an agent's ability to make independent decisions on how to reach its objectives, without any influence of external authority such as users, administrators, or other agents. On distributed application level, agents have distinct roles and behaviors, and they need to negotiate and share their resources – the system behavior emerges through the actions and interactions of the autonomous or partially autonomous individual agents, with the guidance of an orchestrator or through a choreography of the autonomous participants. In open systems, such as the Internet of Things (IoT), a MAS often needs to dynamically reorganize itself to adapt and to evolve in response to changes in the participating agents or in the environment. These aspects ultimately facilitate individual and collaborative learning to improve operations towards common goals and proactive behavior(s).

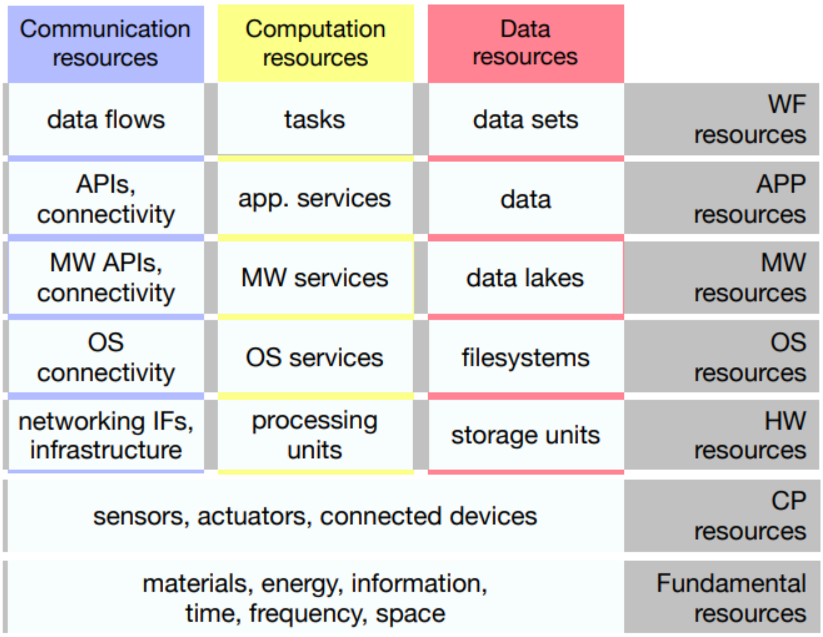
## 4.2 Orchestration paradigms

Computational resources span over the network infrastructure, all the way from a central data center to the user at the edge of the network. Several architectural approaches, including fog and edge computing, mobile cloud computing (MCC), and multi-access edge computing (MEC) (Taleb, 2017), (Ranaweera;Jurcut;& Liyanage, 2021), take advantage of these resources, expanding the cloud computing paradigm. Although each approach encompasses its own paradigm and requirements, they bring services closer to the user while simultaneously addressing challenges inherent in edge application deployments, such as latency requirements, bandwidth constraints or energy utilization. The cloud, with ample resources for computing and storage, is still often a necessity, calling for hybrid edge–cloud architectures (Xiong, 2018), (Brabakaran, 2020), (Yuan, 2020), or even a continuum of computational resources between the devices and the cloud, where applications can choose the best resource usage policy based on current needs (Balouek-Thomert, 2019), (Dustdar;Pujol;& Donta, 2022). Accordingly, in this document, we collectively refer to edge and fog computing, MEC, and other similar distributed computing approaches with heterogeneous and opportunistic resources by the term computing continuum.

Synthesizing the taxonomies in a number of recent works (see e.g. (Mampage;Karunasekera;& Buyya, 2022) (Costa, 2022) (Hong, 2020) (Toczé & Nadjm-Tehrani, 2018) (Zhong, 2021)), we take a holistic approach to the resources in the computing continuum. These resources, as depicted in Figure 2, are present on a number of levels, ranging from fundamental resources such as energy or time (see (Mämmelä;Riekki;Kotelba;& Anttonen, 2018)), through cyber-physical (CP), hardware (HW), operating system (OS), middleware (MW) and application (APP) resources, finally to workflow (WF) resources catering to the highest-level application business logic or clients. In this hierarchy of levels, higher level resources rely on the lower ones to fulfill their function. Further, from hardware level up, the resources can be divided in three distinct categories, namely, communication, computation, and data-related. It should be noted that this hierarchy of levels does not constitute a layered architecture in the sense that a level would only be aware of its immediate lower level. For example, data sets on the workflow level may be sourced from sensors on the cyber-physical level.

This holistic viewpoint is not emphasized in many of the related studies. These studies often refer to entities on HW and OS levels as resources, and entities on MW and application levels as services. However, there is considerable ambiguity in these conventions, and we find that an explicit consideration of *Everything as a Resource* (EaaR) simplifies the overall view.

The cyber-physical resources of a computing continuum may comprise sensors, actuators, and other connected user devices which have a physical form and function. They may act as sources (sensors) or sinks (actuators) of data flows in the computing continuum. Hardware resources in the communication category comprise, for example, network interfaces, access points, and base stations. Computational hardware resources refer to processing units (e.g., CPU, GPU, AI related accelerators), whereas data-related hardware resources include, for example, hard drives and SSDs. OS resources include, for example, connections (and related abstractions such as sockets), OS services such as processes (threads), and filesystems, as well as support for virtual machines (VM) and containers.



| Communication resources | Computation resources | Data resources | |
|---|---|---|---|
| data flows | tasks | data sets | WF resources |
| APIs, connectivity | app. services | data | APP resources |
| MW APIs, connectivity | MW services | data lakes | MW resources |
| OS connectivity | OS services | filesystems | OS resources |
| networking IFs, infrastructure | processing units | storage units | HW resources |
| sensors, actuators, connected devices | | | CP resources |
| materials, energy, information, time, frequency, space | | | Fundamental resources |

Figure 2. Resources in the computing continuum.

A simplified example of two IoT applications in the computing continuum is depicted in Figure 3. The workflows of the applications start with data lifted from sensors. The data is processed in a sequence of tasks, running on containerized services in edge devices or cloud-based serverless functions. The containerization frameworks and the serverless functions are provided by two mobile network operators and two cloud providers, respectively. Both workflows end on actuators. In the depicted example, the applications share some of their sensors with each other.

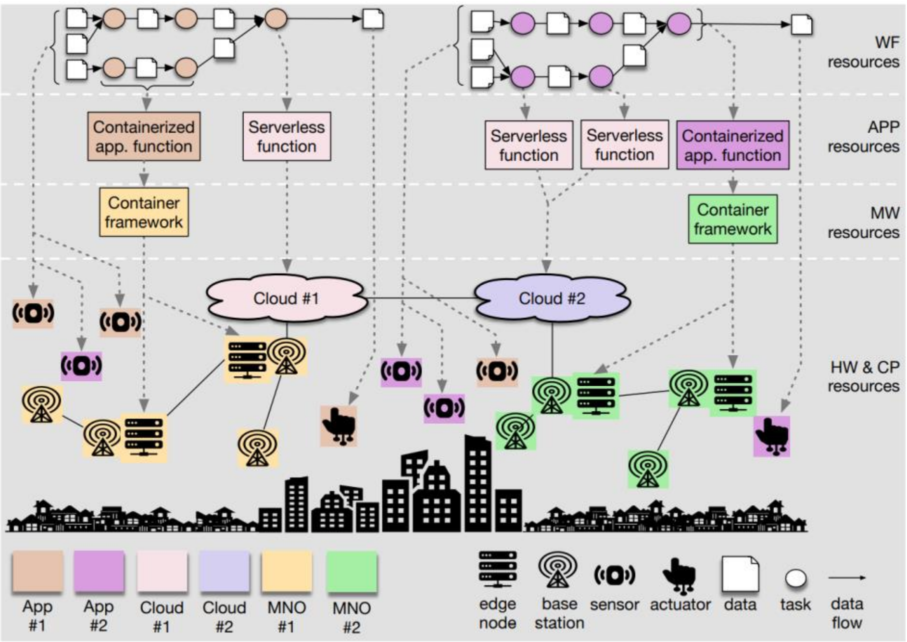


Figure 3. Two example IoT applications in the computing continuum.

Managing these resources in the computing continuum is often referred to as orchestration. In more detail, the term orchestration is used to refer to functions such as the automated management (i.e., configuring and coordination) of complex services, dynamic resource allocation, efficient and optimized resources utilization, control of functions, or real-time service delivery (Guerzoni, ym., 2017) (Taleb, 2017). However, related work often scopes orchestration to certain aspects of the continuum, such as networks and connections, application services, or tasks and workflows. Network orchestration refers to the configuration and management of communication networks. In contrast, service orchestration refers to the management and configuration of the life cycle of application components encapsulated as services. With EaaR, we can holistically define orchestration as the management of resources in the computing continuum. As presented in Figure 4, it can be divided into a number of functions such as lifecycle management or monitoring, as well as overarching attributes such as security or privacy. Orchestration aims to reach certain objectives set by a number of possible stakeholders such as end users and infrastructure providers.

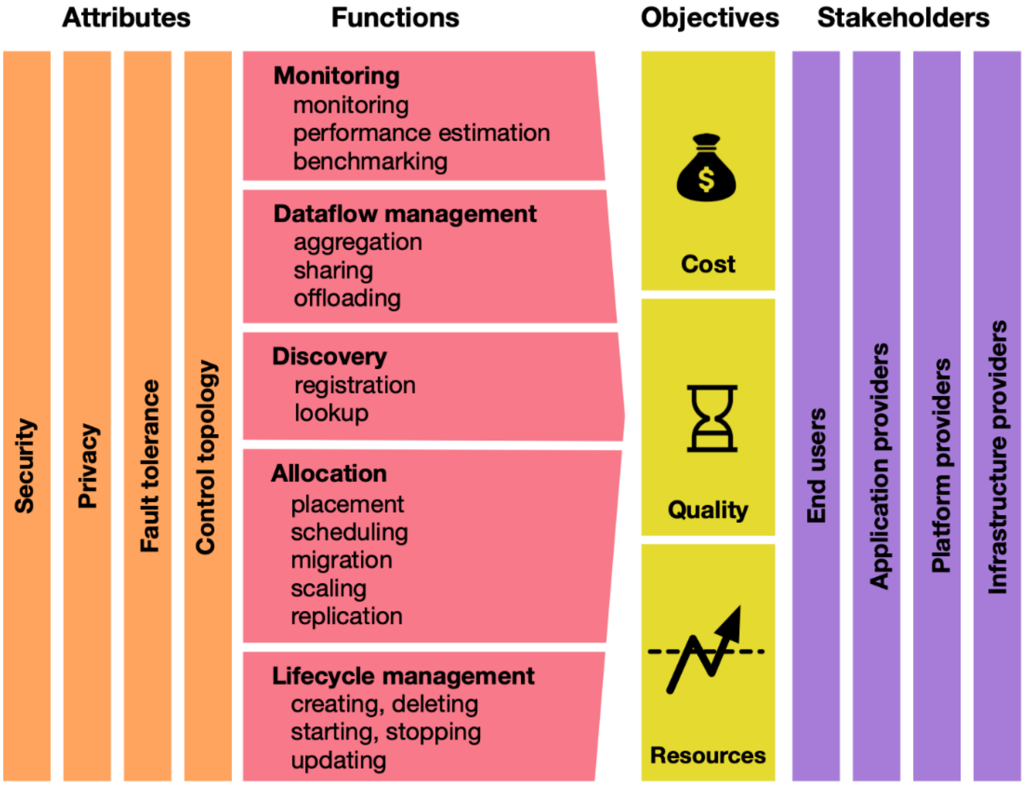


Figure 4. Continuum orchestration taxonomy.

## 4.3 AI methods

The computing continuum poses a number of challenges in realizing the vision of autonomous, intelligent AI agents: communication is intermittent and fluctuating, computation resources are distributed, heterogeneous and opportunistic, and data is distributed, siloed and non-IID. Furthermore, there can be a massive number of resources, applications, and their users, especially in the machine-to-machine (M2M) domain, and these users may have partially conflicting objectives. Finally, the applications may generate data that is highly sensitive, and must not be leaked outside devices or nearby edge servers.

These challenges set requirements for the AI approaches deployed in the computing continuum, as depicted in Figure 5. These approaches must be decentralized or distributed, as the resources are; further, weak coupling (Mämmelä & Riekki, New network architectures will be weakly coupled, 2022) and autonomy allow the approaches to survive alone if connections are severed. Non-IID data requires localized or personalized intelligence, while the numerous stakeholders and tenants present in the continuum demand approaches that support balancing multiple objectives. Finally, the approaches must be privacy-preserving especially in case the applications generate sensitive data.

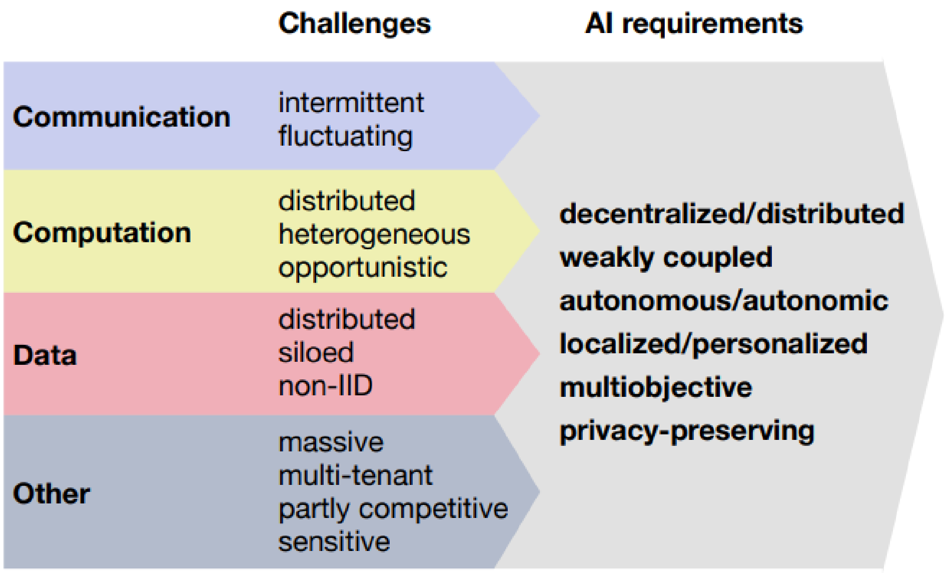


Figure 5. Challenges inherent in the computing continuum, and subsequent characteristics required of the AI approaches.

### 4.3.1 Distributed learning techniques

Orchestration in the computing continuum can be modelled as a hierarchical network of intelligent, autonomous agents that manage the resources of the platform in a decentralized manner. These agents need ML models to make predictions about processes and future states, which supports the agents' decision-making processes. Based on data, the agents must learn different dynamics in their environment to improve their performance and to adapt to the uncertain, evolving environment.

Each agent has access to the data they have collected, but this data may not have enough volume or diversity to train accurate models. In addition, an edge agent may not have enough resources for the training and inference of complex models. Hence, it is inevitable that agents must somehow collaborate with other nearby agents in the training and inference of ML models

Training ML models on edge requires distributed learning architectures and algorithms. Federated learning (FL) has quickly become the de facto training paradigm for distributed model training in edge environment. FL, first introduced by Google (McMahan;Moore;Ramage;Hampson;& Aguera y Arcas, 2017), aims to train a global ML model in a distributed manner. The global model is most typically an ANN model, but it can also be some other parameterized model. Original version of FL, often called vanilla FL, trains a global model in a centralized manner on decentralized data. Each agent participating in the training has their own training data that they use to train a local model. Then, the local parameter updates are sent periodically to a central server that aggregates the updates and sends the resulting global model back to agents.

### 4.3.2 Decision-making

Learning is a part of proactive behavior. It allows the agent to evaluate its actions in the environment and derive and explore new actions with the aim to reach its objectives. To be able to learn, agents store individual knowledge of their (sometimes partially) observable environment and themselves. This knowledge is referred to as models. Based on the agent's experience, the agent builds and adapts these models, that is, learns.

In the highly dynamic computing continuum, decision-making strategies[1] learned by the agents must be stable, but adaptive enough to conform to changes. Agents have to learn strategies based on their own local experience and interactions with neighbouring agents in order to make optimal decisions.

However, Fractal features can only exist through the design of intelligent behaviours for nodes. To illustrate, *adaptability* requires sophisticated methods allowing a node to know how to adapt its behaviour to a set of situations. *Context awareness* might require techniques allowing nodes to merge data from their environment to build a representation of their situation. However, these behaviours are still difficult to develop: firstly because of the limitations of AI algorithms and methods, but also because of their computational cost which can prevent them from being implemented on edge devices.

In order to implement the Fractal features, we need *intelligence on the edge* through the adaptation of AI techniques to devices that have strong physical constraints such as a low computational power. But we also need *intelligence for the edge* (Figure 6) to provide new cognitive capabilities for intelligence resource orchestration.

[1] Note: words 'policy' and 'strategy' are used interchangeably.

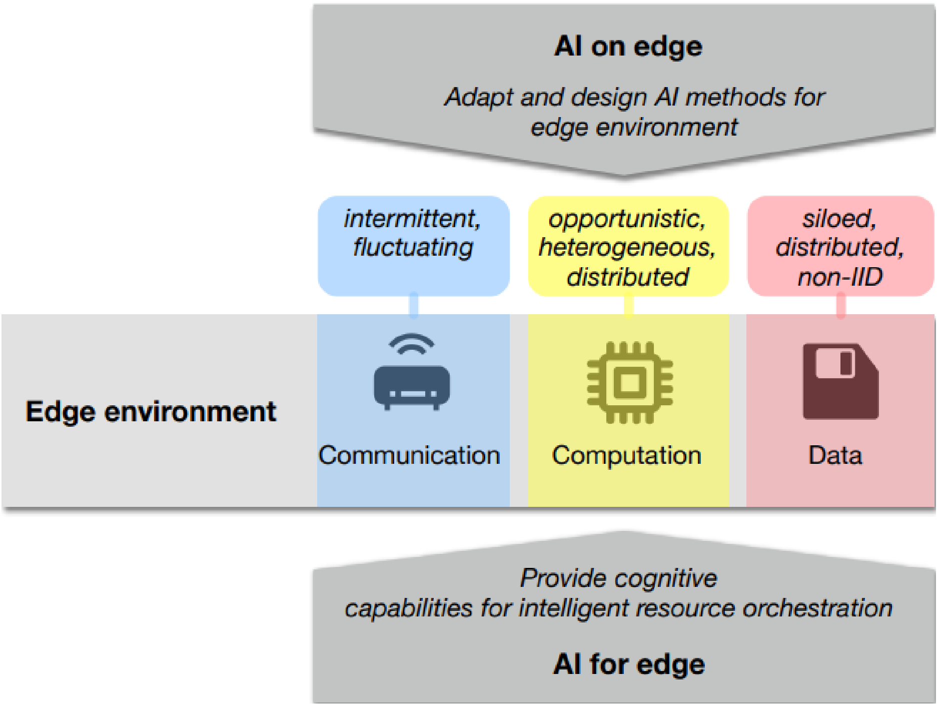

AI on edge
Adapt and design AI methods for edge environment
intermittent, fluctuating
opportunistic, heterogeneous, distributed
siloed, distributed, non-IID
Edge environment
Communication
Computation
Data
Provide cognitive capabilities for intelligent resource orchestration
AI for edge

Figure 6. AI on edge vs. AI for edge.

# 5 Theoretical studies

In this section, we report Fractal scientific reports under research question RQ2: What kind of methodology and intelligent capabilities are required 1) AI for edge and 2) AI on edge. For this deliverable, we have identified the most relevant publications and summarize the inputs from these results. Implementation of the suggested visions and algorithms is an iterative process connected to the development of HW and system architecture. The project is in its midway, and implementation work and analysis of different functionalities of Fractal framework is still ongoing.

## 5.1 AI for edge

Intelligent application workflows on edge set requirements for the edge platform with regard to performance, reliability and privacy. Furthermore, one of the objectives for the Fractal node is to guarantee a set of non-functional properties (dependability, security, timeliness and energy-efficiency). To fulfill such application requirements and node properties, intelligence is needed for the management of the edge platform. Hence, developing AI for edge is required, that is, developing AI techniques for the optimization of edge to make it function in a more intelligent and autonomous manner. This section introduces the theoretical studies that have been conducted in T5.1 to answer the *RQ2: What kind of methodology and intelligent capabilities are required 1) AI for edge*.

### 5.1.1 Vision

*J. Riekki and A. Mämmelä, "Research and Education Towards Smart and Sustainable World," in IEEE Access, vol. 9, pp. 53156-53177, 2021, doi: 10.1109/ACCESS.2021.3069902.*

Riekki and Mämmelä propose a high-level vision for directing research and education in the field of information and communications technology. They define their Smart and Sustainable World vision as follows: "Prosperity for the people and the planet is achieved with intelligent systems that sense their environment, make proactive decisions on actions advancing their goals, and perform the actions on the environment. Sustainable development is emphasized in decision-making, and system performance is optimized to save basic resources. Humans observe the autonomous operation through user interfaces and, when needed, revise the operation or control the systems manually."

Their vision leads to complex systems of systems, where a large number of interconnected devices provide a distributed platform for numerous co-existing intelligent systems that share the platform’s limited resources. This vision encompasses the Fractal system, and a central condition for realizing the vision is the intelligent use of the limited resources. Their ultimate argument is that managing the complexity requires studying system-level research problems, which in its turn requires a research paradigm that combines the conventional reductive view with a holistic systems view.

The idea of reductive view is to start from a conceptual analysis, reduce research problems to simpler problems, then perform experiments, after which the results can be generalized to a theory by induction, abduction, or formation of a hypothesis. Finally, results for the original problem can be derived from the theory by deduction. However, deduction is possible only in mathematically tractable problems, i.e., when the system is a linear system that follows the superposition principle. A complex system, such as the Fractal system, is characterized by nonlinear relationships between the system parts, which leads to emergence (the behavior at a higher hierarchy level of the system is not predictable from the properties at lower levels). This calls for systems thinking, which is more general than reductive thinking. Systems thinking is a form of generalized inference that is needed to replace deduction in mathematically intractable problems. Furthermore, such problems can be studied by simulations and experiments with system prototypes to provide insight into higher hierarchy level properties.

Such an approach where bottom-up reductive experimental research is followed by top-down systems research is required to meet the strict performance requirements in resource-constrained, large-scale systems of systems. Hence, this is the research paradigm that underlies the studies on Fractal AI.

*Henna Kokkonen, Lauri Lovén, Naser Hossein Motlagh, Juha Partala, Alfonso González-Gil, Ester Sola, Iñigo Angulo, Madhusanka Liyanage, Teemu Leppänen, Tri Nguyen, Panos Kostakos, Mehdi Bennis, Sasu Tarkoma, Schahram Dustdar, Susanna Pirttikangas, Jukka Riekki (2022): Autonomy and Intelligence in the Computing Continuum: Challenges, Enablers, and Future Directions for Orchestration, doi: 10.48550/arXiv.2205.01423.*

Kokkonen et al. propose a more concrete vision for edge orchestration: by developing intelligent solutions for edge orchestration, the edge environment will eventually evolve into a coherent device-edge-cloud computing continuum that is able to function in an autonomous, decentralized and decoupled manner, while optimizing and balancing multiple objectives with regard to, e.g., efficiency, reliability and security. The computing continuum will be able to orchestrate its limited computational, network, energy and memory resources in a globally optimized manner while being aware of and ready to adapt to the dynamic environment.

The vision relies on a more holistic view on resources and orchestration in the computing continuum, which is proposed in the paper and included in the Concepts and definitions -section of this deliverable. The architecture of the computing continuum is envisioned as a hierarchical multi-agent system consisting of autonomous, intelligent, self-interested agents, which is in line with the recursive Fractal system architecture. Agents correspond to resources in the computing continuum, and each agent has local autonomy with regard to deciding on when and how to conduct actions related to orchestration functions. However, to avoid the emergence of chaos on a global system level due to the nonlinear interactions of the agents, the system adopts loose (weak) coupling: the agents are nearly autonomous, but fair resource allocation and agent cooperation are ensured via minimal centralized control. In other words, higher levels in the hierarchy control the lower ones, operating at lower resolutions and having wider perspective to the system and the environment. The highest levels can be located in the cloud. This centralized control can be realized through goals and constraints (specifically on resource usage).

In order to reach coordinated orchestration decisions in the system, the agents must often negotiate with each other. Negotiation techniques are essential for mitigating the varying levels of cooperation and local decision-making between the agents and subsystems.

The paper provides an overview of the core challenges with regard to developing AI for edge. These challenges are posed by the inherent nature of the computing continuum: communication is intermittent and fluctuating, computation resources are geographically distributed, heterogeneous and opportunistic, and data is distributed, siloed and non-IID, and at times sensitive. Further, there can be a massive number of resources, applications, tenants and other stakeholders over a number of domains, and these stakeholders may have partially conflicting objectives.

The challenges set requirements for the orchestration of the computing continuum, as also depicted in the Concepts and definitions -section of this deliverable (Figure 5). Orchestration must be decentralized, as the resources are; further, weak coupling and local autonomy allow the approaches to survive alone if connections are severed. Non-IID data requires distributed edge intelligence, with localized learning and decision-making, while the numerous stakeholders and tenants present in the continuum demand approaches that support balancing multiple objectives. Finally, security and privacy must be considered for both APIs and execution, as well as data and AI models. However, implementing AI methods to answer these requirements is very difficult.

The paper provides a roadmap for AI research by introducing the state of the art in AI research fields that will be the key areas of the research for the future computing continuum orchestration solutions; this roadmap also applies to Fractal AI. The paper presents architectures and methods that currently exist for distributed learning, decision-making and negotiation. However, none of the existing approaches alone solves all the challenges that must be overcome to reach a truly intelligent orchestration. The solutions should emerge through joint efforts in these fields.

### 5.1.2 Analysis

*L. Lovén, E. Peltonen, E. Harjula and S. Pirttikangas, "Weathering the Reallocation Storm: Large-Scale Analysis of Edge Server Workload," 2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), 2021, pp. 317-322, doi: 10.1109/EuCNC/6GSummit51104.2021.9482593.*

Lovén et al. analyze and compare four different strategies for workload reallocation on edge servers: reallocation to cloud (cloud strategy), reallocation to another edge server based on promixity (choose the closest edge server), bottom-up (choose the edge server with the lowest workload) or random strategy (choose randomly). They have two main findings. First, a reallocation storm with a large number of superfluous reallocations is triggered when a task is reallocated to an edge server the capacity of which is exceeded within the duration of the task. Superfluous reallocation refers to a reallocation decision that causes another reallocation at the target edge server. Second, superfluous reallocations vanish when the edge server capacity is increased above a certain threshold. According to their experiments, the proximity strategy consistently results in the highest number of superfluous reallocations, and the random strategy is the most recommendable for dense edge server deployments.

Reallocation is one part of edge orchestration. Avoiding superfluous reallocations is important, as they increase the network burden and the latency of task processing. The study by Lovén et al. shows that it is important to study the conditions behind the reallocation storms more carefully and develop novel, intelligent reallocation strategies.

### 5.1.3 Method

*Lovén L, Lähderanta T, Ruha L, Peltonen E, Launonen I, Sillanpää MJ, Riekki J, Pirttikangas S. EDISON: An Edge-Native Method and Architecture for Distributed Interpolation. Sensors (Basel). 2021 Mar 24;21(7):2279. doi: 10.3390/s21072279. PMID: 33805187; PMCID: PMC8037329.*

Lovén et al. propose methods and an architecture for edge-native spatio-temporal data interpolation, called EDISON. They concentrate on interpolation models which extend the observations of a sparse sensor network to those areas and points in time where no observations are available. EDISON brings data pre-processing techniques (namely data interpolation) to the edge, distributing spatio-temporal interpolation models, their computations, and the observed data vertically and horizontally between device, edge, and cloud layers. On the device layer, mobile and fixed sensors collect data, while IoT gateways provide connectivity and local data storage for the mobile sensors. The edge layer has edge servers, placed at the fixed sensor locations, providing local computational capacity. The cloud provides centralized large-scale computational capacity.

During distributed training in EDISON, the cloud is responsible for partitioning the training set into subsets of observations around each edge server, aiming for subsets that are maximally independent. The cloud then sends the partitioned training set to all edge servers, rasterized to reduce transmission burden. Edge servers then train a local, spatio-temporal interpolation model for the observations in the edge server's subset of the training set. During distributed inference, each edge server first finds the right edge server for each new mobile observation from IoT gateways that have passed by, and then sends the observations to the selected servers. Edge servers then apply the local interpolation model with the data collected by the sensors. EDISON is designed to particularly address the challenges related to large-scale data and mobile, low-capability devices.

EDISON provides an edge-native way for interpolating the new observations over the unobserved timeslots and locations. Proper interpolation improves the subsequent data analysis, which in its turn improves situation awareness and decision-making. Furthermore, the architecture of EDISON is also applicable to edge-native predictive analysis in general.

*Nguyen, H., Nguyen T., Leppänen, T., Partala, J., Pirttikangas, S. (2022): Situation Awareness for Autonomous Vehicles Using Blockchain-based Service Cooperation, doi: 10.48550/arXiv.2204.03313.*

Nguyen et al. propose a blockchain-based method for enhancing context-awareness, fault tolerance and decision-making in vehicular networks. Their system comprises a trusted, collaborative platform where data storage and exchange, as well as service access are based on smart contracts between the service providers' edge servers. The system has two layers, one comprises the communications between vehicles and edge servers, and the other comprises the communications between the edge servers. Each edge server manages a region by conducting three main tasks: (1) collecting vehicles' information, (2) interpreting from vehicles' information, and (3) maintaining a blockchain with other edge servers. In other words, the vehicles at a specific region share data related to the environment with the edge server, which gathers vehicles' information and analyzes it to understand the region's state. This state is broadcasted to other edge servers and formulated as a new block in the shared blockchain. Each edge server can then share this new state with the vehicles in their region.

Because the roadside edge servers maintain a unique blockchain by which the data is distributed and stored at every part of the network, the system provides a secure and fault-tolerant way for vehicles to share contextual information with each other, which improves the decision-making capabilities of the vehicles. Further, the utilization of smart contracts facilitates the cooperation of different stakeholders on edge.

### 5.1.4 Synthesis

Table 3 provides a summary of the studies on AI for edge, stating their category, main contribution and connection to RQ2.

| Article | Category | Essential contribution | Connects to |
|---|---|---|---|
| (Riekki & Mämmelä, 2021) | Vision | Research paradigm for Fractal AI | Required methodology |
| (Kokkonen, ym., 2022) | Vision | Holistic view on resources and orchestration on device-edge-cloud continuum, vision of intelligent orchestration, extensive overview of state of the | Required methodology and intelligent capabilities, decision-making, learning and data architectures, mitigating varying levels of cooperation |

| | | | |
|---|---|---|---|
| | | art in distributed learning, decision-making and negotiation | and local decision-making, deciding about the level of autonomy |
| (Lovén;Peltonen;Harjula;& Pirttikangas, 2021) | Analysis | Comparison of current strategies for workload reallocation | Need for intelligent capabilities |
| (Lovén, ym., 2021) | Method | Methods and an architecture for distributing interpolation models, their computations, and the observed data vertically and horizontally between device, edge, and cloud layers | Novel architecture, enhanced decision-making |
| (Nguyen;Nguyen;Leppänen;Partala;& Pirttikangas, 2022) | Method | System for secure data sharing in vehicular networks | Data architecture, interoperability, enhanced decision-making |

Table 3. Summary of AI for edge contributions in Fractal T5.1

## 5.2 AI on edge

The Fractal use cases are in the central role for realizing AI on edge in Fractal project. In this section, we list the publications and reports related to the use cases. In other words, this section answers the *RQ2: What kind of methodology and intelligent capabilities are required 2) AI on edge* reflecting the scientific publications and reports published in the Fractal project. It should be noted that not all the material related to the use cases has been published in scientific papers.

This section also briefly lists HW related papers of Fractal even though they are not directly in the focus of Task 5.1. The publications are clustered to emphasize the sustainability properties of the solutions: safety, energy-efficiency and fault-tolerance. The related research questions are: *How can we increase energy-efficiency, safety, security and fault-tolerance of the Fractal system? How does energy-efficient chip design help us in the overall sustainability of the Fractal system?* Linking to other deliverables can be found in Section 6 of this document.

The overall project description was published in 2020 at Euromicro Conference (Lojo, ym., 2020).

*A. Lojo et al., "The ECSEL FRACTAL Project: A Cognitive Fractal and Secure edge based on a unique Open-Safe-Reliable-Low Power Hardware Platform," 2020 23rd Euromicro Conference on Digital System Design (DSD), 2020, pp. 393-400, doi: 10.1109/DSD51259.2020.00069.*

Lojo et al. presented the FRACTAL project and its expected benefits, in other words, the project's contribution to the current literature as well as industrial perspectives. This paper was published in the beginning of the project and it set the ambition level for the FRACTAL node to be able to provide cognitive skills through an internal and external architecture that allows to forecast its internal performance and the state of the surrounding world. The paper proposes two platforms applicable to serve as FRACTAL nodes, one commercial and one open RISC-V based. The main areas of FRACTAL ambitions are shown in Figure 7 (from (Lojo, ym., 2020)).

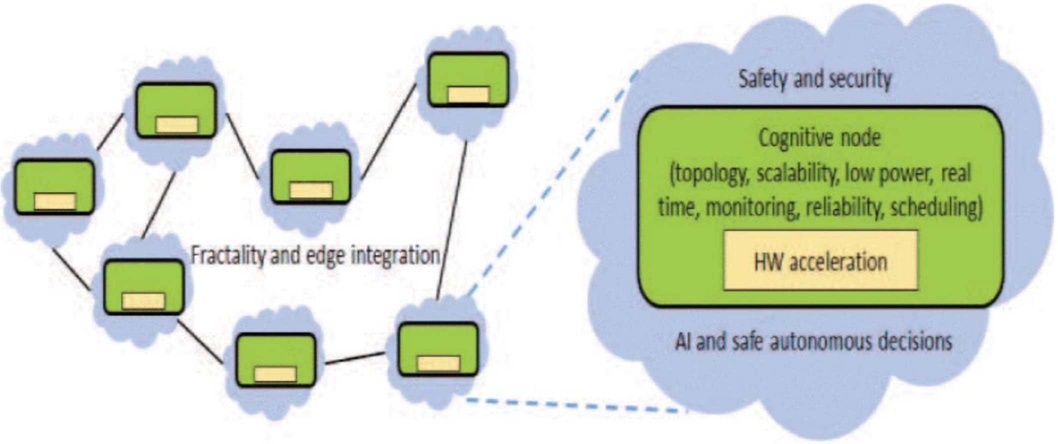

Fractality and edge integration
Safety and security
Cognitive node
(topology, scalability, low power, real time, monitoring, reliability, scheduling)
HW acceleration
AI and safe autonomous decisions

Figure 7. The main areas of FRACTAL ambitions.

The main functionalities of AI on edge are developed and tested through use cases. In this deliverable, we list the use cases and any related scientific publications.

*Use case* ***AI requirements*** are the functional and non-functional needs as they have been captured in the project. They are listed in the following table.

| | |
|---|---|
| Use case | UC1: Improving the quality of engineering and maintenance work through drones |
| Method | Algorithm capable of distinguishing between active and non-active cracks of a wide range of pathologies registered in concrete structures. Semantic segmentation. |
| Platform | UAV for data collection, Fractal node for processing |
| Related theoretical studies | *Ignacio Garrido Botella, David Sanz Muñoz. "A Cognitive Node to assist in the Inspection and Maintenance of Structures", internal report by INDRA, 2022.*<br>The designed crack detection model is based on a CNN with U-Net architecture. In addition, to improve the image segmentation |

| | |
|---|---|
| | model, a dataset has been created with real images taken on site with a UAV, and some image augmentation techniques have been implemented as well (change of brightness, overlap the images with textures). Further, the output is refined with techniques such as opening, closing, and a double threshold algorithm, so it is easier to measure the length of a crack/families of cracks. |

| | |
|---|---|
| Use case | UC2: Improving the quality of automotive air control |
| Method | Algorithm for implementing an intelligent control system that will reduce emissions. Reinforcement learning and regression modelling. |
| Platform | Not known yet. |
| Related theoretical studies | No related scientific publications published, yet. |

| | |
|---|---|
| Use case | UC3: Smart meters for everyone |
| Related theoretical studies | No AI focus. |

| | |
|---|---|
| Use case | UC4: Low-latency Object Detection as a generic building block for perception in the edge for Industry 4.0 applications |
| Method | HW acceleration for edge computing. Object detection with TinyYOLOv3. |
| Platform | ARIANE |
| Related theoretical studies | No related scientific publications published, yet. |

| | |
|---|---|
| Use case | UC5: Increasing the safety of an autonomous train through AI techniques |
| Method | Incorporating AI and high-performance computational capabilities. for increased dependability and safety. Autonomous train functionalities: stopping precision, odometer view, rolling stock coupling operation, person and obstacle detection using YoloV3/YoloV4. |
| Platform | Versal |
| Related theoretical studies | No related scientific publications published, yet. |

| | |
|---|---|
| Use case | UC6: Elaborate data collected using heterogeneous technologies (intelligent totem) |
| Method | Building an AI-based smart mobile totem, for advertisement and customer support inside shopping malls. Image recognition, speech recognition using neural Networks, CNN, LLM, YoloV4. |
| Platform | Versal |

| | |
|---|---|
| Related theoretical studies | *Di Mascio, T., Fantozzi, P., Laura, L., Rughetti, V. (2022). Age and Gender (Face) Recognition: A Brief Survey. In:, et al. Methodologies and Intelligent Systems for Technology Enhanced Learning, 11th International Conference. MIS4TEL 2021. Lecture Notes in Networks and Systems, vol 326. Springer, Cham. https://doi.org/10.1007/978-3-030-86618-1_11* (Di Marccio;Fantozzi;Laura;& Rughetti, 2021)<br>This paper presents a comparative overview of the state-of-the-art approaches which estimate age and gender from human faces, some of them proposing novel network architectures or the addition of new components to already known models.<br>(Di Mascio;Peretti;Caruso;& Cassioli , 2022) |

| | |
|---|---|
| Use case | UC7: Autonomous robot for implementing safe movements |
| Method | Integrating the Cognitive Edge Node in autonomous robot SPIDER and evaluate its applicability for performing computational intensive relevant vehicle functions of variable complexity at the edge of the network (near the source of the data). The use case leverages the LEDEL AI library. |
| Platform | NOEL-V |
| Related theoretical studies | No related scientific publications published, yet. |

| | |
|---|---|
| Use case | UC8: Improve the performance of autonomous warehouse shuttles for moving goods in a warehouse |
| Method | Leveraging swarm intelligence for autonomous shuttles. Improving the throughput, availability and safety of the system. |
| Platform | KV260, Versal |
| Related theoretical studies | No related scientific publications published, yet. |

### *5.2.1* Fractal features: Safety

Fractal publications related to hardware safety aspects are summarized below (Alcaide, ym., 2022), (Mazzocchetti, ym., 2022).

*Sergi Alcaide, Guillem Cabo, Francisco Bas, Pedro Benedicte, Fabio Mazzocchetti, Francisco J. Cazorla, Jaume Abella, "Unboxing the Sand: on Deploying Safety Measures in the Programmable Logic of COTS MPSoCs" ERTS^2 2022 11th European Congress on Embedded Real Time Software and Systems Toulouse (France), June 1-2 2022*

This paper proposes using programmable logic (PL) to provide hardware support for implementing safety measures efficiently. Providing sufficient hardware support for functional safety is a key enabler for using Commercial Off-the-Shelf (COTS) MPSoCs in safety-related systems that need high assurance levels. The goal is not to master PL from the cores solely, but also allow PL to provide monitoring (e.g., contention, diversity, watchdogs) and control (e.g., configuring QoS features) capabilities for the purpose of realizing a safety concept atop. The work presented in this paper provides specific monitoring, diversity, and controlling strategies to allow PL to take over safety-related functionalities.

*Fabio Mazzocchetti, Sergi Alcaide, Francisco Bas, Pedro Benedicte, Guillem Cabo, Feng Chang, Francisco Fuentes, Jaume Abella, "SafeSoftDR: a Library to Enable Software-Based Diverse Redundancy for Safety-Critical Tasks" FORECAST 2022 - Functional Properties and Dependability in Cyber-Physical Systems Workshop (held with HiPEAC conference 2022), Budapest (Hungary), June 21 2022*

Abstract not available, yet.

### 5.2.2 Fractal features: sustainability, energy-efficiency and low power

Papers representing sustainability (Benz;Bertaccini;Zaruba;Schuiki;& Gürkaynak, 2021), energy-efficiency and low power aspects of Fractal hardware are summarized below (Nambinina;Onwuchekwa;Ahmadian;Goyal;& Obermaisser , 2021), (Muoka;Onwuchekwa;& Obermaisser, 2022), (Tuzov, ym., 2021), (Rutishauser;Scherer;Fischer;& Benini, 2022), (Wistoff;Schneider;Gürkaynak;Heiser;& Benini, 2022), (Lua;Onwuchekwa;& Obermaisser, 2022), (Alshaer;Lua;Muoka;Onwuchekwa;& Obermaisser, 2022), (Muoka;Umuomo;Onwuchekwa;& Obermaisser, 2022), (Rogenmoser;Wistoff;Vogel;Gurgaynak;& Benini, 2022).

*T. Benz, L. Bertaccini, F. Zaruba, F. Schuiki, F. K. Gürkaynak and L. Benini, "A 10-core SoC with 20 Fine-Grain Power Domains for Energy-Proportional Data-Parallel Processing over a Wide Voltage and Temperature Range," ESSCIRC 2021 - IEEE 47th European Solid State Circuits Conference (ESSCIRC), 2021, pp. 263-266, doi: 10.1109/ESSCIRC53450.2021.9567755.*

This paper presents Thestral, a 10-core RISC-V chip for energy-proportional parallel computing manufactured in 22 nm FD-SOI technology. Thestral contains a control core and a nine-core computer cluster. This paper proposes a fast and fine-grain power gating architecture with much finer granularity than the state of the art for multi-core computing platforms.

*R. Nambinina, D. Onwuchekwa, H. Ahmadian, D. Goyal and R. Obermaisser, "Time-Triggered Frequency Scaling in Network-on-Chip for Safety-Relevant Embedded Systems," 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), 2021, pp. 1-7, doi: 10.1109/SMARTGENCON51891.2021.9645782.*

This paper presents models and algorithms for low power techniques in Networks on Chip based on dynamic frequency scaling for safety-relevant real-time systems. The novel technique is based on time-triggered frequency scaling to enable power and energy efficiency while preserving safety and real-time guarantees.

*Muoka P, Onwuchekwa D, Obermaisser R. Adaptive Scheduling for Time-Triggered Network-on-Chip-Based Multi-Core Architecture Using Genetic Algorithm. Electronics. 2022; 11(1):49. [https://doi.org/10.3390/electronics11010049](https://doi.org/10.3390/electronics11010049)*

In this work, an algorithm for path reconvergence in a multi-schedule graph, enabled by a reconvergence horizon, is presented to manage the state-space explosion problem resulting from an increase in the number of scenarios required for adaptation.

*Tuzov, Ilya & Andreu, Pablo & Medina, Laura & Picornell Sanjuan, Tomás & Robles, Antonio & Lopez, Pedro & Flich, José & Hernández, Carles. (2021). Improving the Robustness of Redundant Execution with Register File Randomization. 1-9. 10.1109/ICCAD51958.2021.9643466.*

This paper shows that Staggered Redundant Execution does not effectively protect the system against a wide range of faults and thus, new mechanisms to increase the diversity of homogeneous cores are needed. This paper proposes Register File Randomization (RFR), a low-cost diversity mechanism that significantly increases the robustness of homogeneous multicores in front of common-cause faults (CCFs) and register file wear out.

*G. Rutishauser, M. Scherer, T. Fischer, L. Benini, "Ternarized TCN for μJ/Inference Gesture Recognition from DVS Event Frames.", Design, Automation and Test in Europe Conference (DATE 2022), online, March 14–23, 2022*

This paper proposes an event frame-based approach to the classification of DVS video data. Ternary video frames are assembled from the event stream and processed with a fully ternarized Temporal Convolutional Network which can be mapped to CUTIE, a highly energy-efficient Ternary Neural Network accelerator.

*N. Wilstoff, M. Schneider, F. K. Gürkaynak, G. Heiser, L. Benini, "Systematic Prevention of On-Core Timing Channels by Full Temporal Partitioning", [https://arxiv.org/abs/2202.12029](https://arxiv.org/abs/2202.12029)*

This work leverages the open and extensible RISC-V instruction set architecture (ISA) to introduce the temporal fence instruction fence.t, which provides the required mechanisms by clearing vulnerable microarchitectural state and guaranteeing a history-independent context-switch latency. This paper proposes and discusses three different implementations of fence.t and implements them on an experimental version of the seL4 microkernel and CVA6, an open-source, in-order, application class, 64-bit RISC-V core.

# 6 Links to other deliverables and implementation

In the previous sections, a thorough analysis of the Fractal AI theoretical framework has been conducted by reviewing all the scientific publications related to (1) Fractal AI, and (2) the hardware Fractal Platform. As a result, a complete set of tools that enable the development of robust and functional technology stacks based on the Fractal paradigm has been provided. Throughout the project, partners providing the use cases have been utilizing and including the Fractal framework in their own technological stacks. They are implementing the Fractal platforms and AI tools to solve their respective use cases, thus bringing the academic research to an actual implementation. All this work to bring the theoretical aspects of the project from the purely academic to the engineering realm has been compiled and is reflected in a variety of deliverables and technical reports.

In this section, the link between the academic publications and the actual implementations, which are reflected on the various deliverables of the project, is described. Work package related deliverables describe the actual validation of the research in more detail and is not a focus in this deliverable.

A description of the deliverables which provide the implementation details from the academic publications is given below.

### 6.1.1 T5.2 Fractal AI Platform

**D5.2 Intermediate platform for Federated AI**

This deliverable describes the development of the runtime platform where the AI algorithms are deployed, tested, and run. The FRACTAL AI Platform is an independent Edge-oriented platform which eventually would require Cloud support for heavy resource-demanding tasks like video-processing, heavy ML model training or large data storage (historical data, for example). In D5.2, a technical and functional description of the Cloud architecture components is provided, taking the FRACTAL Use Case requirements as a starting point. AI algorithms and processes coming from the scientific publications can be directly implemented in the Fractal Cloud Platform through the data pre-processing and storage capabilities, performing model training, model quantization and validation. The resulting models can then be deployed in the Edge Fractal platform.

**D5.4 Platform and building blocks for Fractal AI, toolkits, and custom and pre-trained models for AI-based Algorithms developed in the other tasks**

This deliverable complements D5.2, by describing the implementation done in D5.2 from a technical perspective, providing detailed explanations about how the Fractal Cloud Components are installed and utilized. As a result, all the Cloud components, which are required to have a fully-functional Cloud-Edge architecture, are integrated together and are then ready to be used in the use cases. Four Test Cases are proposed in this document, detailing the functioning through four experiments that demonstrate the capabilities of the Fractal Cloud Platform, ranging from data capturing to the final model deployment at the Edge.

### 6.1.2 T5.3 Applied Fractal AI

**D5.1 Specification of AI methods for use case applications**

D5.1 is one of the most important deliverables when considering the implementation of the theoretical Fractal system. It provides the first approach to defining the AI methods and tools that will be available in the Fractal environment. It consists of a wide collection of AI tools and methods to be used by the use cases for the development of their applications. This collection will then be integrated together to become a robust framework where end-to-end AI processes can be built and executed.

First, an overview of the main AI methods characterizing the Fractal system is given. It addresses the difference between learning and inference, centralized and decentralized learning, and other key aspects of the Fractal functioning like context awareness and model preparation for Edge deployments.

Then, the functional and non-functional requirements of the use cases are detailed. These state the necessity of a variety of tools to cover each of the requirements. Finally, D5.1 provides a complete collection of AI methods and AI tools that allow the development and implementation of all the algorithms and strategies previously investigated. This collection includes the algorithms needed to tackle each of the use cases, video analyzing processes, neural networks, training strategies, as well as model training and deployment libraries and data science libraries. As a result, the Fractal environment has at its disposal a plethora of tools to bring all the theoretical research into engineering, allowing the creation of models and its proper handling.

**D5.6 Mechanisms for AI transparency interactions**

Initially, the objective of this deliverable was to detail the interaction mechanisms between humans and AI, and to discuss how these mechanisms can be controlled and regulated so that the response of the ML models to human responses is legally and ethically correct.

After the completion of D5.1, it was decided by the T5.3 partners that even though the collected list of tools and frameworks was complete, the project was still in an early development phase and new tools or necessities could emerge during the course of the project that would remain uncovered. For this reason, it was decided to address these emerging tools and requirements in D5.6, which will also include thorough technical descriptions of the use cases. These descriptions cover the model design, working, optimization, and the architectural deployment of the data collection and processing tools, as well as the models themselves.

### 6.1.3 T6.1 Edge node design and implementation

**D6.1 FRACTAL processing node design and implementation**

In this deliverable, the practical implementation details of all the necessary tools to build a Fractal Platform are detailed. D6.1 takes as starting point D2.1 which gathers all the functional and non-functional requirements of the use cases, and describes all the steps to implement them technically on a physical platform, by targeting the three reference architectures: Low-end node (PULP), mid-node (NOEL-V or RISC-V) and high-end node (VERSAL). The importance of D6.1 resides in the fact that it presents the actual construction of the Fractal Platform in physical systems, covering all the reference platforms and utilizing the AI tools selected in D5.1. Hence, it allows the knowledge obtained from Fractal AI research to be used in practical implementations.

It also provides instructions to create virtual machines to emulate the reference architectures for developing purposes, which can help to validate the developed items before taking them into the actual physical hardware.

# 7 Conclusions

According to WP2, the most interesting Fractal features in relation to AI on the edge and AI for the edge are decentralization, context-awareness and adaptability. Decentralization refers to the distribution of the system operation, which can be achieved with, e.g., a multiagent paradigm. Also, by having weakly coupled Fractal system components, we can bring more adaptability into the nodes' behavior. Decentralized behaviors require distributed mechanisms to operate, such as distributed learning which allows agents to learn, infer and adapt to uncertain and evolving environments. *Context-awareness* is a feature referring to the ability of the node to perceive and leverage available information in the environment of the node. This feature can be viewed from the perspective of the use cases, as well as through the system operation. It can be any information that characterizes the situation of the node such as the time, the available data from neighboring nodes, etc. *Adaptability* represents the capacity of a node to change its behavior depending on its objective, its knowledge or even its context. This deliverable lists and synthetizes the Fractal scientific publications that handle different methods and architectures for allowing the system to make intelligent decisions about the management of the distributed resources of the system. Further, the Fractal publications that focus on the AI methods needed for the use cases are listed and summarized in this deliverable. We also list the papers that can shed insight into the practical implementation aspects of efficient and sustainable Fractal nodes. Finally, the deliverable’s links to other deliverables and implementation paths are given. In the project, the integration of different components towards FRACTAL system is ongoing.

# 8 Bibliography

Alcaide, S., Cabo, G., Bas, F., Benedicte, P., Mazzocchetti, F., Cazorla, F., & Abella, J. (2022). Unboxing the Sand: on Deploying Safety Measures in the Programmable Logic of COTS MPSoCs. *11th European Congress Embedded Real Time Systems ( ERTS 2022 ).*

Alshaer, S., Lua, C., Muoka, P., Onwuchekwa, D., & Obermaisser, R. (2022). Deep Learning based Meta-scheduling in real time systems. *IEEE 27th Int Conf Emerging Technologies and Factory Automation (ETFA)* (p. to appear). Stuttgart, Germany: IEEE.

Balouek-Thomert, D. (2019). Towards a computing continuum: Enabling edge-to-cloud integration for data-driven workflows. *The Intl Journal of High Performance Computing Applications*, 1159–1174.

Benz, T., Bertaccini, L., Zaruba, F., Schuiki, F., & Gürkaynak, F. K. (2021). A 10-core SoC with 20 Fine-Grain Power Domains for Energy-Proportional Data-Parallel Processing over a Wide Voltage and Temperature Range. *ESSCIRC 2021 - IEEE 47th European Solid State Circuits Conference (ESSCIRC)*, (pp. 263-266).

Brabakaran, B. S. (2020). EMAP: A Cloud-Edge Hybrid Framework for EEG Monitoring and Cross-Correlation BAsed Real-Time Anomaly Prediction. *ACM/IEEE Design Automation Conference (DAC)*, (pp. 1-6).

Costa, B. (2022). Orchestration in Fog Computing: A Comprehensive Survey. *ACM Comput. Surv*, 1-34.

Di Marccio, T., Fantozzi, P., Laura, L., & Rughetti, V. (2021). Age and Gender (Face) Recognition: A Brief Survey. *International Conference in Methodologies and intelligent Systems for Techhnology Enhanced Learning*, (pp. 105-113).

Di Mascio, T., Peretti, S., Caruso, F., & Cassioli , D. (2022, June 7-9). The "Great Beauty" of Diversity: Smart Totems to Promote Gender Uniqueness. *IEEE Int workshop on Metrology for Undustry 4.0 and IoT*. Trento, Italy: IEEE.

Dustdar, S., Pujol, V. C., & Donta, P. K. (2022). On Distributed Computing Continuum Systems. *IEEE Transactions on Knolwedge and Data Engineering*, 1-14.

Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures.* Irvine: University of California.

Guerzoni, R., Vaishnavi, I., Caparros, D. P., Galis, A., Tusa, F., Monti, P., . . . Szabo, R. (2017). Analysis of end-to-end multi-domain management and orchestration frameworks for software defined infrastructures: an architectural survey. *Transactions on Emerging Telecommunications Technologies*, 1-19.

Hong, C.-H. a. (2020). Resource Management in Fog/Edge Computing: A Survey on Architectures, Infrastructure, and Algorithms. *ACM Comput. Surv.*, 37.

Kokkonen, H., Lovén, L., Motlagh, N. H., Partala, J., González-Gil, A., Sola, E., . . . Riekki, J. (2022). Autonomy and Intelligence in the Computing Continuum: Challenges, Enablers, and Future Directions for Orchestration. *https://arxiv.org/pdf/2205.01423.pdf*, 1-53.

Lampkin, V., Tat Leong, W., Olivera, L., Rawat, S., Subrahmanyam, N., Xiang, R., . . . Locke, D. (2012). *Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry.* IMB Redbooks.

Lojo, A., Rubio, L., Ruano, J. M., Di Marcio, T., Pomante, L., Ferrari, E., . . . Abella, J. (2020). The ECSEL FRACTAL Project: A Cognitive Fractal and Secure edge based on a unique Open-Safe-Reliable-Low Power Hardware Platform. *2020 23rd Euromicro Conference on Digital System Design (DSD)*, (pp. 393-400).

Lovén, L., Lähderanta , T., Ruha, L., Peltonen, E., Launonen, I., Sillanpää, M. J., . . . Pirttikangas, S. (2021). EDISON: An Edge-Native Method and Architecture for Distributed Interpolation. *Sensors*, 2279.

Lovén, L., Peltonen, E., Harjula, E., & Pirttikangas, S. (2021). Weathering the Reallocation Storm: Large-Scale Analysis of Edge Server Workload. *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, (pp. 317-322).

Lua, C., Onwuchekwa, D., & Obermaisser, R. (2022). AI-Based Scheduling for Adaptive Time-Triggered Networks. *11th Mediterranean Conference on Embedded Computing (MECO)*, (pp. 1-7).

Mampage, A., Karunasekera, S., & Buyya, R. (2022). “A Holistic View on Resource Management in Serverless Computing Environments: Taxonomy and Future Directions. *ACM Comput. Surv.*, Just accepted.

Mazzocchetti, F., Alcaide, S., Bas, F., Benedicte, P., Cabo, G., Chang, F., . . . Abella, J. (2022). SafeSoftDR: A Library to Enable Software-based Diverse Redundancy for Safety-Critical Tasks. *FORECAST 2022 Functional Properties and Dependability in Cyber-Physical Systems Workshop (held jointly with HiPEAC Conference)*, (p. to appear). Budapest.

McMahan, B., Moore, E., Ramage, D., Hampson, S., & Aguera y Arcas, B. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. *Proceedings of Machine Learning Research* , (pp. 1273-1282).

Muoka, P., Onwuchekwa, D., & Obermaisser, R. (2022). Adaptive Scheduling for Time-Triggered Network-on-Chip-Based Multi-Core Architecture Using Genetic Algorithm. *Electronics*, 49.

Muoka, P., Umuomo, O., Onwuchekwa, D., & Obermaisser, R. (2022). Effect of Periodic Sampling on Metascheduling for Time-triggered Systems . *IEEE 27th Int Conf Emerging Technologies and Factory Automation (ETFA)* (p. to appear). Stuttgart, Germany: IEEE.

Mämmelä, A., & Riekki, J. (2022). New network architectures will be weakly coupled. *IEEE Future Networks Tech Focus*, 1-8.

Mämmelä, A., Riekki, J., Kotelba, A., & Anttonen, A. (2018). Multidisciplinary and Historical Perspectives for Developing Intelligent and Resource-Efficient Systems. *IEEE Access*, 17464-17499.

Nambinina, R., Onwuchekwa, D., Ahmadian, H., Goyal, D., & Obermaisser , R. (2021). Time-Triggered Frequency Scaling in Network-on-Chip for Safety-Relevant Embedded Systems. *2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, (pp. 1-7).

Nguyen, H., Nguyen, T. H., Leppänen, T., Partala, J., & Pirttikangas, S. (2022). Situation Awareness for Autonomous Vehicles Using Blockchain-Based Service Cooperation. *LNCS, volume 13295*, 501-516.

Ranaweera, P., Jurcut, A. D., & Liyanage, M. (2021). Survey on Multi-Access Edge Computing Security and Privacy. *IEEE Communications Surveys Turorials*, 1078-1124.

Riekki, J., & Mämmelä, A. (2021). Research and Education Towards Smart and Sustainable World. *IEEE Access*, 53156-53177.

Rogenmoser, M., Wistoff, N., Vogel, P., Gurgaynak, F., & Benini, L. (2022). On-demand redundancy grouping: Selectable soft-error tolerance for a multicore cluster. *IEEE Computer Society Annual Symposium on VLSI*, (p. to appear).

Rutishauser, G., Scherer, M., Fischer, T., & Benini, L. (2022). Ternarized TCN for µJ/inference gesture recognition from DVS event frames. *DATE '22: Proceedings of the 2022 Conference & Exhibition on Design, Automation & Test in Europe*, (pp. 736-741).

Saint-Andre, P., Smith, K., & Tronçon, R. (2009). *XMPP: The Definitive Guide. Building Real-Time Applications with Jabber.* O'Reilly Media, Inc.

Shelby, Z. a. (2011). *6LoWPAN: The wireless embedded Internet* (Vol. 43). John Wiley \& Sons.

Siemens. (2018). *https://www.plm.automation.siemens.com.* Retrieved from https://www.plm.automation.siemens.com: https://www.plm.automation.siemens.com/media/global/en/Siemens-MindSphere-Whitepaper-69993_tcm27-29087.pdf

Taleb, T. (2017). On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration. *IEEE Communications Surveys Tutorials*, 1657-1681.

Toczé, K., & Nadjm-Tehrani, S. (2018). A Taxonomy for Management and Optimization of Multiple Resources in Edge Computing. *Wireless Communications and Mobile Computing, special issue Mobile Edge Computing*, 23.

Tuzov, I., Andreu, P., Medina, L., Picornell, T., Robles, A., Lopez, P., . . . Hernandez, C. (2021). Improving the Robustness of Redundant Execution with Register File Randomization. *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, (pp. 1-9).

Weiss, G. (2013). *Multiagent Systems.* Massachusetts, USA: MIT Press.

Wistoff, N., Schneider, M., Gürkaynak, F. K., Heiser, G., & Benini, L. (2022). *Systematic Prevention of On-Core Timing Channels by Full Temporal Partitioning.* https://arxiv.org/abs/2202.12029.

Xiong, J. (2018). Extend Cloud to Edge with KubeEdge. *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, (pp. 373-377).

Yuan, H. (2020). Profit-Maximized Task Offloading with Simulated-annealing-based Migrating Birds Optimization in Hybrid Cloud-Edge Systems. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, (pp. 1218-1223).

Zhong, Z. (2021). Machine Learning-based Orchestration of Containers: A Taxonomy and Future Directions. *ACM Comp. Surv.*, 1-36.

# 9 List of figures

Figure 1. Fractal features (modified from deliverable D2.3). ........................... 7
Figure 2. Resources in the computing continuum. ........................................10
Figure 3. Two example IoT applications in the computing continuum. ............10
Figure 4. Continuum orchestration taxonomy. ............................................11
Figure 5. Challenges inherent in the computing continuum, and subsequent characteristics required of the AI approaches. ................................................12
Figure 6. AI on edge vs. AI for edge. ........................................................14
Figure 7. The main areas of FRACTAL ambitions. ........................................21

# 10 List of tables

Table 1. D5.3 research questions ........................................................ 6

Table 2. Association of AI concepts to Fractal features .................................. 8

Table 3. Summary of AI for edge contributions in Fractal T5.1 .........................20

# 11 List of Abbreviations

| | |
|---|---|
| AI | artificial intelligence |
| APP | application |
| CCF | common-cause faults |
| CP | cyber-physical |
| CPU | central-processing unit |
| EaaR | everything as a resource |
| FD-SOI | fully depleted silicon on insulator |
| FL | federated learning |
| GPU | graphical processing unit |
| HW | hardware |
| IID | independent and identically distributed random variables |
| IoT | Internet of Things |
| ISA | instruction set architecture |
| M2M | machine to machine |
| MAS | multi-agent systems |
| MCC | mobile cloud computing |
| MEC | multi-access edge computing |
| ML | machine learning |
| MW | middleware |
| OS | operating system |
| PL | programmable logic |
| RFR | register file randomization |
| RQ | research question |
| SoC | system on chip |
| SSD | solid-state drive |
| UC | use case |
| VM | virtual machine |
| WF | workflow |