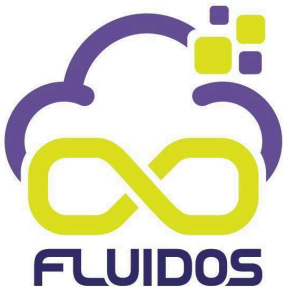
Grant Agreement No.: 101070473
Call: HORIZON-CL4-2021-DATA-01
Topic: HORIZON-CL4-2021-DATA-01-05
Type of action: HORIZON-RIA



D9.2 Release 1 and evaluation report

First release of the integrated meta-OS

Revision: v1.0

Work package	WP 9
Task	Task 9.1, 9.2, 9.3
Due date	29/02/2024
Submission date	29/02/2024
Deliverable lead	ROB
Version	1.0
Authors	Guillem Gari (ROB)
Reviewers	TOPIX

Abstract	First release of the integrated meta-OS
Keywords	Agile Integration, testing, continuous integration, continuous delivery,KPI, evaluation, workflow, performance, playground

DOCUMENT REVISION HISTORY

Version	Date	Description of change	List of contributor(s)
1.0	29/02/2024	First Version	Guillem Gari, Eduardo Canovas

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)*

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

This deliverable describes and explains the release 1 of FLUIDOS and the associated evaluation report. The first section details information about this document, such as structure and scope. The second section provides an overview of the project to help readers gain a better context and understand the scope of the document. The third section describes and explains each component of the comprehended on release 1 and points to repository codes where detailed instructions can be found. In the fourth section, we evaluate each component of Release 1 with the objectives of the project for this stage.

Funded by Horizon Europe Framework Programme of the European Union

Table of Contents

Funded by Horizon Europe Framework Programme of the European Union

LIST OF FIGURES

Funded by Horizon Europe Framework Programme of the European Union

ABBREVIATIONS

CDaaS	Cloud Native Cyber Deception as a Service
CLI	Command Line Interface
CNN	Convolutional Neural Network
CR	Custom Resource
CPU	Central Processing Unit
DLT	Distributed Ledger Technology
DID	Decentralized Identifier
IoT	Internet of Things
KPIs	Key Performance Indicators
KWh	Kilowatt-hour, a unit of energy
LED	Leaf Edge Device
MEC	Multi-access Edge Computing
MQTT	Message Queuing Telemetry Transport
MSPL	Medium-level Security Policy Language
SSI	Self-Sovereign Identity
VDR	Verifiable Data Record
vCred	Verifiable Credential

Funded by Horizon Europe Framework Programme of the European Union

1 Introduction

The current IT landscape is characterised by hyperconnectivity, with devices and information system

1.1 Document Structure

To incorporate the above information and guidelines the report is organised in the following subsections/chapters:

- Chapter 1 provides an overview and structure of the document.
- Chapter 2 provides a brief description of the FLUIDOS project.
- Chapter 3 describes, explains and points to the setup and use instructions of each FLUIDOS component present of Release 1.
- Chapter 4 discusses the evaluation of this release related to the project objectives.

Funded by Horizon Europe Framework Programme of the European Union

2 FLUIDOS DESCRIPTION

The current IT landscape is characterised by hyperconnectivity, with devices and information systems engaging in extensive communication and data exchange across multiple applications. FLUIDOS (Flexible, scalable, secure, and decentralised Operating System) seeks to harness the vast, underutilised processing capacity at the edge, which is distributed among diverse edge devices and cloud services that struggle to seamlessly integrate and form a cohesive computing continuum.

FLUIDOS aims to achieve the following objectives:

- Establish a seamless, decentralised continuum by integrating edge and cloud resources through automatic, autonomous resource discovery and integration, eliminating borders and enhancing fluidity.
- Shift the focus of computing and service provision beyond traditional data centres by creating a cross-provider, community-based fabric using open-source software, thereby redistributing computational gravity.
- Orchestrate services and hyper-distributed applications across multiple devices and domains in a continuous, automated manner, utilising energy-efficient AI learning algorithms for mobility and behaviour prediction as well as traffic forecasting.
- Implement a Zero Trust security paradigm to authenticate and authorise access to geographically dispersed resources, ensuring a high level of security.
- Foster the development of a multi-stakeholder market for edge services and applications, independent of cloud providers, to promote European digital autonomy.

More information can be found on the D2.1 *D2.1 Scenarios, Requirements and Reference Architecture – v.1"*

2.1 RELEASE 1 SCOPE

Since this is an intermediate FLUIDOS release all components presented here are not yet integrated into a single overall FLUIDOS package, this will come on the next releases. The release 1 comprehends all ready to install FLUIDOS components at the time for the Release date.

Funded by Horizon Europe Framework Programme of the European Union

3 Components Description

3.1 FLUIDOS Node

A FLUIDOS node is orchestrated by a single Kubernetes control plane, and it can be composed of either a single device or a set of devices (e.g., a datacenter). Device homogeneity is desired in order to simplify the management (physical servers can be considered all equals, since they feature a similar amount of hardware resources), but it is not requested within a FLUIDOS node. In other words, a FLUIDOS node corresponds to a Kubernetes cluster, but with extra features.
Now we are going to describe components of the FLUIDOS node:

3.1.1 Local ResourceManager

The Local Resource Manager is constructed through the development of a Kubernetes controller. This controller serves the purpose of monitoring the internal resources of individual nodes within a FLUIDOS Node, representing a cluster. Subsequently, it generates a Flavour Custom Resource (CR) for each node and stores these CRs within the cluster for further management and utilisation.

3.1.2 Available Resources

Available Resources component is a critical part of the FLUIDOS system responsible for managing and storing Flavours. It consists of two primary data structures:

1. Free Flavours: This section is dedicated to storing Flavours, as defined in the REAR component.
2. Reserved Flavours: In this section, objects or documents representing various resource allocations are stored, with each represented as a Flavour.

The primary function of Available Resources is to serve as a centralised repository for Flavours. When queried, it provides a list of these resources. Under the hood, Available Resources seamlessly integrates with Kubernetes' etc., ensuring efficient storage and retrieval of data.

This component plays a crucial role in facilitating resource management within the FLUIDOS ecosystem, enabling efficient allocation and utilisation of computing resources.

3.1.3 Discovery Manager

The Discovery Manager component within the FLUIDOS system serves as a critical part of the resource discovery process. It operates as a Kubernetes controller, continuously monitoring Discovery Custom Resources (CRs) generated by the Solver Controller.

The primary objectives of the Discovery Manager are as follows:

- Populating Peering Candidates Table (Client): The Discovery Manager's primary responsibility is to populate the Peering Candidates table. It achieves this by

Funded by Horizon Europe Framework Programme of the European Union

identifying suitable resources known as "Flavours" based on the initial messages exchanged as part of the REAR protocol.
- Discovery (Client): it initiates a LIST_FLAVOURS message, broadcasting it to all known lists of FLUIDOS Nodes.
- Offering Appropriate Flavours (Provider): In response to incoming requests, it will provide Flavours that best match the specific request.

3.1.4 Peering Candidates

The Peering Candidates component manages a dynamic list of nodes that are potentially suitable for establishing peering connections. This list is continuously updated by the Discovery Manager.

Under the hood, Peering Candidates are stored through an appropriate Custom Resource.

3.1.5 REAR Manager

The REAR Manager plays a pivotal role in orchestrating the service provisioning process. It receives a solving request, translates them into resource or service requests, and looks up external suitable resources:

- If no Peering Candidates are found, it initiates the Discovery.
- Optionally, if a suitable candidate is found, it triggers the Reservation phase.
- If this process is successfully fulfilled, resources are allocated, contracts are stored, and optionally can start the Peering phase.

3.1.6 Contract Manager

The Contract Manager is in charge of managing the reserve and purchase of resources. It handles the negotiation and management of resource contracts between nodes:

- When a suitable peering candidate is identified and a Reservation is forged, the Contract Manager initiates the Reserve phase by sending a RESERVE_FLAVOUR message.
- Upon successful reservation of resources, it proceeds to the Purchase phase by sending a PURCHASE_FLAVOUR message. Following this, it stores the contract received.

3.1.7 Github Repository and setup instructions

FLUIDOS node is hosted on FLUIDOS github public repository code where all details of how install are details:
https://github.com/fluidos-project/node

3.2 FLUIDOS Edge

FLUIDOS edge it's a minimal architecture for running the FLUIDOS components at the edge of the network on some low resources boards like STM boards, leveraging KubeEdge.

Funded by Horizon Europe Framework Programme of the European Union

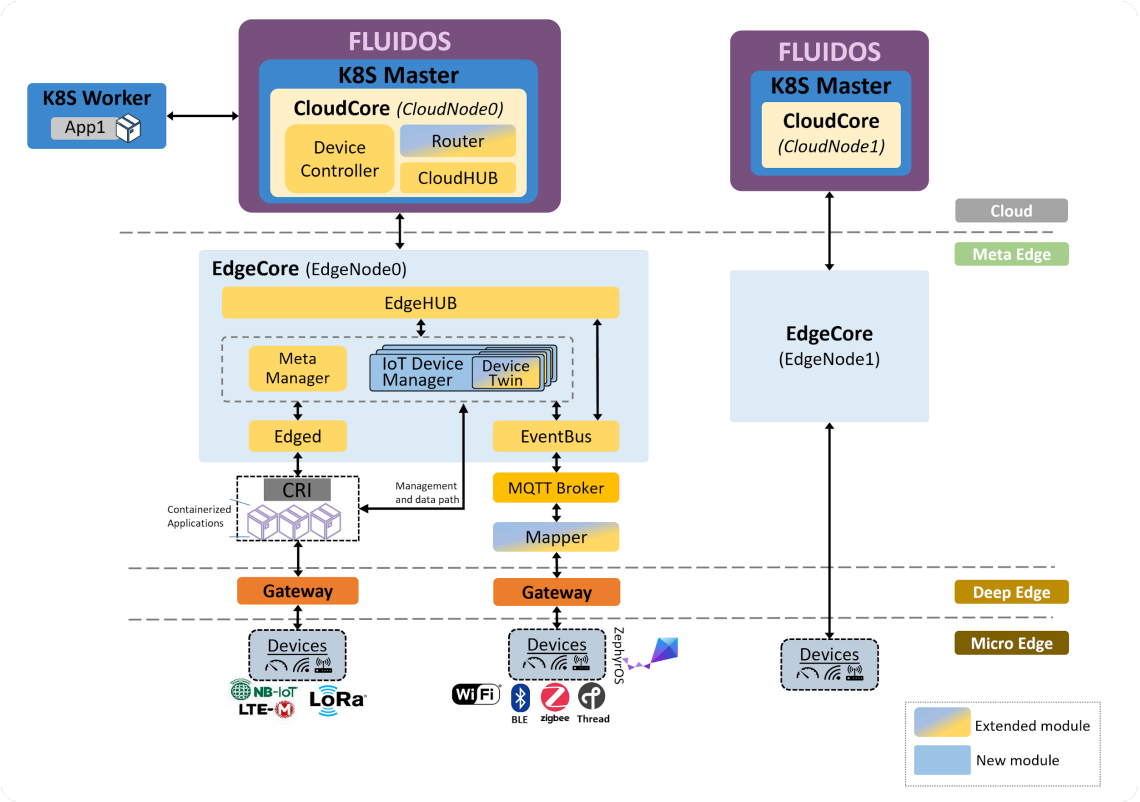


Figure 1: Fluidos Kube-edge architecture

FLUIDOS Edge is composed of several components in order to work properly including Cloud Core, Edge Core, MQTT Broker which are prerequisites to enable accessing and managing Edge nodes (Meta Edge) and devices (Deep Edge and Micro Edge).

The validation of the setup could performed by the following:

- Deploying pod at the Meta Edge
- Reading sensors values from a Micro Edge device over Bluetooth (via kubectl kubernetes CLI tool from cloud layer)
- Using Mapper (via kubectl kubernetes CLI tool from cloud layer)

FLUIDOS Edge is possible to use in a more advanced scenario where many cloud applications will share a Leaf Edge Device (LED) , which is part of Micro Edge, and selectively read data from an LED's sensors.

The layers of FLUIDOS Edge will be described below:

3.2.1 Cloud Core

This is the upper layer that moves toward distant resources, with the components called Fog Computing (usually at the customer's premises) and Multi-access Edge Computing (MEC), till to the Cloud, which brings large data centres into the picture.

3.2.2 Meta Edge

Funded by Horizon Europe Framework Programme of the European Union

Meta Edge is used to describe edge computing architectures that involve a layer of computing resources located between the edge and the cloud. The meta edge may be responsible for aggregating and processing data from multiple deep edge devices, and it may be located closer to the edge than the cloud, but further from the data source than micro edge or deep edge devices.

3.2.3 Deep Edge

Deep Edge refers to the intermediate layer of the hierarchy, which is located between the micro edge and the meta edge. This layer is responsible for more advanced processing and decision-making, and it may include devices such as gateways and edge servers that are capable of handling more complex tasks than the devices at the micro edge. It includes IoT devices and processing, network computing units and intelligent controllers.

3.2.4 Micro Edge

Micro Edge refers to the lowest level of the hierarchy, which includes the devices and sensors that are located at the edge of the network. These devices typically have limited processing and storage capabilities, and they rely on the other layers of the hierarchy for communication and decision-making. These services include sensing/actuating, connectivity, intelligent processing.

3.2.5 Github Repository and setup instructions

FLUIDOS Edge is hosted on FLUIDOS github public repository code where all details of how install are details:
https://github.com/fluidos-project/fluidos-edge

3.3 FLUIDOS MetaOrchestration

FLUIDOS meta-orchestration provides functionality to perform intent-based meta orchestration of workloads within FLUIDOS continuum. The component relies on the functionality provided by the FLUIDOS node to perform resource discovery and acquisition. The project itself is extensible, allowing the definition of specific models, or rule/heuristics, for the orchestration of the deployed workloads.

Funded by Horizon Europe Framework Programme of the European Union

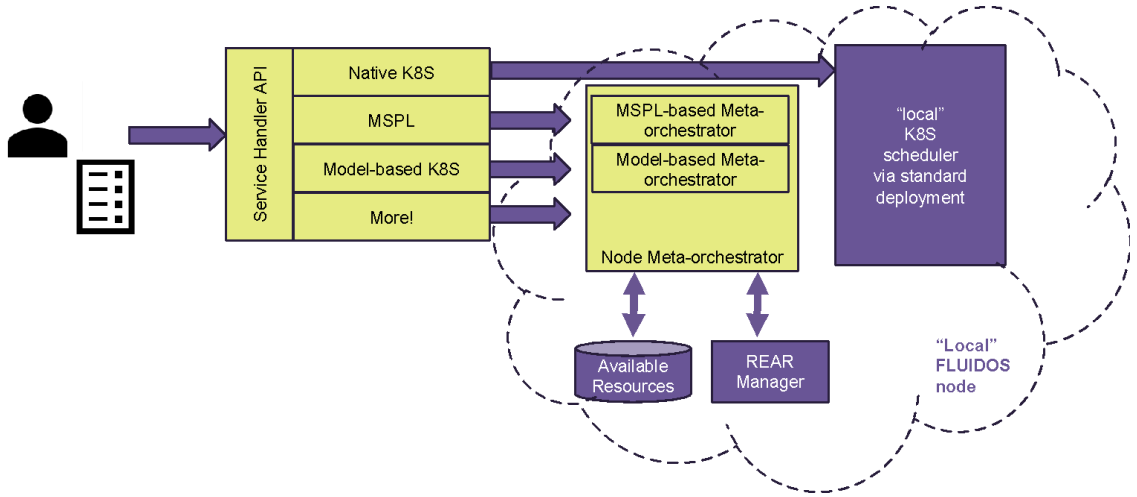


Figure 2: Meta-orchestrator conceptual architecture

On the current release the FLUIDOS MetaOrchestrator focuses on the Model Base component.

FLUIDOS model-based meta-orchestrator interacts with the local Kubernetes cluster through the operator paradigm, as presented in Figure XX. Namely, a Kubernetes Operator is a software component that reacts to events generated within Kubernetes as Resources are created, deleted, and updated.

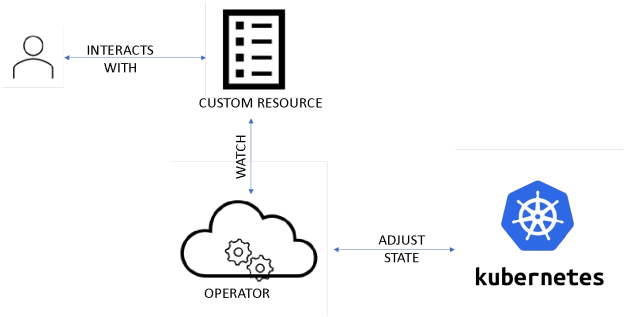


Figure 3: High level Kubernetes Operator Paradigm

From an operational point of view, this component translates the workload information, including container image information, and explicit and implicit intents to a feature vector that is in turn used to perform a prediction with the provided model. The first version of the model can process explicit intents, such as CPU, Memory, Latency, Throughput and Location.

3.3.1 Github Repository and setup instructions

FLUIDOS MetaOrchestor is hosted on FLUIDOS github public repository code where all details of how install are details:
https://github.com/fluidos-project/fluidos-modelbased-metaorchestrator

Funded by Horizon Europe Framework Programme of the European Union

3.4 FLUIDOS Kubectl Plugin

kubectl FLUIDOS plugin provides an extension (plugin) to kubectl to seamlessly interact with FLUIDOS components, namely meta-orchestrator(s). The project is developed using Python, and it acts as a bridge between traditional kubernetes requests and the one processed by the model-based meta orchestrator. Note that the plugin also allows interaction with the MSPL-based meta-orchestrator, thus providing a developer a single tool for transparently interacting with the FLUIDOS components.

3.4.1 Github Repository and setup instructions

FLUIDOS kubectl plugin is hosted on FLUIDOS github public repository code where all details of how install are details:
https://github.com/fluidos-project/fluidos-modelbased-metaorchestrator

3.5 FLUIDOS Cyber Deception

The Cyber Deception feature, part of the FLUIDOS Cyber Security services designed to improve the overall security of the ecosystem, is willing to provide Cloud Native Cyber Deception as a Service (CDaaS) integrated into the FLUIDOS continuum.

A description of a first use-case follow:

"By leveraging FLUIDOS, the owner of a local domain will see a continuum across local and remote resources. Thanks to CDaaS he will also get additional advantages from a given remote domain which offers those security capabilities, so as to take advantage of it and protect the workload running on the FLUIDOS continuum.

At some point a cloud-native application distributed across the two domains is transparently protected by decoys running in the remote cluster, which are created out of the original application components in order to intercept a possible malicious attack."

The research and development activities currently targets the following KPIs:

- Improved integration of Cyber Deception with FLUIDOS and delivery as a service
- Additional monitoring functionalities and extensions to attack tracing and threat intelligence capabilities

The Cyber Deception service currently relies on the features offered by the Decepto tool (https://gitlab.fbk.eu/cyber-deception/decepto), which is a system that creates decoys as clones of existing services.

Given an application graph (sets of micro-services and data-flows across them) Decepto decides the services to clone as decoys and where to deploy them based on optimization metrics such as the availability of resources.

As shown in the below picture it runs in a Kubernetes cluster and could use multiple external algorithms to take decisions and perform actions.

Funded by Horizon Europe Framework Programme of the European Union

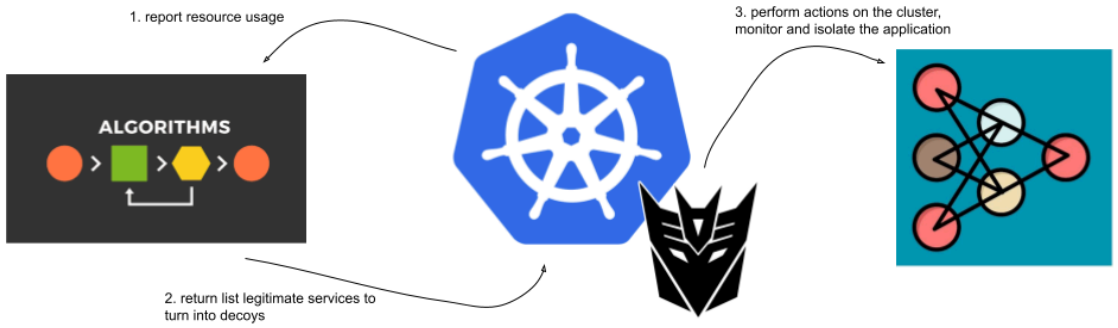


Figure 4: Cyber Deception architecture

Decepto is being defined with a fully open approach, which facilitates the participation of actors that are outside the FLUIDOS community.

3.5.1 Github Repository and setup instructions

FLUIDOS Cyber Deception is hosted on FLUIDOS github public repository where all details and instructions on how to integrate Decepto in the FLUIDOS ecosystem will be available: https://github.com/fluidos-project/cyber-deception

3.6 FLUIDOS Identity Management

To create the FLUIDOS decentralised, immutable, secure, interoperable and traceable overlay we leverage the Hyperledger Aries Agent seamlessly integrated with the Hyperledger Fabric. On the one hand, Aries is a set of tools and libraries designed to provide a shared, reusable, interoperable infrastructure for decentralised identity and blockchain-interoperable projects. It focuses on enabling the exchange of verifiable credentials and the establishment of decentralised identity systems. On the other hand, Fabric is a permissioned blockchain platform for developing enterprise-grade applications. It provides a modular architecture with a pluggable consensus mechanism, allowing FLUIDOS to tailor the network to their specific requirements. To further elevate the desirable capabilities, the p-ABC implementation https://github.com/fluidos-project/p-abc was incorporated to the solution. It provides cutting-edge cryptography that enables the creation of derived verifiable presentations and zero-knowledge proofs. The synergy of these components has converged to form a powerful interface, empowering the solution with the indispensable capabilities needed for seamless identity management within the FLUIDOS scenarios.

To seamlessly integrate Hyperledgers, enabling communication between Hyperledger Aries and Fabric, using Fabric as a Verifiable Data Record (VDR), and providing functionalities such as DID generation, enrolment, authentication and authorisation the following modules has been implemented:

Funded by Horizon Europe Framework Programme of the European Union

- DID Management smart contracts: To enable the use of Hyperledger Fabric as the VDR for the identity data, we implemented smart contracts for registering and resolving DIDs.
- VDR Agent: This component is a specialised VDR client that leverages the network of Hyperledger Fabric nodes. In practical terms, nodes use this module to expose their DIDs publicly and resolve DIDs within the ledger to retrieve the corresponding documents through calls to the appropriate smart contracts.
- SSI Management: This component defines methods leveraging the Aries agent's capabilities to address identity management requirements in FLUIDOS. Within this component we can find the following implemented processes:
    - Generate DID: Create DID with keys/purposes as specified in request, and register it in the DLT. Any entity in the scenario can thus self-manage its identification data, and particularly its keys.
    - DoEnrolment: Do an enrolment process against an issuer, obtaining a new vCred. With this method, a node will be able to obtain a verifiable credential that asserts its identity attributes for its later use during authentication and authorisation processes.
    - Generate Verifiable Presentation: Generate a Verifiable Presentation of a vCred for an authorisation process.
    - VCredential/VPresentation verification: Verify a Verifiable Credential or a Verifiable Presentation, returning a boolean of the verification result to complete an authorisation process.

3.6.1 Github Repository and setup instructions

FLUIDOS Identity manager is hosted on FLUIDOS github public repository code where all details of how install are details:
https://github.com/fluidos-project/idm-fluidos-aries-framework-go

3.7 FLUIDOS Energy Predictors

FLUIDOS energy predictor is a component that foresees the energy demand for a FLUIDOS node. It uses a neural network that takes as input, given a certain machine. It computes this predictions using the following data:

- Past workload of the machine
- Power profile of the machine

The outputs are the predicted energy demand for the following day.

FLUIDOS Energy predictor provides additional components which are concerned with automating the parsing of datasets and the drawing of result graphs but are not fundamental to the inner workings of the application.
First, the data manipulation component is tasked with properly preparing the incoming workload data and power curves, transforming them into a form suitable for consumption from the learning and prediction components. In particular, the workload data is aggregated by machine ID, reordered, and a moving average is applied to reduce the amount of data points to one every 15 minutes. This is done both for CPU and memory data; the rest of the information is scrapped.

Funded by Horizon Europe Framework Programme of the European Union

Then, sequences are prepared by further aggregating the data points over eight-day periods, of which the first seven are kept; using the power curves, the last day is instead transformed into a single number. This figure represents the amount of energy (in KWh) that the node consumed over that single day, and it will be used as a ground truth for the model. The seven days of data plus the energy consumption of the eighth day are then packed and saved together, to be used as a sequence by the other components. Further sequences are generated in the same manner, but by shifting the time window by a customizable amount, which defaults to an hour. With this approach, for example, nine days yield twenty-four different sequences to be used by the model.

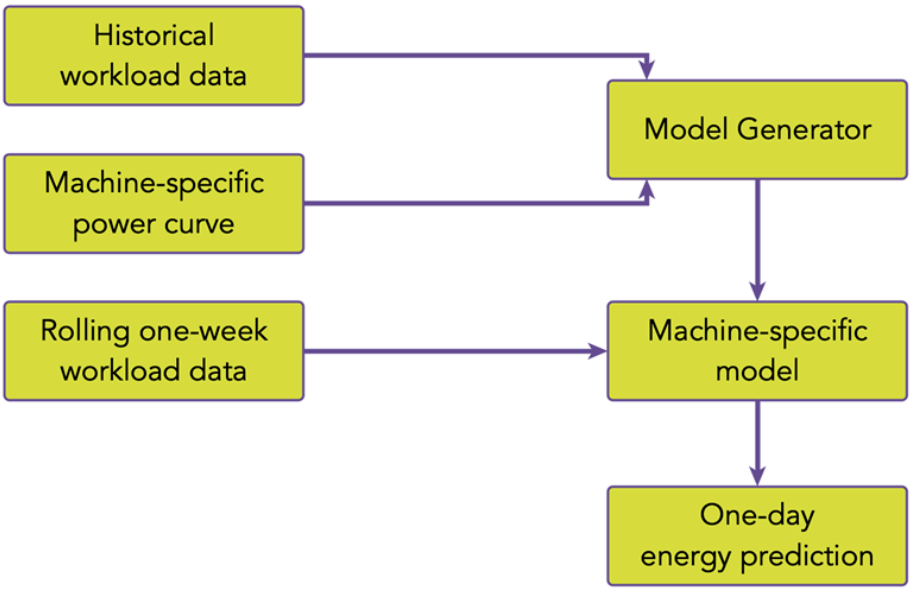


Figure 5: Load prediction model workflow

The learning and prediction components are the key part of the application. The former takes in input a certain amount of data sequences (along with some user-provided parameters) and is tasked with training a deep learning model. In its current form, the ML model is a fully connected Convolutional Neural Network (CNN), composed of three convolutional layers and a final dense layer. As an input, the first layer takes 672 data points over two dimensions, i.e., seven days' worth of data every 15 minutes, both for CPU and memory usage. As an output, a dense layer condenses the model's information into a single value.
With that in mind, the component outputs a single number, representing the energy (in KWh) the model estimates the machine will consume in the following day. It is then printed to the user, along with a collection of metrics that evaluate the performance of the model over that particular set of data. In particular, the users are presented with the R2 score (coefficient of determination), the mean squared error, and the mean absolute error.

3.7.1 Github Repository and setup instructions

FLUIDOS Energy Predictor is hosted on FLUIDOS github public repository code where all details of how install are details:
https://github.com/fluidos-project/fluidos-energy-predictor

Funded by Horizon Europe Framework Programme of the European Union

4 Evaluation Report

The FLUIDOS project has adopted a unified CI/CD pipeline platform, marking a significant advancement in its development methodology. This platform extends beyond basic automation, offering a comprehensive solution that fosters both holistic testing and seamless integration of the project's diverse components.

This shift towards a CI/CD platform significantly impacts the project's development efficiency, effectiveness, and overall success. The platform streamlines workflows, enhances collaboration among team members, and expedites the delivery of high-quality software. Additionally, a comprehensive evaluation of the platform's implementation aims to identify areas for improvement, optimize development processes, mitigate risks, and ensure standardized and efficient deployment across all work packages and components. Ultimately, this evaluation serves as a valuable tool for continuous improvement and innovation within the FLUIDOS project, contributing to its overall success and impact in the field of research and technology.

The CI/CD platform facilitates holistic testing through a multi-tiered approach. Individual components undergo rigorous testing to ensure their functionality. Subsequently, the platform orchestrates integrated system testing, verifying seamless collaboration within the larger system. This proactive approach enables the early identification and resolution of potential issues, leading to a more robust final product.

Seamless integration is achieved through the platform's centralized integration hub. This hub enables continuous integration of newly developed or modified components into the broader system, allowing for early detection and resolution of integration challenges. This proactive approach prevents such issues from cascading into later stages of development, while the consistent and automated nature of the process minimizes human error and ensures reproducible builds.

Furthermore, the CI/CD platform contributes to enhanced efficiency and risk mitigation by automating repetitive tasks like testing and deployment. This automation streamlines development processes, significantly increasing development velocity and reducing the likelihood of errors. Additionally, the platform facilitates early issue identification and resolution, effectively mitigating risks associated with late-stage integration problems.

The platform also plays a crucial role in continuous feedback and improvement. By collecting performance and functionality data, the platform establishes a feedback loop that empowers developers to iteratively refine and enhance both individual components and the overall system, ultimately leading to a more robust and efficient final product.

4.1 Fluidos Code Repository

The FLUIDOS project prioritises streamlining version control, fostering collaboration, and optimising workflows for efficient task and deliverable execution. Recognizing these goals, the selection of GitHub (https://github.com/fluidos-project), the world's largest open-source code platform, as the central repository proves strategic, offering a robust solution that aligns well with project requirements.

Funded by Horizon Europe Framework Programme of the European Union

GitHub's extensive adoption ensures seamless integration with various CI/CD tools and services. By leveraging GitHub Actions or other CI/CD pipelines directly within the platform, the project team can efficiently automate testing, build processes, and deployment workflows. This integration streamlines development, enhancing productivity and guaranteeing a smooth and reliable software delivery pipeline.

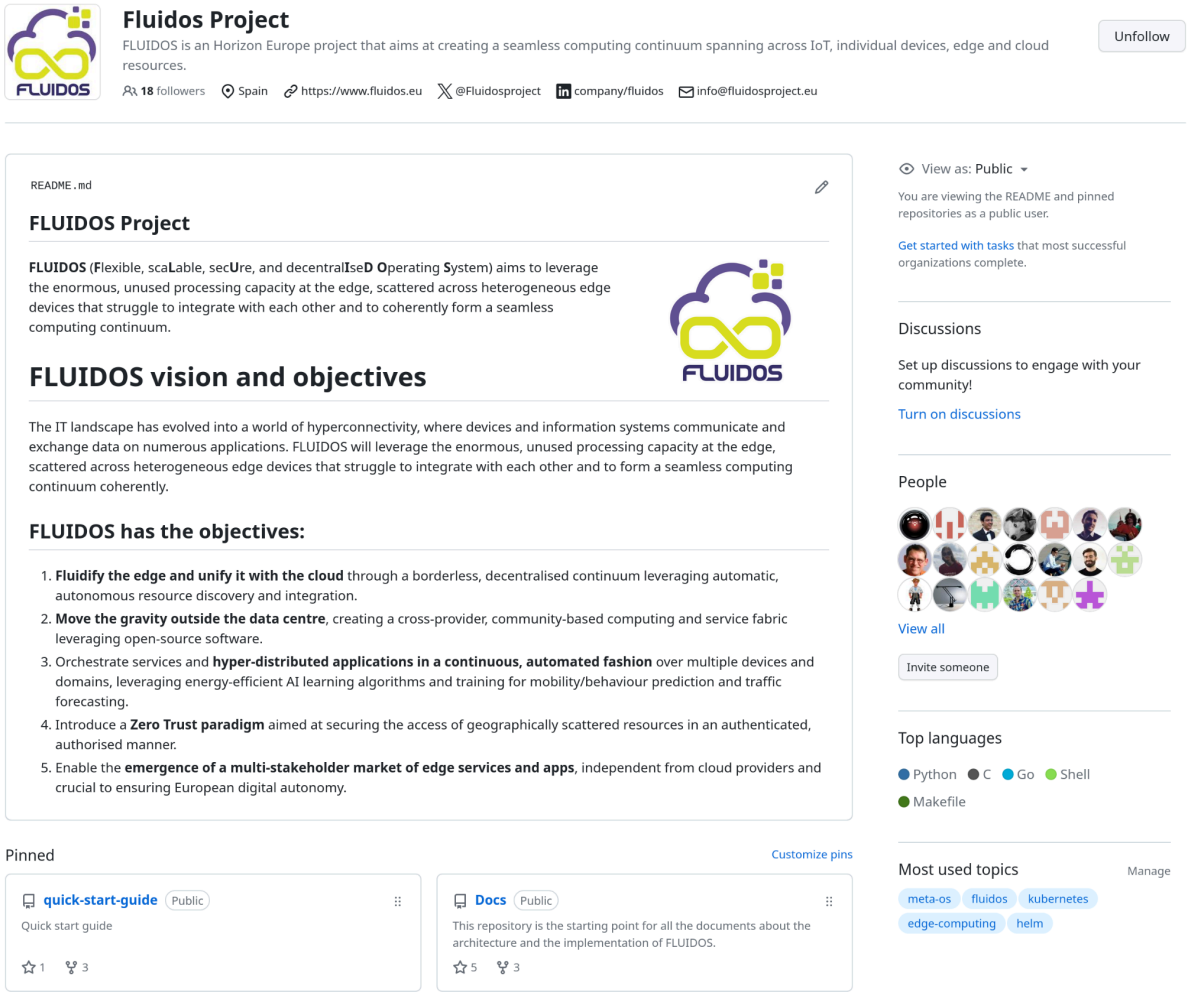


Figure 6: Github organization

Furthermore, GitHub's unique strengths contribute significantly to FLUIDOS' goals of reaching a broader technical audience within the European landscape:

- Amplified Reach and Visibility: GitHub's immense user base, boasting millions of developers and technical enthusiasts globally, provides a unique opportunity for FLUIDOS to amplify its reach and visibility within the European technical community. By utilizing GitHub, the project can effectively disseminate information, engage with potential collaborators, and attract valuable contributions from a diverse pool of experts across Europe.
- Open Collaboration and Knowledge Sharing: GitHub fosters a thriving culture of open collaboration and knowledge sharing, perfectly aligning with FLUIDOS' commitment to transparency and accessibility. By hosting the project on GitHub, the team promotes open access to code, fosters collaborative development practices,

Funded by Horizon Europe Framework Programme of the European Union

and encourages valuable feedback from the wider technical community. This not only benefits project development but also serves as a testament to its commitment to open science principles.

- Enhanced Collaboration: GitHub's robust set of collaboration features significantly benefits FLUIDOS. The platform facilitates effective communication and teamwork through functionalities like pull requests, code reviews, issue tracking, and project boards. This collaborative environment fosters transparency, accountability, and knowledge sharing within the team, ultimately leading to improved code quality and accelerated development cycles.

Additionally, GitHub's extensive ecosystem of third-party integrations and tools enhances the project's flexibility and scalability. The GitHub marketplace offers a wide range of solutions, from project management tools to continuous integration services, that can be seamlessly integrated into the development workflow. This flexibility empowers the project team to effectively tailor their development environment to their specific needs and preferences.

4.2 FLUIDOS CI/CD PIPELINES

The FLUIDOS project has strategically chosen GitHub Actions as its CI/CD platform due to its seamless alignment with the project's existing GitHub-based version control and collaboration workflows. This powerful automation tool enables continuous integration and deployment directly within the familiar GitHub environment, taking advantage of the platform's native repository integrations.

GitHub Actions simplifies setup and configuration by allowing developers to define workflows using YAML files directly in their repositories. This streamlined approach improves visibility, traceability, and reproducibility of CI/CD processes, leading to more efficient and reliable software delivery. Moreover, GitHub Actions offers a vast selection of pre-built actions and workflows that can be easily customized to fit specific project needs, from running tests and building artifacts to deploying applications.

The FLUIDOS project benefits from the scalability and cost-effectiveness of GitHub Actions as it allows teams to scale CI/CD pipelines based on demand, ensuring quick feedback loops, rapid iterations, and timely delivery of high-quality software. This approach contributes to a more efficient development process, enhancing collaboration and resource optimization.

In Work Package 9, "Politecnico di Torino" has taken an essential step towards strengthening the organization's CI/CD capabilities by providing self-hosted runners accessible through the Action Runner Controller from Kubernetes. These dedicated compute resources offer optimal performance and reliability for all FLUIDOS project partners.

This initiative uses Kubernetes as the underlying orchestration platform, further enhancing the efficiency and flexibility of self-hosted runners. Kubernetes' scalable and resilient environment seamlessly manages containerized workloads, enabling the deployment and operation of CI/CD pipelines across various repositories within the organization.

By leveraging self-hosted runners with the Action Runner Controller from Kubernetes, partners can execute CI/CD tasks efficiently and consistently, ensuring reliable delivery of high-quality software across the FLUIDOS project. This approach streamlines development

Funded by Horizon Europe Framework Programme of the European Union

processes, fosters collaboration, standardization, and resource optimization within Work Package 9.

In regard to the context provided, KPI-OBJ-SCHED is fulfilled through effective CI/CD pipelines that ensure efficient FLUIDOS release management and contribute to the successful execution of Objectives 1 (an extensible, modular FLUIDOS node with resource-agnostic abstraction capabilities) and 6 (fostering and consolidating a user/developer community). The pipelines promote seamless software development and management by streamlining workflows, enhancing collaboration, and optimizing resource utilization to guarantee high-quality software delivery across FLUIDOS project components. These efforts lead to the fulfilment of KPI KPI-OBJ-SCHED, ultimately paving the way for a successful Release 1 that is both efficient and flexible.

Funded by Horizon Europe Framework Programme of the European Union