Grant Agreement No.: 101070473
Call: HORIZON-CL4-2021-DATA-01
Topic: HORIZON-CL4-2021-DATA-01-05
Type of action: HORIZON-RIA

D7.2 Business Use Cases V1

Revision: v.1.0

Work package	WP 7
Task	Tasks T7.2, T7.3, T7.4
Due date	March 31st 2024
Submission date	April 30th 2024
Deliverable lead	TER
Version	1.0
Authors	Elisa Albanese, Luca Zuanazzi, Roberta Terruggia (RSE) Andy Edmonds, Panagiotis Gkikopoulos (TER) Guillem Garí (ROB)
Reviewers	Francesco Cappa (POLITO) Malvina Catalano Gonzaga (CYSEC)

Funded by Horizon Europe Framework Programme of the European Union

Abstract	This deliverable reports on the activities of T7.2, T7.3, T7.4 and details the progress in delivering each use case.
Keywords	Computing continuum; liquid computing; FLUIDOS use cases

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1		1st version of the template for comments	Margherita Facca (MAR)
V0.2		Version submitted for review	Andy Edmonds (TER)
V1.0		Final Version after integration of review comments	Francesco Cappa (POLITO) Malvina Catalano Gonzaga (CYSEC)

Disclaimer

Copyright Notice

Project co-funded by the European Commission in the Horizon Europe Programme		
Nature of the deliverable:	R	
Dissemination Level		
PU	Public, fully open, e.g., web	X
SEN	Sensitive, limited under the conditions of the Grant Agreement	
Classified R-UE/ EU-R	EU RESTRICTED under the Commission Decision No2015/ 444	
Classified C-UE/ EU-C	EU CONFIDENTIAL under the Commission Decision No2015/ 444	
Classified S-UE/ EU-S	EU SECRET under the Commission Decision No2015/ 444	

** R: Document, report (excluding the periodic and final reports)*

Funded by Horizon Europe Framework Programme of the European Union

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

DATA: Data sets, microdata, etc.

DMP: Data management plan

ETHICS: Deliverables related to ethics issues.

SECURITY: Deliverables related to security issues

OTHER: Software, technical diagram, algorithms, models, etc.

Funded by Horizon Europe Framework Programme of the European Union

EXECUTIVE SUMMARY

FLUIDOS aims to redefine the cloud-edge-IoT compute continuum paradigm, offering a seamless and decentralised operating system that integrates the cloud with the edge and IoT, enhancing efficiency, scalability, and security.

FLUIDOS leverages existing operating systems, enriched by Ubuntu, Kubernetes, Liqo, and ARCA Trusted OS, to enable existing, legacy and new applications to run on the FLUIDOS platform, which can be dynamically orchestrated based on user intents and network conditions.

FLUIDOS encompasses three use cases that demonstrate its potential across various sectors: intelligent power grid, smart viticulture, and robotic logistics. These use cases illustrate how FLUIDOS can address specific challenges and opportunities in each domain, such as optimising the management of distributed energy resources, enhancing crop management and disease detection, and improving task orchestration and resource allocation among autonomous robots.

The work follows a structured and iterative process to extract scenarios, actors, requirements, and architecture changes for each use case, aligning them with the technical features and advancements of FLUIDOS. FLUIDOS also provides a detailed description of the implementation and integration steps for each use case, highlighting the testbeds, workloads, intents, and metrics involved.

FLUIDOS sets the foundation for a wide range of applications that can benefit from decentralised, efficient, and secure computing at the edge, fostering a more interconnected, resilient, and sustainable digital infrastructure. It delivers the concept of a fluid, distributed cloud continuum that spans from the small IoT device, through the edge to the cloud.

Funded by Horizon Europe Framework Programme of the European Union

TABLE OF CONTENTS

Funded by Horizon Europe Framework Programme of the European Union

Funded by Horizon Europe Framework Programme of the European Union

LIST OF FIGURES

Funded by Horizon Europe Framework Programme of the European Union

LIST OF TABLES

Funded by Horizon Europe Framework Programme of the European Union

GLOSSARY

- **AMM (Autonomous Mobile Manipulator)**: A type of robot that combines the mobility of a mobile platform (AMR) with the dexterity of a robotic manipulator arm.
- **AMR (Autonomous Mobile Robot)**: A type of robot that can navigate and move through its environment without human control or direct oversight. AMRs are equipped with sensors and software that allow them to understand and respond to their surroundings, and they can perform tasks autonomously without the need for human intervention.
- **ARCA Trusted OS**: A hardened operating system combined with a secure Kubernetes orchestrator to contain system intrusion and avoid data compromission.
- **Cloud Continuum**: A virtual space spanning diverse technological domains irrespective of their physical locations, enabled by FLUIDOS.
- **Crate**: The edge hardware device used to deliver on-premises service to the customer in the Smart Viticulture use case.
- **FLUIDOS**: A project that aims to redefine the cloud-edge-IoT compute continuum paradigm by offering a seamless and decentralized operating system that integrates the cloud with the edge and IoT.
- **GSE (Grid State Estimation)**: An algorithm that accurately estimates the operating conditions of the electrical grid.
- **ICT (Information and Communication Technology)**: Infrastructure that plays a pivotal role in facilitating real-time monitoring, control, and optimization of the power grid.
- **Intent-Based Orchestration**: A feature of FLUIDOS that automatically orchestrates workloads based on user intents and network conditions.
- **Kubernetes**: A technology that abstracts underlying physical resources and capabilities, used by FLUIDOS.
- **Liqo**: An open-source project which extends the Kubernetes abstractions to multi-cluster scenarios, used by FLUIDOS.
- **PDC (Phasor Data Concentrator)**: A data hub responsible for collecting, aggregating, and aligning the timestamps of geographically scattered PMU measurements.
- **PMU (Phasor Measurement Unit)**: A measuring device attached to the power grid, strategically positioned to measure electrical quantities, and equipped with a GPS module that guarantees precise time synchronization.
- **ROS (Robot Operative System)**: A robotic software middleware used by the Robotnik RB-THERON autonomous robot in the Robotic Logistics use case.

Funded by Horizon Europe Framework Programme of the European Union

- **TEE (Trusted Execution Environment)**: A secure area of a main processor that ensures that sensitive data is stored, processed, and protected in a trusted environment.
- **TerraviewOS**: A cloud-native service for agriculture that helps end-users manage their property inputs and outputs.
- **TVOS-Core (TerraviewOS-Core)**: The core part of the TerraviewOS system, deployed on cloud resources.

Funded by Horizon Europe Framework Programme of the European Union

1 INTRODUCTION

FLUIDOS encompasses a wide array of use cases spanning various sectors, illustrating the versatility and comprehensive approach of the initiative. At its core, FLUIDOS aims to redefine the cloud-edge-IoT compute continuum paradigm, offering a seamless and decentralised operating system that integrates the cloud with the edge and IoT, enhancing efficiency, scalability, and security. To exercise the FLUIDOS technology, three use cases are used.

In the energy sector, FLUIDOS addresses the growing complexity and decentralisation of energy systems, particularly with the proliferation of renewable energy sources. The project seeks to optimise the management of distributed energy resources, enhancing grid stability and efficiency through real-time data processing and control at the edge. This not only reduces the reliance on centralised cloud infrastructures but also ensures more resilient and responsive energy systems capable of handling dynamic demands and potential disruptions.

The agricultural domain, specifically viticulture, benefits from FLUIDOS through the smart management of agricultural data and operations. By deploying edge computing capabilities, FLUIDOS enables real-time data analysis and decision-making directly on-site, minimising the latency and bandwidth issues associated with cloud computing. This use case demonstrates how FLUIDOS can transform agricultural practices, offering more precise and timely insights for crop management, disease detection, and resource optimization, leading to enhanced productivity and sustainability.

Finally, in the logistics sector, FLUIDOS introduces an innovative solution for managing fleets of autonomous robots. The project facilitates efficient task orchestration and resource allocation among robots, improving operational efficiency and reducing overheads. This use case highlights FLUIDOS's potential to enhance logistics operations, enabling more agile and cost-effective supply chain management, and enhancing customer experiences in retail environments.

These use cases collectively showcase FLUIDOS's potential across various industries, underlining its role in advancing cloud-edge-IoT compute continuum technologies and fostering a more interconnected, efficient, and resilient digital infrastructure. FLUIDOS not only addresses specific sectoral challenges but also sets the foundation for a wide range of applications that can benefit from decentralised, efficient, and secure computing at the edge.

Funded by Horizon Europe Framework Programme of the European Union

These use cases have been selected to "(i) involve project stakeholders (RSE, ROB & TER) in the definition of the project pillars as early as possible (ground-up architecture), (ii) to experiment with different usage scenarios and business models, and (iii) to validate and assess FLUIDOS in rather different operating conditions." (FLUIDOS, DoA)

Finally, these use cases and their experience and knowledge will form a basis of those new use cases that will contribute to FLUIDOS from the Open Call.

Funded by Horizon Europe Framework Programme of the European Union

2 USE CASES

We have 3 core use cases to demonstrate FLUIDOS technology, which are now briefly introduced. Each of these use cases are now clearly described on the FLUIDOS website[1].

Figure 1: FLUIDOS Use Cases

2.1 UC1: INTELLIGENT POWER GRID (ENERGY, RSE)

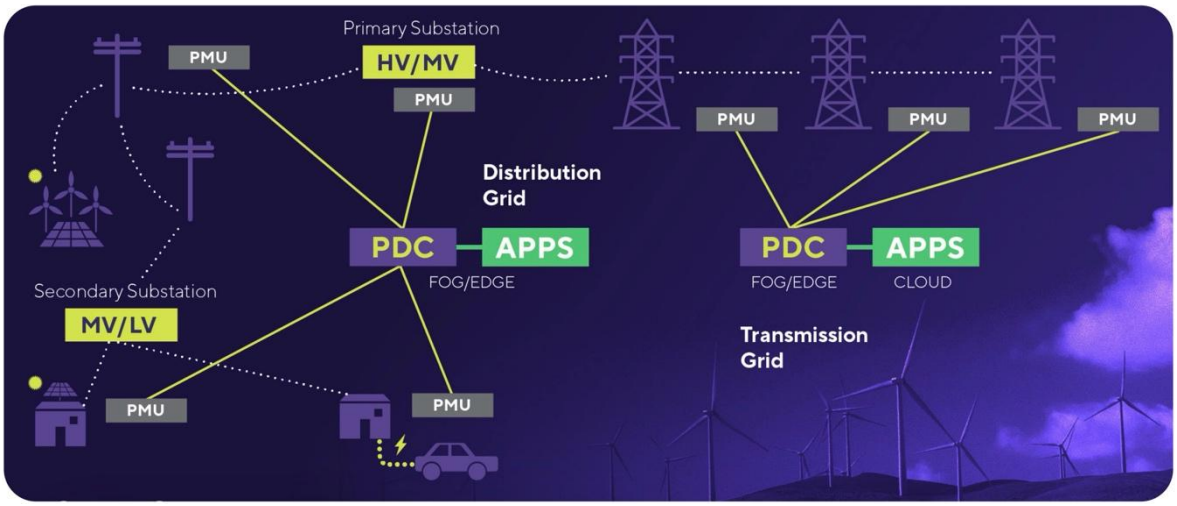

Figure 2: Intelligent Power Grid Use Case Overview

The growth of small renewable energy sources, like solar and wind power, means we need to monitor and control the power grid better, both for big transmission lines and local distribution networks. This means using more tech, like devices to measure and control , and setting up networks to handle all the data. This helps electricity to homes and businesses, even when demand or supply change suddenly. Having lots of small energy sources the

[1] https://www.fluidos.eu/use-cases/

Funded by Horizon Europe Framework Programme of the European Union

integration of technology for critical functions also brings potential risks., including system faults and cyber-attacks that could disrupt power supply. Moreover, making sure all these devices work together smoothly poses a significant challenge. Balancing the benefits of renewable energy and the complications of relying on so much technology, highlights the need for accurate planning and management. Further details of this use case are discussed in section 5.1.

2.2 UC2: SMART VITICULTURE (AGRICULTURE, TER)

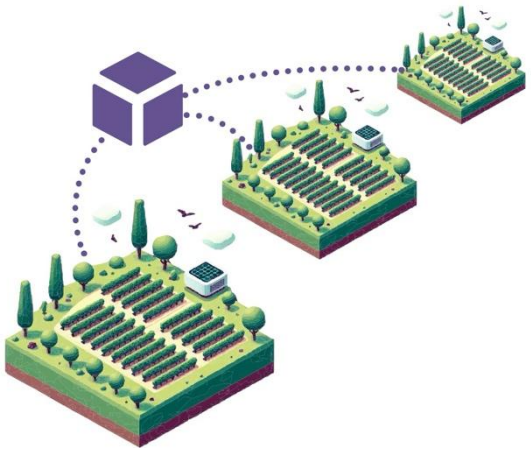

Figure 3: Smart Viticulture Use Case Overview

Terraview OS is an innovative tool for vineyard management, bringing together data from various sources to provide valuable recommendations and insights like crop estimates, smart watering recommendations, and early warnings for diseases. However, since it is cloud-based, it has difficulties when connecting with computers and IoT devices in areas with poor internet connectivity. For instance, tasks like using drones to survey fields can be slow, with delays in delivering results to users, leading to frustration. One way to tackle this for Terraview is to customise TerraviewOS for different setups, like cloud, on-site or hybrid options. However, this isn't ideal due to the complexity and costs involved in developing and maintaining such tailored versions. To overcome these hurdles, Terraview needs a system that handles various deployment setups, taking the pressure off their software team. This is something that FLUIDOS will provide. Further details of this use case are discussed in section 5.2.

Funded by Horizon Europe Framework Programme of the European Union

2.3 UC3: Robotic Logistics (Industry 4.0, ROB)

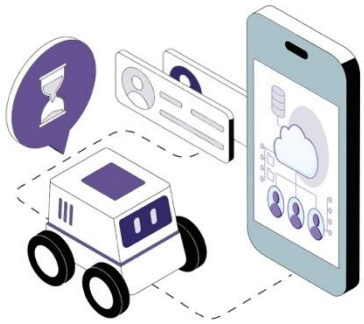

Figure 4: Robotic Logistics Use Case Overview

Mobile robots in Industry 4.0, smart logistics, and retail are battery-powered devices with limited resources. They operate in spaces shared with humans, other IoT devices, and fellow robots. The efficiency of these systems relies on robots' ability to perform tasks, battery life, and coordination. For example, in smart retail, robots interact with on-site devices to minimise data transmission to the cloud and ensure business continuity during network outages. Most of a robot's computing power is used for logistics, location tracking, basic movement, and communication with other devices, making it difficult to add new tasks without affecting performance or battery life. Challenges include real-time fleet coordination, requiring specialised devices, and complex deployment procedures compared to managing cloud devices. Further details of this use case are discussed in section 5.3.

Funded by Horizon Europe Framework Programme of the European Union

3 USE CASE DEVELOPMENT PROCESS

In order to leverage FLUIDOS technology effectively within these use case domains, a thorough analysis was required to initiate the delivery process. This began with the initial elicitation process.

WP7 follows a structured and iterative process to extract scenarios, actors, requirements, and architecture changes for all the use cases, aligned with processes from WP9. This process ensures that the developed solutions are both technically viable and closely aligned with the actual needs of end-users and stakeholders across different use cases.

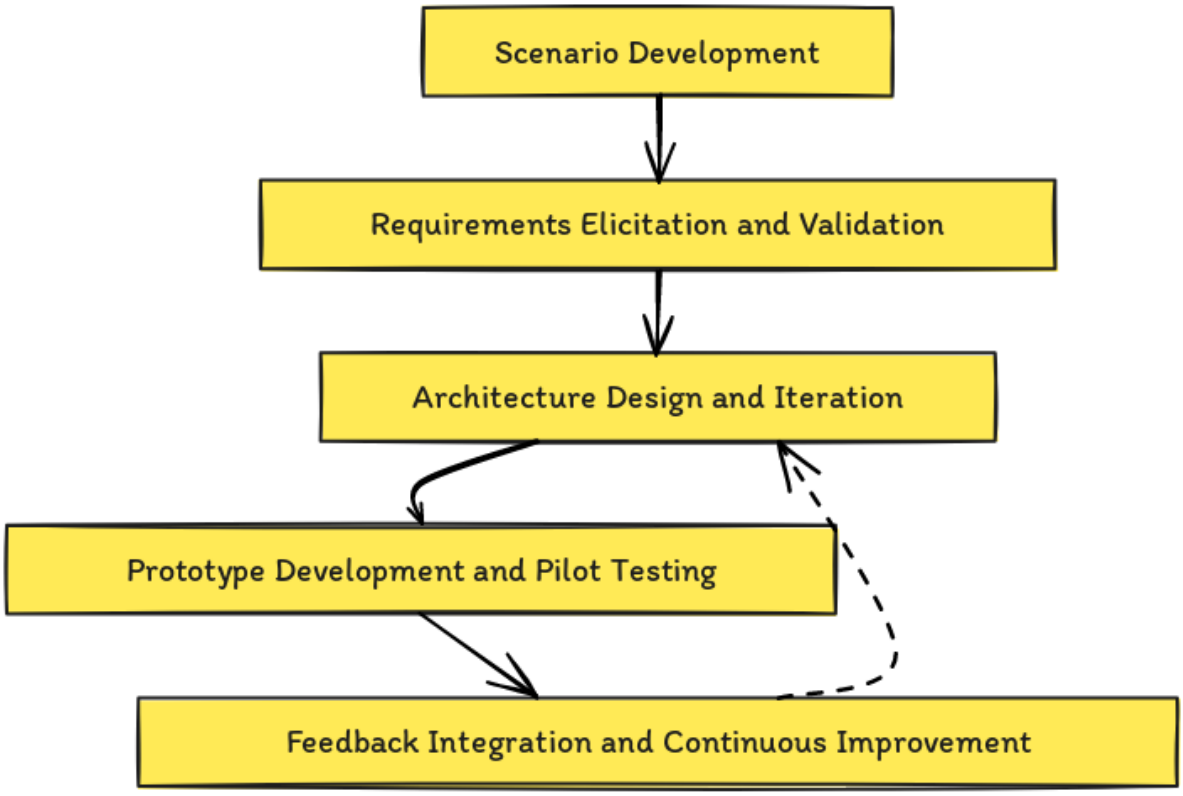

Figure 5: WP7 Process

The process involves the following stages:

Funded by Horizon Europe Framework Programme of the European Union

3.1 Scenario Development

- **Use Case Scenarios**: Develop scenarios for each use case that illustrate how FLUIDOS would be used in real-world contexts. Scenarios include descriptions of the environment, tasks, user interactions, and expected system responses.
- **Actor Identification**: Identify all actors involved in each scenario, including both direct users (e.g., system operators, farmers, logistics managers) and indirect users (e.g., customers, regulatory bodies).

3.2 Requirements Elicitation and Validation

- **Functional and Non-functional Requirements**: from the developed scenarios, extract detailed functional requirements (specific features and capabilities of the FLUIDOS system) and non-functional requirements (performance, usability, security, scalability).
- **Initial Requirements Validation**: as the requirements came from experienced domain stakeholders of the application to leverage FLUIDOS this part of the process was implicit. With this experience the requirements accurately reflect stakeholder needs and are technically feasible.

3.3 Architecture Design and Iteration

- **Initial Architecture Design**: based on the validated requirements, develop an initial architecture design based on the FLUIDOS platform that outlines key components, interfaces, data flows, and technologies. This for many of the use cases meant, not a complete redesign but more a modification of an existing architecture.
- **Iterative Refinement**: refine the architecture iteratively based on feedback, technological advancements, and pilot testing results. This happens as further understanding and refinement of the main FLUIDOS architecture happens.

3.4 Prototype Development and Pilot Testing

- **Develop Prototypes**: build functional prototypes of the FLUIDOS use cases based on the refined architecture to demonstrate the feasibility and effectiveness of the solution in addressing the use case scenarios.

Funded by Horizon Europe Framework Programme of the European Union

- **Pilot Demonstration**: demonstrate the use cases for stakeholders to validate the system's performance, usability, and impact.

3.5 Feedback Integration and Continuous Improvement

- **Gather Feedback**: collect detailed feedback from pilot tests, focusing on system performance, user experience, and any gaps or challenges encountered.
- **Incorporate Feedback**: integrate feedback into the development process, making necessary adjustments to the system's features, architecture, and user interfaces.

This process is cyclical and may be repeated multiple times throughout the FLUIDOS's lifecycle, allowing for continuous refinement of the system based on evolving user needs, technological advancements, and feedback from real-world application. This adaptation is in line with what is defined in WP9.

Funded by Horizon Europe Framework Programme of the European Union

4 USE CASE WIDE ARCHITECTURAL NOTES

Throughout all use cases and their related prototypes, the FLUIDOS architecture is seamlessly integrated, aligning the use case architectures with it. The DoA emphasizes the principle of: "*Do not reinvent the wheel: leverage existing operating systems, enriched by Kubernetes.*" Within WP7 the same approach was followed, and by doing so not only are the benefits of Kubernetes exploited, but also the advantages built-in the FLUIDOS architecture. Adopting FLUIDOS does not disrupt the architecture of the use cases, rather it enhances them due to FLUIDOS' additive nature; which adds new useful features to Kubernetes, Liquo and ARCA Trusted OS. This is a great advantage and reduces the risk of new technology adoption.

Moreover, as highlighted in the FLUIDOS DoA, "legacy containerised applications can run unchanged on FLUIDOS, according to the lift and shift approach." This means that transitioning from an existing architecture to one delivered via FLUIDOS is straightforward, particularly for simpler cases. Should an application be a monolith, then perhaps leveraging some of FLUIDOS' advantages might be difficult, although this scenario has not been encountered yet.

Fortunately, many use cases in WP7 are sensibly designed and already incorporate containerisation, or are in the process of doing so. Therefore, transitioning to FLUIDOS' underlying technologies does not pose a significant challenge, requiring primarily initial engineering effort. This is a HUGE[2] benefit of FLUIDOS.

4.1 Common Use Case Requirements

WP7 stands upon three core technical WPs (3, 4 and 5), guided by the architectural framework established in WP2. Each use cases in WP7 required a deep understanding of which features from those core technical WP would be used. Following analyses and discussions within WP7 and across the relevant work packages, the following prioritised list of common features was assembled. This shows the core technical features of FLUIDOS needed to support the specific requirements of each use case and their respective prototypes.

[2] Capitalisation is intentional.

Funded by Horizon Europe Framework Programme of the European Union

Table 1: FLUIDOS Core Technical Features for Use Cases

Priority	ID	SubID	FLUIDOS Feature	UC1 - SViti	UC2 - Energy	UC3 - Robotnik	Notes	Component Match	WP
1	TC1		Controlling a large number of (Edge) servers/clusters	yes	yes	yes		Node	WP3
2	TC2	Main	Connect applications hosted on different clusters	yes	yes	**yes**	network fabric	Node	WP3
4	TC8	main	Secure Workload Container Independence	yes	yes	yes		TEE, ArcaOS (Node)	WP5
5	TC9	Main	Intent-based Orchestration	yes	yes	yes		WP4 Orch	WP4
6	TC9a	01	Provide Intent Policy Prototypes	yes	yes	**yes**	use case to provide these after understanding policy language		WP4
7	TC9b	02	Intent orchestrator	yes	yes	**yes**		MetaOrchestrator	WP4
3	TC3	main	Cloud bursting	yes	yes	yes	dynamic resource provisioning	Broker & Node	WP3
8	TC6b		Migration of stateless applications across clusters	yes	yes	**yes**		node	WP3
	~~TC7~~		~~Access to Data spaces~~	~~no~~	~~no~~	~~no~~			
	~~TC4~~		~~Decentralised Federation~~	~~no~~	~~no~~	~~no~~			

From this point on in the document, the focus will be on the specifics of each use case and their status from the point of view of the prototypes that each are developing.

Funded by Horizon Europe Framework Programme of the European Union

5 INDIVIDUAL USE CASES

In this section of the document, the detailed descriptions of activities carried out per use case are presented.

5.1 UC1: INTELLIGENT POWER GRID

5.1.1 Use Case Description

The power grid is undergoing a paradigm shift, driven by technological innovation and the pressing need to integrate renewable energy sources. This transformation fosters a more dynamic and intricate energy system, characterised by fluctuating and geographically dispersed generation, bidirectional energy flows, and an amplified focus on energy efficiency. As the traditional, centralised model transitions towards a distributed and inherently unpredictable system, robust monitoring capabilities become increasingly critical. In this context, Information and Communication Technology (ICT) infrastructures play a pivotal role in facilitating real-time monitoring, control, and optimization of the power grid. By leveraging the potential of advanced sensors, sophisticated data analytics, and robust communication networks, ICT empowers grid operators to respond swiftly to fluctuations in supply and demand. This translates to enhanced grid reliability and seamless integration of diverse energy resources.

Within the emerging power grid architecture, two key components play a vital role: Phasor Measurement Units (PMUs) and Phasor Data Concentrators (PDCs). PMUs are measuring devices attached to the power grid, strategically positioned to measure electrical quantities, and equipped with a GPS module that guarantees a precise time synchronisation. These measurements are then transmitted at high sampling frequencies (from 10 to 100 frames per second depending on the application) to PDCs, following the protocol IEEE C37.118-2011[3] or IEC 61850-90-5[4]. PDCs function as data hubs, responsible for collecting, aggregating, and importantly, aligning the timestamps of geographically scattered PMU measurements.

Ultimately, the synchronised data collected by PMUs and managed by PDCs serves as crucial input for the Grid State Estimation (GSE) algorithm, which is the basis for real-time applications such as monitoring, control and protection of the grid, and offline applications such as archiving and offline analysis.

[3] "IEEE Standard for Synchrophasor Measurements for Power Systems," IEEE Std C37.118.1-2011, pp. 1-61, 2011.

[4] "Communication networks and systems for power utility automation - Part 90-5: Use of IEC 61850 to transmit synchrophasor information according to IEEE C37.118," IEC/TR 61850-90-5:2012.

Funded by Horizon Europe Framework Programme of the European Union

Traditionally, PMUs have been deployed within the transmission grid. However, the evolving power grid demands a more comprehensive monitoring approach, including the distribution level. To address this, PDCs can be strategically located in primary and secondary substations, as well as control centers. These PDCs can then be interconnected to form a hierarchical architecture, enabling a more holistic view of the grid.

However, adapting this infrastructure to the distribution grid comes with new challenges. Notably, the number of nodes requiring monitoring increases significantly (from hundreds to tens of thousands). Additionally, communication between these nodes and the control centre becomes less reliable. Unlike the transmission grid, where fiber optic cables ensure consistent connectivity, the distribution grid often relies on less stable connections, such as LTE. Therefore, in distribution grids, the importance of scalability and resiliency is even more crucial than in the transmission grid.

5.1.2 Actors

The main entities involved in a typical Intelligent Power Grid use case can be broadly categorised as follows:

- Power Grid Administrator: the actor that is responsible for the administration of the power grid, from technical to economic optimizations.
- Grid Maintenance Operator: the actor that is responsible for the maintenance of the power grid, for example in case of failures.
- ICT Operator: the actor that is responsible for developing, operating, and maintaining the private ICT infrastructure.
- Telecommunication Provider: the entity that offers and administers the public communication infrastructure.

In the following Figure 6, a graphical representation of the Intelligent Power Grid use case is presented.

Funded by Horizon Europe Framework Programme of the European Union

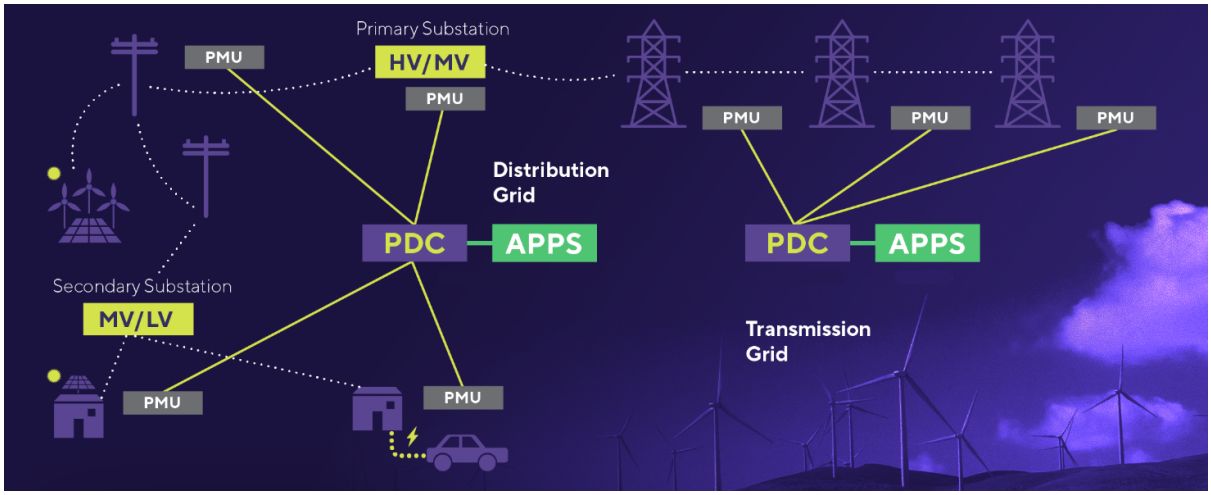

Figure 6: Intelligent Power Grid use case

The use case objectives are the following:

- **Real-Time Processing**: the system must support large-scale data processing in real-time, meeting the specific latency constraints of the targeted applications.
- **Resilient Grid Monitoring**: seamless phasor data concentration and grid state computation are required under conditions of multiple ICT outages, minimising the potential for electrical grid downtime.
- **Reconfiguration Continuity**: during grid reconfiguration, phasor data concentration must seamlessly migrate to prevent data loss and maintain uninterrupted grid state estimation.
- **Simplified Maintenance**: service offloading to nearby nodes should be facilitated to simplify ICT maintenance.
- **Local Processing for Fault Tolerance**: local processing capabilities should be implemented to prevent error escalation in scenarios of communication breakdown with the remote ICT infrastructure.
- **Hardware Efficiency**: hardware duplication should be no longer needed for effective redundancy.

5.1.3 Use Case Specific Requirements

Considering these use case objectives, a list of requirements has been provided to the FLUIDOS technical partners to deploy specific features. A preliminary version of these

Funded by Horizon Europe Framework Programme of the European Union

requirements was already listed in the deliverable D2.1[5], while a more extended one is reported below:

- **High Availability**: FLUIDOS must continuously monitor hardware and communication health and provide survival capabilities to ensure uninterrupted phasor data processing and grid state analysis even during outages.
- **Data Integrity**: data streaming should be preserved by FLUIDOS to prevent PMU data loss.
- **Maintenance**: FLUIDOS should handle maintenance activities without service interruptions so that seamless phasor data concentration and grid state computation in the presence of maintenance activity are obtained.
- **Monitoring**: FLUIDOS should provide dashboards and logs for real-time monitoring and post-incident analysis.
- **Alerting**: FLUIDOS should trigger alerts for critical events, enabling situational awareness for ICT and grid operators.
- **Orchestration Based on ICT Network Quality**: FLUIDOS should be aware of network quality so that workload placing can be automatically optimised to reduce the latency in the control loop decisions in case network quality requirements are not satisfied.
- **Orchestration Based on Electrical Grid Reconfiguration**: FLUIDOS should allow grid administrators to define workload locations (edge/cloud) so that workload placing can be optimised in case of electrical grid reconfiguration.
- **Hierarchy Consistency**: FLUIDOS should preserve hierarchy when moving workloads so that services are located consistently with the logical hierarchy.
- **Local Processing**: FLUIDOS should allow deployment on the local edge infrastructure services that are typically executed on central systems, guaranteeing excellent survivability properties also in case of widespread ICT outages.
- **Archiving**: FLUIDOS should facilitate archiving past data, stored on the edge, to the cloud so that storage resources on the edge are preserved.
- **Persistent Storage**: persistent storage between FLUIDOS nodes should be enabled to recover data and configurations in case of failure.

5.1.4 Architecture

The architecture of the Intelligent Power Grid Use Case is composed of PMUs, PDCs and the Grid State Estimation application. PMUs, strategically placed on the distribution grid,

[5] F. Risso, M. Iorio, S. Galantino, J. Marino, F. Valenza, R. Sisto, A. Cannarella, V. Coroama, N. Asadov, S. Braghin, A. Rawat, M. Suliman, A. Edmonds, A. Skarmeta and A. Zarca, "FLUIDOS D2.1 SCENARIOS, REQUIREMENTS AND REFERENCE ARCHITECTURE – V.1," Founded by Horizon Europe, 2022-2025.

Funded by Horizon Europe Framework Programme of the European Union

send synchrophasor measurements through the IEEE standard C37.118-2011[6] protocol to the PDC applications. Traditionally, PDCs are monolithic applications running on dedicated hardware. In recent years, experimental efforts have been underway to use virtualization and orchestration techniques for the deployment of PDCs and real-time analysis applications at the edge[7]. The Intelligent Power Grid Use Case leverages a containerized version of the PDC service, based on the OpenPDC[8] software administered by the Grid Protection Alliance. The containerized version of OpenPDC requires a MySQL database for the storage of configurations which is also containerized and can therefore be executed in a fluidified environment. Data collected by PDCs is sent to the Grid State Estimation algorithm that accurately estimates the operating conditions of the electrical grid. In the Intelligent Power Grid Use Case, the Grid State Estimator algorithm is presented in the form of microservices that can in turn be orchestrated.

In the context of orchestration, the adoption of the FLUIDOS computing continuum is a significant progress. The primary goal of FLUIDOS is to seamlessly integrate processing capabilities scattered across numerous edge devices, servers, and on-premises data centers, thereby establishing a cohesive computing continuum. The FLUIDOS computing continuum is not simply the deployment of microservices across multiple sites; it constitutes a virtual space spanning diverse technological domains irrespective of their physical locations. Inter-service communications are facilitated by FLUIDOS itself, ensuring uninterrupted connectivity regardless of location. FLUIDOS enables each microservice to be scheduled in the most optimal location leveraging intent-based orchestration capabilities and facilitates dynamic optimizations by redeploying services in new optimal places at runtime. A foundational premise of FLUIDOS is its reliance on Kubernetes technology, which abstracts underlying physical resources and capabilities.

A baseline hierarchical architecture encompassed by the Intelligent Power Grid Use Case is depicted in the figure below. Within this structure, the components are distributed across FLUIDOS Nodes, spanning from the edge to the private cloud. This configuration

[6] "IEEE Standard for Synchrophasor Data Transfer for Power Systems," IEEE Std C37.118.2-2011 (Revision of IEEE Std C37.118-2005), pp. 1-53, 2011.

[7] S. Galantino, F. Risso, A. Cazzaniga, F. Garrone, R. Terruggia and R. Lazzari, "An Edge-based Architecture for Phasor Measurements in Smart Grids," in 2022 AEIT International Annual Conference (AEIT), Rome, 2022.

[8] Grid Protection Alliance (GPA) , "OpenPDC," [Online]. Available: https://github.com/GridProtectionAlliance/openPDC.

Funded by Horizon Europe Framework Programme of the European Union

establishes a virtual environment where PDCs and Grid State Estimation algorithms can be orchestrated seamlessly by FLUIDOS.

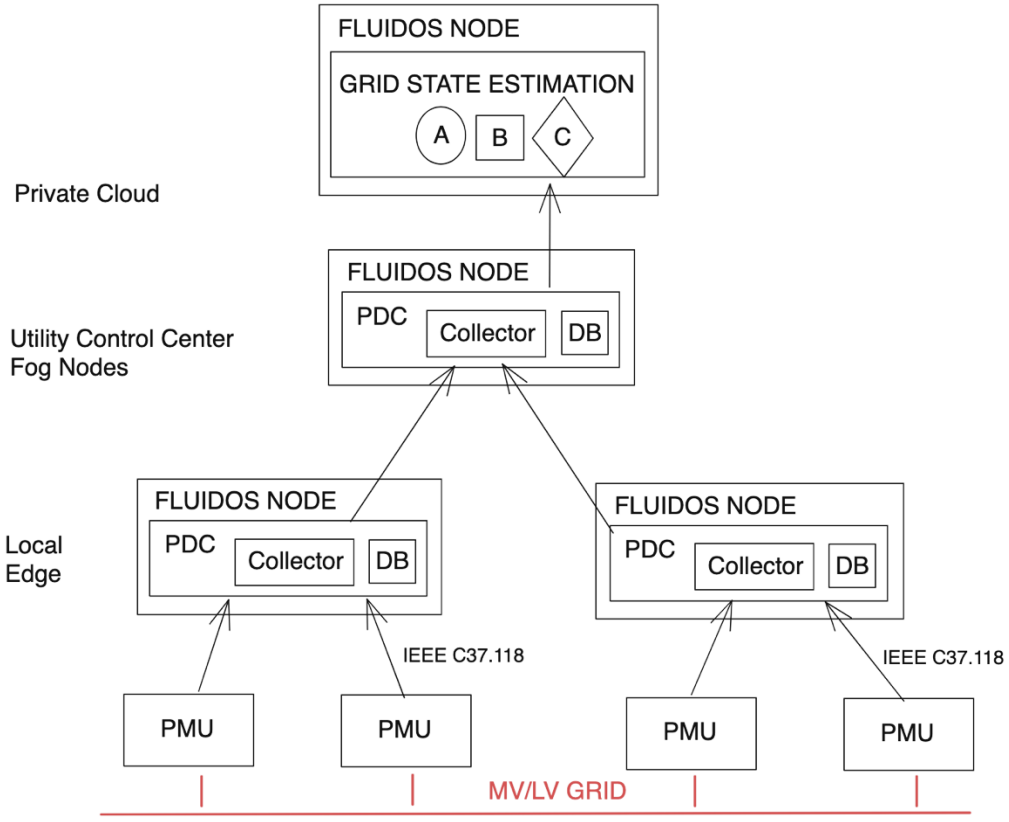

Figure 7: Intelligent Power Grid Use Case architecture composed of physical sensors Phasor Measurement Units (PMUs), Phasor Data Concentrators (PDCs) in a containerized version which includes a collector microservice and a MySQL database (DB) and the Grid State Estimation Algorithm developed in the form of microservices (generic A, B, C). PDCs and Grid State Estimations functions are running on FLUIDOS nodes, taking part in the FLUIDOS computing continuum and leveraging its advanced features.

The main features enabled by FLUIDOS are:

- **Computing Continuum**: FLUIDOS would enable PDCs and analysis applications to continue functioning even in case of faults or communication interruption with control centers by migrating PDC services to an adjacent FLUIDOS node, enhancing the resilience of the proposed architecture.
- **Intent-Based Orchestration**: FLUIDOS can automatically orchestrate PDCs based on the latency between the node and PMUs, thereby improving the power grid state estimate or responding to faults.

Funded by Horizon Europe Framework Programme of the European Union

- **Cybersecurity**: FLUIDOS ensures service isolation from other applications on the hosting node with different usage permissions. It also leverages logging and anomaly detection capabilities and provides survival capabilities in case of a cyber-attack.

Considering the definition of requirements and the main features enabled by FLUIDOS, three interesting use case scenarios are investigated:

- **Maintenance**: the FLUIDOS architecture facilitates simplified ICT maintenance by enabling service offloading to nearby nodes. This capability allows for seamless upgrades to the PDC service or the implementation of generic security patches without incurring data loss or interrupting critical Grid State Estimation computations.
- **ICT Outage**: in the event of an ICT fault occurring within a FLUIDOS node hosting the PDC micro-services, the FLUIDOS computing continuum capability is leveraged. Consequently, the PDC is transitioned to nearby computational resources, ensuring uninterrupted collection of synchrophasor data with seamless continuity, an example of the ICT Outage scenario is presented in Figure 8.
- **Power grid reconfiguration**: the hierarchical structure between PMUs and PDCs may change in the event of a grid reconfiguration. A new PMU, which was previously connected to another PDC, may need to connect to a new PDC, but latency exceeds a threshold value. Being able to migrate applications across clusters accounting for network metrics can lead to a substantial improvement in latency, allowing the grid state estimation algorithm to satisfy the stringent requirements of real-time applications. An example of the grid reconfiguration scenario is presented in Figure 9.

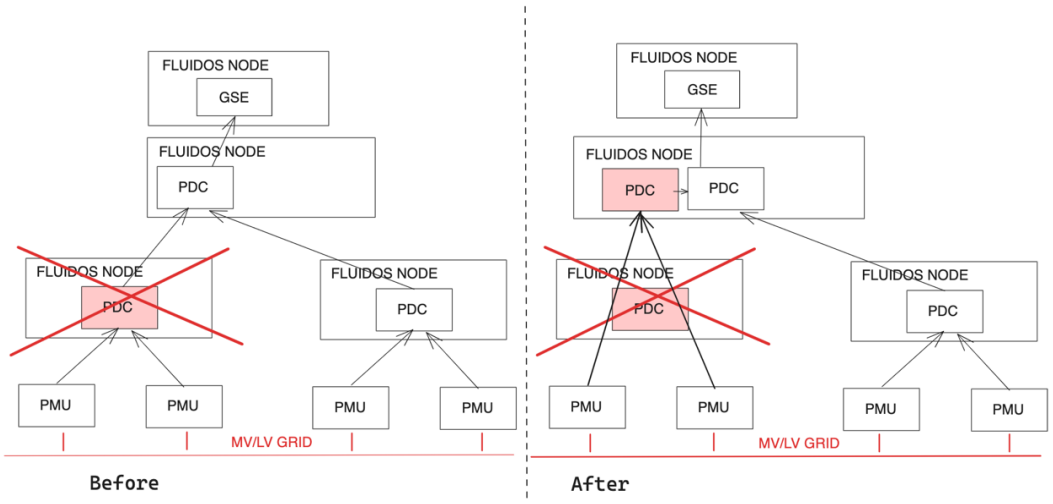

Figure 8: ICT Outage scenario, an ICT fault of a FLUIDOS Node is depicted schematically; FLUIDOS should guarantee that the PDC is migrated to a nearby FLUIDOS node ensuring uninterrupted collection of synchrophasor data with seamless continuity.

Funded by Horizon Europe Framework Programme of the European Union

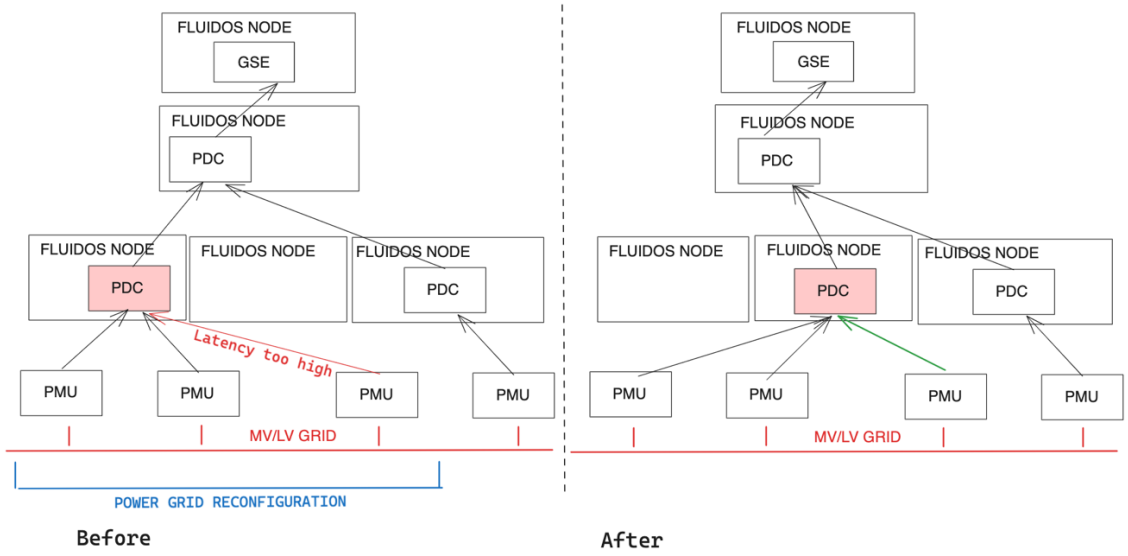

Figure 9 In the event of a grid reconfiguration, where a PMU must transmit phasor measurements to a newly assigned PDC, the latency between the PMU and PDC could be excessive. FLUIDOS steps in to address this issue by strategically scheduling the PDC on an adjacent node, thereby minimising latency and ensuring efficient data transmission.

In summary, the expected advantages of the adoption of FLUIDOS in the Intelligent Power Grid Use Case are:

- **Resilience**: FLUIDOS should enhance the capacity to tolerate and respond to faults, forced disconnections, and maintenance.
- **Application Performance**: the Intent-Based Orchestration of FLUIDOS should enhance the performance of the phasor data collection in terms of an overall reduction of the latency experienced by monitoring and control applications.
- **Scalability**: FLUIDOS should allow to effectively manage a high number of PMUs.
- **Cybersecurity**: the FLUIDOS environment will enhance the security of the ICT infrastructure that supports the Intelligent Power Grid Use Case with advanced capabilities, such for example, authentication, authorization, availability of Trusted Execution Environments (TEE) and Remote Attestation (RA) and anomaly detection features.

5.1.5 Implementation and Integration

5.1.5.1 Testbed

Experimental tests are conducted through the implementation of the presented architecture within the FLUIDOS computing continuum. The testing infrastructure encompasses the Distributed Energy Resources Test Facility (DER-TF)[9], situated at the premises of Ricerca sul

[9] C. Gandolfi, R. Lazzari, M. Zanoni and D. Palladini, "MISSION Project: Design of MV/LV hybrid AC/DC smart-energy grid," in AEIT International Annual Conference, Rome, 2022.

Funded by Horizon Europe Framework Programme of the European Union

Sistema Energetico (RSE S.p.A) in Milan. The DER-TF Facility, shown in Figure 10, is characterised by its experimental low-voltage microgrid, comprising distributed energy sources such as photovoltaic panels and wind turbines, storage systems, and various loads including charging stations for electric vehicles. Strategically positioned within the microgrid are five real PMUs developed on the hardware platform cRIO of NI Instruments on which the OpenPMULabview Project code was installed. The PMUs measure three-phase voltage and current amplitude and phase. These PMUs furnish real phasor data, facilitating comprehensive monitoring of the microgrid's operational status. In general, the DER-TF Facility serves as an experimental testbed for assessing the integration and performance of diverse technologies, alongside the implementation and evaluation of monitoring and control strategies tailored for the management of the grid.

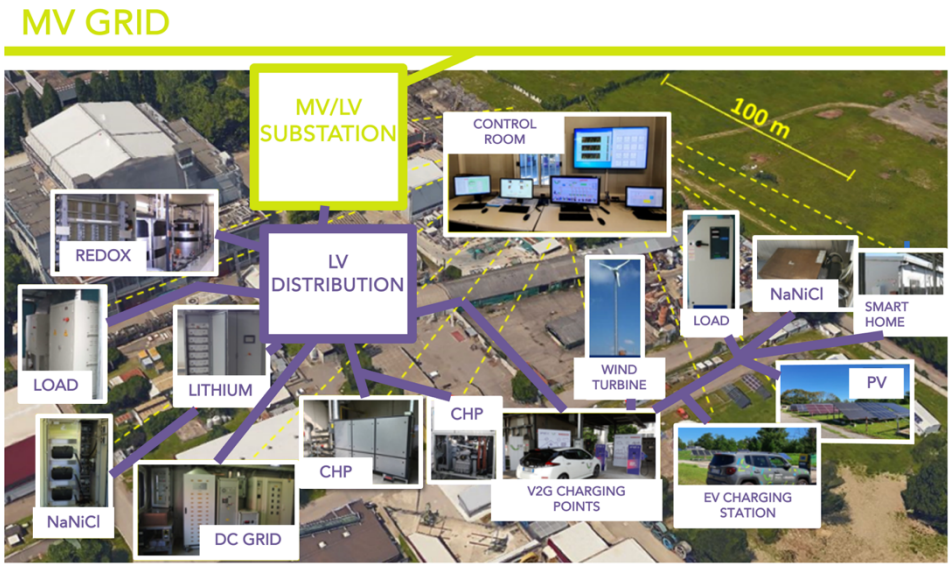

Figure 10: RSE Distributed Energy Resources Test Facility (DER-TF) and its major components (Figure derived from[10]).

The DER-TF seamlessly interfaces with RSE's laboratories through an Ethernet and optical fibre network as well as mobile technologies such as 4G and 5G for added flexibility and connectivity. Within RSE's laboratories, a dedicated private cloud based on OpenStack environment[11] hosts numerous virtual machines, facilitating the execution of various tests and research endeavors. In the Control Room of the RSE DER-TF Facility, industrial computers, and edge nodes, such as Raspberry Pi, are available to test the FLUIDOS environment on hardware with reduced computational resources. The purpose of testing involves the implementation and evaluation of the hierarchical architecture comprising containerized PDCs and the Grid State Estimation algorithm. The PDCs are strategically distributed to form

[10] C. Gandolfi, R. Lazzari, M. Zanoni and D. Palladini, "MISSION Project: Design of MV/LV hybrid AC/DC smart-energy grid," in AEIT International Annual Conference, Rome, 2022.

[11] OpenInfra Foundation, "OpenStack," [Online]. Available: https://www.openstack.org/.

Funded by Horizon Europe Framework Programme of the European Union

an edge/cloud infrastructure, seamlessly orchestrated to create a cohesive computing continuum. With the aim of exploring FLUIDOS scalability, we're exploring the utilisation of virtual PMUs, which can operate alongside their real counterparts. These virtual PMUs will run within RSE's private cloud infrastructure, providing valuable insights into the scalability and performance of our system.

5.1.5.2 Implementation

To explore the full potential of FLUIDOS capabilities, it is necessary to delve into the deployment of the Intelligent Power Grid applications, emphasising a fluid native approach. As proposed in previous studies[12], an initial hierarchical architecture of PDC services was obtained using k3s[13], a lightweight version of Kubernetes, suitable for resource-constrained edge devices such as single board computers, for example, Raspberry Pis available in the RSE DER-TF. The PDC application, based on the containerized version of the openPDC software, needs to store its configuration parameters in a MySQL database. The PDC service also encompasses an OpenPDC manager which is a Windows-only application that offers a UI for the database and configuration of an openPDC instance; a snapshot of OpenPDC Manager is reported in Figure 6.

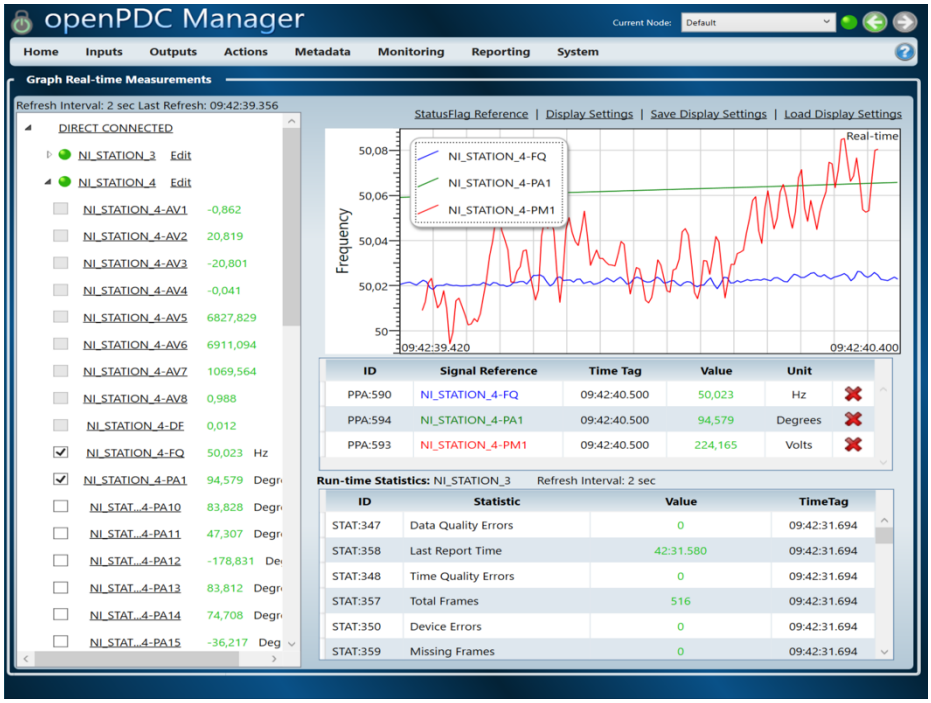

Figure 11: Snapshot of the OpenPDC manager, UI of the containerized version of openPDC service; on the left, PMUs and relative metrics are listed; on the top right, frequency trends measured by

[12] S. Galantino, F. Risso, A. Cazzaniga, F. Garrone, R. Terruggia and R. Lazzari, "An Edge-based Architecture for Phasor Measurements in Smart Grids," in 2022 AEIT International Annual Conference (AEIT), Rome, 2022.

[13] The Linux Foundation, "K3s," [Online]. Available: https://k3s.io/.

Funded by Horizon Europe Framework Programme of the European Union

three different PMUs of the Distributed Energy Resources Test Facility are displayed in real-time; on the bottom right, run-time statics are reported.

Given the potential fault scenarios previously outlined for the PDC application, where it may need to be instantiated on another FLUIDOS node, ensuring data persistence becomes crucial. In the k3s lightweight environment, achieving data persistence has been accomplished through the utilisation of the Longhorn[14] solution. Longhorn ensures data replication across nodes of the same Kubernetes cluster so that information is evenly duplicated across multiple nodes, reducing the risk of data loss in the event of failures or issues with a single node. Additionally, the ability to create volume snapshots provides an effective mechanism for backup and restore management, enabling developers to preserve the data state at a given moment.

Concurrently, with the development of the FLUIDOS node by technical partners of the project, initial components of the FLUIDOS environment have been tested considering Intelligent Power Grid workloads. Preliminary tests have been conducted leveraging the Virtual Fabric Manager, which is the FLUIDOS component responsible for establishing computing continuum abstractions to enable the seamless execution of workloads spread across multiple nodes. The Virtual Fabric Manager component, presented in Deliverable 2.1 of the FLUIDOS project[15], is based on Liqo[16], an open-source project which extends the Kubernetes abstractions to multi-cluster scenarios.

The initial implementation encompasses the creation of two virtual machines with 8GB of RAM, 4 vCPU and a fresh volume of Ubuntu 20.04 LTS on the RSE's private cloud infrastructure. On both VMs, a single-node Kubernetes cluster using k3s v1.23.14 was installed. The RSE Private Cloud is connected to the VLAN of the Distributed Energy Resources Test Facility, giving the possibility to connect the developed services to the real PMUs present in the low-voltage portion of the facility.

The software requirements are Helm[17], which is a package manager for Kubernetes that simplifies the deployment and management of applications, Kubectl, the main command-line tool (comes as part of any distribution) used to interact with Kubernetes clusters, and Liqoctl, which is the command-line tool that simplifies the installation and management of the Virtual

[14] The Linux Foundation, "Longhorn," 2024. [Online]. Available: https://longhorn.io/.

[15] F. Risso, M. Iorio, S. Galantino, J. Marino, F. Valenza, R. Sisto, A. Cannarella, V. Coroama, N. Asadov, S. Braghin, A. Rawat, M. Suliman, A. Edmonds, A. Skarmeta and A. Zarca, "FLUIDOS D2.1 SCENARIOS, REQUIREMENTS AND REFERENCE ARCHITECTURE – V.1," Founded by Horizon Europe, 2022-2025.

[16] Liqo, "Liqo," [Online]. Available: https://docs.liqo.io/en/v0.10.1/.

[17] The Linux Foundation, "Helm," [Online]. Available: https://helm.sh/.

Funded by Horizon Europe Framework Programme of the European Union

Fabric Manager component. The Virtual Fabric Manager of the FLUIDOS Node is then installed on both k3s clusters, namely the consumer and the provider. To guarantee the data persistence inside each k3s cluster the Longhorn storage system was installed, and coherent storage class parameters were configured. Subsequently, leveraging the Virtual Fabric Manager, the peering between the two k3s clusters has been established and the namespace of the consumer cluster has been offloaded to the provider cluster, creating a computing continuum between the two computational resources.

The computing continuum has been tested using the containerized version of the PDC application and the MySQL database. The PDC application is developed in the form of a Kubernetes Deployment in two parts: one which sets up the MySQL database with configurations and the second part which runs the main application. The PDC can be accessed externally through a Kubernetes Service definition. The MySQL database is created as a Kubernetes Deployment which defines a Persistent Volume Claim (PVC) to claim storage resources (Persistent Volumes, PV), specifies the container image to use and finally creates a Kubernetes Service to expose the database externally. The PDC and MySQL deployments were applied in the offloaded namespace and, therefore, scheduled onto the consumer or provider k3s cluster.

As a result of the initial test of the Virtual Fabric Manager FLUIDOS component, two important issues in the original implementation of the Intelligent Power Grid workloads have been highlighted.

Firstly, the MySQL pod is a stateful application which requires persistent storage. The Virtual Fabric Manager creates PVs in the exact location where their associated pods have been scheduled for the first time, if a subsequent scheduling process takes place (e.g., following a restart), a set of automatic policies attracts pods in the cluster where the PV is already created. Thus, preventing the MySQL pod from being scheduled on another node. Consequently, multiple replicas of the MySQL database should be deployed on the two clusters to allow the PDC to migrate seamlessly.

Secondly, inside a k3s cluster, the data persistence was guaranteed by the Longhorn storage system. However, when extending the architecture beyond a single k3s cluster, such as in a multi-cluster environment like that introduced by FLUIDOS, Longhorn is not able to guarantee this property anymore. In a multi-cluster context, it is essential to consider new approaches which guarantee data persistence at an application level. In the Intelligent Power Grid use case, solutions to replicate data between multiple MySQL replicas should be considered.

Funded by Horizon Europe Framework Programme of the European Union

Tools currently under experimentation are MySQL Group Replication[18] and Percona XtraDB Cluster[19] which both provide a high-availability MySQL clustering solution that offers synchronous multi-master replication.

Leveraging the same testing setup, an examination of the system's resilience properties was initiated. The initial focus was placed on a fully stateless application, such as the Grid State Estimation Algorithm. The aim was to evaluate how the system responds to a basic ICT outage scenario. This involved simulating the outage by sequentially powering off the Virtual Machine hosting the provider cluster, followed by the VM hosting the consumer cluster. Subsequently, distinct behaviours were observed as outcomes of these tests:

- If the stateless application pods are scheduled on the consumer cluster and the provider cluster experiences a fault, pods continue to run seamlessly on the consumer cluster.
- If the stateless application pods are scheduled on the provider cluster and the consumer cluster experiences a fault, pods continue to run on the provider cluster leveraging the Virtual Fabric Manager resilience features.
- If the stateless applications pods are scheduled on the provider cluster and the latest experience a fault, pods are rescheduled on the consumer cluster.
- If the stateless applications pods are scheduled on the consumer cluster and the latest experience a fault, pods are not rescheduled on the provider cluster, thus not being able to recover from the fault.

As a result of these observations, collaborative efforts with our technical partners are underway to develop multiple peering architectures between FLUIDOS nodes. This initiative aims to enhance the system's resilience to ICT outages, ensuring continuity of operations even in adverse conditions.

The ongoing initial implementation of Intelligent Power Grid use case is visible in the WP7 – Intelligent Power Grid Use Case repository on the GitHub of the FLUIDOS project.

[18] Oracle, "MySQL Group Replication," [Online]. Available: https://dev.mysql.com/doc/refman/8.0/en/group-replication.html.

[19] Percona, "Percona XtraDB Cluster," [Online]. Available: https://www.percona.com/mysql/software/percona-xtradb-cluster.

Funded by Horizon Europe Framework Programme of the European Union

5.2 UC2: SMART VITICULTURE

5.2.1 Introduction

TerraviewOS, a cloud-native service for agriculture, helps end-users manage their property inputs and outputs. This requires Terraview to install and operate software at both the property premises and in the cloud, which are intended for different operations (e.g., data gathering at the edge, and data processing in the cloud). At the customer premises, each node offers the TerraviewOS service to distinct customers and can be bought or leased from Terraview. The connectivity between nodes and central TerraviewOS service varies based on available network technologies at customer locations. Local nodes handle specific TerraviewOS functions, while the central service provides additional capabilities. Every FLUIDOS node corresponds to a customer managing one or more vineyards, using shared resources from Terraview. Customers operate through "Crate" edge devices, aiming to integrate a central service by FLUIDOS technology to address specific problems.

The current architecture of TerraviewOS is a set of cooperating microservices. These microservices are packaged as Docker containers, are currently deployed, and operate upon Docker Swarm as the orchestration technology. In the following diagram, the TerraviewOS architecture is shown, before the introduction of the FLUIDOS technology.

Funded by Horizon Europe Framework Programme of the European Union

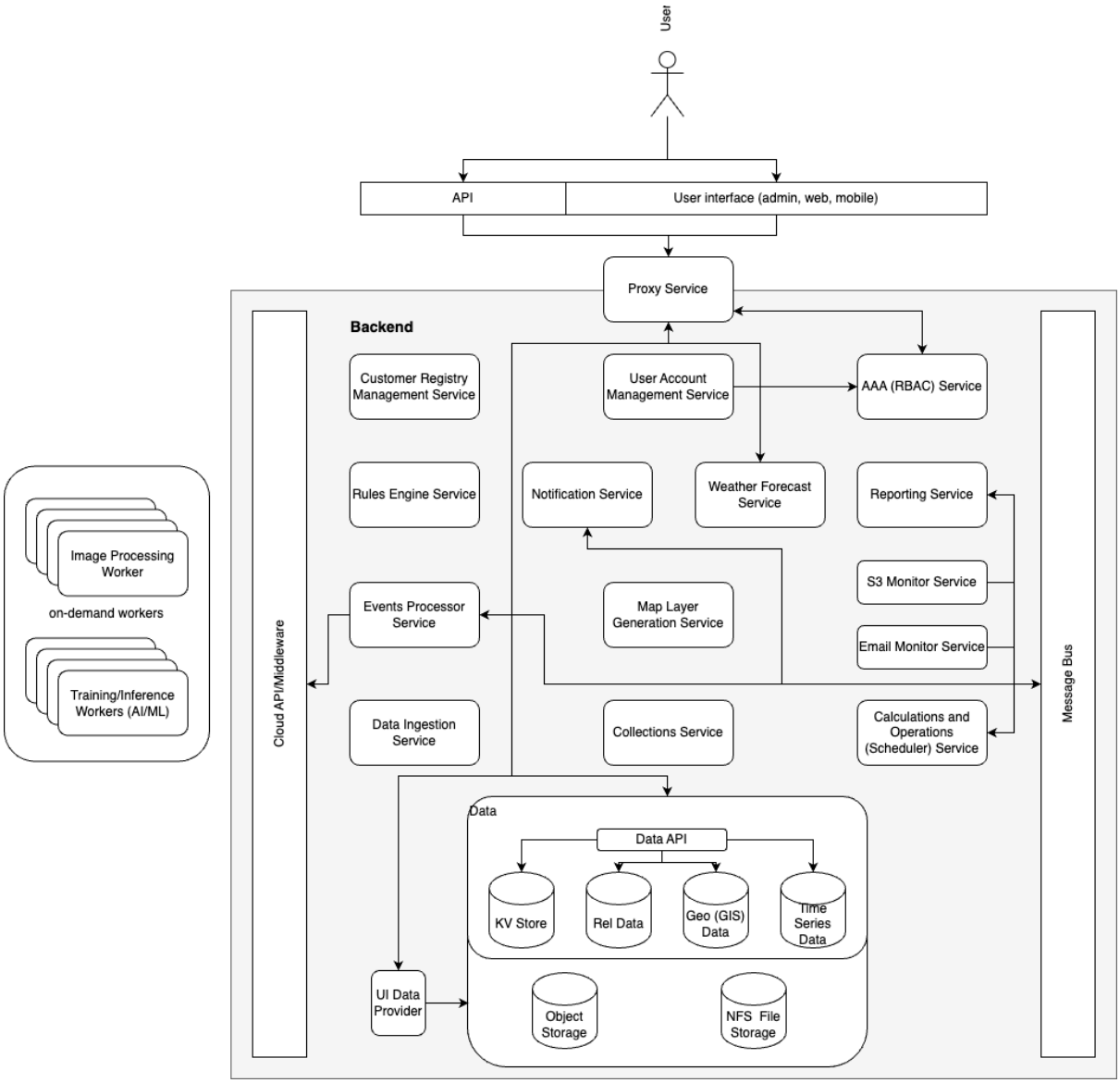

Figure 12: TerraviewOS Architecture Before FLUIDOS Adaptation

All the microservices execute using docker technology and the orchestration technology has been Docker swarm. To use FLUIDOS technology and its innovations, it was necessary to migrate the complete application stack from Swarm to Kubernetes, which required the reengineering of our application stack.

With the current architecture and implementation presented, we'll now recap some of the core aspects of the system, starting with the users (actors) that use the system. Understanding the actors and then the customisations needed to be made to the application stack to match the initial application scenario, then the changes that have been made to the migrated Kubernetes application stack will be presented.

Funded by Horizon Europe Framework Programme of the European Union

5.2.2 Actors

The actors in this use case have remained the same since last reported in D2.1. For reference, those actors are the following:

Table 2: Smart Viticulture Actors

Name	Description
Admin (TVAdmin)	The administrator of the TerraviewOS system that includes core cloud and edge deployments
Customer (End user)	The end user of the system. They typically interact with the system through a user interface run inside their web browser.
Crates	This is the edge hardware device used to deliver on-premises service to the customer
Developer	The actor that is responsible for the implementation and management of one or more components (services) of TerraviewOS

5.2.2.1 Tenants

Each customer that has an account on the TerraviewOS system is deemed to be a tenant. Each tenant corresponds to the billable entity and is the organisation that is contracted with and consequentially receives the bill for services rendered. All tenants need the system to ensure that they all operate upon the TerraviewOS system as isolated entities. This is important to maintain as TerraviewOS transitions to an edge-core deployment.

The different nodes should not be able to interfere with each other. The backend services should run in the cloud, however, the use case should not have one backend instance per customer edge node, so workloads and data that are in the cloud must be securely shared among FLUIDOS nodes. As such, workloads are required to run on TEE-equipped systems, which will be defined in the application descriptors through intents.

5.2.2.2 Locations

The design of the use case is location independent. For demonstration purposes, the tenant (customer) edge nodes (3) are separate from each other, however, they are both

Funded by Horizon Europe Framework Programme of the European Union

located at the offices of Terraview GmbH. All tenants share the TerraviewOS core (TVOS-Core), which is a managed Kubernetes instance provided by Microsoft Azure. Upon this TVOS-Core Liqo is installed.

5.2.3 Use Case Specific Requirements

The requirements for this use case have not changed as reported in D2.1, however, the list has now been prioritised to provide the first prototype of the use case. Below are the 3 key use case-specific requirements:

Table 3: Key Use Case Requirements

ID	UC Na	Reporter	Title	As a...	I want...
WP7-ViticultureUC-20	SV	andy@terraview.gmbh	**Workloads on TEEs**	Crates	execute all processes supported by the use of a TEE using FLUIDOS technology
WP7-ViticultureUC-5	SV	andy@terraview.gmbh	**Offload compute**	Crates	move unused but dependent functionality from the edge to the core using FLUIDOS
WP7-ViticultureUC-7	SV	andy@terraview.gmbh	**Declare per-app compone**	Developer	to be able to specify the location of my workload execution (edge/cloud) using FLUIDOS

Continued…

So That...	Category	Tech/UC	Priority	Y2 Priority
I am assured that my workloads are secure and cannot be tampered with		TC	high	high
there is more local processing available or given there's less processing, there's less power consumption		TC	medium	high
Workload placing can be optimised based on costs and network limitations		TC	high	high

Once these requirements are satisfied, other requirements will be implemented in forthcoming releases of the use case.

5.2.4 Architecture

The current architecture of TerraviewOS has been revised to identify necessary additions, removal and changes required to support the use case. Specifically, the components to be

Funded by Horizon Europe Framework Programme of the European Union

added/removed/modified were determined to support the initial prototype and meet both the technical requirements and the use case-specific requirements as detailed above.

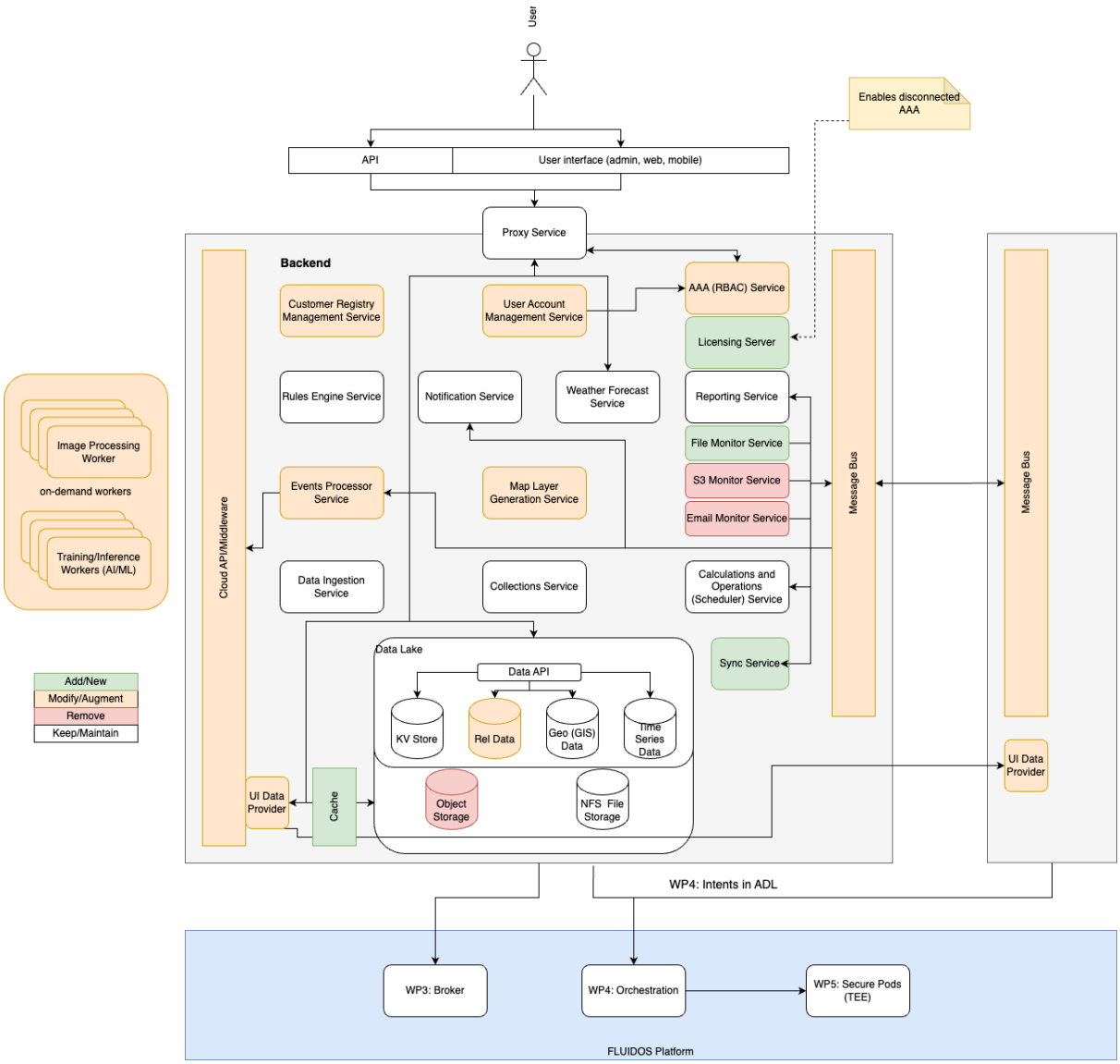

Figure 13: Terraview Architecture Adapted to FLUIDOS

In the architecture figure above, it is important to note the colors assigned to each component in the diagram. Green represents new components to be added to ensure complete "edgification" of Terraview OS, orange indicates the components that need to be modified, in red are the ones to be removed and white are those to remain and unmodified.

The new items to be added are:

Funded by Horizon Europe Framework Programme of the European Union

- **Cache**: this feature will store data locally on the device after it has been requested and retrieved from the remote core TerraviewOS instance.
- **Licensing Server**: this components performs periodic checks to the core TerraviewOS instance to validate the customer's license, ensuring continued access to the service on the device.
- **File Monitoring Service**: this service monitors for new imagery provided to the edge node, typically resulting from drone surveys. The processing of this imagery occurs locally rather than remotely.
- **Sync Service**: this service ensures that locally held data is synchronised with the authoritative data maintained on the core TerraviewOS instance.

Components that remain, are to be removed or modified will not be detailed for reasons of brevity.

The key FLUIDOS features that will be leveraged are:

- Orchestration
- Broker
- TEE

Their relationship to the TerraviewOS architecture is shown in figure 13.

5.2.4.1 Workloads

The workloads (functionalities of TerraviewOS) that will be initially supported have been described in Deliverable 2.1[20]. However, the table below provides an overview of the selected functionalities to be provided in the first prototype.

Table 4: Target TerraviewOS Workloads

Name	Description	Selection

[20] F. Risso, M. Iorio, S. Galantino, J. Marino, F. Valenza, R. Sisto, A. Cannarella, V. Coroama, N. Asadov, S. Braghin, A. Rawat, M. Suliman, A. Edmonds, A. Skarmeta and A. Zarca, "FLUIDOS D2.1 SCENARIOS, REQUIREMENTS AND REFERENCE ARCHITECTURE – V.1," Founded by Horizon Europe, 2022-2025.

Funded by Horizon Europe Framework Programme of the European Union

YPS	This carries out Yield prediction of vine plots under the management of the customer	This will not be used in the first prototype.
Image processing	This processes drone imagery collected from customer vine plots.	This will be used in the first prototype as this currently introduces large amounts of data transfers from the customer to the core.
Aquaview	This service calculates soil moisture levels of a particular customer vine plot.	This will be not used as the processing is largely based on satellite imagery and does not make sense to

From the processing perspective, the key processing workloads of TerraviewOS are considered. The ones that are most resource-intensive are:

- **Image Processing**: this processing is related to all image processing that results in agronomic information (indices) that inform the end user (vineyard operator) of crop health aspects. The imagery sources are mainly from satellites and drones.
- **Yield Prediction**: this process uses an AI/ML workflow to derive a prediction of crop yield for the vineyard plots under management by the customer.
- **SMM**: this workload calculates the soil moisture over the set of vineyards to produce information needed to drive strategic decisions on irrigation.

For the initial work, the Image processing workload will be used as a workload to offload to the Crate (edge unit). The main reason is that the information is not time-critical and does not require GPU hardware on the Crate edge unit.

The functionalities listed below will be extended as the development of the prototype continues beyond year two.

- Implementation of additional features for the edge device, including the offloading of additional processing

Funded by Horizon Europe Framework Programme of the European Union

- Improvement of the dynamic deployment strategy and configurability of the setup, allowing the end user to specify data privacy and locality requirements. The system will then respond accordingly with the placement of services and computation.

5.2.4.2 Use Case Relevant Intent Metrics/Attributes

For the initial prototype, orchestration will be carried out by the intent-based orchestrator coming from WP4. To effectively use this orchestration, the prototype must define its intents. Initially, the set of intents are a relatively simple set of attributes that each node should satisfy. The key intent, and currently the only one, is as follows:

- The edge node where the workload is deployed must have TEE capabilities. This is a key enabler for privacy and data locality.
- The edge node where the workload is deployed must have hardware resource specifications that match at a minimum those that are defined for the edge node hardware.

There is no anticipation of needing intents based upon network characteristics like bandwidth, latency etc. Additionally, there are no expectations of any intents that require dynamic changes or monitoring to be respected.

5.2.4.3 Scenario: Demonstrating Secure Intent-based Cloud Fluidity.

Every FLUIDOS node corresponds to a customer managing one or more vineyards, using shared resources from Terraview. Customers operate through "Crate" edge devices, aiming to integrate a central service by FLUIDOS technology to address specific problems.

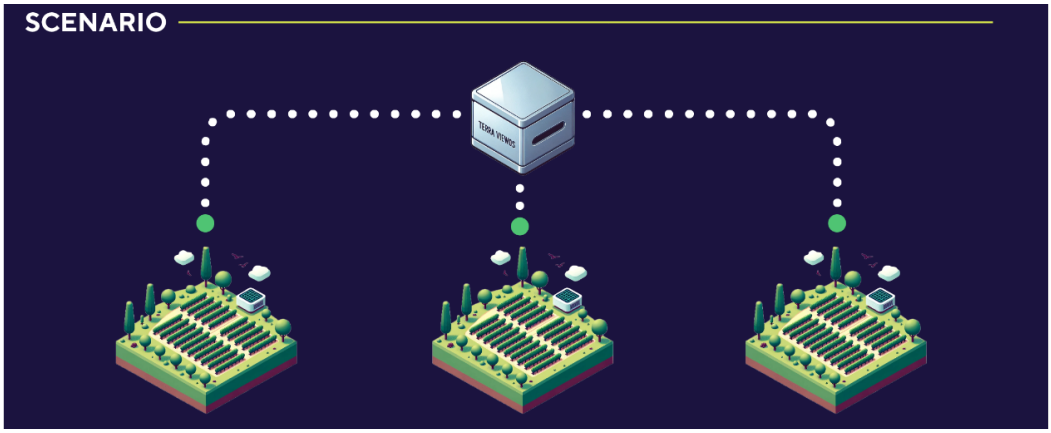

Figure 14: Smart Viticulture Scenario Overview

This scenario and related work enable the basic operations of a fluid and liquid cloud continuum. This will enable the business of Terraview Crates.

Funded by Horizon Europe Framework Programme of the European Union

The deployment of the application stack begins from the core and extends to the edge. New service components are deployed to the edge based on the application configuration. What is deployed onto edge units are a subset of the entire TerraviewOS stack. Currently in the prototype, those components are the UI and image processing.

Figure 15: Smart Viticulture Hybrid Deployment

These workloads execute within a hybrid deployment model. At the core, utilizing Microsoft's Azure Managed Kubernetes service, there is no TEE capability, however, workloads deployed from a central management interface to the edge are exclusively placed on edge devices with TEE capabilities. This selective placement is guaranteed by the FLUIDOS orchestrator, ensuring that edge deployments only occur on hardware that aligns with Terraview's requirements.

The use case scenario involves the deployment of the edge devices and their communication with the core system in the cloud. The workload needs to be split between the computation that is handled on the edge devices themselves, and what is performed by the centralised core system. The selection of edge workloads depends on their **processing requirements**, prioritizing time-sensitive computations for the edge. **Privacy and data locality** are also key considerations, ensuring that raw data stays encrypted and does not need to be transmitted to the central system. This setup improves the overall quality of service, especially in cases where the edge nodes experience intermittent network connectivity.

Below is an overview of the multi-tenant prototype that is under development. The key aspect here is that all services available on the edge system can be deployed locally or on the core deployment without the end user needing to be aware of or concerned with this distinction.

Funded by Horizon Europe Framework Programme of the European Union

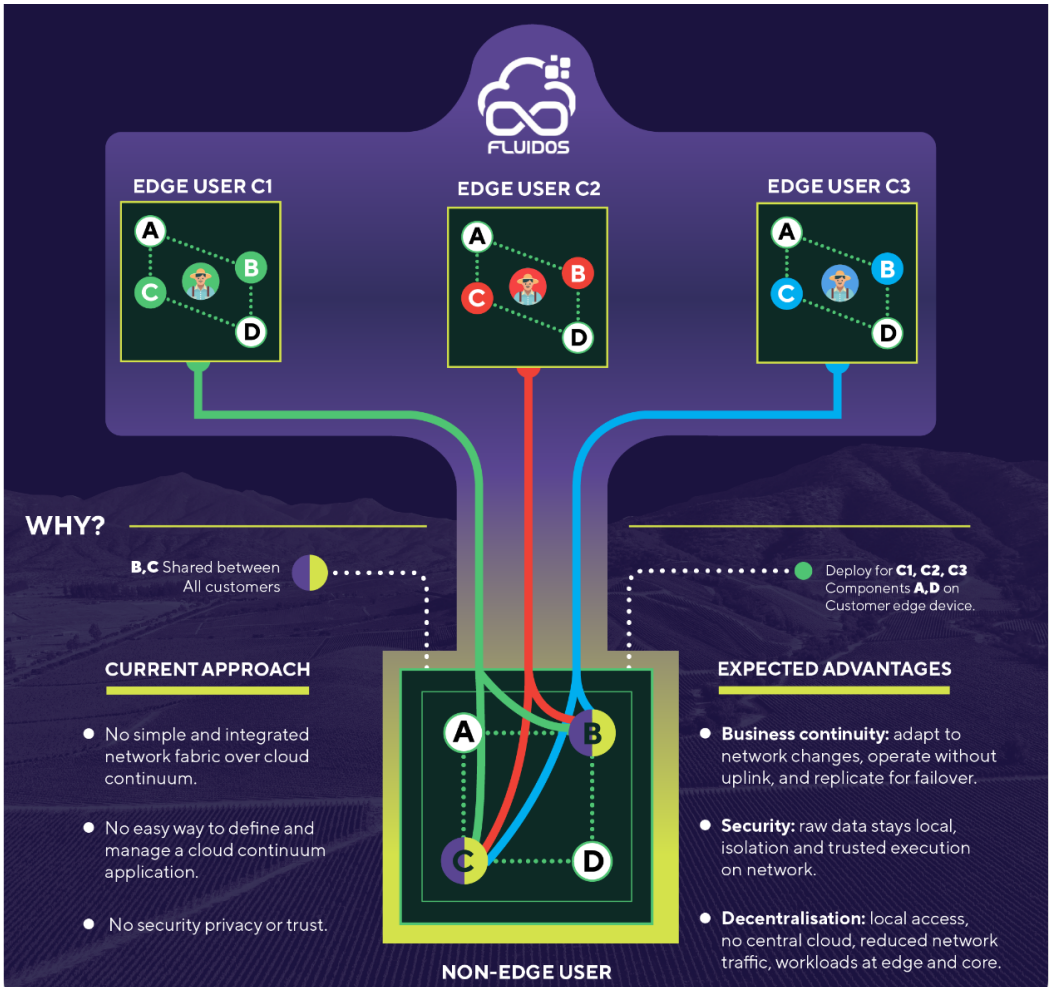

Figure 16: Smart Viticulture Scenario Details

The following sequence diagram provides a brief overview of the scenario from the perspective of key FLUIDOS architectural components.

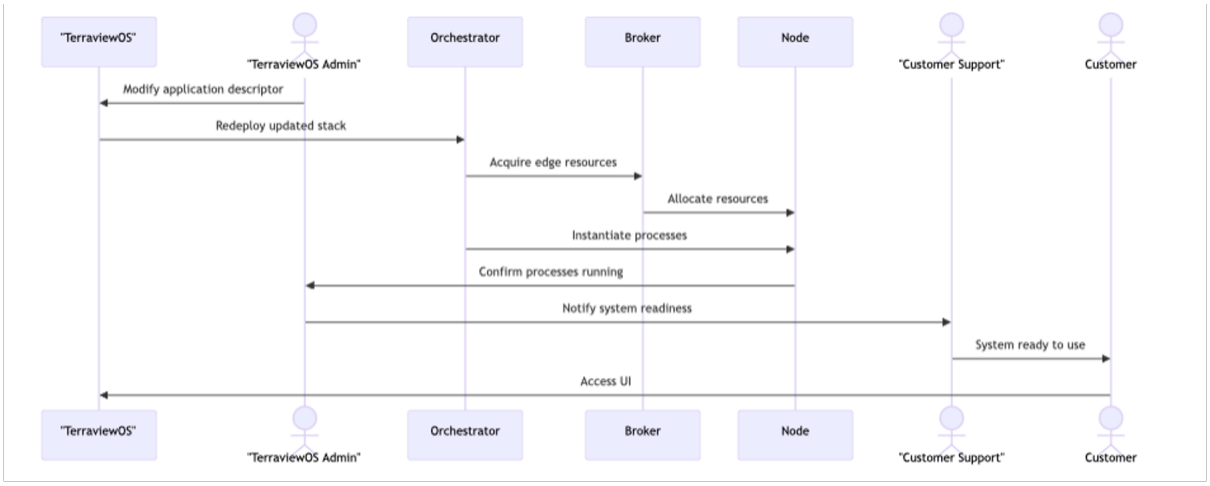

Figure 17: Smart Viticulture Scenario Platform Interactions

Funded by Horizon Europe Framework Programme of the European Union

5.2.5 Implementation and Integration

5.2.5.1 Testbed

The first integration step is to set up the edge testbed. The setup comprises 4 edge nodes, each a ruggedised PC, upon which the technology stack is deployed.

The hardware that is hosting the FLUIDOS prototype is an Advantech UNO-2372G V2 edge computing node.

Figure 18: Avantech Compute Unit UNO-2372G V2[21]

Table 5 gives an overview of its specifications:

Table 5: Advantech Compute Unit Specifications

System Hardware	Memory	Built-in 4GB DDR3L 1333MHz (Up to 8GB)	Built-in 4 GB DDR3L, 1866 MHz (Up to 8 GB)
	Graphics Engine	Intel HD Graphics	Intel HD Graphics 500
	Ethernet	Intel® i210/ Realtek i8119i GbE, 802.1Qav, 802.1AS, 802.3az	
	Hardware Security	TPM2.0 (Optional)	
	Storage	1 x mSATA shared with mPCIe socket 1 x 2.5" SATA HDD drive bay (Max. height 9.5 mm)	
	Expansion	2 x Full size mPCIe slots	
	Bios	AMI UEFI (64 Mbit)	
	Watchdog Timer	Programmable timer with 255 intervals (1 ~ 255 sec)	

The specifications table above indicates that the unit can be upgraded to 8GB of RAM. This upgrade has been implemented, and testing has been conducted on the image processing software to ensure compatibility with this specification.

The logical deployment of the hardware and software is described in the following diagram.

21 https://www.advantech.com/en-eu/products/1-2mlj9a/uno-2372g-v2/mod_18e89fa0-bdfa-4c7d-b431-138213855968

Funded by Horizon Europe Framework Programme of the European Union

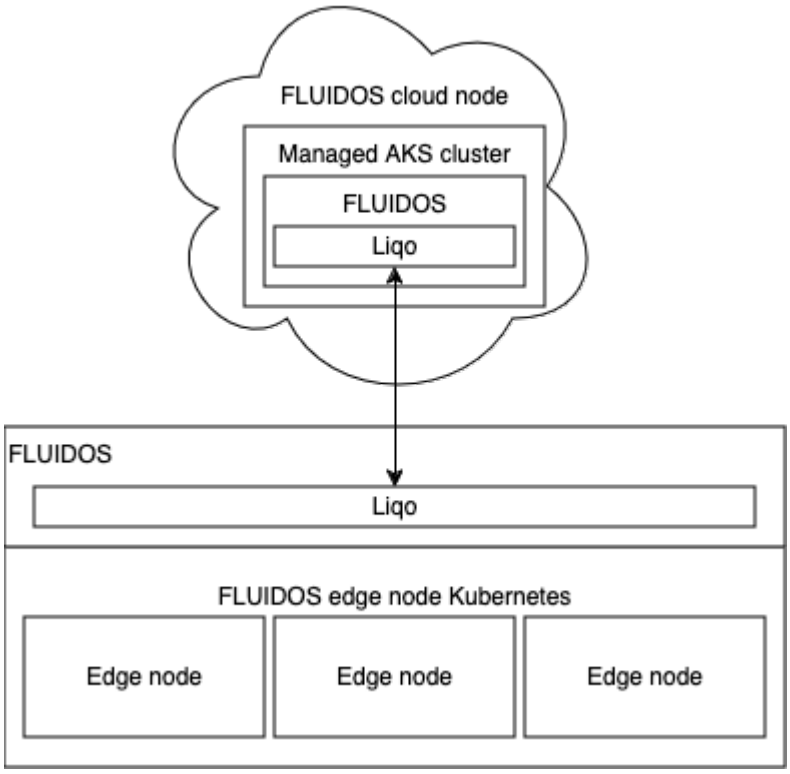

Figure 19: Logical Deployment of Hardware and Software

The edge nodes can either function as an edge cluster or as 4 single-node Kubernetes clusters, depending on the needs of the overall system. However, for this use case and prototype, each node represents one tenant and therefore each is a FLUIDOS node. Liqo is the FLUIDOS component that provides the network fabric capability.

5.2.5.2 TEEs & ARCA Trusted OS

Before the rest of the technology stack could be deployed, a Trusted Execution Environment (TEE) had to be set up on the test bed. To achieve this, the Arca Trusted OS (v1.9.1) was installed on all nodes as their operating system. This operating system also includes pre-installed utilities for setting up a Kubernetes cluster. A Kubernetes cluster was thus set up per edge unit using the 3 edge devices as FLUIDOS nodes.

5.2.5.3 Kubernetes & TerraviewOS

The next step was to prepare the Terraview service stack for deployment to the new setup. For this, the stack needed to be migrated from Docker Swarm to Kubernetes (k3s v1.23.14). The migrated stack was first validated on one edge node and then converted into a Helm chart, a packaging format for Kubernetes' package management system, Helm. This chart was then deployed onto Microsoft Azure, forming the cloud part of the hybrid setup.

Funded by Horizon Europe Framework Programme of the European Union

In more detail, from the original deployment scripts of the TerraviewOS platform, an edge optimised AIO (All-in-one) deployment workflow was created. This deployment included everything needed to run the platform and access the data but did not include any of the computationally heavy processing. These workloads are to be deployed selectively as described above.

This AIO deployment was still managed in a Docker swarm environment and deployed with Docker Compose. This was then converted to a Kubernetes deployment. This was no mean feat and generated over 90 separate Kubernetes manifests. Those manifests are available in the FLUIDOS repository[22]. A test cluster was created on a single node, and the Docker Compose was translated into Kubernetes manifests.

After verifying the operation of the Kubernetes version of the AIO deployment, it was converted into a Helm chart, to ease operations, and redeployed on our target cloud environment, an Azure AKS managed Kubernetes cluster. It was then further extended with additional features not present in the AIO, to reach feature parity with the core, centralized version of TerraviewOS.

Now, both the edge and core versions of the platform are deployed using the Helm chart, and the stack is deployed on the edge nodes of the testbed, running on ARCA Trusted OS.

5.2.5.4 TerraviewOS & FLUIDOS

5.2.5.4.1 Liqo

The hybrid setup consists of the deployment on Azure, connected to the deployment on the edge, using Liqo (v0.10.2) for providing and managing the network fabric per tenant spread across the continuum between the edge and the cloud (core).

Liqo provides the networking layer, allowing secure communication of the core cluster and each of the edge clusters.

This hybrid setup, augmented by the secure platform provided by ARCA Trusted OS and the seamless secure networking provided by Liqo, ensures that functionalities can be split between the computationally heavy workloads done in the data centre (cloud, within Azure

[22] https://github.com/fluidos-project/smart-viticulture-uc

Funded by Horizon Europe Framework Programme of the European Union

in this case), and the secure and private part handled on the edge, executed directly on the edge nodes. Those nodes are selected based on their declared intents.

5.2.5.5 Progress

The current state of the prototype has all the basic requirements now satisfied. The focus has shifted to the deployment of the key FLUIDOS components for orchestration. Once this is completed, the creation of the needed inputs to the FLUIDOS orchestrator per tenant will be carried out. This will then allow the examination of the deployment process and the understanding of the system functions, from image processing on the edge node to the visualisation of those results on the locally deployed user interface. A key area for discovery and feature-enablement will include discussions with CYSEC on how best to leverage ARCA Trusted OS in the context of TEE and securing workloads in operation, data in transit and at rest. In parallel, efforts will be directed towards securing the deployment with a licensing service and implementing content caching mechanisms for tenant use.

Funded by Horizon Europe Framework Programme of the European Union

5.3 UC3: Robotic Logistics

5.3.1 Introduction

Mobile robots for Industry 4.0, smart logistics, and retail are resource-constrained and battery-powered mobile robots that operate in shared spaces with humans and other IoT devices, such as elevators, automatic doors, and other mobile robots. For autonomous operation, mobile robots are equipped with sensors and high-performance computational algorithms that

(1) need considerable computing power and
(2) need to be executed at the highest speed.

5.3.1.1 Mobile Robotics Problems

This kind of robotics, by definition, use battery-powered devices, unlike stationary robot arms. This adds constraints to an already complicated application. In robotics logistics, idle robots are not productive. And intrinsically, robots need to charge their batteries to continue working. During this time, the robot is not available to move things around.

The robot's software is complex and requires heavy computation, so it is mandatory to equip the robot with powerful enough computing engines that, however, should limit the consumed power. Onboarding an ultra-low-resource device will lead to a robot that is not capable of processing the required tasks, and onboarding an ultra-high-resource device will deplete the battery very quickly.

There is an additional relationship between the computer load and the battery drain, heavy computational tasks drain the battery faster.

5.3.1.2 Problem Definition

The FLUIDOS continuum, e.g., the capability to transparently use computing resources nearby, can increase the overall system's productivity by intelligently and dynamically externalising the robotics workload to other devices and using the robot's idle time (i.e., when the robot is locked in the battery charging station) to increase the entire system's computational capabilities. This approach will lead to a significant decrease in battery usage and an increase in the robot's computational capabilities beyond its onboard limits.

Additionally, FLUIDOS will achieve this goal without interfering with the robotics task, so robot developers will be able to run their applications without changing their way of working.

Funded by Horizon Europe Framework Programme of the European Union

5.3.1.3 Traditional robotics problems

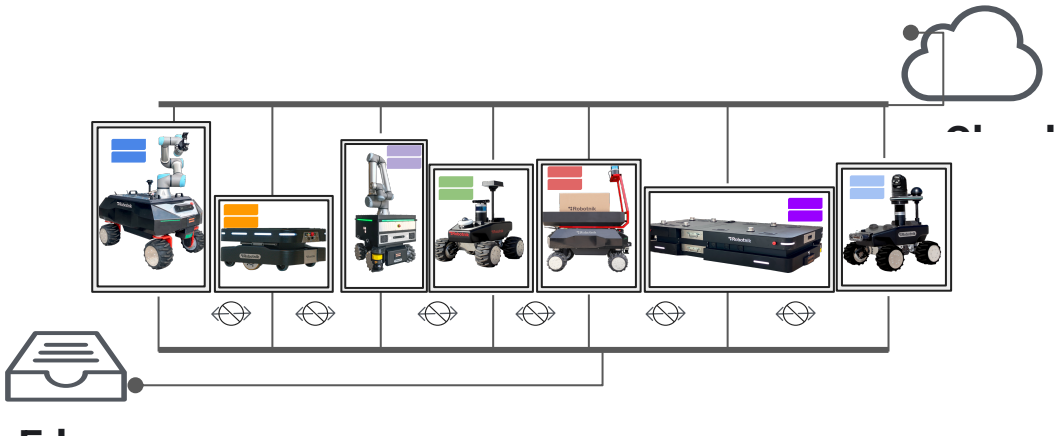

Figure 20: Robotic Fleet

5.3.1.3.1 Centralised communication

The robot fleet follows a centralised communication architecture, meaning that all communication between the robots and the fleet manager (or monitor system) is routed through an edge or cloud device. This means that even if the network infrastructure allows it, the robots do not communicate directly with each other.

5.3.1.3.2 Individual computation

For simplicity, robotic developers typically compute everything on the same robot (they already have a lot of robotic development and configuration to do). This leads to situations where some robots are idle and underused, while others are overloaded with computation and on the verge of collapsing their onboard computers.

5.3.1.3.3 Energy-unaware Usage

Mobile robots do not optimise their workloads based on energy parameters (e.g., when energy is more sustainable), and all the robotics workloads are running all the time, regardless of the robot state, which consumes energy from the battery source. This approach can lead to increased energy consumption and lower productivity.

5.3.2 Actors

- **Robot**: for this use case, an Autonomous Mobile Robot (AMR) or Autonomous Mobile Manipulator (AMM) will be considered as a robot. An autonomous mobile robot (AMR) is a type of robot that can navigate and move through its environment without human control or direct oversight. AMRs are equipped with sensors and software that allow them to understand and respond to their surroundings, and they can perform tasks autonomously without the need for human intervention. An

Funded by Horizon Europe Framework Programme of the European Union

autonomous mobile manipulator (AMM) is a type of robot that combines the mobility of a mobile platform (AMR) with the dexterity of a robotic manipulator arm. It's important to stress that in this case the robot is battery powered so energy management is a key element for its performance. In this case the robot it's composed from the hardware perspective of at least x86_64 PC, DC motors, Battery, Battery Manager, USB device(s), Ethernet device(s) and router with switching, mobile and wireless connectivity capabilities. From the software perspective, the robot has some computational workloads that must be done in the robot and there are others that could be executed somewhere else.

- **Robotic Fleet**: a robotic fleet refers to a group or collection of autonomous mobile robots that work together to perform tasks in a coordinated manner. The fleet can consist of multiple robots that operate within a specific environment, such as a warehouse or manufacturing facility, and are managed by a centralised system or software.
- **Fleet Manager**: the robot fleet manager is a supervisory software that controls the tasks and objectives of multiple autonomous mobile robots operating in a warehouse or other environments. The fleet manager oversees the operation of the fleet, assigns work tasks to individual robots, and optimises the movement and coordination of the robots to maximise efficiency and productivity. This software will be placed on the edge, in the warehouse facilities, at least for the moment.

5.3.2.1 Tenants

In this use case, only a single tenant will be considered. The owner of the robotic fleet will be a single and only tenant of all the machines of the fleet including all edge devices and robots.

5.3.2.2 Locations

The physical environment where the robotic fleet and the edge devices will be considered as "single geographical location". However, when examining the locations from a computing perspective:

- All grid powered machines available for FLUIDOS will compose the "edge" locations.
- Each robot will be considered a different location.

5.3.3 Use Case Specific Requirements

The robotic logistic use case requirements for FLUIDOS can be summarized as follows:

Funded by Horizon Europe Framework Programme of the European Union

- Robotic workload must be containerized, which will enable more efficient use of resources and better management of workloads across different devices.
- Kubernetes will be used to orchestrate the containerized robotic workloads, ensuring that each robot has its own mastery over its workload even if it does not have direct communication with other robots or edge devices.
- FLUIDOS will provide additional computing capabilities for each robot in the robotic fleet and the edge devices, which will help to increase autonomy and efficiency of the whole fleet.
- Robots must inform FLUIDOS orchestration about their specific metrics such as battery level, current latency among remote devices, or robot state, which will help FLUIDOS to improve the efficiency of the whole fleet.
- FLUIDOS will attend energy and sustainability variables to orchestrate the robotic workloads across the continuum.
- FLUIDOS must improve the efficiency of the whole fleet by providing an orchestration of the robotic workloads across the continuum, which will help to decrease the deployment time of the robots and increase the autonomy of the robots.
- FLUIDOS must increase the autonomy of the robots by offloading and/or switching off using an intent-based orchestration among the continuum, which will help to reduce energy consumption and improve sustainability.
- FLUIDOS will attend to the robotic specific metrics such as battery level, current latency among remote devices, or robot state, which will help to improve the efficiency of the whole fleet.

5.3.4 Architecture

With FLUIDOS, the cloud continuum computing approach can be applied by considering each robot as an edge device and intelligently and dynamically outsourcing robotics workloads to other robots or devices depending on the environment.

Funded by Horizon Europe Framework Programme of the European Union

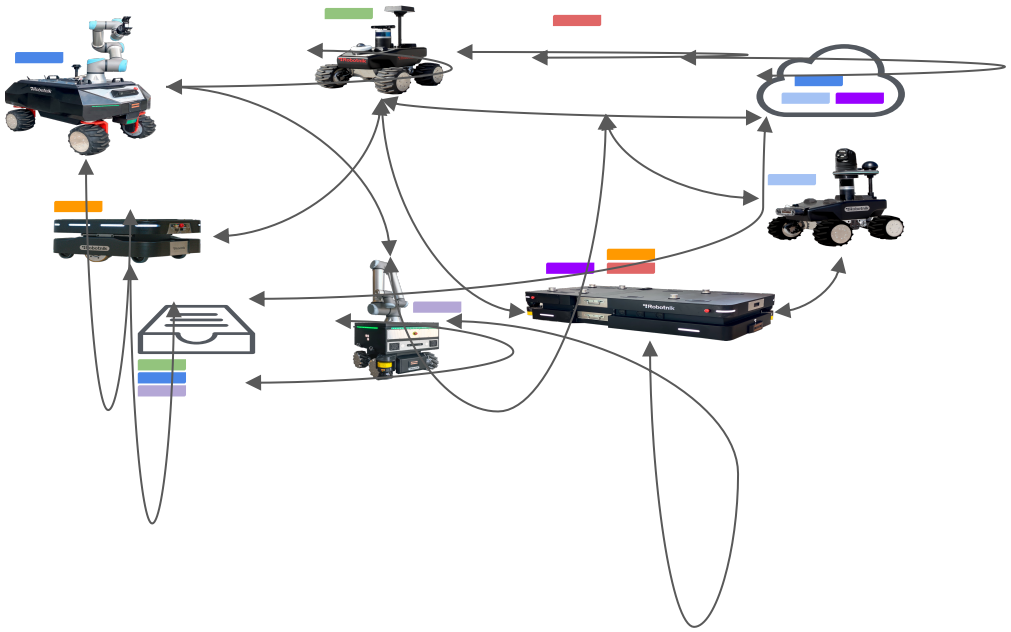

Figure 21: Robotic Logistics Architectural Overview

Instead of using monolithic bare-metal workloads, the robots will use cloud-native technologies like containerization and Kubernetes to split and dynamically place the workloads in different devices. All robots will be treated as edge devices that can accept or externalize workloads, instead of being isolated devices.

Due to the battery-powered nature of the robots, communications among the devices must be wireless. With this level of interaction, simpler architectures that consider a global control plane (on stationary devices) and treat the robots as drones or worker nodes without self-management capabilities cannot be used. Wireless communications are prone to temporary blackouts or low-quality zones where the robots move, potentially rendering the robot inoperative if communication is lost.

To mitigate this issue, FLUIDOS uses a multi-cluster federation architecture with Liqo, which allows all robots (or other servers) to remain autonomous even during periods of communication blackouts.

Funded by Horizon Europe Framework Programme of the European Union

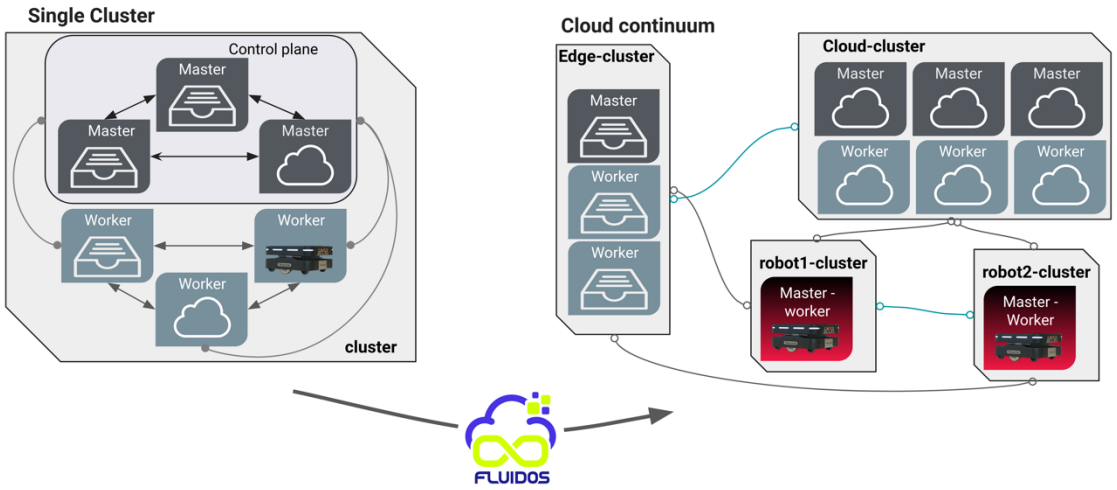

Figure 22: Multi-cluster federation architecture allowing all robots to remain autonomous

5.3.4.1 Use Case Objectives

FLUIDOS's intent orchestration allows to specify a set of objectives to achieve with the robot fleet, beyond the reach of Kubernetes. FLUIDOS will then intelligently and dynamically place the workloads in the optimal location to achieve these objectives.

For example, objectives could include:

- Maximize the battery life of the robots.
- Minimize the time it takes for the robots to complete their tasks.
- Ensure that all the robots are evenly utilized.
- Avoid overloading any individual robot.

FLUIDOS would then use these objectives to make decisions about where to place the workloads. For example, if a robot is low on battery, FLUIDOS might move its workloads to other robots with more battery power. Or, if a robot is overloaded, FLUIDOS might move some of its workloads to other robots that are less busy.

By using FLUIDOS to orchestrate the workloads in the robot fleet improves performance, efficiency, and reliability of the entire system.

5.3.4.2 Workloads

In this use case different types of workloads with different requirements are considered:

Funded by Horizon Europe Framework Programme of the European Union

5.3.4.2.1 Non-offloadable Workloads

In the robotic field it is common to have workloads that need to be executed on a specific machine. Usually, they are related to data acquisition or commanding actuators.

Examples are:

- DC motor interface
- Actuators interface
- Fleet manager/telemetry agents
- Battery managers
- Video sensor interfaces
- LIDAR sensor interface
- Other USB devices interface
- Other Local Network devices interface

5.3.4.2.2 Offloadable Workloads

In the general case, a defined workload can be executed in location, under certain constrictions. This can be a pure robotic task or general-purpose task.

Example of robotic tasks are:

- Localization workloads
- Navigation workloads
- Mapping workloads
- Traversability workloads

5.3.4.2.3 Realtime workloads

Especially on robotic tasks there are some workloads where the data must be processed below a certain time since it's produced.

Example of this task:

- Localization workloads
- Navigation workloads
- Traversability workloads
- Obstacle avoidance workloads

5.3.4.2.4 Non-real-time Workloads

In the general case the workload does not need that data is processed below a certain amount of time. There are several example robotic workloads in this category:

- Mapping workloads.
- Fleet Management workloads.
- Inventory workloads

Funded by Horizon Europe Framework Programme of the European Union

5.3.4.2.5 Non-Persistent Data Workloads

In the general case the robotic workloads do not require a permanent storage, it only needs some configuration parameters to work.

5.3.4.2.6 Persistent Data Workloads

Some robotic workloads do require persistent storage like:

- Mapping workloads
- Localization workloads

5.3.4.3 Fleet Manager

All the robots communicate with the fleet manager in a bidirectional way. Each robot conveys its status to the fleet manager through an agent, and in turn, the fleet manager assigns specific missions to each robot.

5.3.4.4 Use Case Relevant Intent Metrics/Attributes

The robotic logistic use case will need the FLUIDOS orchestrator to place workloads regarding some hardware, network, and robotic specific parameters. This list is preliminary, and more parameters could be added in the future, and some could be removed.

5.3.4.4.1 Hardware

- Location (for non-offloadable workloads)
- CPU
- RAM
- Storage
- Special hardware (GPU)

5.3.4.4.2 Network

- Latency among devices
- Throughput
- Physical medium (wired, wireless, mobile, etc)

5.3.4.4.3 Energy

- Battery presence
- Battery level
- Current consumption
- Charging status
- Remaining working time
- Remaining charging time
- Grid energy source (Nuclear/Carbon/Solar/Wind/other-renewable)

Funded by Horizon Europe Framework Programme of the European Union

5.3.4.4.4 Robotic specific

- Robot status (Idle/Ready/In Mission/Charging)

Following is an example of some high-level rules that can be applied to robot status:

Table 6: Robotic High-level Rules

Robot state	Application Specification	Own Workloads Specifications	Foreign Workloads Specifications
idle	stopped and not need to go to the charger	not running anywhere	can accept foreign workloads
ready	stopped and not need to go to the charger	all running (wherever)	cannot accept foreign workloads
in mission	transporting goods to somewhere or going to charge	all running (wherever)	cannot accept foreign workloads
charging	charging	no matters (minimum load)	can accept foreign workloads

Other examples could be:

- "When the robot is idle remove all possible workloads in order to save as much battery as possible".
- "When a mission is assigned, place all the off-loadable workloads outside the robot and the workloads that time response critical, place it where the latency is below XX ms and the throughput is above YY Mbps."
- "When the robot is less than ZZ % of battery, modify the parameters of the workload in order to reduce battery usage."

Funded by Horizon Europe Framework Programme of the European Union

5.3.5 Implementation and Integration

5.3.5.1 General approach

To achieve the final use case for the success of FLUIDOS, adapting the robotic code and operative system of the robots is necessary.

For the use case, Robotnik RB-THERON autonomous robot (AMR) is utilized, specifically designed for robotic logistics with pick and place capabilities. This robot uses the Robot Operative System (ROS) as robotic software middleware, and it can be simulated using Gazebo robotic simulation Environment.

The first step involves migrating the Robot stack from Robot Operative System (ROS) version 1 to version 2. The version 1 is in the last stages of support and it has intrinsic limitations that are overcome in version 2. This migration should be done on the real robot and simulated environment.

Once the stack is migrated the next step is to split the monolithic way of launching the robotic software in a fragmented schema. This way, each component can be launched in different instances.

To allow the use of this robotic software, it needs to be containerized allowing the run of robotic instances in other machines that might not have all required underlying software installed.

But the containerized software is not enough, it is mandatory to have some kind of manifest to specify how the containers run and what are the relationships among them. For doing that, since FLUIDOS is Kubernetes based and the robots have enough computation power to run as Kubernetes, the way to run the fragmented ROS2 robotic containers is via Kubernetes Manifest files.

Once the vanilla robot software runs on the Kubernetes cluster, it is required to inform FLUIDOS orchestrators about robotic metrics, such as robot, battery, and network status. This is done so they can interact with the intent based FLUIDOS orchestration of the robotic workloads. In that regard in collaboration with WP4 the Kubernetes manifests need to add the intent constraints of each workload (if can be offloaded or not, network requirements, required on X robot status, etc…)

Funded by Horizon Europe Framework Programme of the European Union

On the other hand, the robot operative systems need to be updated in order to run Kubernetes and Liqo. And it's required to give a systematic way to install these base requirements in several robots or edge devices.

With all these points achieved the use case is ready for demonstration. But it recommended to try, tweak, and fix all the components in a controlled environment. The last step will be to test the use cases with real robots.

5.3.5.2 Progress

At the moment the migration of the ROS2 for the Robotnik RB-THERON is already done. The monolithic code is also split in several instances. The robot software is containerized in docker images. ROS2 RB-THERON simulations for (Humble distro) are publicly available in docker hub.

Also, Kubernetes manifest for running the fragmented simulation is available in the FLUIDOS github, with web visualization.

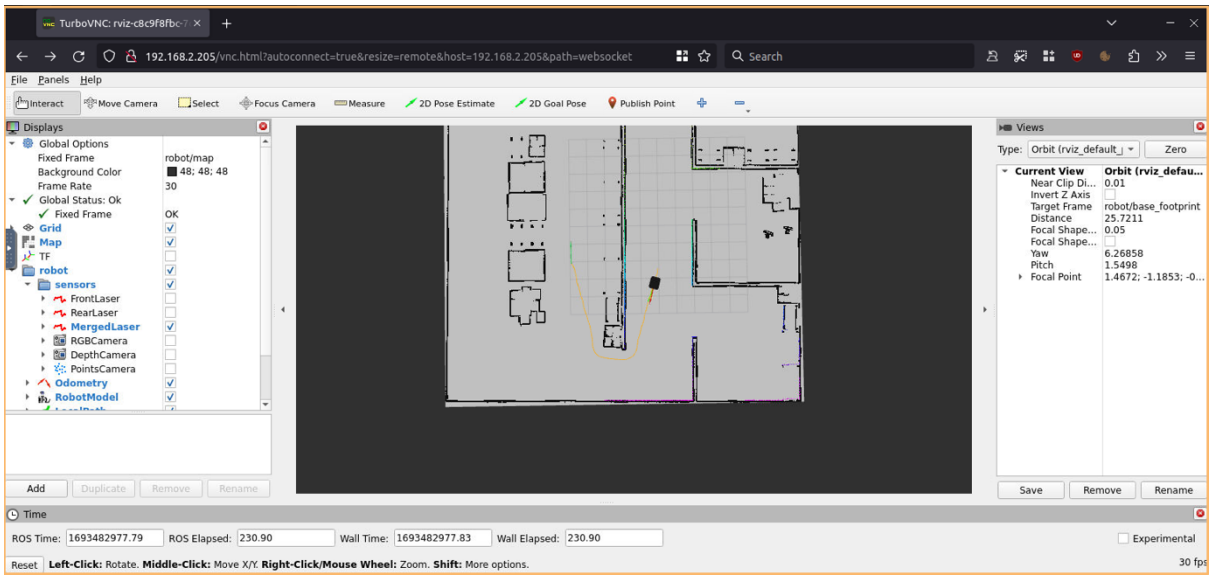

Figure 23: RViz Simulation

Funded by Horizon Europe Framework Programme of the European Union

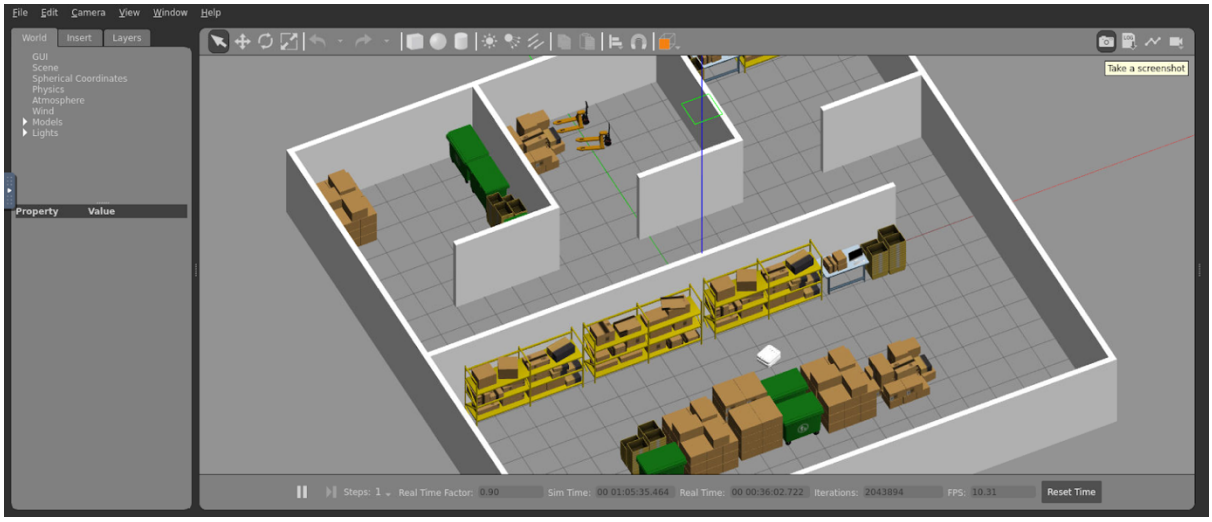

Figure 24: Gazebo Rendering of the Simulation within a Warehouse

For deploying Kubernetes and Liqo on Ubuntu 22.04 machine (like the one onboarded on the robots) in unattended machine available in the FLUIDOS Github repository.

Currently, the challenge lies in solving the issue with UDP multicast used in ROS2 DDS underlying communications and the interaction with Kubernetes CNI, how to bypass these problems and allow inter-cluster communication.

Funded by Horizon Europe Framework Programme of the European Union

6 CONCLUSIONS

The FLUIDOS project aims to redefine the cloud-edge-IoT compute continuum paradigm by offering a seamless and decentralized operating system that integrates the cloud with the edge and IoT. This enhances efficiency, scalability, and security across various sectors, as demonstrated by the three use cases: intelligent power grid, smart viticulture, and robotic logistics. These use cases illustrate how FLUIDOS can address specific challenges and opportunities in each domain, such as optimizing the management of distributed energy resources, enhancing crop management and disease detection, and improving task orchestration and resource allocation among autonomous robots.

This document provides the initial prototypical steps of delivering use case realisations and the challenges in doing so. From the experience to date, the major challenges have not been of adjusting the FLUIDOS technology and platform, but rather the specifics of each use case, for example state synchronisation in Use Case 1 or tackling issues of UDP broadcasts in Use Case 3. All these challenges are being worked on and solutions to this and others will be found and reported upon.

Fortunately, the use cases in WP7 have been designed in a sensible manner and are already largely containerized. As a result, transitioning to FLUIDOS technologies is not a significant obstacle, but rather an initial engineering endeavour. This is a considerable advantage of FLUIDOS. This allows adopters of the technology focus on their business challenges and not that of the platform and leverage the new innovative features of FLUIDOS.

Funded by Horizon Europe Framework Programme of the European Union