Ref. Ares(2024)2330993 - 28/03/2024

GLACIATION

Green responsibLe privACy preservIng dAta operaTIONs

Deliverable 7.2 – Use Case Integration, Validation, and Demonstration Report - intermediate

GRANT AGREEMENT NUMBER: 101070141

This project has received funding from the European Union's HE research and innovation programme under grant agreement No 101070141

Project acronym:	**GLACIATION**
Project full title:	**Green responsibLe privACy preservIng dAta operaTIONs**
Call identifier:	**HORIZON-CL4-2021-DATA-01-01**
Type of action:	**RIA**
Start date:	**01/10/2022**
End date:	**30/09/2025**
Grant agreement no:	**101070141**

D7.2 – Use Case Integration, Validation, and Demonstration Report			
Executive Summary:	This document reports on the work done in Work Package 7 “Validation and Use cases”; it is aligned to T7.1 “Use cases planning and definition of the methodological approach for validation”		
WP:	WP7		
Author(s):	Pournima Sonawane, Jonas Böhler, Rafy Benjamin, Konstantin Tatarnikov, Daniela Piernera Palmieri, Alessandro Innocenti, Paolo Tartara		
Editor:	DELL		
Leading Partner:	EISI		
Participating Partners:	MEF, SAP, SOGEI, HIRO		
Version:	1.0	**Status:**	Submitted
Deliverable Type:	(R) Report	**Dissemination Level:**	(PU) Public
Official Submission Date:	31/03/2024	**Actual Submission Date:**	31/03/2024

Disclaimer

The GLACIATION consortium consists of the following partners:

No.	Partner Organisation Name	Partner Organisation Short Name	Country
1	MINISTERO DELL'ECONOMIA E DELLE FINANZE	MEF	IT
2	EMC INFORMATION SYSTEMS INTERNATIONAL UNLIMITED COMPANY	EISI	IE
3	HIRO MICRODATACENTERS B.V.	HIRO	NL
4	GOTTFRIED WILHELM LEIBNIZ UNIVERSITAET HANNOVER	LUH	DE
5	THE LISBON COUNCIL FOR ECONOMIC COMPETITIVENESS ASBL	LC	BE
6	UNIVERSITA DEGLI STUDI DI MILANO	UNIMI	IT
7	UNIVERSITA DEGLI STUDI DI BERGAMO	UNIBG	IT
8	GEIE ERCIM	ERCIM	FR
9	EURECOM	EURECOM	FR
10	SAP SE	SAP SE	DE
11	UNIVERSITY COLLEGE CORK - NATIONAL UNIVERSITY OF IRELAND, CORK	UCC	IE
12	SOGEI-SOCIETA GENERALE D'INFORMATICA SPA	SOGEI	IT
13	LAKESIDE LABS GMBH	LAKE	AT
14	ENGINEERING - INGEGNERIA INFORMATICA SPA	ENG	IT
15	EIDGENÖSSISCHE TECHNISCHE HOCHSCHULE ZÜRICH	ETH	CH

Document Revision History

Version	Description	Contributions
0.1	Table of Contents	SAP
0.2	Executive Summary, Introduction, WP7 Overview	DELL
0.3	Use Cases Objectives, Validation, Tests Workloads	DELL, SAP, SOGEI
0.4	Test Methodology, Test Overview	HIRO
0.5	Validation, Test descriptions for all Use Cases	DELL, SAP, SOGEI
0.6	Revised Test and workload details for all Use Cases	DELL, SAP, SOGEI
0.7	Conclusion, Future Work	DELL
0.8	Review Work	DELL, SAP, SOGEI, UNIBG, LAKE
0.9	Final check, improvements, and preparation for submission	MEF
1.0	Submitted	MEF

Authors

Author	Partner
Pournima Sonawane	EISI
Jonas Böhler	SAP
Konstantin Tatarnikov	HIRO
Rafy Benjamin	HIRO
Daniela Piernera Palmieri	Sogei
Alessandro Innocenti	Sogei
Paolo Tartara	Sogei

Reviewers

Author	Partner
Aidan O' Mahony	EISI
Stefano Paraboschi	UNIBG
Chepizhko Oleksandr	LAKE
Jonas Böhler	SAP
Daniela Piernera Palmieri	SOGEI
Pournima Sonawane	EISI
Alessio Chellini	MEF
Laura Coticoni	MEF
Leonardo Cherubini	MEF

Table of Contents

List of Figures

List of Tables

List of Terms and Abbreviations

Abbreviation	Description
GLACIATION	Green responsibLe privACy preservIng dAta operations
EU	European Union
HEU	Horizon Europe
WP	Work Package
Mxy	Month ‘xy’ of project’s duration
UC	Use Case
GDPR	General Data Protection Regulation
DKG	Distributed Knowledge Graph
KPI	Key Performance Indicator
PET	Privacy-Enhancing Technology
MPC	Multi-party Computation
DP	Differential Privacy
COBOTS	Collaborative Robots
HRC	Human Robot Collaboration
AGV	Automated Guided Vehicle
HRC	Human Robot Collaboration
OS	Operating System
PDU	Power Distribution Unit
RCA	Root Cause Analysis
HIPAA	Health Insurance Portability and Accountability Act
SVM	Support Vector Machine
ICMP	Internet Control Message Protocol
SCP	Secure Copy Protocol
RBAC	Role-Based Access Control
REST	Representational State Transfer
API	Application Programming Interface
ACL	Access Control List
OPA	Open Policy Agent
MFA	Multi-Factor Authentication
SSO	Single Sign-On
AES	Advanced Encryption Standard
IDS	Intrusion Detection System
KPI	Key Performance Indicator
RGBD	Red, Green Blue Depth
LIDAR	Light Detection and Ranging
AI/ML	Artificial Intelligence/Machine Learning
PA	Public Administration
MEF	Ministry of Economy and Finance

Executive Summary

The rapid growth of big data analytics, spanning from edge to cloud, has raised concerns about energy consumption. National grids, which produce significant carbon emissions, are particularly affected. To tackle this issue, the GLACIATION (Green Responsible Privacy-Preserving Data Operations) project focuses on energy-efficient and privacy-preserving data operations.

GLACIATION leverages expertise from public administration services, manufacturing processes, edge-core-cloud technologies, privacy-enhancing methods, AI/ML, and swarm intelligence. Its goal is to design and develop a novel edge-core-cloud architecture that minimizes energy consumption by reducing data movement operations, moving service delivery to a local (decentralized) level through edge technology and improving accuracy of analytical computations.

GLACIATION aims to showcase its achievements through implementation and deployment in three distinct use case scenarios such as: Public Administration, Manufacturing, and Enterprise Collaboration. In these contexts, it addresses energy consumption and carbon emissions associated with edge-core-cloud technologies, emphasizing energy-efficient data operations, reduction of central computation and incorporating privacy and trust considerations.

The Deliverable D7.2 comprises an intermediate report, that outlines the requirements of Use Cases, including integration, validation, and demonstration. It corresponds to the M12-M20 activity within Task 7.1 which focuses on the planning activities for the realization of the use cases as well as their validation. Also defining methodological approach to formalize the requirements and execution regarding the validation process. Throughout development, continuous monitoring will ensure alignment with subsequent tasks: Task 7.2 includes all the activities and efforts related to integration and adaption of GLACIATION platform, spanning from M16-M36 and Task 7.3 comprises all the activities related to the final integration, validation, evaluation, and demonstration of Use cases, running from M20-M36.

In the upcoming version of this deliverable, which will be the final report corresponding to Deliverable D7.3 for use case validation, integration, and demonstration, we anticipate presenting it by month 36, at project close.

1 Introduction

The vision of the GLACIATION project is to enhance the efficiency and adoption of reliable digital technologies to meet the requirements of individuals, private entities, and public organizations by addressing privacy, confidentiality, and environmentally friendly data operations—specifically related to energy, carbon, and material footprint.

This document presents Deliverable D7.2 “Use Case Integration, Validation, and Demonstration Report – Intermediate” with the planning activities for realization and validation of the Use cases:

- UC1: Edge-decentralized data management
- UC2: Data-driven energy-efficient manufacturing
- UC3: Privacy-preserving cross-company analytics

The purpose of Deliverable D7.2 is to provide an update on the progress made by WP7 during the first half of the project. It serves as an interim report which will be followed by the final report in Deliverable D7.3 at the project closure.

This deliverable describes in detail the validation strategy for each use case based on the validation criteria, requirements analysis from the implementation architecture and tests scenarios. In addition to the use case information, this deliverable includes validation objectives, test cases, and workload descriptions for use cases. This marks the next step towards the implementation process and future work.

The D7.2 document outlines the main technical outcomes of this deliverable and is divided into 7 Chapters. It starts with a brief introduction in Chapter 1, followed by an overview of WP7 in Chapter 2. The document summarizes all the chapters, followed by the Conclusion and Future Work in the last sections.

Chapter 3 delivers an overview of each use case and provides details about the validation environment for each one.

Chapter 4 outlines the validation objectives, which includes addressing the requirements for each use case, such as the validation aspects and criteria.

Chapter 5 focuses on the test structure, which defines the test methodology, platform testing, and describes the test workflow & workload for each use case.

Chapter 6 discusses future work that will be included in the final report by the end of project.

Chapter 7 concludes by discussing the subsequent work within the GLACIATION,

2 WP7 Overview

The WP7– Validation and Use Cases, is one of the eight work packages that frame the project GLACIATION work plan.

Work Package 7 outlines the necessary steps for the validation process, including the integration and adaptation of all modules, components, languages, APIs, and algorithms that were designed and implemented in previous work packages. This package comprises all the activities related to the final integration, validation, evaluation, and demonstration of the GLACIATION platform using use cases. The aim is to evaluate the platform's performance against use case requirements in a fully integrated environment.

The purpose of this WP is to define a method for evaluating the outcomes of the GLACIATION project, taking into consideration various performance indicators (KPIs) related to different scenarios (use cases), and the entire platform. The results of this evaluation will provide valuable input for improving the GLACIATION ecosystem and exploring its potential benefits.

WP7 is organized in 3 tasks running from M12 to M36, mentioned as below:

- Task 7.1: Use cases planning and definition of the methodological approach for validation (M12-M20).
- Task 7.2: System integration & testing of GLACIATION platform (M16-M36)
- Task 7.3: Integrate, validate, and demonstrate use cases in integrated environments (M20-M36)

Within WP7, there are 3 deliverables scheduled to be submitted between M18 and M36 – D7.1 (Complete platform), D7.2 (Intermediate report) and D7.3 (final report)

The objectives of WP7 include monitoring the project achievements against the state of the art, refining the application requirements, adapting the test bed infrastructure to fulfil the requirements for each use case.

In summary, WP7 aims to assess GLACIATION's effectiveness, learn from it, and enhance the overall system.

3 Use Case Overview

GLACIATION aims to examine energy efficiency and analytical improvements in three different environments: the Italian Ministry of Economy and Finance, Dell Manufacturing and SAP Enterprise Collaboration and these environments are supported by three use cases from different domains and end user needs, covering public administration, manufacturing, and enterprise collaboration.

The project's results have been successfully deployed in three different use cases, as illustrated in **Error! Reference source not found.Error! Reference source not found.** below. These use ca ses demonstrate the platform's robust capabilities in facilitating data-intensive, high-performance, and heterogeneous applications with varying data characteristics.

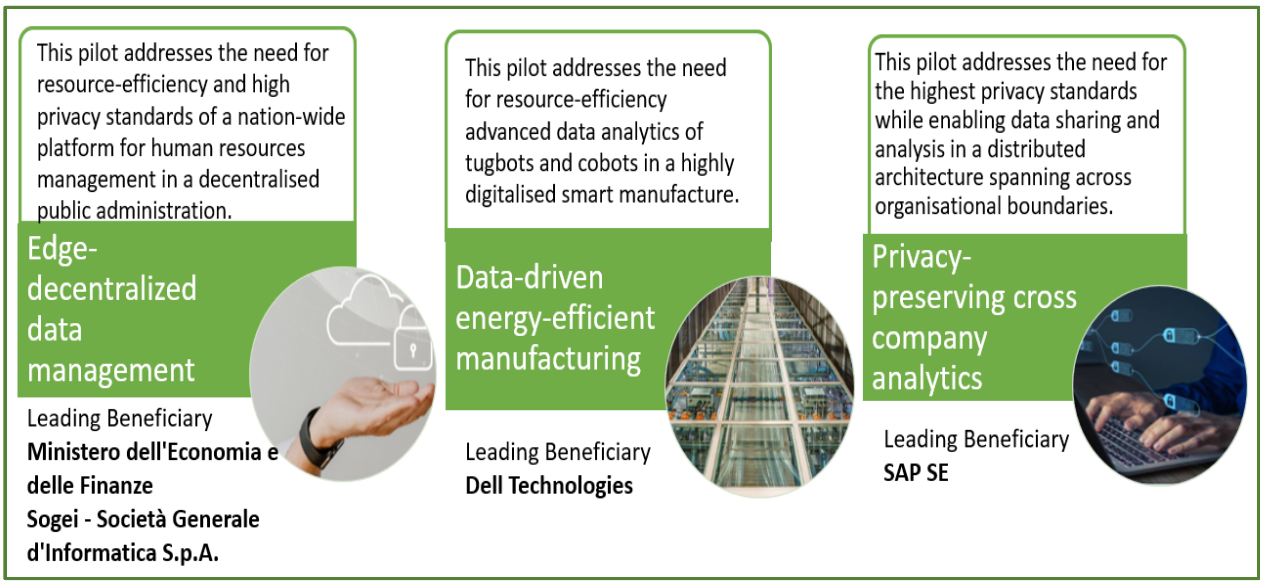


Figure 1: Pilot Use Cases

Each pilot works to deploy its own use case, implementing them with GLACIATION tools in various domains, thereby contributing value to the project. Since each use case has its specific scope and requirements, the coverage and evaluation of GLACIATION are thoroughly comprehensive. An overview of the pilot UCs can be found in the next section.

3.1 UC1: Edge-decentralized data management

The Department of General Administration, Personnel and Services (DAG) of the Ministry of Economy and Finance (MEF) provides IT services, information, and data management services to the Ministry, as well as services to over 80 Italian Public Administrations through the "NoiPA" digital platform. Within the NoiPA platform, there is the Time Management module that offers solutions to enable Public Administrations to manage the time and attendance management process.

Time Management includes self-service options, for employees, to view their attendance and absence tracking in real time, and to manage other related requests (e.g., requests for vacation, etc).

Main features:

- Time and Attendance Management - monitoring of staff attendance at work through the management of timestamp acquisition with turnstile system
- Overtime and allowance - management of monthly hours and overtime authorizations, allowances and food vouchers.

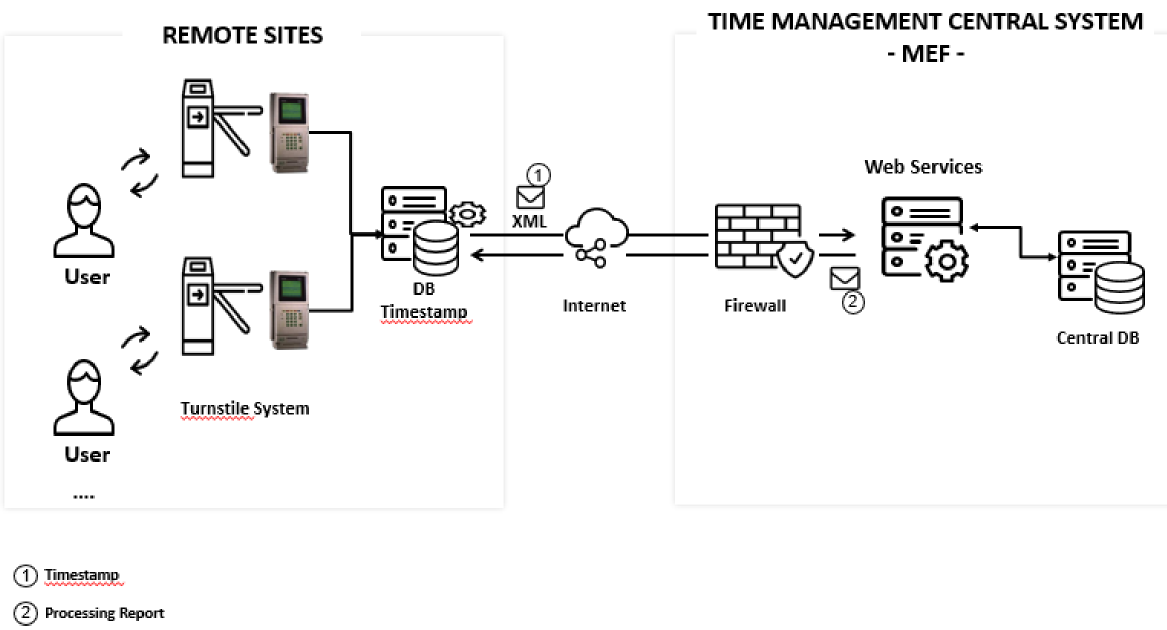


Figure 2: NoiPA system architecture

NoiPA collects time and attendance data through the turnstiles present in various administration offices distributed throughout Italy. The application processes of analysis, normalization and data processing are currently carried out entirely on the centralized information system hosted in MEF data center.

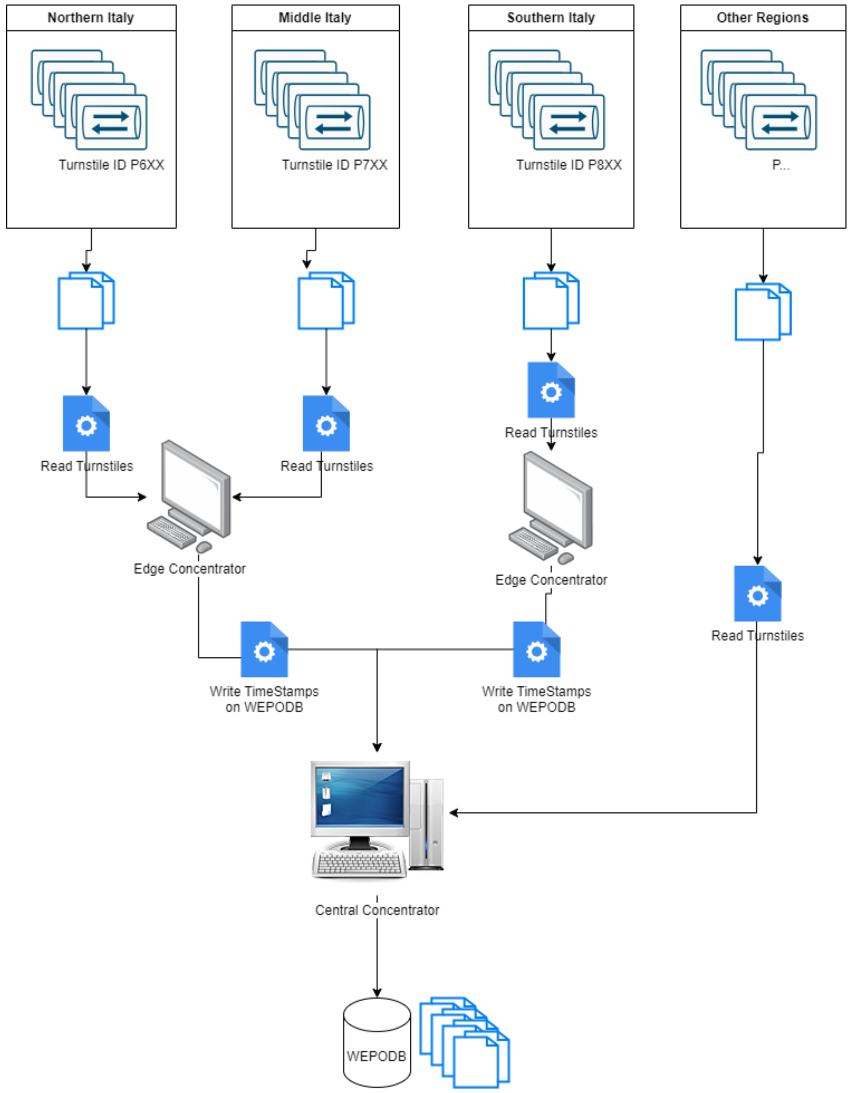


Figure 3: NoiPA Timestamps collection architecture

The proposed use case involves the adoption of the GLACIATION platform to improve the efficiency of NoiPA services through balancing decentralized and centralized processing and reducing of data movement, with AI/ML technologies, ensuring the reduction of energy consumption and processing time.

The proposed solution is designed to optimize the processing of employee attendance data captured through turnstile timestamps. It begins by accepting these timestamp files as input, then proceeds to process the received files, preparing them in a format suitable for application of algorithms aimed at reconciling the employee's daily attendance records. This preparation is crucial for ensuring the data can be effectively used in the subsequent steps of the process. The solution provides valuable information for the GLACIATION system which includes historical telemetry data, the processed timestamp files, and the baseline configuration. This baseline configuration encompasses the initial setup details, featuring the characteristics of the Kubernetes cluster and the configurations of the pods involved.

The solution lies in the creation of a container-based edge component to process timestamps tracked in the remote site, balancing:

- Edge computing resources
- Network Traffic between remote site and central data center
- Central data center computing resources.

The edge is a Kubernetes container and the GLACIATION platform is applied to orchestrate the containers and the resources allocated, so to make energy consumption more efficient.

The target architectural design is shown in Figure 4: Use Case 1 architecture:

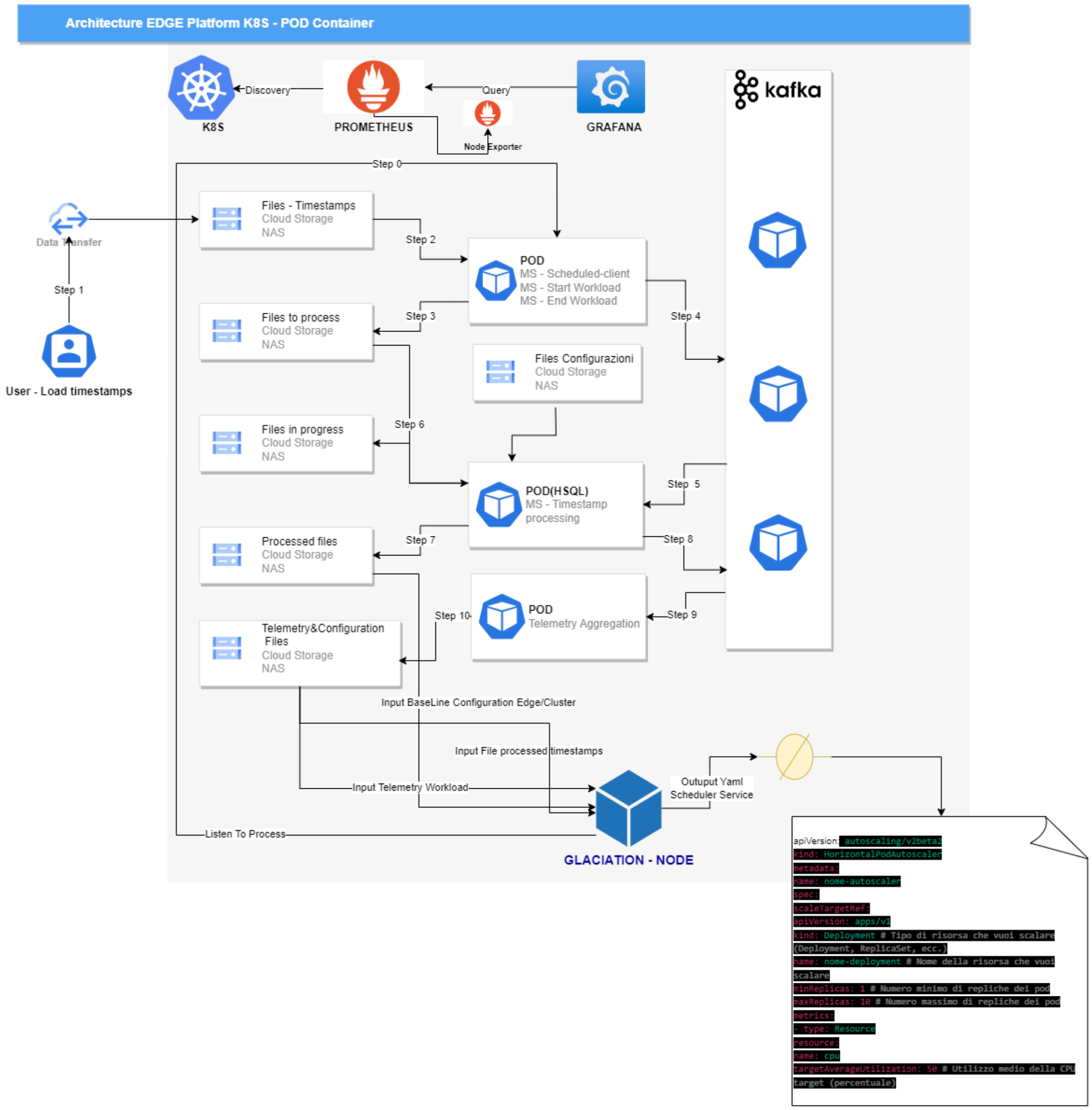


Figure 4: Use Case 1 architecture

<u>K8s Cluster</u> where applications run (REST services, @Springboot/Rest Controller):

- Pod with 3 API
  - API – “Start Workload”, service exposed to start the timestamp acquisition process, invoked with postman (API platform for building and using APIs)
  - API – “STOP Workload”, service exposed to stop the acquisition process
  - API – “Scheduled Client”, service for automatic start-up, scheduled as needed, of timestamp acquisition process

- Pod with API that runs the timestamp processing
- Pod with API for telemetry aggregation, based on Controller/Prometheus, of temporal logs for each acquisition process, in terms of
    - Number of Files Processed
    - KB/MB of each File subject to acquisition
    - Cluster resources used in processing (CPU, RAM, Network).

Kafka Cluster, to make the entire process asynchronous.

Storage Persistence Shared Volume – Components necessary for the management of raw files and result of final processing both in terms of processed timestamps and temporal logs of telemetry on the resources used for processing. Furthermore, the state of the initial configuration is also stored.

GLACIATION Platform: for optimization of cluster resources from an energy point of view, guaranteeing performance.

UC1 aims to provide the following input to the GLACIATION platform:

- Initial cluster configuration
- Timestamp File with its dimension
- Processed timestamp file
- Logs of the entire processing from the start of the workload (scheduled or user activated) with the following information:
    - Started process
    - Identifier file 1 timestamps/…File n timestamps
    - Size of each file
    - Resources used for pod/service/components (cpu/ram/network/ etc…).

The expected output from the platform is as follows:

- The initial cluster configuration corresponds to the needs in terms of energy savings
- Proposal for a new cluster configuration in terms of resource scheduling with corresponding parameters below is an example (indicative) of the expected dictionary as output:

```
services:
  minReplicas: 8
  maxReplicas: 10
```

```
metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 50
  - type: Resource
    resource:
      name: memory
      target:
        type: AverageValue
        averageValue: 500Mi.
```


It will be possible to validate and verify obtained output in terms of energy savings and unchanged application performance.

3.1.1 Validation Environment

The physical architecture for UC1 is a k8s environment with the configuration reported below:

- 3 virtual machines with Linux Ubuntu server 22.04 LTS for the kubernetes cluster, the technical characteristics are given in Table 1: VM Technical characteristics

CPU	RAM	Storage	Network
4	16GB	50GB	• SSH access enabled • Internet access enabled • Middleware Kafka access

Table 1: VM Technical characteristics

VMs will be divided as follows:

- N.1 VM, K8S control plane manager, with port 8443 HTTPS enabled for Grafana dashboard
- N.1 VM, K8S worker
- N.1 VM K8S worker
- Volume Storage (NFS) of 50GB
- Middleware Kafka
- Prometheus/node exporter.

The validation environment was designed to simulate the operating conditions present in the NoiPA centralized information system with the aim to:

- Move part of the timestamps file processing application process to the edge
- Provide an environment compliant with the GLACIATION platform
- Identify, using GLACIATION platform, the correct resource configuration of the pods used to decentralize steps of timestamps processing workload and gaining in reduction of central computation and energy savings, maintaining performance needed for business process.

3.2 UC2: Data-driven energy-efficient manufacturing

Dell Technologies' use case is based in Cork Campus Manufacturing facility. The use case analyses the data generated from Cobots (Collaborative Robots) and Tugbots (Autonomous Mobile Robots) functioning within the facility and the power consumed by these Robots.

Dell Technologies develops edge computing solutions tailored to the challenges and opportunities manufacturers face at the edge. The Dell validated design for Manufacturing Edge helps manufacturers enhance efficiencies, reducing waste, and accelerate the integration of advanced technologies at the manufacturing edge. An example of this validated design driving Smart Manufacturing towards more efficient operations, reducing waste, and the ability to run advanced applications such as digital twins, machine vision, as shown in Figure 5**Error! Reference source not found. Error! Reference source not found.**below.

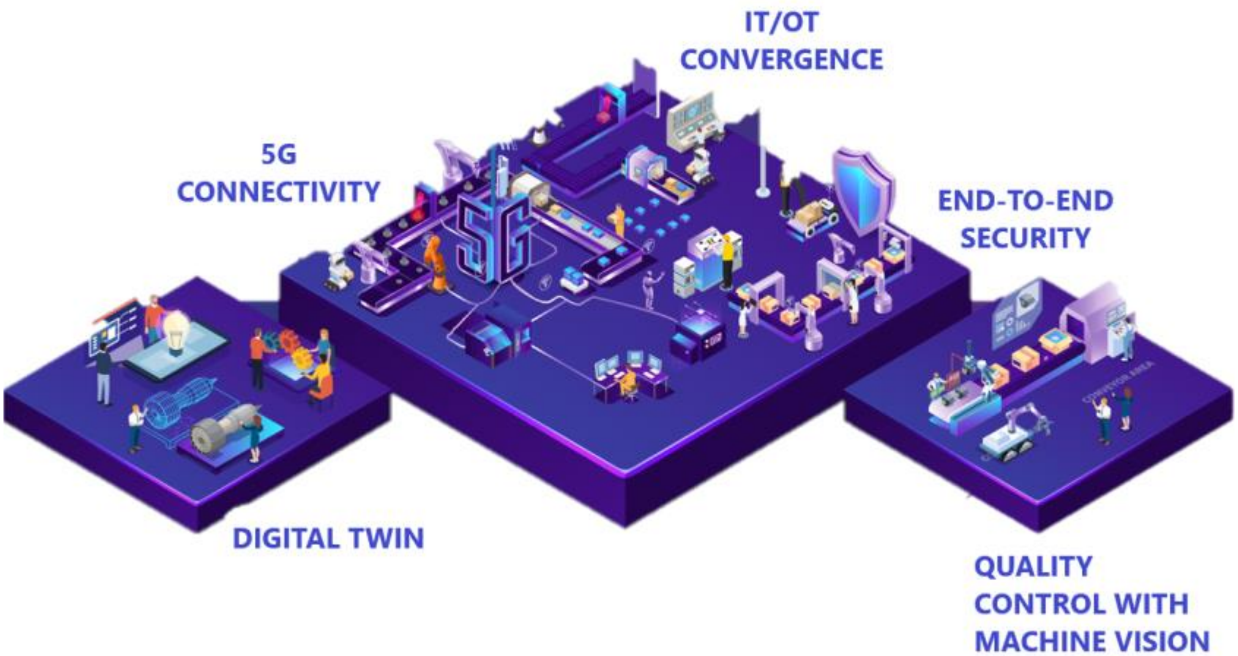


Figure 5: End-to-End Smart Manufacturing Solution

The problem identified is within the facility, many cobots and tugbots are functioning, and generating data related to the tasks that they carry out but the data about the robots themselves is not currently being analysed. The Pilot involves software that analyses data generated by robots in the facility, running on the GLACIATION platform. The platform will ingest all data from robots including diagnostic data of the robots themselves alongside the functional data, which the robots are designed for.

The use case generates large amounts of data from various sources, including sensors, robots, and other IoT devices. This data will be collected, stored, and analysed in such a way as to optimize the use of renewable energy sources and minimize energy consumption. The manufacturing use case is designed to improve the energy efficiency of data analytics within a manufacturing setting.

This pilot addresses the need for the resource-efficiency nature of the robots and the advanced analytics of the robots in a highly digitalised smart manufacturing setting. Overall, this use case of the GLACIATION platform demonstrates the advanced data analytics in improving the efficiency and sustainability of manufacturing processes, while also protecting personal and sensitive information through robust privacy measures.

3.2.1 Validation Environment

The GLACIATION manufacturing use case UC2 is centred around the various hardware and software components integrated to facilitate IoT services in manufacturing. The UC2 platform involves establishing a test bed environment by configuring multiple hardware and software components, involving deployment of two servers, three gateways, to form a five-node cluster, incorporating smart PDUs, cameras, sensors, robots, and the platform utilizing Kubernetes (K8s) cluster orchestration, as shown below in Figure 6 and Figure 7.

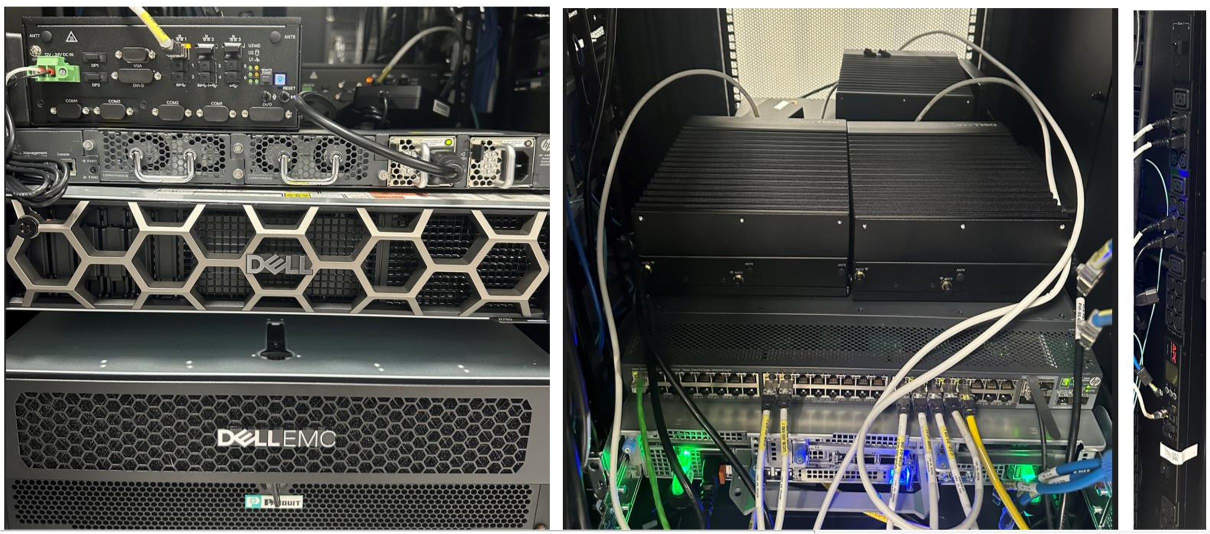


Figure 6: Servers and PDUs for Test Bed

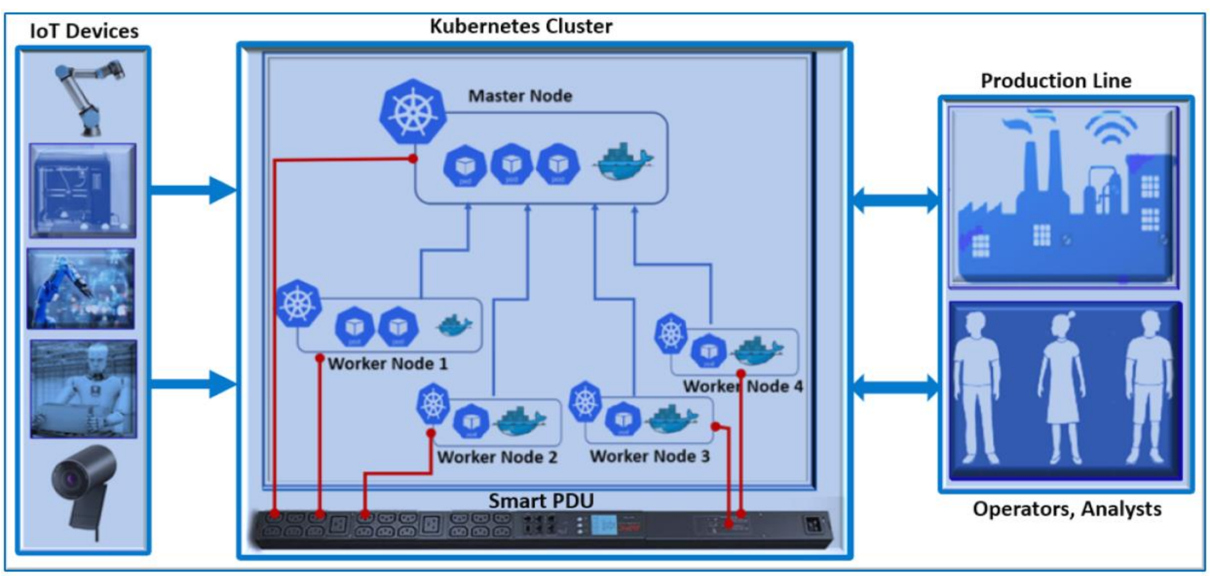


Figure 7: UC2 Test Bed setup

This process involves creating a Kubernetes pod with a container image, developing an Excel reader application utilizing Python libraries like Pandas, building the Excel reader application into a Docker image, pushing the Docker image to a container registry accessible by the Kubernetes cluster, and deploying the pod to access data from an Excel sheet where the test data is documented.

The Kubernetes cluster on Ubuntu OS 22.04 orchestrates energy measurement functions across servers and Edge Gateways. The inclusion of the smart PDUs, robots, sensors, cameras enhance the ability to capture, contribute real-world data for analytics. The integration of these components ensures a comprehensive evaluation of energy usage, allowing for optimization and energy management.

The testbed facilitates interaction between robots, sensors, and cameras collecting and sharing the information. It aims to create a physical validation environment to support cooperative. Below is a description of all the components and the figure shows the UC2 test bed set up:

Servers – Two PowerEdge rack servers with high processing power, memory, and storage options, suitable for demanding workloads:

- PowerEdge R760 Server, deployed as a Master node
- PowerEdge XR12 Server, deployed as a Worker1 node

Edge Gateways – Three Edge Gateway 5200 with Core i5-9500TE, 2x DDR4 SO-DIMMs, up to 64 GB memory, Ubuntu OS support, collect data directly from edge endpoints to facilitate seamless communication and data exchange within the environment, are deployed as:

- Worker Node 2
- Worker Node 3
- Worker Node 4

Smart PDUs – Two APC Metered Rack PDUs with ZeroU 2G – 42 output connectors, manage, and monitor efficient power distribution to connected devices within a rack.

Robots – Multiple robots with advanced sensors and actuators, interact with the environment, execute tasks, contribute to real-world simulations, and provide data for analytics.

Camera – Intel Real Sense Depth Camera D435.

OS – Ubuntu OS 22.04 serves as the host for servers and gateways.

Kubernetes – Cluster running on Kubernetes v1.28.2, serves as the orchestration platform, to manage the deployment and communication with servers and gateways, to ensure seamless operations and containerized workloads.

Yolo – You Only Look Once, an object detection algorithm for real-time video analysis and object recognition, enhances the capabilities of robots and cameras for sophisticated data capturing and analysis.

The validation environment for UC2 is designed to support the Power Measurement Framework testbed, equipped with advanced hardware, software components, and additional resources, to conduct several tests to ensure efficiency in the operations, identify errors/ bugs, and improve performance.

By leveraging the latest technologies, the environment serves as a testing ground for the performance, scalability, and efficiency of energy measurement applications within a Kubernetes-driven cluster, a platform that streamlines the validation process and helps to ensure a cohesive and effective validation process with the environment.

GLACIATION utilizes a secure hardware environment to make protocols more efficient for a test bed, creating a trust, secure, and privacy-preserving environment for sensitive data handling.

3.3 UC3: Privacy-preserving cross-company analytics

The focus of this use case is to foster data-driven business collaborations holistically. First, we will provide a general overview of this scenario and its current problem, then we will detail a more specific instance of this scenario.

Generally, the problem is that sharing data across organizational boundaries, i.e., cross-company, comes with regulatory, privacy, and competitive concerns that hinder such data-based collaborations. At the same time, there is a clear benefit from sharing data, especially to gather meaningful statistical insights across multiple data sources and to improve various machine learning models that either require vast amounts of training data or access to enough valuable, high-quality data.

Now, privacy-enhancing technologies, such as secure multi-party computation and differential privacy, allow to perform computation on encrypted data and anonymize the data, respectively. Thus, these privacy-enhancing technologies allow sharing insights without needing to share data from which the insights have been derived. Specifically, our aim is to protect each company's data while reaping the benefits of a joint data analysis. Given the broad applicability of this new paradigm – *share insights, not data* – we envision multiple business scenarios that could be enhanced by an efficient solution that achieves this paradigm which we envision to further investigate.

Specifically, we aim to focus on cross-company asset performance management. Here, an asset manufacturer for, e.g., production equipment and machinery, wants to gather insights from different customers deploying its assets. Customers are reluctant to participate as such data collection enables to learn usage patterns and details regarding their manufacturing process that could divulge business-sensitive production details and allow inference of a company's short-term business situation and health. At the same time, improved analytics help to improve the design, service, and maintenance of the assets.

The cross-company collaboration, in the form of asset performance management, is illustrated in **Error! Reference source not found.**Figure 8. Here, a manufacturer A produces industry equipment, illustrated as industrial pumps, which are used by the customers C1, C2, …, Cn of manufacturer A. Notably, this is applicable to different industry sectors, such as automotive, and not restricted to the illustrated example. The collaborative computation is enabled and protected by privacy-enhancing technologies with a focus on secure multi-party computation.

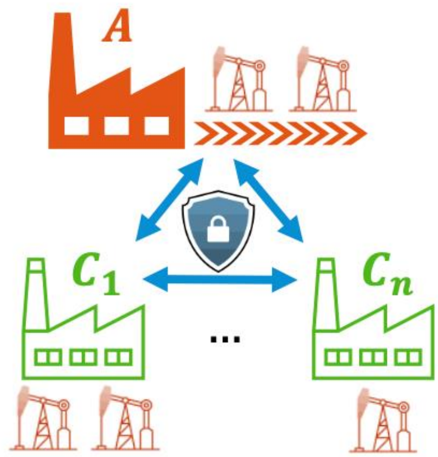


Figure 8: UC3 illustration for cross-company collaboration with manufacturer A & its customers C1, …, Cn.

Asset performance management encompasses multiple tasks. Our focus is on providing securely computed outputs from distributed data. Thus, one task is to learn statistical insights of asset health, i.e., typical monitor equipment behaviour and analyse common issues, which can be satisfied by securely computing statistical analysis on the distributed data. Here, the goal is to leverage the insights to improve production processes and designs, to improve sustainability efforts by reducing waste and eliminating inefficiencies.

Another important task is predictive maintenance, to increase asset availability and help predict failures early, which can be realized by collaboratively building machine learning models for forecasting. Here, the goal is to use the distributed data to collaboratively train such a model to improve the model by incorporating a more diverse and larger set of training data. The aim is to improve model accuracy by having access to previously unavailable, siloed data, i.e., provide better forecasting results for informed decision making.

3.3.1 Validation Environment

SAP's use case aims to foster cross-company collaborations in general and is illustrated for a specific scenario outlined above. The vision is to enable general, privacy-preserving data sharing between companies with the help of privacy-enhancing technologies. To find general acceptance and provide the basis for large-scale uptake by business customers of SAP solutions, the use case realization must be suitable for cloud-native deployment. Specifically, SAP Business Technology Platform, the cloud solution optimized for SAP applications, must be supported. The SAP Business Technology Platform (BTP) is a platform-as-a-service (PaaS) provider. Its primary purpose is to serve as a robust foundation for the development and deployment of applications that seamlessly integrate with various SAP services in the cloud, such as S4/HANA.

Regarding BTP-suitable deployment, we note that BTP provides multiple environments for deploying business solutions. For example, Cloud Foundry provides runtime options for creating cloud-native applications in languages like Java. Also, Kyma provides a managed Kubernetes cluster, i.e., with a set of predefined components for, e.g., monitoring and load balancing. For our purpose, however, we focus on Gardener. Gardener is a tool developed by SAP which simplifies the creation of Kubernetes environments on different cloud providers, e.g., AWS and Azure. It is also used to deploy the Kyma runtime for BTP.

To realize computation on encrypted data, we focus on secure multi-party computation (MPC). Briefly, we chose MPC due to its many security models (e.g., malicious security ensuring the correct computation was performed if so desired), maturity (i.e., long line of research), collaboration-suitable distributed computation model, and resistance to attacks that hardware-based solutions (e.g., trusted execution environments) suffer from[1].

For MPC, we distinguish between an MPC backend (running the computation on encrypted data) and MPC orchestration (cloud-native deployment of computation parties running MPC, networking, encrypted storage, etc.). For MPC orchestration, we use open source MPC cloud stack Carbyne Stack from Bosch Research (https://carbynestack.io/), where SAP is also a contributor, which leverages cloud-native technologies and Kubernetes. For MPC backend, we follow Carbyne Stack, which uses the well-regarded MPC framework MP-SPDZ (https://github.com/data61/MP-SPDZ).

In summary, we use the following technologies and software components to validate UC3:

- **Gardener** for BTP-suitable cloud-deployment
- **Carbyne Stack** for cloud-native, scalable MPC orchestration
- **MP-SPDZ** for MPC backend, i.e., computation on encrypted data

[1] In more detail, MPC ensures that data is never decrypted in use. Trusted execution environments (TEE), i.e., hardware-based protection in a CPU like Intel SGX, on the other hand, decrypt the data in use. In more detail, in TEEs data is decrypted in a restricted environment shielded from other parts of the hardware and software, which suffers from certain side-channel attacks as the operations are applied on plaintext data. Another alternative is fully homomorphic encryption (FHE); however, it currently still suffers from high computational requirements, and MPC, with multiple computing parties, better fits our collaboration scenario. Notably, the different technologies, FHE, TEE, MPC, can be considered in combination. For example, Carbyne Stack allows to protect federated learning models from clients via TEEs and hides client's gradient from the server by using MPC for secure aggregation.

4 Validation Objectives

We split validation objectives in the following main aspects:

1. Requirements for the Use Cases
2. Integration
3. Demonstration
4. Expected Outcomes & KPIs

For each use case, we discuss requirements in the first subsections (4.1.1, 4.2.1, and 4.3.1) and group integration, demonstration, and expected outcomes in the second subsections (4.1.2, 4.2.2, and 4.3.2). Given the validation objectives, we also define their validation criteria and metrics, i.e., successful validation and their measurement in subsections (4.1.3, 4.2.3, and 4.3.3). Regarding requirements, we categorize them by their priorities as shall, should, and may.

For the demonstration aspect, the validation objectives are the objectives defined in the proposal. Specifically, objective 9 – KPI20 and KPI21, which we quote for context below:

*"**Objective 9**: Testing and demonstration of the effectiveness and generality of the GLACIATION approach to Edge-Core-Cloud computing, the innovative computing architectures for energy efficient, privacy preserving and autonomous operation with an emphasis on AI-assisted data movement by evaluating multiple real-world use cases and scenarios that exhibit the required compliance, privacy preservation, green and responsible data operations.*
GLACIATION considers three demanding use cases from different domains and end user needs, covering different domains: (1) public administration, (2) manufacturing, and (3) enterprise. The deployment of the project's results in the three use cases will showcase the strong capabilities of the platform in enabling data-intensive, highly demanding in terms of performance, extremely heterogeneous in terms of data characteristics applications.

***KPI20** Number of applications successfully demonstrating GLACIATION capabilities: 3.*
***KPI21** Number of different domains in which GLACIATION demonstrates its potential: 3".*

For the expected outcomes & KPIs, the validation objectives are the use case KPIs defined in the proposal. Specifically, the associated parts of use case descriptions in Section 1.3.3 which are recalled for context below:

*"**UC1**: The final goal the MEF would like to ensure is to reduce the energy consumption, the waste of processing time and taxpayer money, optimizing data movement and adopting distributed calculation, while guaranteeing privacy preservation and data security across the entire data process lifecycle. To do so, its three main KPIs are: (1) Saving in centrally processing and storage of large amount of data (2) proof that new operating conditions are consistent with important business parameters such as reliability, accuracy, cost structure; (3) Amount of data with a higher degree of privacy protection thanks to the edge computing of the raw data."*

*"**UC2**: There are three primary outcomes for validating the GLACIATION platform through the Dell use case: (1) energy reduction of manufacturing through transition of data analytics to edge-*

core-cloud platform combined with AI/ML technologies; (2) improved predictive analytics through ingestion of data with metadata annotations into distributed graph of GLACIATION; and (3) reduced power consumption of analytics and prediction through optimized data movement throughout the manufacturing GLACIATION platform."

*"**UC3**: "The expected, interconnected outcomes for the collaborative, privacy-preserving computation are as follows: (1) higher analytics accuracy than single-data or general-purpose privacy-preserving solutions, (2) reduced waste due to collaborative data-driven forecasting and planning".*

4.1 UC1 Objectives

4.1.1 Requirements

MEFs final goal is to reduce the energy consumption, the waste of processing time, optimizing data movement and adopting distributed calculation, while guaranteeing actual operating condition in terms of performance, privacy preservation and data security across the entire data process lifecycle.

To do so, the first step is to obtain the analytics requirements, both technical and non-technical.

The analysis of the requirements has been broken down into the following:

Validation ID	Requirement Type	Title	Description
VID_UC1_1	SHALL	Optimize data consumed near the producer	To improve efficiency in computational and energetic resources required by the service. Saving in central amount of data
VID_UC1_2	SHALL	Distribution of calculation based on the historical trends	So that computational central requirement can be reduced gaining in energy saving and avoiding linear growth with data and users
VID_UC1_3	SHALL	Prioritize Low-cost energy workload placement (Machine Learning/AI)	So that the energy consumption is optimized in terms of possible reduction
VID_UC1_4	SHOULD	Energy Monitoring	To make informed decisions and reduce environmental impact

Table 2: UC1 Requirements

4.1.2 Integration, Demonstration & KPIs

There are three primary outcomes for validating the GLACIATION platform through UC1 MEF use case:

1. Saving in centrally processing and storage of large amount of data
2. Proof that new operating conditions are consistent with important business parameters such as reliability, accuracy, cost structure
3. Amount of data with a higher degree of privacy protection thanks to the edge computing of the raw data

The validation aspects are described in Table 3: UC1 Integration & Demonstration, as below:

Validation ID	Aspect	Title	Description
VID_UC1_5	Integration	Integration in GLACIATION platform	Provide seamless and frictionless integration of UC1
VID_UC1_6	Demonstration	Demonstration of GLACIATION capabilities	Successful demonstration of capabilities within GLACIATION context
VID_UC1_7	Demonstration	Demonstration for Use Case domain	Successful demonstration for use case requirements domain
VID_UC1_8	Expected Outcomes & KPIs	Reduce processing & storage	Saving in centrally processing and storage of large amount of data
VID_UC1_9	Expected Outcomes & KPIs	Ensure fit for use	Proof that new operating conditions are consistent with important business parameters such as reliability, accuracy, cost structure
VID_UC1_10	Expected Outcomes & KPIs	Privacy protection	Amount of data with a higher degree of privacy protection thanks to the edge computing of the raw data

Table 3: UC1 Integration & Demonstration

4.1.3 Validation Criteria

The Validation Criteria and Metrics of all the requirements of UC1 are described in Table 4: UC1 Validation Criteria:

Validation ID	Aspect	Title	Validation Criteria and Metric
VID_UC1_1	Requirements	Optimize data consumed near the producer	Proof the possibility to maintain a set of data on Edge (near the producer) processing them locally without moving to the central site creating the conditions for a higher degree of privacy protection where applicable.
VID_UC1_2	Requirements	Distribution of calculation based on the historical trends	Measure computational resource before using the platform (baseline) and after (baseline improvement)
VID_UC1_3	Requirements	Prioritize Low-cost energy workload placement (Machine Learning/AI)	Measure energy consumption of workloads before using the platform (baseline) and after
VID_UC1_4	Requirements	Energy Monitoring	Power measurement framework should maintain energy meters, gathering data, develop methods for computations and estimates and report results
VID_UC1_5	Integration	Integration in GLACIATION platform	Evaluate the platform artifacts to ensure they meet requirements and standards and fit its intended purpose
VID_UC1_6	Demonstration	Demonstration of GLACIATION capabilities	Use case prototype deployed, running in context of GLACIATION e.g., consuming services)
VID_UC1_7	Demonstration	Demonstration for Use Case domain	Use case prototype deployed, running and satisfying majority/all Use Case validation criteria
VID_UC1_8	Expected Outcomes & KPIs	Reduce processing & storage	Measure computational and storage before using the platform (baseline) and after (baseline improvement)
VID_UC1_9	Expected Outcomes & KPIs	Ensure fit for use	Verify that use of the platform keeps or improves the preexistent operating conditions
VID_UC1_10	Expected Outcomes & KPIs	Privacy protection	Verify that use of the software keeps or improves previous privacy conditions on data

Table 4: UC1 Validation Criteria

4.2 UC2 Objectives

4.2.1 Requirements

To gain insights into the analytic needs of the manufacturing use case, the first step is to obtain the analytics requirements, both technical and non-technical.

The use case requirements come from the characteristics of HRC, applications, robotic workstations, and operational scenarios from the production floor area. Further requirements were extracted from the interviews that were conducted with the manufacturing team to extract all necessary use case requirements.

Validation Criteria and Metrics were gathered using various sensors and monitoring systems that are integrated with the industrial equipment to measure the energy performance, consumption, the production patterns of Cobots and AGVs, the edge-cloud devices carrying out the data operations, including the amount of energy used for movement and operations. The data can be transmitted in real-time to a central management system for analysis, visualization, and reporting.

To adapt the methodology as per the needs of the GLACIATION platform, a list of functional and non-functional requirements is selected based on their relation and applicability to the platform.

- Functional requirements – are the system requirements which are needed for running the Use Cases and the application scenarios.
- Non-functional requirements – includes the system requirements, which are not mandatory for running the platform components but concern the proper usability, performance, and expandability of the system.

The main characteristics required in an efficient manufacturing are flexibility and adaptability, flexible system that learns from new tech tools, new conditions in real-time, adapts and runs the entire production process achieving asset efficiency, quality, low costs, safety, and sustainability.

For the manufacturing use case, we have user requirements collections and initial classification, for the derived approach are shown in Table 5 as below:

Validation ID	Requirement Type	Title	Description
VID_UC2_1	SHOULD	Edge & IoT devices	Deploy edge compute hardware & IoT devices
VID_UC2_2	SHOULD	Network connectivity	Required robust communication network
VID_UC2_3	SHOULD	Flexibility and scalability	Should be flexible, adaptability to embrace emerging technologies and scalable to accommodate the big data generated from Smart Factories

Validation ID	Requirement Type	Title	Description
VID_UC2_4	SHOULD	Control and monitoring	Should have a control and real time monitoring system through apps or sensors to provide visibility into factory's performance, quality control
VID_UC2_5	SHALL	AI and ML capabilities	Leverage AI & ML capabilities for the platform to perform workloads seamlessly at the edge
VID_UC2_6	SHALL	Manufacturing cloud computing	Leverage cloud-based solutions to help with planning, execution and managing the entire manufacturing operations
VID_UC2_7	SHOULD	Data movement	Should provide fast, efficient, and secured movement of data generated from robots, cobots, and other machines
VID_UC2_8	SHALL	Data wrapping	Ingestion mechanism shall implement data wrapping functions as per the defined data governance model and different levels of sensitivity and risk
VID_UC2_9	SHALL	Data assessment	Allow data assessment of completeness ensuring data quality
VID_UC2_10	SHOULD	Real-time data streaming	Ingestion mechanism should support real-time stream data handling and ingestion from multiple concurrent sources
VID_UC2_11	SHALL	Batch handling	Ingestion mechanism shall support batch data handling, which will enable the best level of automation
VID_UC2_12	SHOULD	Data sharing / restrictions	Platform should allow data providers to share data or restrict with a particular data consumer or with multiple stake holders as needed while protecting sensitive data
VID_UC2_13	SHALL	Data aggregation and filtering	Able to aggregate and filter data at the edge reducing dependency to central system, and network bandwidth requirements
VID_UC2_14	SHALL	Distributed data processing	Distributed data processing system to enable data processing and analytics across multiple nodes or clusters to handle large volumes of data
VID_UC2_15	SHOULD	Workload movement	Should be able to receive and send workloads efficiently between platform and applications
VID_UC2_16	SHOULD	Access control levels and authorization	Should support access control and authorizations mechanisms, to grant and revoke access as required, implement

Validation ID	Requirement Type	Title	Description
			access levels based upon user's role & requirements
VID_UC2_17	SHALL	Standardized data formats	Shall use standardized data formats to ensure that data can be shared and analysed across multiple nodes or factories seamlessly
VID_UC2_18	SHALL	Data ownership and control	Shall provide clear data ownership and control mechanisms to ensure that each factory maintains control over its own data
VID_UC2_19	SHOULD	Deletion process	Platform should provide guarantees over data deletion following necessary protocols
VID_UC2_20	SHOULD	Policy configuration	Policies for data sets and platform users should be configurable by the data provider
VID_UC2_21	SHOULD	Data Protection	Data should be protected at rest and in transfer, parameters should be configurable by the data owner protecting data controller rights
VID_UC2_22	SHOULD	Data Risk assessment	System should be designed to evaluate the data provided by each contributor to determine the relative sensitivity of the data elements provided
VID_UC2_23	SHOULD	Data Governance & License models	Platform should define data governance models for the data sets with clear language and set of definitions and must support Licensed model
VID_UC2_24	SHALL	Data Storage	Shall be capable of secure and reliable storage, including local, cloud, or hybrid and retrieval of large volumes of data from edge computing devices and other systems
VID_UC2_25	SHOULD	Storage architecture	Should have a scalable storage architecture that will consolidate and share data with every smart manufacturing application, including AI, computer vision and data analytics
VID_UC2_26	SHOULD	Predictive analytics	Should have the ability to perform predictive analytics in real-time to forecast future events and identify potential issues before they occur
VID_UC2_27	SHALL	Alerts and notification	Shall have alerts and notification system in place capturing issues in real-time, will help for faster response and resolution to issues minimizing downtime

Validation ID	Requirement Type	Title	Description
VID_UC2_28	SHOULD	Fault-tolerance	Should have fault-tolerance mechanisms in place to ensure system can continue to operate even in the event of any failures
VID_UC2_29	SHALL	Load balancing	Shall have load balancing mechanisms in place to ensure tasks are distributed evenly across nodes or clusters, reducing the risk of node or cluster overload
VID_UC2_30	SHALL	Identify Root Cause Defects	Shall help investigate the exact cause of quality issues occurred during the production processes
VID_UC2_31	SHOULD	Monitor KPIs	Should monitor KPIs from machines or applications, evaluate on regular basis to check the efficiency and performance levels at different times and limiting latency
VID_UC2_32	SHALL	Interoperability	Shall be able to communicate, interact and integrate with devices using standard protocols and interfaces to enable interoperability and data exchange
VID_UC2_33	SHOULD	System management	Should have secure local and remote management to help with updates and monitoring system health
VID_UC2_34	SHOULD	High availability	Should be highly reliable with minimal downtime and faster recoveries in the event of any disruptions or failures
VID_UC2_35	SHOULD	Secured Login	User logon should be secure and validated for their role, secure access to all persisted data and appropriate security controls for applications
VID_UC2_36	SHOULD	Security	Should implement robust security measures to protect against threats/attacks to ensure the confidentiality, integrity and availability of data across multiple nodes or clusters or network
VID_UC2_37	SHOULD	Privacy	Should adapt a model-based approach to support the specification of privacy requirements and identify violations of private information
VID_UC2_38	SHOULD	Energy monitoring	Should use sensors, smart measurement metrics and energy performance indicators (EPI) to monitor, measure and analyse energy consumption in real-time
VID_UC2_39	SHOULD	Energy efficiency	Should implement energy-efficient technologies and deploy systems to reduce

Validation ID	Requirement Type	Title	Description
			the amount of energy required optimizing power usage
VID_UC2_40	SHALL	Energy awareness	An energy aware system at edge, utilizing energy-efficient components to measure and analyse consumption usage
VID_UC2_41	MAY	Support for Swarm agent	Platform may Support Swarm agent orchestration
VID_UC2_42	SHOULD	Worker records (read worker data only)	Worker's full details should be recorded for identification and allowing only authorised personnels work with robots, assembly line and other floor areas as needed

Table 5: UC2 Requirements

4.2.2 Integration, Demonstration & KPIs

There are three primary outcomes for validating the GLACIATION platform through UC2, Dell use case:

1. Energy reduction of manufacturing through transition of data analytics to edge-core-cloud platform combined with AI/ML technologies.
2. Improved predictive analytics through ingestion of data with metadata annotations into distributed graph of GLACIATION.
3. Reduced power consumption of analytics and prediction through optimized data movement throughout the manufacturing GLACIATION platform.

The validation aspects are described in Table 6, as below:

Validation ID	Aspect	Title	Description
VID_UC2_43	Integration	Integration with platform	Provide seamless and frictionless integration of UC2
VID_UC2_44	Demonstration	Demonstration of Glaciation capabilities	Successful demonstration of capabilities within Glaciation context
VID_UC2_45	Demonstration	Demonstration for Use Case domain	Successful demonstration for manufacturing domain
VID_UC2_46	Expected Outcomes & KPIs	Energy reduction	Energy reduction through transition of data analytics to edge-core-cloud platform combined with AI/ML technologies
VID_UC2_47	Expected Outcomes & KPIs	Improve accuracy	Improved predictive analytics through ingestion of data with metadata annotations into distributed graph of GLACIATION
VID_UC2_48	Expected Outcomes & KPIs	Optimized data movement	Reduced power consumption of analytics and prediction through optimized data movement throughout the manufacturing GLACIATION platform

Table 6: UC2 Integration & Demonstration

4.2.3 Validation Criteria

The Validation Criteria and Metrics are described in Table 7, as below:

Validation ID	Aspect	Title	Validation Criteria and Metric
VID_UC2_1	Requirements	Edge & IoT devices	Deploy edge compute hardware and IoT devices
VID_UC2_2	Requirements	Network connectivity	Reliable, secured, and low latency data transmission
VID_UC2_3	Requirements	Flexibility and scalability	Flexible, adaptable to embrace latest technologies and scalable to accommodate huge demands
VID_UC2_4	Requirements	Control and monitoring	Use Real time monitoring tools to Implement quality control measures, equipment, and production status
VID_UC2_5	Requirements	AI and ML capabilities	Infrastructure to support AI & ML capabilities, including generative AI
VID_UC2_6	Requirements	Manufacturing cloud computing	Tracking production data, material usage, quality counters, timestamping finished goods, and much more
VID_UC2_7	Requirements	Data movement	Define requirements, protocols, encryption. Latency, Authentication and Authorization
VID_UC2_8	Requirements	Data wrapping	Understanding organization's data governance model, define data wrapping functions,
VID_UC2_9	Requirements	Data assessment	Define quality rules, optimize observability, track & secure relevant data
VID_UC2_10	Requirements	Real-time data streaming	Implement techniques to define clear and measurable data quality objectives and criteria
VID_UC2_11	Requirements	Batch handling	Use automated testing tools to help verify batch processing logic, data quality, and performance
VID_UC2_12	Requirements	Data sharing / restrictions	Data Sharing framework with specific legal agreement in place covering privacy compliance
VID_UC2_13	Requirements	Data aggregation and filtering	Use APIs and monitoring or alerting tools to apply filtering and aggregation
VID_UC2_14	Requirements	Distributed data processing	Use distributed data processing framework ensuring it meets data processing objectives
VID_UC2_15	Requirements	Workload movement	Define workload requirements and data movement strategies
VID_UC2_16	Requirements	Access control levels and authorization	Implement strong identity authentication policies, access control, user & role mechanism

Validation ID	Aspect	Title	Validation Criteria and Metric
VID_UC2_17	Requirements	Standardized data formats	Apply some best practices and tools throughout the data lifecycle, adopt standardization ensuring data consistency
VID_UC2_18	Requirements	Data ownership and control	Data owners should monitor metrics such as accuracy, completeness, consistency, timeliness, validity, and report regularly, identify and resolve any issues that arise
VID_UC2_19	Requirements	Deletion process	Develop data retention policies and schedules ensuring secure data removal in compliance with data protection regulations
VID_UC2_20	Requirements	Policy configuration	Set access rules, define policy objectives, implement robust configuration management, penetration testing, regular audits
VID_UC2_21	Requirements	Data Protection	Implement structured framework, set best practices for encryption, access controls, backups, compliance
VID_UC2_22	Requirements	Data Risk assessment	Develop a risk assessment implementation plan and matrix to establish high-risk findings
VID_UC2_23	Requirements	Data Governance & License models	Ensure data in compliance with legal requirements and licensing agreements, comply with industry regulations and standards
VID_UC2_24	Requirements	Data Storage	Define data storage objectives, architecture, replication and redundancy measures, security, performance, integrity
VID_UC2_25	Requirements	Storage architecture	Evaluate the performance of data systems and processes, configurations of tools & technologies
VID_UC2_26	Requirements	Predictive analytics	Use advanced analytics tools and techniques such as neural networks which use a combination of regression, classification, clustering, and time series models.
VID_UC2_27	Requirements	Alerts and notification	Identify the key metrics to be monitored, hardware, performance like CPU, Memory, disk spaces or other processes and design the monitoring system
VID_UC2_28	Requirements	Fault-tolerance	Must incorporate fault tolerant mechanisms to handle hardware failure or software issues and ability to operate effectively even in the absence of fault tolerance
VID_UC2_29	Requirements	Load balancing	Measure count and type of connections, error rate, hosts status, throughput, response time
VID_UC2_30	Requirements	Identify Root Cause Defects	Gather logs from the timestamp of the issue occurred and analyse using relevant tools / techniques

Validation ID	Aspect	Title	Validation Criteria and Metric
VID_UC2_31	Requirements	Monitor KPIs	Define objectives, strategies to achieve KPIs and create dashboards to keep track of performance in real time
VID_UC2_32	Requirements	Interoperability	Identify components, compatibility, and develop a framework to analyse, measure and assess the interoperability among processes
VID_UC2_33	Requirements	System management	Verify system specifications and requirements, check compliance, establish strategies to monitor system health
VID_UC2_34	Requirements	High availability	Design the system to have no single point of failure and recover from the failure, good to stay close as 99.99%
VID_UC2_35	Requirements	Secured Login	Use secured authentication process such as 2FA, 3FA, MFA SSO
VID_UC2_36	Requirements	Security	Define objectives and metrics, establish standards and compliance regulation for data security and privacy
VID_UC2_37	Requirements	Privacy	Define objectives and metrics, establish standards and compliance regulation for data security and privacy
VID_UC2_38	Requirements	Energy monitoring	Maintain energy meters, gathering data, develop methods for computations and estimates and report results
VID_UC2_39	Requirements	Energy efficiency	Assess energy efficiency by calculating the ratio of energy output to energy input. Multiply the result by 100 to obtain the efficiency percentage
VID_UC2_40	Requirements	Energy awareness	Analyse data to identify energy usage patterns, validate energy awareness systems to ensure efficient behaviours
VID_UC2_41	Requirements	Support for Swarm agent	Develop architecture, where each host runs a swarm agent, and one host runs a Swarm manager
VID_UC2_42	Requirements	Worker records (read worker data only)	Ensure workers records are protected to guard against loss and unauthorized access use, modification, or disclosure
VID_UC2_43	Integration	Integration in Glaciation platform	Evaluate the platform artifacts to ensure they meet requirements and standards and fits its intended purpose
VID_UC2_44	Demonstration	Demonstration of Glaciation capabilities	Use case prototype deployed, running in context of Glaciation (e.g., consuming services)
VID_UC2_45	Demonstration	Demonstration for Use Case domain	Use case prototype deployed, running and satisfying majority/all Use Case validation criteria

Validation ID	Aspect	Title	Validation Criteria and Metric
VID_UC2_46	Expected Outcomes & KPIs	Energy reduction	Identify energy-saving opportunities
VID_UC2_47	Expected Outcomes & KPIs	Improve accuracy	Using quick analytics, smart dashboards, for real time monitoring
VID_UC2_48	Expected Outcomes & KPIs	Optimized data movement	Reduce unnecessary data transfer, consider parallel processing, fewer computational resources, renewable energy sources etc

Table 7: UC2 Validation Criteria

4.3 UC3 Objectives

In this section, we first give the requirements for validation of Use Case 3. Then, we focus on aspects of integration, demonstration and KPIs. Finally, we present validation criteria, overviewing what successful validation entails.

4.3.1 Requirements

The validation requirements for UC3 are detailed next.

Validation ID	Requirement Type	Title	Description
VID_UC3_1	SHALL	Strong privacy and security guarantees	Support protection of sensitive values
VID_UC3_2	SHALL	Practical performance	Ensure sufficient performance for target tasks
VID_UC3_3	SHALL	Data collection, analysis, and sharing capabilities	Capability for foundational data movements
VID_UC3_4	SHOULD	Parameterization of privacy-enhancing technologies	Allowing to adjust accuracy/efficiency/performance trade-offs
VID_UC3_5	SHOULD	Improve security/privacy posture	Improve upon baseline for protection
VID_UC3_6	MAY	Improve performance	Improve upon baseline for performance of secure collaborative computation
VID_UC3_7	MAY	Relative ease of use, i.e., usability	Hide technological complexities as much as possible
VID_UC3_8	SHOULD	Deployment in business application environment/context	Support configuration and setup of research prototypes in SAP BTP

Table 8: UC3 Requirements

4.3.2 Integration, Demonstration & KPIs

For UC3, the validation in the context of integration, demonstration and KPIs are listed next.

Validation ID	Aspect	Title	Description
VID_UC3_9	Integration	Integration in Glaciation platform	Provide seamless and frictionless integration of research prototypes
VID_UC3_10	Demonstration	Demonstration of Glaciation capabilities	Successful demonstration of capabilities within Glaciation context
VID_UC3_11	Demonstration	Demonstration for Use Case domain	Successful demonstration for enterprise domain
VID_UC3_12	Expected Outcomes & KPIs	Improve accuracy	Higher analytics accuracy than single-data or general-purpose privacy-preserving solutions
VID_UC3_13	Expected Outcomes & KPIs	Reduce waste	Reduced waste due to collaborative data-driven forecasting and planning

Table 9: UC3 Integration & Demonstration

4.3.3 Validation Criteria

The validation criteria for UC3 are detailed next.

Validation ID	Aspect	Title	Validation Criteria and Metric
VID_UC3_1	Requirements	Strong privacy and security guarantees	Strong security/privacy parameters (e.g., 128-bit for security and epsilon <= 1 for privacy supported)
VID_UC3_2	Requirements	Practical performance	Acceptable computation time (e.g., in minutes) and communication size (e.g., in order of MB) for secure, distributed analytics
VID_UC3_3	Requirements	Data collection, analysis, and sharing capabilities	Capability for foundational data movements
VID_UC3_4	Requirements	Parameterization of privacy-enhancing technologies	Allowing to adjust accuracy/efficiency/performance trade-offs
VID_UC3_5	Requirements	Improve security/privacy posture	Improve upon baseline for security/privacy protection
VID_UC3_6	Requirements	Improve performance	Improve upon baseline for performance of secure collaborative computation
VID_UC3_7	Requirements	Relative ease of use, i.e., usability	Hide technological complexities as much as possible
VID_UC3_8	Requirements	Deployment in business application environment/ context	Support configuration and setup of research prototypes in SAP BTP
VID_UC3_9	Integration	Integration in Glaciation platform	Straightforward interfaces for consumable components, e.g., REST API
VID_UC3_10	Demonstration	Demonstration of Glaciation capabilities	Use case prototype deployed, running in context of Glaciation (e.g., consuming services)
VID_UC3_11	Demonstration	Demonstration for Use Case domain	Use case prototype deployed, running and satisfying majority/all Use Case validation criteria
VID_UC3_12	Expected Outcomes & KPIs	Improve accuracy	Evaluating accuracy improvement of multi-data analytics (e.g., mean squared error)
VID_UC3_13	Expected Outcomes & KPIs	Reduce waste	Evaluating accuracy improvement of collaborative forecasting (e.g., mean squared error)

Table 10: UC3 Validation Criteria

5 Test Methodology

This document serves as an intermediate report, wherein we outline a general test methodology, structure, and preliminary details to guide our ongoing work. We will provide finalized descriptions and detailed information at a later date, acknowledging that these may differ from our current outline. These initial definitions lay the groundwork for Task 7.2, which will further detail and refine our testing and validation efforts.

We have divided our testing methodology into two main areas: platform testing and use case validation. Platform testing aims to ensure the delivery of the platform with the required quality standards, while use case validation focuses on ensuring compliance with the specific requirements of each use case.

5.1 Platform testing

In this section, we would like to define the test methodology for the Glaciation platform. The general strategy is to automate all test cases and do manual testing only as an exception. The automated tests must be in the same repository as the code itself. The automated tests are run on every change such that the regressions are found as early as possible in the development cycle. Every bug must have a test case that verifies the issue.

Below we elaborate methodology for testing functional and non-functional requirements.

5.1.1 Functional testing

The goal of functional testing is to evaluate that the functions of a system are compliant with functional requirements. For functional testing, we follow the test pyramid approach Figure 9.

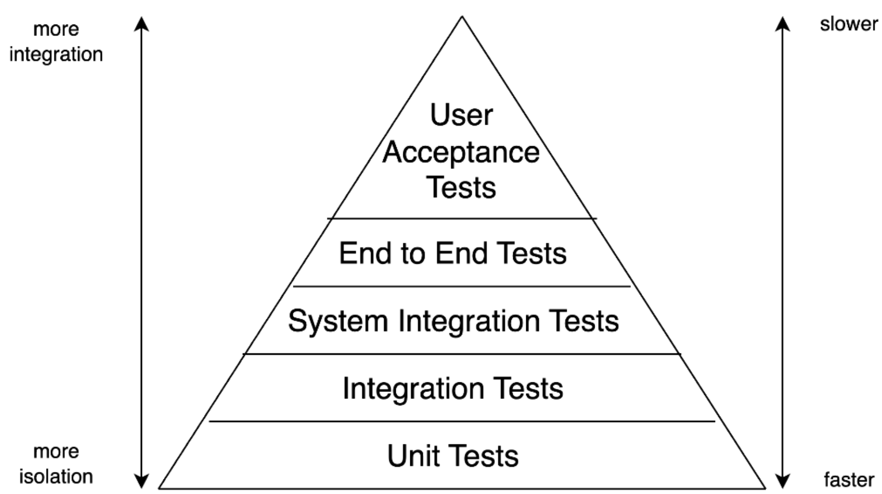


Figure 9: Test Methodology Overview

The testing pyramid defines the several classes of tests. The levels of pyramid are interpreted as a rule for the number of tests of the specific class for the engineers to be able to gradually evolve a big software system and at the same time remain productive, agile, quickly find and troubleshoot issues and maintain software quality at a high level. Thus, the number of tests for each subsequent level up should decrease by the order of 10. This guidance is derived from the two characteristics of the tests. Higher the level of pyramid, more integration is verified by a test. Correspondingly, lower the level, more isolated are the tests. The isolation-integration trade-off influences the effort to write/run and maintain those tests. Additionally tests at the bottom of the pyramid are usually executed faster (seconds) and therefore engineers receive quick feedback if they break anything in the system when they implement a change. On the contrary, execution time of the tests on the top of pyramid is much longer than at the bottom, therefore engineers implement those tests only if it is necessary.

5.1.1.1 Unit tests

In the development we follow test driven development methodology where the feature is developed in a sequence of cycles towards the delivery objective. Each cycle consists of two phases – write code and fix/add tests. This process allows incremental feature development keeping the test quality at the high level.

The scope of the unit test is individual unit level, e.g. Class, or function. According to the test pyramid, most of the tests are unit tests.

When most of the unit tests are ready it is recommended to evaluate test coverage with Python tools like coverage[2]. However, it is recommended to apply coverage only as an auxiliary tool for detecting the edge cases that are not covered with unit tests. It is not desirable to use code coverage as driving tool for the test cases. The driving tool for unit test cases are requirements and features.

5.1.1.2 Integration tests

Integration tests exists to ensure that multiple units/modules are working together according to requirements. Integration testing becomes quite complex because of combinatorial explosion of input vs outputs of the system under test, brittleness of external dependencies and time taken to maintain such tests. To tackle this problem, we split integration testing into two subproblems: 1) integration testing of the core of the module/service without external dependencies and 2) testing that external dependencies behave according to expectations of core of the module/service.

Following this strategy, integration of the core of the module without external dependencies becomes more affordable and no massive effort is needed for implementation of test infrastructure of the core. In fact, the tests of the core do not differ much from the unit tests and are just more advanced version of them.

[2] https://coverage.readthedocs.io/en/7.4.1/

A bit more test infrastructure is required to build integration tests of external dependencies – mainly dependent services. This requires interacting with external modules/services and usually involves network/block device IO against external dependencies running in docker containers. It is important that each service owner prepares a docker image of the service with minimalistic configuration for usage in integration tests.

To be able to apply the methodology described above the architecture of each service must be adopted for testability. There are several methodologies that are in fact different perspectives of the same idea – partitioning the module into "inside" and "outside" and testing those in isolation. For detailed information about this approach one can refer to "clean architecture"[3].

5.1.1.3 System integration tests

System integration testing exists to test one (or several) services/submodules together.

We simplify system integration testing by following a contract-first development where the API of a service is defined using OpenAPI specification, and then compiler can generate client and server code responsible for handling requests and serialization/deserialization. Automatic generation eliminates corresponding subclasses of integration errors by turning runtime-level integration problem into compile-time checks.

The system integration tests are implemented in the test application placed next to the application code and is executed every time service is deployed to an integration environment.

The test application consists of test cases written in the frameworks such as pytest, robot, unit test etc. Each test case must set up each subsystem/service/data participating in the test and correspondingly tear down after use.

Maintenance of system integration tests takes more time than maintenance of unit tests and ordinary integrations tests, therefore system integration tests should cover only basic scenario. And thorough coverage should be done by the unit tests.

5.1.1.4 End to end tests

End-to-end testing is a type of testing that verifies an entire software application from start to finish, including all the systems, components, and integrations involved in the application's workflow. In Glaciation the automated end-to-end tests are covering the platform level excluding specifics of the use cases and ensure that entire system is working end-to-end. The tests are implemented using the same framework as system integration tests with the only difference in the scope of tests.

[3] https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html

5.1.1.5 User acceptance tests

User acceptance testing ensures that solutions meet user requirements. In Glaciation user acceptance tests are specified and validated per use case and therefore documented in the corresponding sections.

5.1.2 Non-functional testing

The goal of non-functional testing is to verify non-functional platform requirements. The sections below provide guidelines for validation of security, scalability, performance and other non-functional requirements generically for a platform. In some cases, no quantitative objectives exist. For those cases we report objective system capabilities achieved during testing.

5.1.2.1 Security

For security testing we follow the general approach where we periodically perform security audit. During audit we identify the security assets, possible threats and vulnerabilities, identify risks and perform a remediation.

During requirements clarification and initial audit, we came up with the following set of guidelines to minimize the security risks:

- The use of hardened base docker images with latest security fixes
- Perform periodic image security scan to detect new vulnerabilities.
- The network communication should use TLS for the traffic going outside of the Glaciation node.
- Authentication and Authorization of the Glaciation users
- Data should be encrypted wherever required by the use cases.

5.1.2.2 Scalability

The Glaciation platform is expected to be scalable from several perspectives. It should be possible to add new Glaciation clusters and it should be possible to fully load the clusters with workloads. Under those conditions the Glaciation platform should remain available and functional characteristics of the system should not degrade.

5.1.2.3 Performance

For performance verification we check if a fully loaded Glaciation cluster remains responsive, stable and functional characteristics of the system do not degrade.

5.1.2.4 Usability

The usability is tested by collecting feedback for the most common scenarios from a group of users that did not participate in original definition of the user interface and queries. The judgement is qualitative based on the user feedback.

5.2 Use case testing

The use case testing for Glaciation varies per use case and test methodology is given in the corresponding validation criteria section of a use case or in a specific test case.

In the following sections, we will outline tests per use case. Specifically, we first provide a table containing the following:

- test **id** for reference, e.g., T_Ucx_y for y'th test of use case x
- test **coverage**, i.e., which VIDs are covered
- if the test is a **blocker** that hinders progress
- the test **scenario**, e.g., if it is concerned with energy metrics
- the main **components** involved in the test

Notably, the test ids have been assigned in order of how they are envisioned to be performed, e.g., initial component deployments happen before testing of component outputs.

Then, we provide additional textual descriptions for the tests, overviewing the goal of the tests and providing further details.

Finally, we provide workload description, which cover the following aspects

- *data*: potential data set to be used
- *task*: general computations to be performed on the data
- *algorithms*: more specific algorithms, e.g., analytical queries

We use the structures below for the test definitions and test procedures:

Test Definitions (detailed per UC in next sections)

- Test scenarios: aspects of the use cases to test, required components (test coverage items) and their interactions.
- Test data: typical data for identified scenarios (with valid/invalid inputs, edge cases)
- Prioritization: order of test, blocking/non-blocking; notably, the test IDS follow this order

Test Procedure (per test case)

- Execution: define testbed environment and testing schedule, record results and deviations from test cases
- Acceptance/Validation Criteria: a list of specific validation criteria for a test where available
- Reporting: document test results, report potential issues with reproduction step
- Review: discuss test results, update tests and process, fix issues, re-execute to check fixes

5.3 UC1 Tests

The energy consumed by a microservice depends on several factors, including the hardware resources used, the workload of the service, and the efficiency of the service code itself. The reference parameters are:

- Resources used: what hardware resources are allocated to the microservice: cpu, memory, storage space and network bandwidth. **Specifically, the resource configuration baseline is considered as Input**
- Resource usage measurement: measure the actual usage of resources by the microservices defined in the Use Case. **This includes cpu usage, the amount of memory used, the network traffic generated**. In the specification, a workload history will be provided based on the timestamps acquired and processed with the baseline configuration.
- Depending on the resources used the GLACIATION platform will calculate the estimate of specific energy consumption per clock cycle/calculation operation, memory unit, storage and network.

Test ID	VID Coverage	Blocker	Test Scenario	Main Components
T_UC1_1	VID_UC1_1 / 2 / 3 /4	NO	Setup	Use Case Workload
T_UC1_2	VID_UC1_1 / 2 / 3 /4 /10	NO	Setup	Use Case Workload
T_UC1_3	VID_UC1_1 / 2 / 3 /4 /10	NO	Setup	Use Case Workload
T_UC1_4	VID_UC1_1 / 2 / 3 /4 /10	NO	Setup	Use Case Workload
T_UC1_5	VID_UC1_1 / 2 / 3 /4 /10	NO	Setup	Use Case Workload
T_UC1_6	VID_UC1_1 / 2 / 3 /4 /10	NO	Setup	Use Case Workload
T_UC1_7	VID_UC1_1 / 2 / 3 /4 /10	NO	Setup	Use Case Workload
T_UC1_8	VID_UC1_1 / 2 / 3 /4 /10	NO	Setup	Use Case Workload
T_UC1_9	VID_UC1_5 / 6 / 7 / 10	NO	Integration within Platform	Platform
T_UC1_10	VID_UC1_5 / 6 / 7 / 10	NO	Deployment in business context	Platform
T_UC1_11	VID_UC1_5 / 6 / 7 / 10	NO	Deployment in business context	AI Movement Engine, Data Management services
T_UC1_12	VID_UC1_8 / 9 / 10	NO	Deployment in business context	Glaciation Management Services

Table 11: UC1 Test Definitions

5.3.1 Test Description

This section describes the tests developed within UC1, it covers the description, workflow and achieved results as well as helping to evaluate the performance with regard to varying complexity of all components.

The UC1 targets to develop 14 Test Scenarios to be examined at the test phase:

T_UC1_1-8 – Preliminary phase

Test Group for building historical series needed by the Glaciation Platform.

In this phase workload will be started to process timestamp files corresponding to data collected in different time slots and days.

Preliminary test covers the creation of telemetry logs for processing of a collection of timestamps files with a baseline configuration of the Kubernetes Cluster and test labs resources.

Objective: To ensure the creation of historical telemetry data, serving as the foundation for realizing UC1. Our goal is to demonstrate the feasibility of maintaining a dataset at the Edge (close to the data source), allowing for local processing without the need to transfer data to a central site. This approach aims to enhance privacy protection where it is applicable.

Test Case Workflow:

- Ingestion timestamps files
- Start workload (scheduled or command line tools)
- Timestamps processing
- Telemetry files collection
- Timestamps saved in volume storage
- Telemetry files saved in volume storage
- End workload

Test Result:

- Processed timestamps files
- Telemetry files collections

Test execution details in Table 12: UC1 Preliminary Test:

Test ID	Test Inputs	Input Description	Description
T_UC1_1	timestamps_1 timestamps_2 timestamps_3	3 types of timestamps in daily interval 08:00-10:00a.m	Functions involved: - Start workload - Timestamp processing - Telemetry files collection - End workload Output generated: - Processed timestamps files - Telemetry files collections
T_UC1_2	timestamps_1 timestamps_2 timestamps_3	3 types of timestamps in daily interval 10:00-12:00a.m	Functions involved: - Start workload - Timestamp processing - Telemetry files collection - End workload Output generated: - Processed timestamps files - Telemetry files collections
T_UC1_3	timestamps_1 timestamps_2 timestamps_3	3 types of timestamps in daily interval 12:00-03:00p.m	Functions involved: - Start workload - Timestamp processing - Telemetry files collection - End workload Output generated: - Processed timestamps files - Telemetry files collections
T_UC1_4	timestamps_1 timestamps_2 timestamps_3	3 types of timestamps in daily interval 03:00-06:00p.m	Functions involved: - Start workload - Timestamp processing - Telemetry files collection - End workload Output generated: - Processed timestamps files - Telemetry files collections
T_UC1_5	timestamps_1 timestamps_2 timestamps_3	3 types of timestamps in daily interval 06:00p.m-06:00a.m	Functions involved: - Start workload - Timestamp processing

Test ID	Test Inputs	Input Description	Description
			- Telemetry files collection - End workload Output generated: - Processed timestamps files - Telemetry files collections
T_UC1_6	timestamps_1 timestamps_2 timestamps_3	3 types of timestamps in daily interval (normal traffic day)	Functions involved: - Start workload - Timestamp Processing - Telemetry files collection - End workload Output generated: - Processed timestamps files - Telemetry files collections
T_UC1_7	timestamps_1 timestamps_2 timestamps_3	3 types of timestamps in daily interval (high traffic day)	Functions involved: - Start workload - Timestamp processing - Telemetry files collection - End workload Output generated: - Processed timestamps files - Telemetry files collections
T_UC1_8	timestamps_1 timestamps_2 timestamps_3	3 types of timestamps in daily interval (low traffic day)	Functions involved: - Start workload - Timestamp processing - Telemetry files collection - End workload Output generated: - Processed timestamps files - Telemetry files collections

Table 12: UC1 Preliminary Test

T_UC1_9-11 – Resource phase

In this phase the Integration test with the GLACIATION Platform will be executed to identify a target configuration of the k8s cluster and other test lab resources for energy and resource consumption savings.

Objective: Ensure the creation of the target configuration data to use on T_UC1_12 execution. Demonstrate the feasibility of maintaining a set of data on edge (near the producer) processing them locally without moving to the central site, creating the conditions for a higher degree of privacy protection where applicable.

Test Case Workflow:

- Ingestion of 3 (or more based on test results) timestamps files (for each kind of workday)
- Start workload (Scheduled or command line tools)
- Timestamps processing
- Telemetry files collection
- Timestamps saved in volume storage
- Telemetry files saved in volume storage
- End workload
- Load/send Input files, telemetry files and output files to GLACIATION
- Load/send baseline configuration of the k8s cluster to GLACIATION

Test Result:

- Processed timestamps files
- Target green edge configuration files

Test execution details in the following Table 13: UC1 Resource Phase:

Test ID	Test Inputs	Input Description	Description
T_UC1_9	timestamps_1 timestamps_2 timestamps_3 ……………_N BaselineConf.yaml; Telemetry.log	3 types of timestamps in daily interval (normal traffic day) Baseline configuration file K8S Telemetry data previous execution	Functions involved: - Start workload - Timestamp Processing - Telemetry files collection - End workload - Load/send input files, telemetry files and output files to GLACIATION - Load/send baseline configuration of the k8s cluster to GLACIATION Output generated: - Processed timestamps files - Target green edge configuration files

Test ID	Test Inputs	Input Description	Description
T_UC1_10	timestamps_1 timestamps_2 timestamps_3 ……………_N BaselineConf.yaml; Telemetry.log	3 types of timestamps in daily interval (high traffic day) Baseline configuration file K8S Telemetry data previous execution	Functions involved: - Start workload - Timestamp processing - Telemetry files collection - End workload - Load/send Input files, telemetry files and output files to GLACIATION - Load/send baseline configuration of the k8s cluster to GLACIATION Output generated: - Processed timestamps files - Target green edge configuration files
T_UC1_11	timestamps_1 timestamps_2 timestamps_3 ……………_N BaselineConf.yaml; Telemetry.log	3 types of timestamps in daily interval (low traffic day) Baseline configuration file K8S Telemetry data previous execution	Functions involved: - Start workload - Timestamp processing - Telemetry files collection - End workload - Load/send Input files, Telemetry files and output files to GLACIATION - Load/send baseline configuration of the k8s cluster to GLACIATION Output generated: - Processed timestamps files - Target green edge configuration files

Table 13: UC1 Resource Phase

T_UC1_12 – Validation phase

In this phase will be validated the green configuration received by GLACIATION.

Objective: Validation of the target configuration with a measurement of energy savings percentage for single timestamp processing executed in the previous phases.

Test Case Workflow:

- Ingestion of 3 (or more based on test results) timestamps files of one type of workday
- Start workload (Scheduled or command line tools)
- Timestamps processing

- Telemetry files collection
- Timestamps saved in volume storage
- Telemetry files saved in volume storage
- End workload
- Load/send Input files, telemetry files and output files to GLACIATION
- Load/send baseline configuration of the k8s cluster to GLACIATION

Test Result:

- Processed timestamps files
- Target green edge configuration files

Test is successful if the comparison between energy consumption recorded in the execution of the workloads with the baseline configuration and in the execution with the green configuration given by GLACIATION show a reduction at least in between the range of values of 2% and 5% for each processing execution.

Test execution details in the following Table 14: UC1 Validation phase:

Test ID	Test Inputs	Input Description	Description
T_UC1_12	timestamps_1 timestamps_2 timestamps_3 ……………_N BaselineConf.yaml; Telemetry.log	3 types of timestamps in daily interval (normal traffic day) Baseline configuration file K8S Telemetry data previous execution	Functions involved: - Start workload - Timestamp processing - Telemetry files collection - End workload - Load/send input files, Telemetry files and output files to glaciation - Load/send baseline configuration of the k8s cluster to GLACIATION Output generated: - Processed timestamps files - Target green edge configuration files Demonstrate that the green configuration determines a reduction in energy consumption of each execution at least in the 2%-5% range

Table 14: UC1 Validation phase

5.3.2 Workload Description

For UC1, the workload will cover timestamps processing computations over smaller to bigger datasets.
For UC1, test data will be real data access collected by NoiPA system and test data.

Workload Description

Objective: the objective of this workload is to take as an input multiple samples of timestamps file, and process them to prepare for reconciliation step, necessary to prepare the employee payroll, and return useful information to Glaciation platform (e.g. historical trends, telemetry, initial configuration baseline). The expected outcome with the integration of Glaciation is the proposal for a new cluster configuration tailored on the data collected during the workload execution, optimized in terms of resource and energy saving.

Task: UC workload is based on the following step:

Step	Description	Component	Input	Output
0	Start the GLACIATION platform in listenmode on the k8s cluster	GLACIATION	NA	Start GLACIATION
1	Uploading files to the cluster	SSH, volume storage	Timestamps files	Timestamps file in volume storage
2	pod for timestamps acquisition start	pod (with started or scheduled client), volume storage	Timestamps files	Acquisition process started
3	Timestamps file is stored in the “file to process” area of the volume storage	pod (with started or scheduled client), volume storage	Timestamps files	Timestamps file in volume storage – “file to process”
4	Write Kafka/topic to start timestamps processing process	Pod (with started or scheduled client), volume storage	NA	Topic Kafka for starting timestamps processing
5/6	Timestamps processing	Pod (timestamps processing), volume storage	Timestamps files	Timestamps data under processing in volume storage – “file in progress”
7/8	Timestamps processing	Pod (timestamps processing); Kafka, volume storage	Processed timestamps	Processed timestamps in volume storage – “processed file”; topic Kafka for

Step	Description	Component	Input	Output
				end of processing
9/10	Telemetry aggregation	pod (telemetry aggregation), volume storage, Kafka	Logs of single workload for timestamps file processing	Telemetry stored in volume storage

Table 15: UC1 Workload steps

Data: see previous table.

Algorithms: take input raw timestamps, normalize them and prepare data for next process that computes the employee presence/absence data over every single workday.

5.4 UC2 Tests

We have collected, classified, and scored the user requirements for ranking and scoring for testing purposes. The aim of the scoring is the selection of the most important requirement that are to be implemented in the use-case in a sequential order, in a scale from 1=first to be commenced, to 12=to be done at the close, as shown below in the Table 16Table 16.

Test ID	VID Coverage	Blocker	Test Scenario	Main Components
T_UC2_1	VID_UC2_1	No	Deployment	Edge Devices / Edge Zone
T_UC2_2	VID_UC2_2	No	Network	Platform
T_UC2_3	VID_UC2_7 / 8 / 9 / 10 / 11 / 12 / 13 /14/ 15 / 17 / 18 / 24 / 25	No	Data Management	Data Management Services
T_UC2_4	VID_UC2_16 /21 / 23 / 35/36 / 37/42	Yes	Security and Privacy	Security
T_UC2_5	VID_UC2_20	Yes	Policy Management	Policy model and language
T_UC2_6	VID_UC2_19 / 22	Yes	Risk Management	Secure Collaborative Computation
T_UC2_7	VID_UC2_3 / 4/ 26 / 27 / 28 / 29 / 30 / 31 / 32 / 33 / 34	No	Performance	Observability Services

Test ID	VID Coverage	Blocker	Test Scenario	Main Components
T_UC2_8	VID_UC2_5 / 6 / 41	No	Technology	AI Movement Engine
T_UC2_9	VID_UC2_ 43	Yes	Integration within platform	Cluster and federation forming
T_UC2_10	VID_UC2_38 / 39 / 40	No	Energy metrics	Energy
T_UC2_11	VID_UC2_ 44 / 45	No	Demonstration	Platform
T_UC2_12	VID_UC2_ 46 / 47 / 48	No	Accuracy of analytics	Platform

Table 16: UC2 Test Definitions

5.4.1 Test Description

This section describes the tests developed within UC2, it covers the objective, workflow, achieved results and algorithms or tools used. The UC2 targets to develop 12 Test Scenarios to be examined at the test phase, following ISO/IEC/IEEE 29119 [1] series of software testing standards, as described below:

T_UC2_1 – Deployment

Objective: The objective of test case T_UC2_1 is to validate the hardware components deployment and the setup of the Cluster within the GLACIATION platform, ensuring the robust and reliable function of all hardware components while supporting software and firmware upgrades, and scalability of the cluster.

Test Case Workflow:

- Install all the hardware components: 2 x servers, 3 x gateways and 2 x Smart PDUs.
- Set up a Kubernetes cluster with all 5 x servers.
- Test the stability of the cluster for any hardware or performance issues.
- Assess the support for software and firmware upgrades.
- Assess the scalability of the hardware deployment to accommodate increasing workloads.

Figure 10**Error! Reference source not found.** below illustrates the sequence of steps for executing test case T_UC2_1.

Figure 10: T_UC2_1 Workflow

Test Results:

- All 5 servers are successfully installed and connected to the network, enabling seamless communication among devices.
- Both smart PDUs connected to all servers, functional, and provide real-time monitoring of power metrics from the servers, as shown in the Figure 11 below.
- Cluster has been configured using Kubernetes orchestration.
- The cluster is operating in a healthy state, as shown in the Figure 12**Error! Reference source not found.** below.
- The cluster supports software and firmware upgrades, ensuring compatibility and minimal disruption to running workloads.
- It supports both horizontal and vertical scaling, maintaining a balanced workload distribution.

Algorithms or Tools used*:* Ansible, Kubernetes, Docker, Prometheus, Grafana, Web portal.

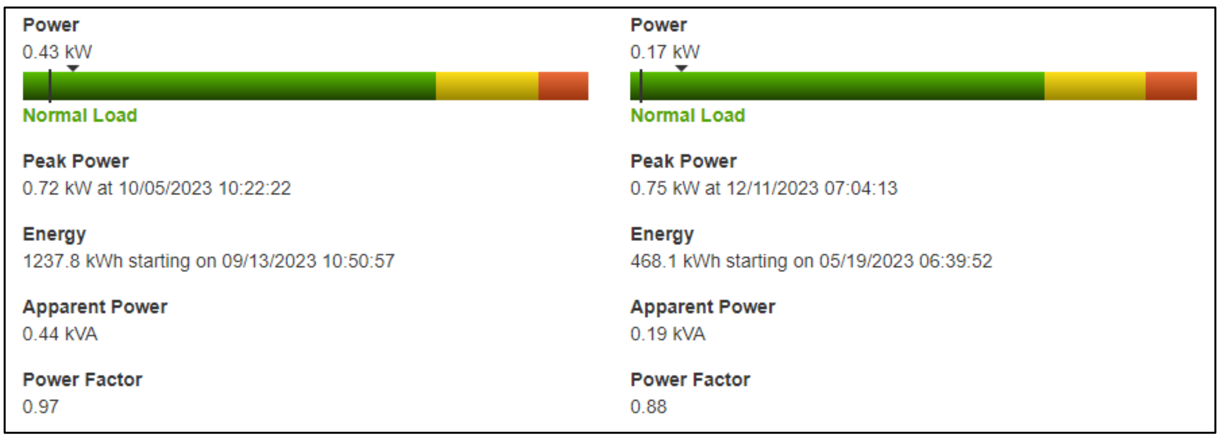


Figure 11: PDU-1 (left) and PDU-2 (right) User Interface

```
ubuntu@GLACIATION-Mast01:~$ sudo dmidecode | grep -i 'system information' -A4; echo -e "\e[91mOS version\e[0m"; uname
--version; echo -e "\e[91mNodes running on Cluster\e[0m"; kubectl get nodes; echo -e "\e[91miperf3 version\e[0m"; ipe
System Information
        Manufacturer: Dell Inc.
        Product Name: PowerEdge R760
        Version: Not Specified
        Serial Number: [redacted]
OS version
Linux GLACIATION-Mast01 5.15.0-89-generic #99-Ubuntu SMP Mon Oct 30 20:42:41 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
Kubernetes version
Client Version: v1.28.2
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
Server Version: v1.28.2
Docker version version
Docker version 24.0.5, build 24.0.5-0ubuntu1~22.04.1
Nodes running on Cluster
NAME                  STATUS   ROLES           AGE    VERSION
glaciation-mast01     Ready    control-plane   119d   v1.28.2
glaciation-worker01   Ready    <none>          116d   v1.28.2
glaciation-worker02   Ready    <none>          119d   v1.28.2
glaciation-worker03   Ready    <none>          119d   v1.28.2
glaciation-worker04   Ready    <none>          119d   v1.28.2
iperf3 version
iperf version 2.1.5 (3 December 2021) pthreads
```

Figure 12: Kubernetes Cluster setup

T_UC2_2 – Network Management

***Objective*:** The objective of test case T_UC2_2 is to verify network connections, and communications between IoT devices on the platform and evaluate the network traffic for any performance issues such as Latency and timeouts.

***Test Case Workflow*:**

- Prepare the test environment by identifying the specific network connections.
- Select preferred tools for network connectivity, latency checks, bandwidth testing, such as Wireshark to intercept network traffic between the client and the application.
- Execute Tests by running ping tests between servers and PDUs using Internet Control Message Protocol (ICMP) ping, measuring data transfer rates between servers, verify open ports.
- Monitor logs for any packet loss during transmission, and latency or timeouts during network interactions.
- Assess the results and investigate any latency spikes or consistently high latency or the cause of timeouts.

Figure 13 below illustrates the sequence of steps for executing test case T_UC2_2.

Figure 13: T_UC2_2 Workflow

Test Results:

- All devices successfully established connections with each other, Test messages sent and received successfully.
- Performed stress tests and latency checked under varying network conditions, got 99.82% of all the CPUs, as shown below in Figure 14.
- Wireshark captures and analyses network traffic between client and application, no high latency observed.

Algorithms or Tools used: Wireshark, Ping utility, Traceroute, Stress, iPerf, SCP.

```
ubuntu@GLACIATION-Mast01:~$ date;iperf3 -c [redacted] -t 10
Fri 16 Feb 15:24:55 UTC 2024
Connecting to host [redacted], port [redacted]
[  5] local [redacted] port [redacted] connected to [redacted] port [redacted]
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  3.17 GBytes  27.2 Gbits/sec    0   1.37 MBytes
[  5]   1.00-2.00   sec  5.11 GBytes  43.9 Gbits/sec    0   1.44 MBytes
[  5]   2.00-3.00   sec  5.15 GBytes  44.3 Gbits/sec    0   1.81 MBytes
[  5]   3.00-4.00   sec  5.15 GBytes  44.2 Gbits/sec    0   2.69 MBytes
[  5]   4.00-5.00   sec  5.35 GBytes  45.9 Gbits/sec    0   2.69 MBytes
[  5]   5.00-6.00   sec  5.09 GBytes  43.7 Gbits/sec    0   2.69 MBytes
[  5]   6.00-7.00   sec  5.47 GBytes  47.0 Gbits/sec    0   2.69 MBytes
[  5]   7.00-8.00   sec  5.45 GBytes  46.8 Gbits/sec    0   2.69 MBytes
[  5]   8.00-9.00   sec  5.36 GBytes  46.0 Gbits/sec    0   4.25 MBytes
[  5]   9.00-10.00  sec  5.17 GBytes  44.4 Gbits/sec    0   4.25 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  50.5 GBytes  43.3 Gbits/sec    0             sender
[  5]   0.00-10.04  sec  50.5 GBytes  43.2 Gbits/sec                  receiver

iperf Done.
ubuntu@GLACIATION-Mast01:~$ stress-ng --matrix 0 -t 1m --times
stress-ng: info:  [4189691] setting to a 60 second run per stressor
stress-ng: info:  [4189691] dispatching hogs: 128 matrix
stress-ng: info:  [4189691] successful run completed in 60.03s (1 min, 0.03 secs)
stress-ng: info:  [4189691] for a 60.03s run time:
stress-ng: info:  [4189691]    7684.17s available CPU time
stress-ng: info:  [4189691]    7667.90s user time   ( 99.79%)
stress-ng: info:  [4189691]       0.79s system time (  0.01%)
stress-ng: info:  [4189691]    7668.69s total time  ( 99.80%)
stress-ng: info:  [4189691] load average: 81.52 25.71 14.95
```

Figure 14: T_UC2_2 Network Test

T_UC2_3 – Data Management

Objective: The objective of the test case T_UC2_3, is to validate Data Management Services, ensuring secure and efficient data movement and storage, across the Edge to Cloud continuum.

Test Case Workflow:

- Define the test environment and install Data Management services within the cluster.
- Create test workloads to interact with the Data Management services.
- Perform data movement tests, across the edge-core-cloud continuum.
- Monitor the data movement process for latency, integrity, overall performance.
- Record and assess the test results and address identified issues.

Figure 15 below illustrates the sequence of steps for executing test case T_UC2_3.

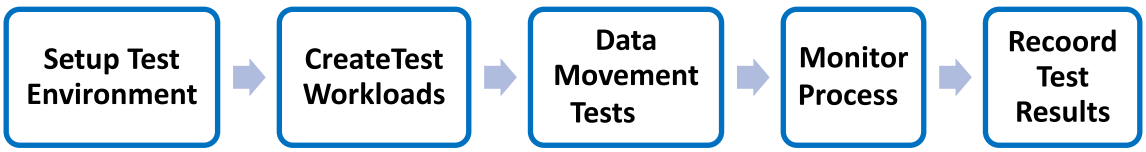


Figure 15: T_UC2_3 Workflow

Test Results:

- Test workloads deployed as per the requirements.
- Data movement tests completed successfully within the timeframe.
- Data Management Services capable of handling data movement and workload placement.

Algorithms or Tools used: SCP, RBAC, REST API, Command-line tools.

T_UC2_4 – Security and Privacy

Objective: The objective of the test case T_UC2_4 is to ensure the implementation of strong data security, privacy, compliance with industry regulations, and adherence to the EU privacy regulation (GDPR). It aims to implement strong authentication and authorization policies, secured access mechanism standards, supports Role-based access control (RBAC) and tracking security event lifecycle in real-time.

Test Case Workflow:

- Review the security and privacy requirements for the platform, including GDPR requirements for handling personal data along with other industrial regulations.
- Implement the specified data protection and security to protect sensitive data at rest and in transit, data retention, and deletion policies as in compliance with GDPR and other industry regulations.

- Test the privacy and access control mechanisms, and scan for potential leaks of sensitive data.
- Monitor the system and security event logs for any alerts, incident notifications, security threats or breaches.
- Generate test reports summarizing the assessment results and provide recommendations to strengthen security and privacy controls.

Figure 16 below illustrates the sequence of steps for executing test case T_UC2_4.

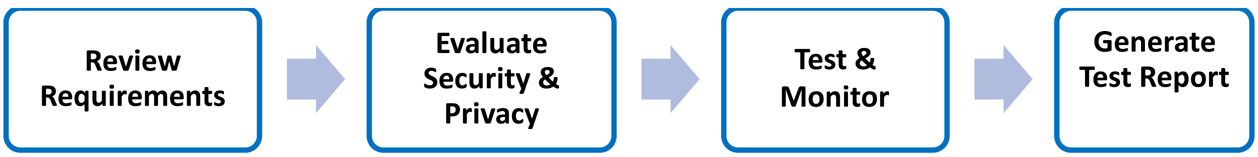


Figure 16: T_UC2_4 Workflow

***Test Results*:**

- System demonstrates strong data security practices, GDPR compliance, and effective security event tracking.
- RBAC is enforced to restrict access to sensitive data based on user roles and permissions.
- Strong identity authentication policies and secured access mechanisms are implemented including Multi-Factor Authentication (MFA), Single Sign-On (SSO)
- Sensitive data is encrypted in transit and at rest using industry-standard encryption algorithms, such as Advanced Encryption Standard (AES-256).
- Real-time security event monitoring is enabled using intrusion detection systems (IDS), and log analysis tools such as syslog.

***Algorithms or Tools used*:** IDS, Syslog

T_UC2_5 – Policy Management

***Objectives*:** The objective of T_UC2_5 test case is to validate the policy configuration, to ensure the policies are clear, correctly defined and implemented as aligned with the platform requirements, ensuring policy statements addressing security, privacy, and compliance requirements.

***Test Case Workflow*:**

- Identify and define the policies relevant to the platform.
- Implement the defined policies aligned with the application's policy model as per the rules and requirements.
- Evaluate the effectiveness of the policies enforced.

- Execute the tests to verify that policies are correctly configured and monitor audit logs for any policy events in real-time.
- Generate reports on policy compliance.

Figure 17 below illustrates the sequence of steps for executing test case T_UC2_5.

Figure 17: T_UC2_5 Workflow

***Test Results*:**

- Policies are clearly defined and configured based on the Access Control List (ACLs) and authentication process.

***Algorithms or Tools Used*:** Open Policy Agent (OPA) and formal methods to check policies.

T_UC2_6 – Risk Management

***Objective*:** The objective of T_UC2_6 test case is to proactively identify and address the potential risks, enhancing the system resilience against security threats and ensuring the confidentiality, integrity, and availability of sensitive data and resources.

***Test Case Workflow*:**

- Identify potential risks such as security attacks, data leakage, data and policy breaches, and unauthorized access.
- Evaluate the identified risks and develop risk mitigation strategies to reduce the impact of identified risks.
- Execute test cases to assess the risk mitigation strategies.
- Monitor the system for potential security incidents, vulnerabilities, unauthorized access.
- Incident Report detailing the procedures of investigation and resolution.

Figure 18**Error! Reference source not found.Error! Reference source not found.** below illustrates the sequence of steps for executing test case T_UC2_6.

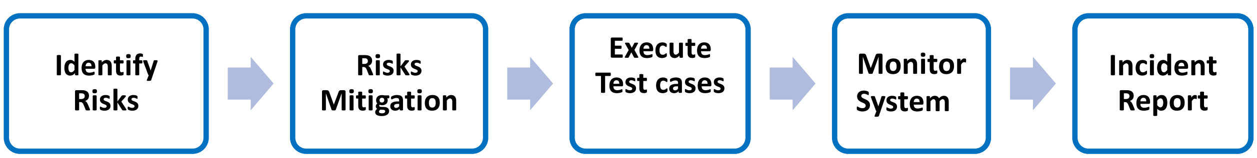


Figure 18: T_UC2_6 Workflow

***Test Results*:**

- Risk mitigation measures are implemented successfully, preventing or mitigation of identified risks under various situations such as data sharing, and tampering.
- System able to handle simulated attacks and security breaches.

***Algorithms or Tools used*:** Vulnerability Management scanning, clustering algorithm.

T_UC2_7 – Performance

***Objective*:** The objective of test case T_UC2_7 aim to assess cluster's stability, latency, throughput, and fault tolerance capabilities, ensuring cluster's performance as expected.

***Test Case Workflow*:**

- Identify the metrics and key performance indicators (KPI) to be monitored and define thresholds.
- Implement Monitoring services to collect telemetry data to analyze performance metrics.
- Define alerting rules and performance thresholds to detect anomalies.
- Execute tests to measure system performance under various load conditions.
- Analyze results, metrics and error rates, latency and fault tolerance mechanisms, system response to hardware failures, network issues, and application errors.
- Conduct Root Cause Analysis (RCA) for performance incidents and outages to identify underlying issues and contributing factors.

Figure 19 below illustrates the sequence of steps for executing test case T_UC2_7.

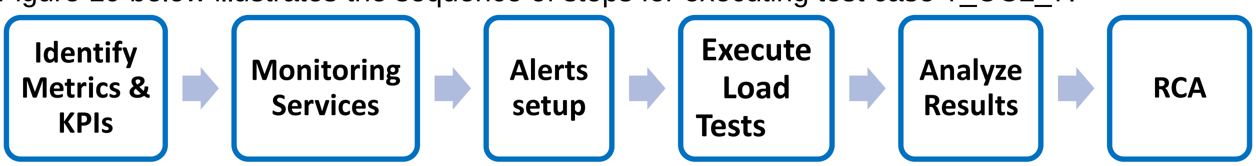


Figure 19: T_UC2_7 Workflow

Test Results:

- The system can handle peak loads without degrading performance.
- Target Latency is 100 milliseconds and the average latency during load testing for the successful transactions was 80 milliseconds.
- The failure rate of 0.1% falls within acceptable parameters. Any value greater than this requires investigation.

Algorithms or Tools used: KPI, Grafana, Node Exporter, Prometheus, formal log analysis, machine-learning-based anomaly detection.

T_UC2_8 – Technology

Objective: The objective of test case T_UC2_8, is to validate the integration of advanced tools and technologies into the system, such as, AI movement engines, AI & ML capabilities, cloud-based solutions, and Swarm agent orchestration to enhance the cluster's performance.

Test Case Workflow:

- Set up the testing environment with the required infrastructure compatible with the preferred solutions such as AI movement engines, AI & ML capabilities, Swarm agent orchestration, or cloud-based platforms.
- Gather training data for AI/ML movement engines or configuration parameters for Swarm agents.
- Design test scenarios and build datasets to train ML models.
- Execute test scenarios and monitor AI movement engines' response, assess ML model, and observe swarm agent behaviour.
- Assess the results and Document the integration process.

Figure 20 below illustrates the sequence of steps for executing test case T_UC2_8.

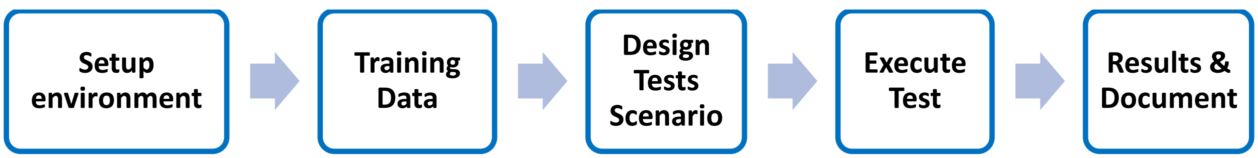


Figure 20: T_UC2_8 Workflow

Test Results:

- Testing demonstrates the successful integration of advanced tools and technologies.
- ML model achieved 95% accuracy in identifying objects, with the average response time of 100 milliseconds.

Algorithms or Tools used: Python, Open AI, Prometheus, Grafana

T_UC2_9 – Integration within the platform

Objective: The objective of test case T_UC2_9 is to validate the integration of components within the platform while ensuring compatibility and functionality of platform artifacts, seamless communication, and data exchange within components.

Test Case Workflow:

- Identify components that need to be integrated within platform and interfaces through which these components will communicate.
- Design integration flow between the components.
- Setup the environment with the required infrastructure and tools.

- Execute integration tests as defined test plan and test cases.
- Monitor the integration tests and analyse the results to identify any issues or failures and document results.

Figure 21 below illustrates the sequence of steps for executing test case T_UC2_9.

Figure 21: T_UC2_9 Workflow

Test Results:

- Platform artifacts demonstrate compatibility across clusters and federation boundaries.
- Integration testing confirms seamless communication and data exchange between clusters within the platform, facilitating workload migration and resource sharing.

Algorithms or Tools used: Cluster, ML, Deep Learning, and prediction algorithms.

T_UC2_10 – Energy Metrics

Objective: The objective of test case T_UC2_10 is to validate the effectiveness of energy-efficient data movement, cluster node management, resource allocation, and task scheduling within the Kubernetes cluster, aligning with energy-efficient data operations initiatives and energy conservation goals.

***Test Case Workflow*:**

- Setup the environment with the required hardware and Kubernetes deployment.
- Configure the cluster to be energy-aware and data management policies and tools to ensure energy-efficient data movement, optimize data storage, and resource allocation within the cluster.
- Task scheduling and Data movement techniques to minimize energy consumption and prioritize energy-efficient nodes balancing workload distribution to minimize idle resource consumption.
- Resource allocation mechanism based on workload patterns.
- Monitor energy metrics such as power consumption, CPU utilization, and memory usage across the cluster.

Figure 22 below illustrates the sequence of steps for executing test case T_UC2_10.

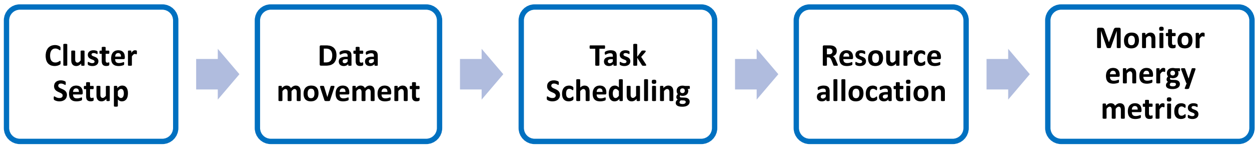


Figure 22: T_UC2_10 Workflow

Test Results:

- The platform aligns with energy-efficient data movement initiatives.
- Data caching, Task scheduling, and resource allocation techniques meet workload demands and improve energy efficiency performance, reducing energy usage by 3% during data movements. Figure 23 below is the piece from Grafana, the Kepler exporter shows power consumption in last 30 minutes.

Algorithms or Tools used: Data caching, compression, deduplication, Kubernetes autoscaling, Power Measurement Framework with components such as Kepler.

Figure 23: Kepler Exporter Dashboard

T_UC2_11 – Demonstration

Objective: The objective of test case T_UC2_11 is to demonstrate the platform and its key achievements.

***Test Case Workflow*:**

- Set up the platform environment with all the hardware components and software applications.
- Identify the key achievements such as security enhancements, performance improvement, and energy efficient data movement.
- Demonstrate the key features, achievements, and functionalities of the platform.

- Measure and present the performance metrics such as error rates, response time, memory consumption, resource utilization.

Figure 24 below illustrates the sequence of steps for executing test case T_UC2_11.

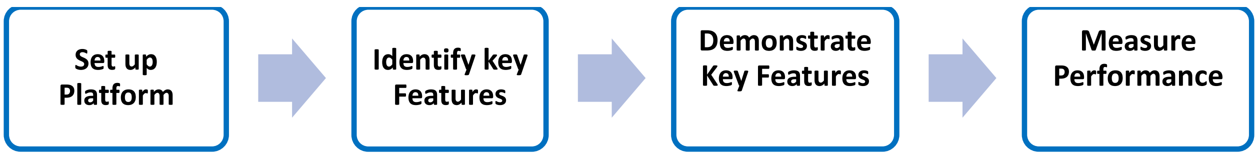


Figure 24: T_UC2_11 Workflow

Test Results:

- The platform demonstration showcases the key achievements, performance, feature upgrades.
- The KPI shows an average response time of 100 milliseconds, throughput of around 500 requests per minute.

Algorithms or Tools used: Helm, Kubernetes, GitLab, Prometheus and Grafana

T_UC2_12 – Accuracy of Analytics

Objective: The objective of test case T_UC2_12 aims to validate the reliability and integrity of data generated and used for analytics, providing accurate insights and predictions.

Test Case Workflow:

- Gather data from servers, robots, cameras, sensors, and logs and prepare the training data for analysis.
- Select the analytics model and implement it on the prepared training data.
- Test the accuracy of the analytics models by comparing their predictions or insights with the expected outcomes.
- Compare the performance of different analytics models and algorithms to identify the most accurate and effective one measuring the performance metrics, such as error rates, latency, CPU utilization, memory usage, network throughput, disk I/O.
- Record the results and analyze the effectiveness and robustness of the most accurate model.

Figure 25 below illustrates the sequence of steps for executing the test case T_UC2_12.

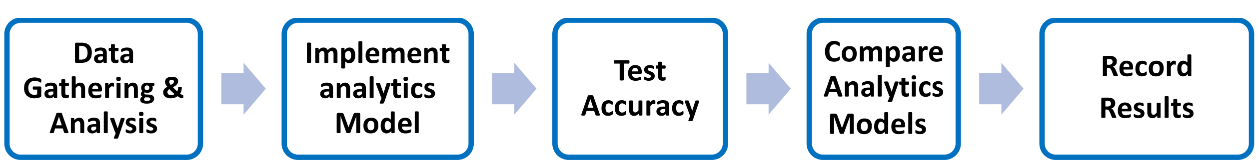


Figure 25: T_UC2_12 Workflow

Test Results:

- The platform exhibits the highest accuracy and consistent performance across different datasets, with accuracy score around 95%.

Algorithms or Tools used: Python libraries such as Panda, ML, Prometheus, Grafana

5.4.2 Workload Description

The workload associated with UC2 offers valuable insights into the manufacturing process together with the deployment of edge hardware within the GLACIATION platform, aligns with the capabilities of edge infrastructure and the workload requirements. In UC2, we are using pre-recorded datasets from manufacturing which involves tasks such as predictive maintenance, quality control, resource management.

Robots, cameras, and sensors are deployed in the manufacturing environment to gather various types of data and Edge gateways, servers deployed to process & analyse the gathered data. Kubernetes pods are deployed using containerized applications to read the processed data for further analysis or integration.

Below Figure 26, shows the relationships between these components indicating the flow of data and operations in the manufacturing use case workloads within the GLACIATION environment.

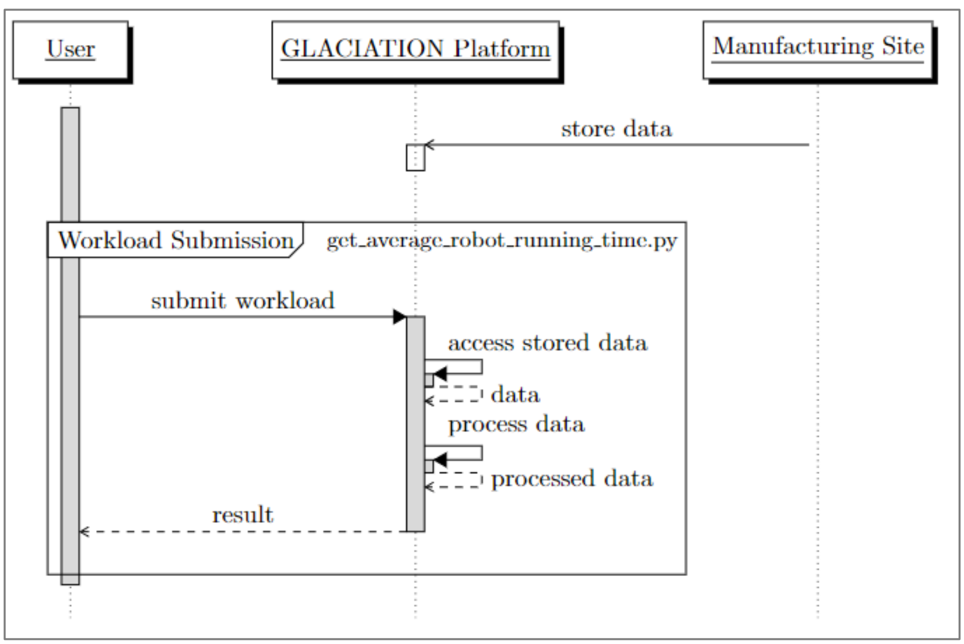


Figure 26: UML diagram for UC2 workload

UC2 targets to develop 3 workload scenarios, as described below:

W_UC2_1

Objective – The objective of workload W_UC2_1 is to monitor and optimize the manufacturing process by analyzing the lot sizes and robot working hours to improve productivity and resource utilization.

Data – The data generated is structured in a tabular format, as shown below in Table 17, with each row representing a specific production batch and columns capturing metrics such as lot size and robot hours, as shown in the Table below:

- Production – This represents production batch order; each row corresponds to a specific production run or batch.
- Lot_Size – Indicates the size of each production batch, the number of items produced in the batch.
- Robot_Hour – Records the time (in hours) spent by robots working on the production run for each batch.

Production Batch	Lot_Size	Robot_Hour
1	30	54.75
2	20	37.5
3	60	96
4	80	127.5
5	40	65.25
6	50	81
7	60	101.25
8	30	51.75
9	70	111
10	60	99

Table 17: Workload1 Dataset

Task – The manufacturing process involves dividing production into batches or lots, each with a specific size (lot size). Robots are deployed in the production environment to perform various tasks, such as assembly, handling, or quality control. The time spent by robots on each production batch is recorded (Robot_Hour). The dataset involves analyzing lot size and robot hours to identify patterns in production efficiency for manufacturing process optimization, predictive analysis, monitoring and quality control purposes.

Algorithms –

- Machine Learning algorithms could be used, training on the datasets for prediction purposes.
- Clustering algorithms could be employed to group similar production batches to identify patterns or anomalies.
- Neural Network for predictive analysis of robot hours based on lot size.

W_UC2_2

Objective – The objective of this workload is to monitor and assess the quality of components produced on the manufacturing floor over a period of 10 consecutive days, to provide insights into the quality of components produced on the manufacturing floor, detect defects in a timely manner, to enhance overall product quality. This workload focuses on inspection, testing, defect handling and data processing tasks in a manufacturing environment. The inspection aims to identify major and minor defects in the components.

Data – Data is collected from inspections and testing of the components based on a random selection of a component every 15 minutes for 10 consecutive days for inspection and testing purposes. The number of defects found for each selected component at each time interval is recorded and saved. The results obtained are presented in a tabular format with three columns, where each row represents a specific inspection event, with the corresponding day, time, and the number of defects found during that inspection. As shown in Table 18 below:

- Day – This represents the order of the data collection per day.
- Time – Records the time of a component selected with an interval of 15 minutes.
- Defects – Records the number of defects found during inspection and testing of a randomly selected component from the production line.

Day	Time	Defects
1	8:15	12
1	8:30	8
1	8:45	9
1	9:00	11
1	9:15	9
1	9:30	10
1	9:45	12
1	10:00	9
………………………………………		
………………………………………		
10	14:15	14
10	14:30	11
10	14:45	8
10	15:00	15
10	15:15	10
10	15:30	8
10	15:45	9
10	16:00	13

Table 18: Workload 2 dataset

Task – The dataset seeks to identify both major and minor defects in the components. Major defects detected must be addressed immediately to prevent adverse impact on product performance or safety. Timely intervention and corrective actions are essential to maintain quality standards and prevent defective products from reaching customers. Analyze the data collected, identify patterns, trends, and anomalies in defect occurrences over time. Also, prioritize the major defects, which must be addressed immediately to avoid any significant impact on the performance, ensuring product quality.

Algorithms –

- Machine learning algorithms, such as clustering, can be used for defect prediction and anomaly detection.
- Support Vector Machines (SVM), and Neural Networks can be used for predictive maintenance, quality control, tasks.
- Prioritization algorithms can help prioritize defects based on severity and impact on product quality.

W_UC2_3

Objective – The objective of this workload is to analyze and study pedestrian behavior in crowded indoor environments by deploying the crowdbot [2] dataset, and to understand mobility robot interactions with pedestrians. The crowdbot dataset captures outdoor pedestrian tracking using onboard sensors on a personal mobility robot named Qolo. Qolo is equipped with a reactive navigation control system and operates in shared-control or autonomous mode while navigating on the streets [3].

Data – Workload involves running the crowdbot dataset within a Kubernetes cluster, as shown in Figure 27. The workload also runs inside a container using the nginx image, and the dataset directory is mounted in the same nginx container, as shown below in Figure 28..

- The dataset includes data from the following sensors:
    - Frontal and rear 3D LIDAR (Velodyne VLP-16) at 20 Hz.
    - Frontal RGBD camera (Real Sense D435).
    - Force/Torque sensor (Botasys Rokubi 2.0) for contact sensing.
- The data covers over 250,000 frames of recordings in crowds with varying densities, more than 200 minutes of recording data.

- Each recording includes approximately 120 seconds of data in both rosbag[4] format (for Qolo's sensors) and npy[5] format.

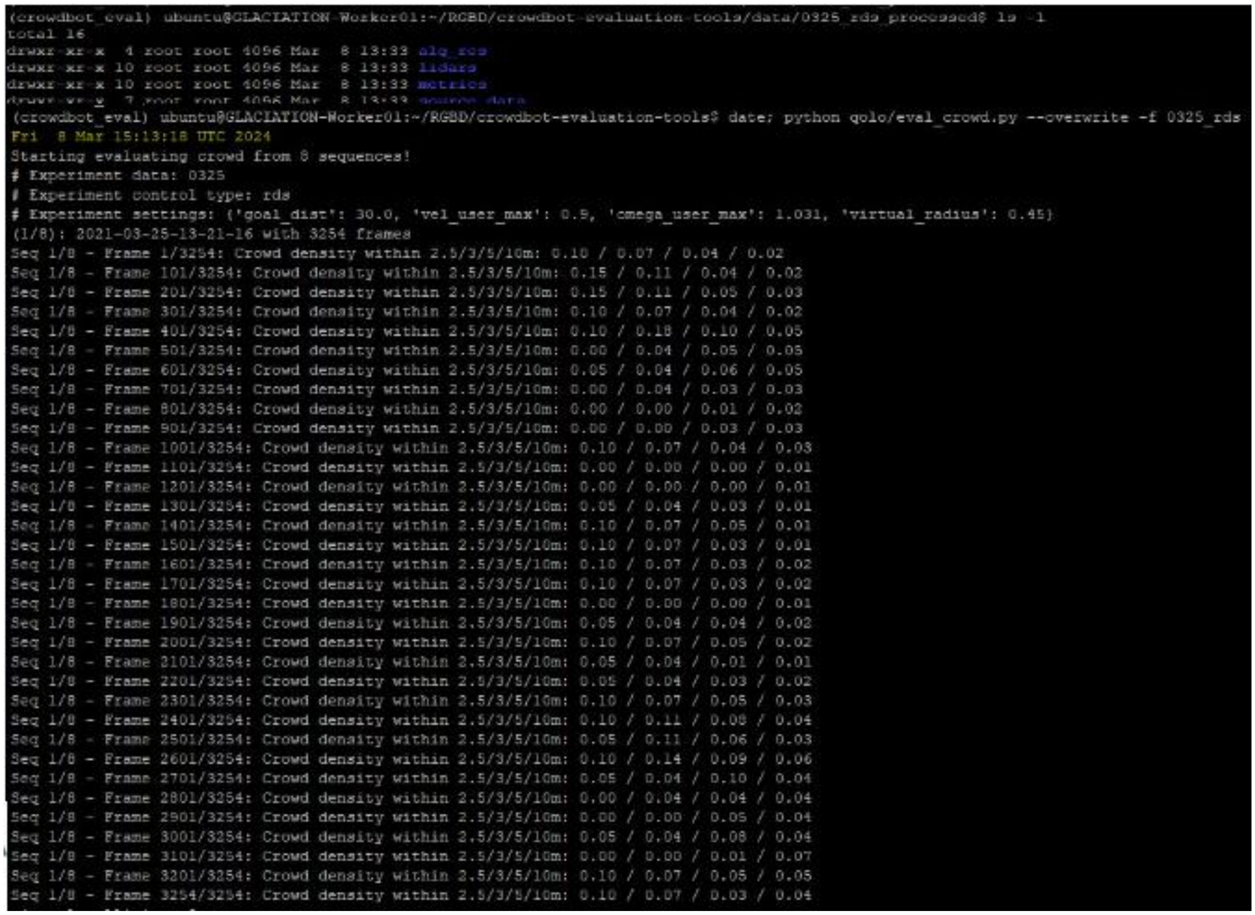
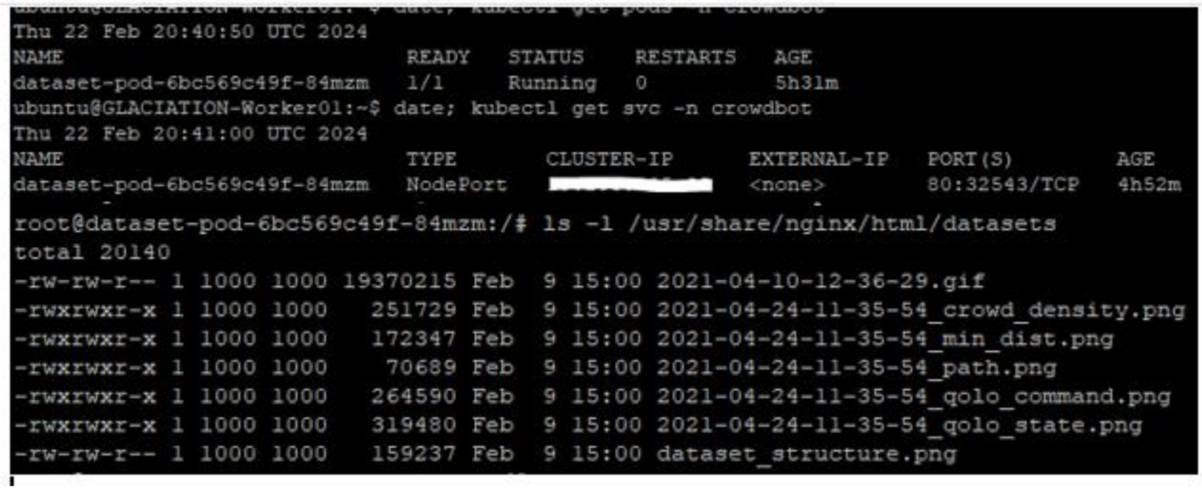


Figure 27: Crowdbot dataset-1

[4] A rosbag or bag is a file format in ROS for storing ROS message data. These bags are often created by subscribing to one or more ROS topics and storing the received message data in an efficient file structure. Rosbag can be activated from the command line or C++ or Python using the code API.

[5] NPY format is a simple format for saving numpy arrays to disk with the full information about them. The .npy format is the standard binary file format in NumPy for persisting a single arbitrary NumPy array on disk. The format stores all the shape and dtype information necessary to reconstruct the array correctly even on another machine with a different architecture. The format is designed to be as simple as possible while achieving its limited goals.



```
ubuntu@GLACIATION-Worker01:~$ date; kubectl get pods -n crowdbot
Thu 22 Feb 20:40:50 UTC 2024
NAME                            READY   STATUS    RESTARTS   AGE
dataset-pod-6bc569c49f-84mzm    1/1     Running   0          5h31m
ubuntu@GLACIATION-Worker01:~$ date; kubectl get svc -n crowdbot
Thu 22 Feb 20:41:00 UTC 2024
NAME                            TYPE       CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
dataset-pod-6bc569c49f-84mzm    NodePort   [redacted]   <none>        80:32543/TCP   4h52m
root@dataset-pod-6bc569c49f-84mzm:/# ls -l /usr/share/nginx/html/datasets
total 20140
-rw-rw-r-- 1 1000 1000 19370215 Feb  9 15:00 2021-04-10-12-36-29.gif
-rwxrwxr-x 1 1000 1000   251729 Feb  9 15:00 2021-04-24-11-35-54_crowd_density.png
-rwxrwxr-x 1 1000 1000   172347 Feb  9 15:00 2021-04-24-11-35-54_min_dist.png
-rwxrwxr-x 1 1000 1000    70689 Feb  9 15:00 2021-04-24-11-35-54_path.png
-rwxrwxr-x 1 1000 1000   264590 Feb  9 15:00 2021-04-24-11-35-54_qolo_command.png
-rwxrwxr-x 1 1000 1000   319480 Feb  9 15:00 2021-04-24-11-35-54_qolo_state.png
-rw-rw-r-- 1 1000 1000   159237 Feb  9 15:00 dataset_structure.png
```

Figure 28: Crowdbot dataset-2

Task – The main task is pedestrian detection and tracking in crowded environments. Additional tasks include estimating crowd density, robot navigation, assessing proximity to the robot, and evaluating the path efficiency of the robot controller. An example of a visual aspect of the dataset related to the crowd tracking and robot performance, can be accessed externally, as shown below in Figure 29..

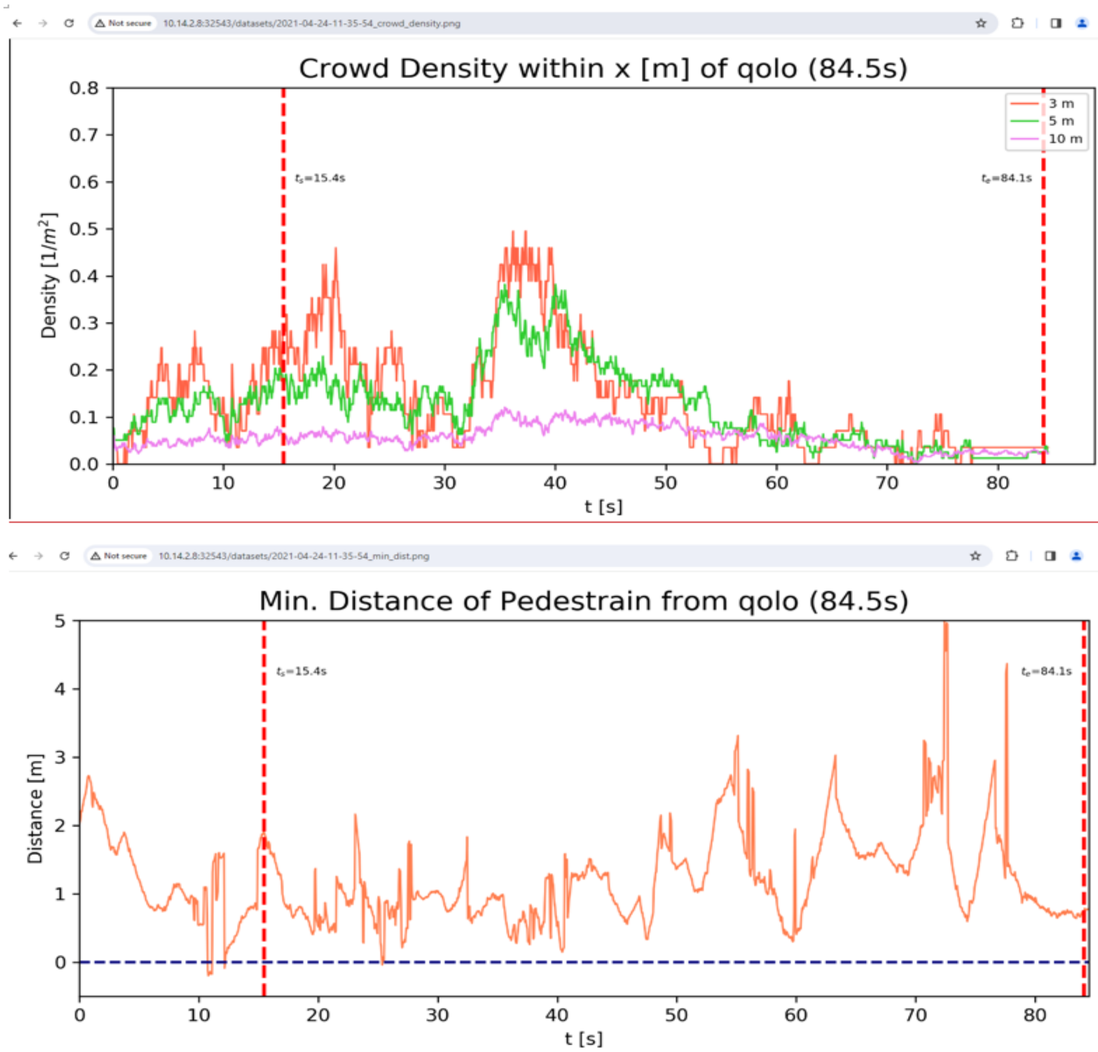


Figure 29: Visual aspect of the Dataset

Algorithms –

- DrSPAAM detector: Provides real-time people detection and tracking from onboard sensing.
- AB3DMOT: Labels 3D point clouds with people classes.
- Yolo: Outputs people detections.
- Tracker: Handles tracking tasks.
- Botasys Rokubi 2.0: Measures contact forces.
- Path efficiency metrics: Evaluate robot controller performance (time to goal, path length, virtual collisions).

5.5 UC3 Tests

We overview the main tests to ensure proper orchestration, integration, parameterization, demonstration and overall deployment and performance of the main component of UC3, i.e., the secure collaborative computation.

Test ID	VID Coverage	Blocker	Test Scenario	Main Components
T_UC3_1	VID_UC3_1, 5	Yes	Setup	Secure Collaborative Computation
T_UC3_2	VID_UC3_9	Yes	Integration	Secure Collaborative Computation
T_UC3_3	VID_UC3_4, 7	No	Parameterization	Secure Collaborative Computation
T_UC3_4	VID_UC3_11	No	Demonstration	Secure Collaborative Computation
T_UC3_5	VID_UC3_2, 6	No	Performance	Secure Collaborative Computation
T_UC3_6	VID_UC3_8	No	Deployment in business context	Secure Collaborative Computation
T_UC3_7	VID_UC3_12, 13	No	Accuracy	Secure Collaborative Computation

Table 19: UC3 Test Definitions

5.5.1 Test Description

Next, we provide a general description of the tests and the required aspects. As this deliverable is an intermediate report, we will improve and augment this description as needed in further deliverables to reflect the changes and alterations required to specify the tests. Notably, Carbyne Stack, one of the main technological enablers of UC3, is still an ongoing work with active development and changes are to be expected. For example, SAP is contributing to the usability of Carbyne Stack, aiming to provide simplified and automatic setup.

T_UC3_1 – Setup

This test covers the setup of the protection techniques, includes the following aspects: Setting up, starting, connecting secure collaboration parties in cloud environment.

Objective: ensure robust and reliable setup as a basis of the realization of UC3.

Test Case Workflow:

- Setup Kubernetes clusters
- Setup Carbyne Stack
- Test connectivity and functionality of compute/storage services of Carbyne Stack

Test Result:

- Setups successful
- Availability and connectivity of services

Note that the following tests assume the above-mentioned setups, if not noted otherwise, when discussing "Setup required components".

T_UC3_2 – Integration

The purpose of this test is to ensure that the UC3 components are effectively integrated in the platform.

Objective: ensure that the provided interfaces are sufficient for the requirements by testing interfaces for secure computation/collaboration component.

Test Case Workflow:

- Setup required components or use sufficient mock-ups of the associated services
- Test availability of services
- Test functionality of services by ensuring expected outcomes are provided when invoking them on pre-defined test sets with known outcomes

Test Result:

- Components successfully setup and availability ensured to allow their accessibility from the platform
- Components demonstrate effective integration and provisioning of services

T_UC3_3 – Parametrization

This test covers the parametrization of the components to be sufficient to realize UC3.

Objective: testing configurability and flexibility by checking support for required parameterizations to ensure generality of the solution

Test Case Workflow:

- Setup required components or use sufficient mock-ups of the associated services

- Test defined set of parameters to configure the components

Test Result:

- Flexible configuration and parametrization of components ensured

T_UC3_4 – Demonstration

This test is concerned with testing that the demonstration of the UC is as required.

Objective: demonstration of UC3 can be successfully run.

Test Case Workflow:

- Setup required components
- Test service availability and functionality
- Run demonstration on example workload

Test Result:

- Setup and service tests successful
- Demonstration running as expected

T_UC3_5 – Performance

This test covers performance and associated measurements.

Objective: ensure sufficient performance for UC requirements.

Test Case Workflow:

- Setup required components
- Test service availability and functionality
- Run performance measurements, measuring required KPIs and metrics, focusing on measuring computation time of the distributed collaboration

Test Result:

- Setup and service tests successful
- Performance measurements are within acceptable ranges to realize UC3, i.e., results available within minutes (for low latency networks, e.g., <5ms ping) or hours (for high-latency networks, e.g., 50ms)

T_UC3_6 – Deployment in business context

This test covers deployment in a business context.

Objective: deployment in business-suitable cloud environment provided by SAP Business Technology Platform.

Test Case Workflow:

- Configure setup for business context by ensuring BTP-suitable deployment method via Gardener
- Setup Kubernetes clusters
- Setup Carbyne Stack
- Test connectivity and functionality of compute/storage services of Carbyne Stack

Test Result:

- Setups successful in business context suitable for BTP deployment
- Availability and connectivity of services

T_UC3_7 – Accuracy

This test covers accuracy aspects of the analytics to be computed in UC3.

Objective: ensure sufficient accuracy to realize UC3.

Test Case Workflow:

- Setup sufficient mock-ups of the associated services. Notably, accuracy computation does not require full setup for encrypted computation as accuracy measurements on our used non-sensitive data can be done in clear.
- Split test data set to simulate distributed computation for different numbers of assumed input providers, i.e., data owners
- Run analytical computation with accuracy measurements per data owner, to simulate local processing without collaboration, as well as on their joint data, to simulate joint collaboration
- Compute error of the analytical computations

Test Result:

- Results for local and collaborative analytical processing available

- Collaborative analytical computation on joint data from multiple input providers improves upon local data from single providers by at least 5%

5.5.2 Workload Description

For UC3, the workload will cover analytical and statistical computations over small to medium sized data sets. Due to the potentially sensitive nature of business data, additional protection guarantees are required. Hence, UC3 leverages privacy-enhancing technologies. A prime example of privacy-enhancing technologies is secure multi-party computation where data remains encrypted even in use. UC3 focuses on a distributed setting, where data from multiple parties is used to perform a joint computation on the combined data. For the evaluation, we will use non-sensitive data to perform the tasks. For UC3, test data will be academic/public research data, e.g., from UCI ML repository. Specifically, the following workloads are relevant for UC3.

Workload W_UC3_1

Objective: the objective of workload W_UC3_1 is to optimize analytical tasks by incorporating multiple samples from a larger and potentially more diverse set of data sources to learn insights about typical usage and loads. For example, typical runtime or load of certain equipment, their typical operation conditions (say, temperature during processing, operational speed), throughput of production machinery, etc.

Task: collaborative analytics based on historic data from multiple parties

Data: the data will be of a generic numerical type, consisting, e.g., of a set of numbers. We will discuss a potential data set in W_UC3_2 that is also applicable for W_UC3_1.

Algorithms: aggregate statistics such as counting queries, minimum, maximum, median or other percentiles (aka *k*-th ranked element)

Workload W_UC3_2

Objective: the objective of workload W_UC3_2 is to improve forecasting and prediction results for industrial settings by incorporating data from multiple parties. For example, improving predictive maintenance by leveraging not only one's own data but data from users of the same asset type, e.g., production equipment.

Task: collaborative forecasting based on historic data from multiple parties
Data: the data will be of tabular form with numerical entries. Here, columns model different attributes and one target, and rows represent samples for these column-specified attributes and targets. In very general terms, during training, the goal is to construct a model that predicts the target in a way that the prediction minimizes an associated metric with regards to the actual target. During inference, the goal is to predict a target from a sample where only attribute values are provided, i.e., target is unknown.

For example, a potential data set that is publicly available is the "AI4I 2020 Predictive Maintenance Dataset" [4] where a subset of columns is presented below for illustration purposes:

UDI	Product ID	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Machine failure
1	M14860	298.1	308.6	1551	42.8	0	0
2	L47181	298.2	308.7	1408	46.3	3	0
3	L47182	298.1	308.5	1498	49.4	5	0
4	L47183	298.2	308.6	1433	39.5	7	0
5	L47184	298.2	308.7	1408	40.0	9	0
6	M14865	298.1	308.6	1425	41.9	11	0
7	L47186	298.1	308.6	1558	42.4	14	0
8	L47187	298.1	308.6	1527	40.2	16	0
9	M14868	298.3	308.7	1667	28.6	18	0

Figure 30: UC3 Example Data Structure

For predictive maintenance, possible attributes can be product id, air temperature, process temperature, rotational speed, torque, tool wear. And the target, to be predicted, is machine failure. Alternatively, for demand forecasting, the target to be predicted is the future demand of a quantity, say, spare parts of a machine or number of products one can expect to sell.

Algorithms: suitable and established machine learning models for industrial settings. Specifically, tree-based models such as regression trees. The reason for using tree-based models is two-fold. First, due to their high efficiency compared to machine learning models based on deep neural networks, which is especially important when using secure multi-party computation which incurs additional overhead as the collaborative computation is distributed and operates on encrypted data. Second, due to their high accuracy for such tasks where tree-based models remain state-of-the-art, mostly outperforming neural-network-based solutions in forecasting competitions [5] [6].

6 Future Work

The definition of validation strategy for each use case and next step towards the implementation process has shown satisfactory progress for all 3 use cases, at the midway point of our project (month 18). We expect steady progress in the upcoming months, aiming to meet the defined requirements of the platform within the specified timeframe.

All validation requirements and overall test methodology have been defined, certain aspects might be improved and subject to change (e.g., test description and details are subject to ongoing implementation) considering that this is an intermediate phase of the project.

The GLACIATION consortium has expanded with the inclusion of one more partner and use case, UC4. UC4 will play a pivotal role in the validation, integration, and evaluation processes for the use cases. In the project's second half, we anticipate completing validation efforts for all four use cases. Our goal is to deliver functional demonstrations of these use cases in the final report (deliverable D7.3) and a comprehensive platform with advanced components (deliverable D7.1) by project completion (month 36).

During the validation, we recognized that a predefined workflow covering all the practical demonstration requirements could become more complex. Exception handling may be required, which is currently not accounted for in the risk analysis for the development of each use case, respectively. In the next iteration, we will explore the ability for demonstrator use cases to support slight modifications as needed. Additionally, we will address any requirements that remain uncovered or partially addressed in this deliverable.

Finally, we aim to extend the demonstrator for the current platform, culminating in the generation of a comprehensive Validation and Demonstration Report.

7 Conclusion

Deliverable D7.2 “Use Case Integration, Validation, and Demonstration Report – Intermediate” outlines a comprehensive validation strategy of the use cases, integrating test cases and analysis based on Task 7.1 which focuses on Use Cases’ planning and definition of the methodological approach for validation.

We provided a detailed description of the functionalities and aspects of the validated use cases. Additionally, our test methodology assesses the efficiency and efficacy of the solution. For each use case, a detailed validation description is included enabling proactive decision-making and prevent potential issues.

The deliverable represents the initial outcome of WP7. It takes the GLACIATION architecture design from WP2 as input and defines validation aspects and test methodology for the use case. Additionally, it considers the requirements defined in Task 2.1 for each use case. These use cases serve to validate the GLACIATION platform in both real-life and simulated environments.

One of the key concerns of the GLACIATION platform is the protection of personal and sensitive information. We ensure compliance with relevant privacy laws and regulations such as GDPR and HIPAA through measures such as data anonymization, encryption, and secure data transfer. The platform adheres to industry best practices and standards for data privacy and security.

We identify assets from WP7 that can be transferred and play a pivotal role in future deliverables. During the second cycle of the project, we will delve deeper into the specifications of each of the use cases. The final report, Deliverable D7.3 is expected to be concluded by the end of the project (month 36), which will provide a comprehensive description of the validation strategy in its final form.

References

[1] "29119-1-2021 - ISO/IEC/IEEE International Standard - Software and systems engineering -- Software testing --Part 1:General concepts," 27 January 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9698145.

[2] Y. H. D. G. L. H. A. B. Diego Paez-Granados, "3D point cloud and RGBD of pedestrians in robot crowd navigation: detection and tracking," 27 January 2021. [Online]. Available: https://dx.doi.org/10.21227/ak77-d722.

[3] D. Paez-Granados, Y. He, D. Gonon, L. Huber and A. Billard, "Pedestrian-Robot Interactions on Autonomous Crowd Navigation: Reactive Control Methods and Evaluation Metrics," 03 August 2022. [Online]. Available: https://arxiv.org/pdf/2208.02121.pdf.

[4] S. Matzka, "Explainable Artificial Intelligence for Predictive Maintenance Applications," International Conference on Artificial Intelligence for Industries , 2020. [Online]. Available: https://archive.ics.uci.edu/dataset/601/ai4i+2020+predictive+maintenance+dataset.

[5] T. e. a. Januschowski, "Forecasting with trees," *International Journal of Forecasting 38.4,* 2022.

[6] Y. e. a. Gorishniy, "Revisiting deep learning models for tabular data," *Advances in Neural Information Processing Systems 34,* 2021.