TRANSPARENT APPLICATION DEPLOYMENT IN A SECURE, ACCELERATED AND COGNITIVE CLOUD CONTINUUM

Grant Agreement no. 101017168

Deliverable D6.4
Business, end user and technical evaluation

Programme:	H2020-ICT-2020-2
Project number:	101017168
Project acronym:	SERRANO
Start/End date:	01/01/2021 – 31/12/2023

Deliverable type:	Report
Related WP:	WP6
Responsible Editor:	INB
Due date:	31/08/2022
Actual submission date:	07/10/2022

Dissemination level:	Public
Revision:	Final

Revision History

Date	Editor	Status	Version	Changes
12.07.2022	INB	Draft	0.1	Initial version with Table of Contents
24.08.2022	INB	Draft	0.2	Integrated contributions from IDEKO, CC
26.08.2022	INB	Draft	0.3	INB input Section 4 and Section 2
29.08.2022	INB	Draft	0.4	IDEKO integrated and INB Section 2.1
29.08.2022	INB	Draft	0.5	INB input section 1
30.08.2022	INB	Draft	0.6	Integrated CC contribution and Section 6
06.09.2022	INB	Final	1.0	Final Document
07.10.2022	ICCS	Final	1.1	Review & submission

Author List

Organization	Author
INB	Maria Oikonomidou, Ferad Zyulkyarov
IDEKO	Julen Aperribay, Javier Martin
CC	Marton Sipos, Marcell Feher, Daniel Lucani

Internal Reviewers

Javier Martín (IDEKO)

Silviu Panica (UVT)

Abstract: This deliverable (D6.4) is the first of two reports that are scheduled to present the business, end user and technical evaluation outcomes of the tasks 6.3, 6.4 and 6.5. It presents the initial deployment and integration of the three SERRANO use cases (i.e., Secure Storage, FinTech Analysis, and Anomaly Detection in Manufacturing Settings) into the SERRANO platform. Since the start of WP6 (M13), the use case partners have been preparing and adapting their use case applications to leverage the SERRANO platform advancements, while performing preliminary integration and deployment activities. At this stage of the project, this initial report does not provide actual technical and business evaluation results, since the respective developments are on progress. The final and detailed performance and production evaluation of the use case applications within the SERRANO platform will be reported in deliverable 6.8 "Final version of business, end user and technical evaluation" (M36).

Keywords: SERRANO platform, integrated components, development environment, integration environment, software deployment, verification, validation

Table of Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence
API	Abstract Programming Interface
ARDIA	A Resource reference model for Data-Intensive Applications
BOM	Bill of Materials
CI/CD	Continuous Integration / Continuous Development
CPU	Central Processing Unit
D	Deliverable
DevSecOps	Development, Security, and Operations
DL	Deep Learning
DPO	Dynamic Portfolio Optimization
DPU	Data Processing Unit
DTW	Dynamic Time Warping
EFT	Electronic Funds Transfer
ETF	Exchange-Traded Fund
ETL	Extract, Transform, Load
FFT	Fast Fourier transform
FPGA	Field-Programmable Gate Array
GDPR	General Data Protection Regulation
HPC	High Performance Computing
HW	Hardware
IO	Input Output
IoT	Internet of Things
JSON	JavaScript Object Notation
K8S	Kubernetes
KNN	K-Nearest Neighbors algorithm
KPI	Key Performance Indicators
ML	Machine Learning
MQTT	MQ (IBM MQ) Telemetry Transport
NIC	Network Interface Controller
PCIe	Peripheral Component Interconnect express
REST	Representational State Transfer
ROT	Resource Orchestration Toolkit
SaaS	Software as a Service
SDK	Service Development Kit
TLS	Transport Layer Security
UC	Use Case
YAML	YAML Ain't Markup Language

1 Executive Summary

Deliverable D6.4 "Business, end user and technical evaluation" reports on the initial integrations and testing for each of the three SERRANO use cases. The deliverable presents the use case developments advances on 1) the Secure Storage Use Case that aims at demonstrating the envisioned capabilities of the SERRANO platform in the context of secure file sharing and storage, 2) the Fintech Use Case that leverage the cloud continuum capabilities of the SERRANO project within the context of investment portfolio management, representing automated management of investment portfolios, and 3) the Anomaly Detection in Manufacturing Settings Use Case that aims at predictive maintenance, remaining lifetime assessment and diagnosis of critical machine elements by proposing an approach where data analysis is performed continuously on the SERRANO platform.

In this deliverable, the use cases focus on describing their successful deployment, integration and testing on the initial release of the SERRANO platform. The initial SERRANO platform prototype provides a partial implementation of SERRANO components that correspond to the first development iteration (M7-M18). Technical details about the initial version of the SERRANO platform are reported on D6.3 (M18). The final and detailed performance and production evaluation of the use case applications will be reported in D6.8 "Final version of business, end user and technical evaluation" (M36).

2 Introduction

2.1 Purpose of this document

The purpose of this deliverable is to report on the integration of the SERRANO use cases Secure Storage (Task 6.3), FinTech (Task 6.4), and Anomaly Detection (Task 6.5) into the SERRANO platform. Since the start of WP6 (M13), the use case partners have been preparing and adapting their use case applications to leverage the SERRANO platform by preparing the applications' components to be deployed, executed, and monitored by the SERRANO platform. The document reports their experience and provides feedback about the integration process. The feedback includes the description and the methodology of the integration process, the application development experience when interacting with the SERRANO platform, the benefits and challenges of developing applications for the SERRANO platform, and the business implications such as the advantages and disadvantages that the current version of the SERRANO platform provides for developing, and orchestrating robust business applications. Since most of the effort focused on adapting and migrating the use case applications, this document also reports the successful deployment and execution of unit tests and integration tests for the use case applications into the SERRANO platform.

2.2 Document structure

The present deliverable is split into six major sections:

- The first section is an executive summary.
- The second section serves as an introduction and describes the main goals of the deliverable.
- The third section is dedicated to Secure Storage Use Case (CC) and describes the business, end user and technical evaluation.
- The fourth section is dedicated to FinTech Use Case (INB) and describes the business, end user and technical evaluation.
- The fifth section is dedicated to Anomaly Detection in Manufacturing Settings Use Case (IDEKO) and describes the business, end user and technical evaluation.
- The sixth section briefly summarizes the evaluation of all Use Cases.

2.3 Audience

The deliverable is public and available to anyone interested in the first release of the SERRANO integrated platform and the SERRANO use cases. Moreover, this document can also be useful to the general public for a better understanding of the framework and scope of the SERRANO project.

3 Secure Storage Use Case

The Secure Storage Use Case highlights the SERRANO integrated platform's storage capabilities. Beyond the actual storage of files, it uses the acceleration capabilities developed as part of the project to improve performance and the intelligent orchestration features to meet user requirements closely.

3.1 Use case description

This UC focuses on providing secure, high-performance storage and sharing of files with lower latency than a purely cloud-based approach. It achieves this goal by extending Chocolate Cloud's commercial SkyFlok[1] multi-cloud distributed storage service with on-premises edge devices that can act as storage locations. Medium and large businesses (250+ employees) that are SkyFlok customers would like to extend their use of Chocolate Cloud's SkyFlok service. SkyFlok works great for file-based collaboration and file sharing workflows as well as for data archival purposes.

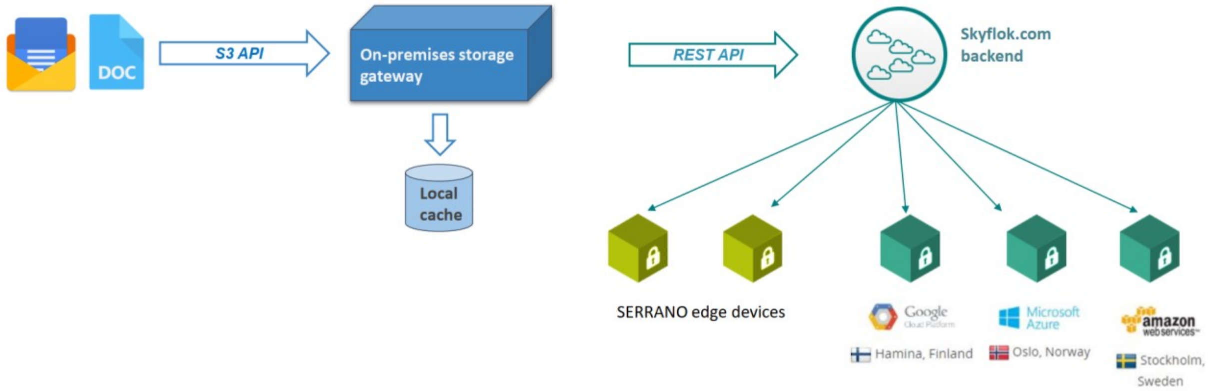

Figure 1: Secure Storage UC components.

However, given its fully cloud-based architecture, it lags behind in terms of latency compared to an on-premises storage solution. Moving data closer to the edge can significantly improve download and upload latencies. While many customers choose SkyFlok over competing solutions thanks to the privacy guarantees it offers, privacy concerns remain a major impediment to the more wide-spread adoption of cloud storage in general. Due to legal requirements or internal policies, enterprises want strong guarantees that their data cannot be accessed by third parties, including the storage provider. Conventional cloud storage can only achieve this to a limited degree. Moving file encryption/decryption process on premises, under full control of the enterprise, is key to providing these guarantees. To make it even more appealing to enterprise customers, files are accessed through an S3-compatible API. This makes it very easy to interact with the storage service, removing one of the obstacles usually

[1] https://www.Skyflok.com

faced by companies when they want to migrate from an existing storage solution to a new one.

3.2 Integration

Most of the features showcased by the UC are provided by the **Secure Storage Service** (also referred to as the SERRANO-enhanced Storage Service in other deliverables), an overview is shown in Figure 1. The service provides an S3 interface to access data distributed to storage locations across the edge-cloud continuum. This objective is crucial to meeting one of the SERRANO platform's objectives of being able to deploy applications across the edge-cloud continuum seamlessly. By deploying a distributed storage solution on the same (or similar) infrastructure as the platform's user applications, performance can be improved. Furthermore, the S3 interface provides a convenient way for existing enterprise applications to connect to the SERRANO platform's Secure Storage Service.

An additional objective for the Secure Storage Service is to support automated storage policy creation based on significantly varying storage task requirements. This objective is linked to the first and fifth objectives of the SERRANO platform. Firstly, the UC will showcase how the Secure Storage Service exposes its storage locations as parts of the platform's resources. Secondly, it will illustrate how a storage policy is created based on the intents of a user application. To demonstrate the benefits of this operation, significantly different storage tasks will be employed. Providing good read and write performance to many concurrent clients is another objective. This objective is closely tied to the previous two. It can be interpreted to some degree as one of the main underlying motivations behind distributing data across the edge-cloud continuum based on stated application requirements. There is also significant alignment with objective 3 of the SERRANO platform because a key element in achieving good performance is the acceleration of data processing operations.

To achieve these objectives, the UC integrates the functionality of many platform components. It relies on the **AI-enhanced Service orchestrator** to deploy the services associated with the edge storage locations and the gateway. The same component is, in turn, used by the UC's users. The **Secure Storage Service** exposes information about cloud and edge locations (status, availability, cost, latency, etc.) to the **SERRANO Telemetry Framework**. This integration is the first step to ensuring that each storage task is placed on the appropriate storage resources/site. Based on it, storage policies can be created automatically using the **Resource Optimization Toolkit** and the formally described requirements of each storage task (through the **ARDIA Framework**). The **Resource Orchestrator & Orchestration Drivers** and the **Local Resource Monitoring** component handle the actual allocation and monitoring of cloud and edge storage locations.

To provide low-latency access to files for a large number of concurrent users, the designed solution will use the **acceleration of encrypted storage using DPU/CTX** (acceleration of TLS connection encryption) by leveraging NVidia Bluefield cards, when available. This will reduce some of the load on the CPU and may increase the number of concurrent supported connections. It will use **FPGA-enabled HW acceleration** and **GPU-enabled HW acceleration** for encryption, erasure coding, and potentially compression using SERRANO tools and services developed in WP4. This includes the **Plug&Chip** API indirectly and should lead to further CPU offloading as well as a reduction of processing time.

3.3 Evaluation

The first use case validates the correct functioning of its components using a set of integration tests performed in the project's CI/CD pipeline. The pipeline was set up by INTRA as part of Work Package 6. At the time of writing this document, the tests cover the storage functionality. They will be expanded to cover the integration with the other platform features and services during the following year. In addition, a set of unit tests is planned for covering endpoints related to storage policies and cloud telemetry. The evaluation methodology centres on measuring 7 different KPIs and comparing the results to the estimated target values.

3.3.1 Testing

Testing is performed using a Jenkins pipeline, shown in Figure 2. It comprises of several stages. The first stage retrieves the source code of the On-premises Storage Gateway. The second stage installs the requirements, a set of python modules. The Unit tests stage is currently empty as S3 endpoints are covered by the unit tests. Each endpoint is the result of integration between the Gateway and several Skyflok.com backend microservices. A set of unit tests will be added to cover the caching functionality on the Gateway, as soon as this is implemented. Following this, Sonarqube is used to scan for vulnerabilities, then a BOM is generated to keep track of dependencies. Finally, a container image is built to run the Gateway. The image is deployed to the INTRA Kubernetes (K8s) cluster, where the integration tests are run using a set of python scripts and the unittest[2] framework. The container is then stopped and the image purged from the cluster. If the tests pass, the image is deployed into the UVT K8s cluster, where it is accessible to enable the integration efforts of all SERRANO partners.

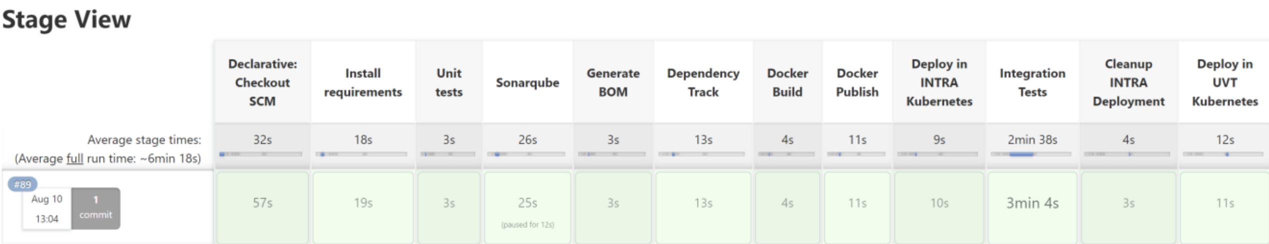

Figure 2: Overview of the Jenkins pipeline for the On-premises Storage Gateway

A total of 44 integration tests have been defined, grouped by the supported S3 endpoints. These are run using the Jenkins pipeline. Table 1 shows the output of the Integration test stage.

[2] Python unittest framework: https://docs.python.org/3/library/unittest.html

Table 1: Output of integration tests for On-premises Storage Gateway

```
test_10_buckets_created (test_s3_create_bucket.TestCreateBucket) ... ok
test_bogus_storage_policy_name (test_s3_create_bucket.TestCreateBucket) ... ok
test_correct_params (test_s3_create_bucket.TestCreateBucket) ... ok
test_duplicate_bucket_name (test_s3_create_bucket.TestCreateBucket) ... ok
test_missing_storage_policy_name (test_s3_create_bucket.TestCreateBucket) ... ok
test_correct_params (test_s3_delete_bucket.TestDeleteBucket) ... ok
test_non_empty_bucket (test_s3_delete_bucket.TestDeleteBucket) ... ok
test_non_existent_bucket (test_s3_delete_bucket.TestDeleteBucket) ... ok
test_correct_params (test_s3_delete_object.TestDeleteObject) ... ok
test_delete_and_put (test_s3_delete_object.TestDeleteObject) ... ok
test_non_existent_bucket (test_s3_delete_object.TestDeleteObject) ... ok
test_non_existent_object (test_s3_delete_object.TestDeleteObject) ... ok
test_overwriten_object (test_s3_delete_object.TestDeleteObject) ... ok
test_zero_byte_object (test_s3_delete_object.TestDeleteObject) ... ok
test_correct_params (test_s3_get_object.TestGetObject) ... ok
test_etag (test_s3_get_object.TestGetObject) ... ok
test_head_object (test_s3_get_object.TestGetObject) ... ok
test_headers (test_s3_get_object.TestGetObject) ... ok
test_non_existent_bucket (test_s3_get_object.TestGetObject) ... ok
test_non_existent_object (test_s3_get_object.TestGetObject) ... ok
test_with_actual_data_2_generations (test_s3_get_object.TestGetObject) ... ok
test_with_actual_data_5k (test_s3_get_object.TestGetObject) ... ok
test_check_new_bucket_shows_ups (test_s3_list_bucket.TestListBuckets) ... ok
test_check_response_body (test_s3_list_bucket.TestListBuckets) ... ok
test_bucket_with_object (test_s3_list_objects.TestListObjects) ... ok
test_bucket_with_three_objects (test_s3_list_objects.TestListObjects) ... ok
test_delim_no_delim (test_s3_list_objects.TestListObjects) ... ok
test_delim_no_prefix (test_s3_list_objects.TestListObjects) ... ok
test_delim_with_prefix (test_s3_list_objects.TestListObjects) ... ok
test_delim_with_prefix_object_only (test_s3_list_objects.TestListObjects) ... ok
test_delim_with_prefix_subfolder_and_object (test_s3_list_objects.TestListObjects)
... ok
test_empty_bucket (test_s3_list_objects.TestListObjects) ... ok
test_etag (test_s3_list_objects.TestListObjects) ... ok
test_non_existent_bucket (test_s3_list_objects.TestListObjects) ... ok
test_prefix_empty_string (test_s3_list_objects.TestListObjects) ... ok
test_prefix_no_match (test_s3_list_objects.TestListObjects) ... ok
test_prefix_with_match_full_object (test_s3_list_objects.TestListObjects) ... ok
test_prefix_with_match_single_char (test_s3_list_objects.TestListObjects) ... ok
test_prefix_with_three_objects (test_s3_list_objects.TestListObjects) ... ok
test_correct_params (test_s3_put_object.TestPutObject) ... ok
test_etag_returned (test_s3_put_object.TestPutObject) ... ok
test_non_existent_bucket (test_s3_put_object.TestPutObject) ... ok
test_overwrite (test_s3_put_object.TestPutObject) ... ok
test_zero_byte_object (test_s3_put_object.TestPutObject) ... ok

----------------------------------------------------------------------
Ran 44 tests in 170.732s
```

In the future, the test suite will be expanded to cover any new S3 features that will be added. Furthermore, tests for endpoints related to storage policies and telemetry will be defined. Finally, the different integrated SERRANO components will be tested using complex scenarios.

3.3.2 Methodology

The evaluation methodology for the technical outcomes of the SERRANO projects has been described in detail in D6.2. In short, the project collected and analyzed the SERRANO platform's requirements as part of Work Package 2. It established a set of good practices for requirement definition and worked on refining them in a two-stage process. The first stage was documented in Deliverable 2.2, with an update presented in Deliverable 2.4.

The three Use Case (UC) providers were asked to extensively describe their future work, focusing on the goals and requirements of each. The UC partners were also involved in discussions with the leaders of the different technical areas covered by the project. To formalize this process, 7 topics were identified based on the distinct areas that the project covers. This led to a very detailed and inclusive set of technical requirements for each group, taking advantage of the technical knowledge of the participating partners and the domain-specific knowledge of the UC providers.

A set of KPIs was identified, some directly linked to each use case, some to the outcomes of the 7 technical groups. Table 2 presents the KPIs associated with the Secure Storage Use Case. In addition, an alternative evaluation method was defined when a requirement's achievement is difficult to measure using a KPI accurately.

Table 2: UC1 technical success criteria of the Secure Storage Use Case

KPI	Success criterion	Estimated target value
Read and write latency reduction with respect to existing cloud locations	Successful integration of edge devices into the SkyFlok and SERRANO ecosystem with the goal of reducing latency.	Reduction of 10 - 50%
Number of applications using the service simultaneously	Demonstration of client applications storing data in the edge/cloud infrastructure using S3 REST API.	20 instances
Reduction in time taken to encode and decode data with respect to a CPU-based solution	Demonstration of GPU- and FPGA- accelerated RLNC encoding and decoding algorithms running on the on-premises storage gateway.	Reduction of 20-30%
Reduction in time taken to encrypt and decrypt data with respect to a CPU-based solution	Demonstration of GPU- or FPGA-accelerated AES-GCM encryption and decryption algorithms running on the on-premises storage gateway.	Reduction of 20-30%

Reduction in time taken to compress and decompress data with respect to a CPU-based solution	Demonstration of GPU- or FPGA-accelerated AES-GCM compression and decompression algorithms running on the on-premises storage gateway.	Reduction of 20-30%
Reduction in CPU load associated with encryption for TLS connections with respect to no hardware acceleration	Using DPU-based hardware acceleration for encryption of TLS connections on the on-premise storage gateway.	Reduction of 10-20%
Storage task execution that involves the creation of a new storage policy without intervention from the user	Transparent operation with regard to the choice of storage locations. Each user application that issues a storage task should state its requirements. The SERRANO resource orchestrator should create/assign a storage policy automatically.	Demonstration successful

The partners who defined the KPIs will measure the KPIs and perform the alternative evaluation methods. In addition, this phase is driven to a large degree by the use cases and a series of demonstrator applications. Therefore, most KPIs will be measured using the integrated SERRANO components, deployed in such a way that they accomplish the goals of a real-world application, represented by one of the use cases.

To this end, several demonstrator applications will be developed. For this use case, CC created an initial demo in time for the mid-project review. It revolves around using a standard S3 client to store and share files using the Secure Storage Service. More details on the technical implementation are described in Section 6.2 of Deliverable 6.2.

3.4 Results and outcome

It is too early in the development of the SERRANO components and their integration to describe all the results and outcomes of the Secure Storage Use Case. As such, we present our expectations and plans in this initial version. Detailed results and outcomes will be detailed, in the final version, namely D6.8.

3.4.1 Technical results

The technical outcomes of the use case centre around the components developed in Task 3.2, namely the On-premises Storage Gateway and the SERRANO edge devices. These will be the core enablers of offering S3-compatible storage that seamlessly blends cloud and edge storage in a single solution. This is augmented by the components developed in Task 3.1, accelerating TLS compression; in Work Package 4, accelerating the main data processing steps; in Work

Package 5, matching user intents with intelligent orchestration solutions, storage deployments, and telemetry.

At the time of this deliverable's submission, the Gateway's core feature set and its integration points with other components are implemented. In the following months, the actual integration will take place, including support for the SERRANO edge devices.

3.4.2 Business results

Chocolate Cloud aims to include edge storage and data sharing to its customers, particularly, with interest in critical applications. CC will integrate the outcomes of the SERRANO project into its Skyflok.com backend to open these new capabilities to companies building applications (via a public S3 REST API developed in the project) and solutions on their backend, but also as part of the optimization for the current SkyFlok file sharing services. This is particularly interesting for CC in its push towards approaching medium to large enterprises with a tailored solution that offers the best of cloud and edge storage.

Our goal is to mature existing technology in the context of edge storage and computing to be able to deliver it as part of our commercial solutions within 24 months after the completion of the project. Secure storage and secure sharing/transfer of data for enterprises is a market set to grow from 2,76 Billion to 10,96 Billion USD in 2023, according to Markets &Markets reports[3]. Disruptive offerings that provide high security and high performance are key to serving the enterprise market's demands. SERRANO is key for CC's strategy to address these demands.

3.4.3 End user feedback

There is potential for Chocolate Cloud to reach out to a selected group of SkyFlok users to ask for feedback on the usability of the solutions developed as part of SERRANO and in particular this use case. This aligns with our expected business result of providing an S3-based object storage solution built around the Skyflok.com backend. The feedback will be evaluated after the end of the project and used to improve our product portfolio.

[3] "Enterprise File Synchronization and Sharing (EFSS) Market by Component (Standalone EFSS Solution, Integrated EFSS Solution, & Services), Deployment Type (Cloud & On-Premises), Organization Size, Industry Vertical, and Region - Global Forecast to 2023," Available online:

https://www.marketsandmarkets.com/Market-Reports/enterprise-file-sharing-and-synchronization-market-149308334.html

4 FinTech Use Case

The Fintech UC demonstrates the automatic optimization of InbestMe's dynamic investment management. The SERRANO project will contribute to the investment management use case by accelerating critical parts of the overall Dynamic Portfolio Optimization (DPO) application and providing a framework that will simplify the deployment, management, operation, and monitoring. Additionally, for the provision of investment management as a service (SaaS), the UC will benefit from the secure storage extension that will keep the data of third parties secure. The UC will also demonstrate the advantages of cloud-based acceleration of various computationally intensive operations. The SERRANO platform will also be beneficial for InbestMe because it will reduce cloud costs and improve the quality of the services. It will be able to easily deploy multiple instances of the investment management platform on local as well as external cloud resources.

4.1 Use case description

This UC aims to demonstrate the cloud continuum capabilities of the SERRANO platform within the context of dynamic investment portfolio optimization. The UC is based on microservices architecture, while leveraging the project developments for transparent deployment on the cloud as well as edge resources that include FPGA or GPU accelerators.

Portfolio optimization (Figure 3) is a very representative and demanding investment management process that can benefit from SERRANO. It starts by getting the market data required for the analysis that are analysed by applying a set of technical calculations. Subsequently, forecasting algorithms and various investment strategies are applied, in parallel, in the investment instruments. The output from the forecasting and the investment strategies is used for creating new investment profiles. The investment profiles are again analysed by applying forecasting methods and back testing. Finally, the investment profiles are rebalanced to match the expected distribution of the investment profiles, but this component is not part of the SERRANO project.

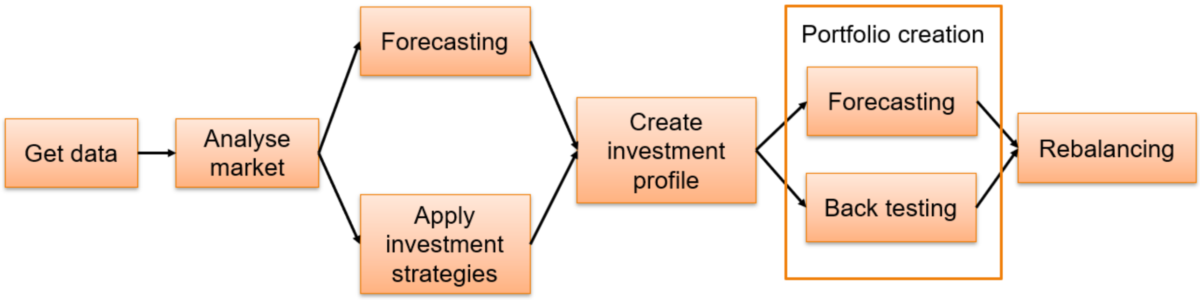

Figure 3: Portfolio optimization workflow.

4.2 Integration

For the Dynamic Portfolio Optimization, the developed services will be demonstrated in a container-based application on the SERRANO platform through the SERRANO SDK. For this application, the data needed are investment profile assets and their market data, which are their historical price data. These data will be stored in Skyflok provided by Chocolate Cloud **(Secure Storage Service)**. The SERRANO platform will be triggered through an API request to start the DPO application. Through this request, the market and investment profile data will be loaded to the SERRANO platform, and the DPO will be executed **(Secure & trusted execution).** The resulting asset profile optimizations will be downloaded locally when the execution is completed. In addition, for the DPO, we will leverage the available acceleration mechanism interfaces and HPC system hardware interfaces to enable their access to the SERRANO-enhanced computational resources, to ensure better performance and optimization of the kernels used in the services **(FPGA-enabled HW, GPU-enabled HW, and Approximate HW accelerator)**.

The DPO currently exposes only one REST method.

```
POST /portfolio_optimization
```

It takes as input a JSON object. The input is described in the textbox below.

```
Input JSON
{
   StartDate: DateTime
   EndDate: DateTime
   Url1ToSkyFlokStorage: String
   Url2ToSkyFlokStorage: String
}

- StartDate:                Historical Asset Data look back date
- EndDate:                  Historical Asset Data look up to date
- Url1ToSkyFlokStorage:     URL to Historical Asset Data file stored in SkyFlock cloud storage
- Url2ToSkyFlokStorage:     URL to Investment Profile Asset Data file stored in SkyFlock cloud
storage
```

4.3 Evaluation

The DPO application with the API has not yet been tested on the SERRANO platform. In the following sections, we describe the current and expected results for the methodology, testing, and outcome.

4.3.1 Testing

The Fintech UC will perform testing with the use of a Jenkins pipeline, which comprises several stages. Initially, a set of python module requirements will be installed. Next is the Unit tests stage. Following this will be Sonarqube, which will be used to scan for vulnerabilities, then a BOM will be generated to be able to keep track of dependencies. Next, the appropriate container image will be built to run the DPO application. The image is deployed to the INTRA K8s cluster, where the integration tests will run using a set of python scripts. The container then will be then stopped and the image will be purged from the cluster. If the tests pass, the image will be deployed into the UVT K8s cluster, where it is accessible to enable the integration efforts of all SERRANO partners.

4.3.2 Methodology

The evaluation methodology for the technical and business outcomes of the SERRANO projects has been described in detail in D6.2 through several KPIs. The KPIs were selected based on factors that affect the business satisfaction and technical requirements. For the business success criteria first, we want to reduce cloud costs by at least 50% by deploying a hybrid cloud infrastructure. Furthermore, we care to increase at least 10% of the portfolio performance, which is to maximize the returns and minimize the risks which are correlated to the improvement of accuracy in forecasting and predictions of the market. Regarding the technical success criteria, we want to successfully mitigate our applications and system to cloud-based containers. Additionally, we target to deploy independent cloud-based instances of our system for third parties. Also, it is very important to be able to rate markets and portfolios by continuous analysis of them. Moreover, technical success will be considered to create real-time orders using live market prices.

Table 3: UC2 business success criteria of Dynamic Portfolio Optimization.

KPI	Success criterion	Estimated target value
Percentage of cloud costs reduction	Reduce cloud costs by deploying a hybrid cloud infrastructure	Reduced by 50% or more
Percentage of portfolio performance increase	Increase portfolio performance (return/risk)	Increased by 10% or more
Percentage of improvement	Improve accuracy of forecasting and prediction	Improved by 10% or more

Table 4: UC2 technical success criteria of Dynamic Portfolio Optimization.

KPI	Success criterion	Estimated target value
Conversion/adaptation to cloud-based containers	Convert/adapt the INB applications and system to cloud-based containers	Conversion/adaptation successful
Independent instances deployment	Deploy independent cloud-based instances of the INB system for third parties	Deployment successful
Real-time orders creation	Create real-time orders using live prices	Real-time orders creation successful

Rate of market analysis	Continuous market analysis	100 financial assets per hour or more
Rate of portfolio analysis	Continuous portfolio analysis	100 portfolios per hour or more

4.4 Results and outcome

The business and technical results will be presented when the deployment of the proposed DPO application is fully integrated into the SERRANO platform. The following sections describe the expected results, which have not been validated yet.

4.4.1 Technical results

The first expected technical result for the Fintech UC will be to successfully mitigate the DPO application and system to cloud-based containers. With the SERRANO platform, it will be possible to implement the high demand in terms of efficiency and complexity of the DPO application. Additionally, we care to deploy independent cloud-based instances of our system for third parties. Also is very important to be able to rate markets and portfolios by continuous analysis of them.

4.4.2 Business results

Inbestme UC represents an automated management of investment portfolios. Investment management is a continuous process of constructing investment portfolios composed of investment instruments such as shares, ETFs, funds, options, etc. With the SERRANO platform InbestMe will reduce cloud costs and improve the quality of services by improving local and cloud resource utilization. By achieving this objective, InbestMe will be able to easily deploy multiple instances of its investment management platform on local as well as external cloud resources. In this way, it will be able to cut significant costs on cloud services. Furthermore, SERRANO integration will facilitate the process of construction portfolios with lower risk and higher returns through more accurate analysis. By achieving this objective, InbestMe will be able to analyse more information and implement prediction and forecasting algorithms with higher precision and accuracy. As a result, it will be able to achieve better returns at lower risks for its clients' investment portfolios.

4.4.3 End user feedback

Since the Fintech UC has not yet fully integrated into the SERRANO platform, we cannot provide feedback regarding the advantages and challenges of using the platform. The integration with SERRANO and the respective evaluation analysis will be detailed in D6.8 after the release of the final version of the SERRANO platform. Feedback and insights from the experience of the integration with the SERRANO platform will be evaluated at the end of the project.

5 Anomaly Detection in Manufacturing Settings Use Case

Downtimes of failed devices in an industrial plant must be kept to a minimum to achieve high system availability. In the very competitive manufacturing world, getting the most out of the machine may be the difference between being competitive or not. Thus, and mainly after the irruption of the Industry 4.0 paradigm, many techniques and methods are being widely applied to meet a simple yet complex goal: keep the machine working most of the time.

Companies that manufacture expensive high added-value parts are very demanding regarding machine availability and quality assurance. As a result, predictive maintenance, remaining lifetime assessment, and diagnosis of critical machine elements are state-of-the-art practices. However, some techniques that are used require the machine to stop before performing the analysis.

The use case will leverage the SERRANO platform to change the state-of-the-art approach and to perform machine component status assessment without stopping the machine. This UC aims to develop a system able to detect anomalies by processing the amount of data generated in real-time by high-frequency sensors. To this end, the use case builds on the components developed in the SERRANO platform, such as acceleration mechanisms, secure storage or data broker, among others.

5.1 Use case description

This UC proposes a deployment approach where data analysis is performed continuously while the hardware equipment keeps running most of the time, and the state of the various independent components, along with the overall status, is continuously reported. Moreover, the UC will focus on a single critical component, ball screws. A ball screw is a mechanical linear actuator that transfers rotational motion to linear motion, moving the machine on the x and y-axis. Ball screws are expensive and critical machine components whose breakage implies stopping the machine for a significant time. Each machine has two ball screws, x and y axis.

The use case is developing a Data Processing Application to analyse real-time signals from the ball-screw sensors and check for anomalies, detecting anomalous behaviours that may affect the part quality and predict imminent failures. This application has been divided into two different services that analyse the data coming from the position sensors and the data from the acceleration sensors (Position Processor service and Acceleration Processor service) of the ball screw. These services are based on AI algorithms and techniques.

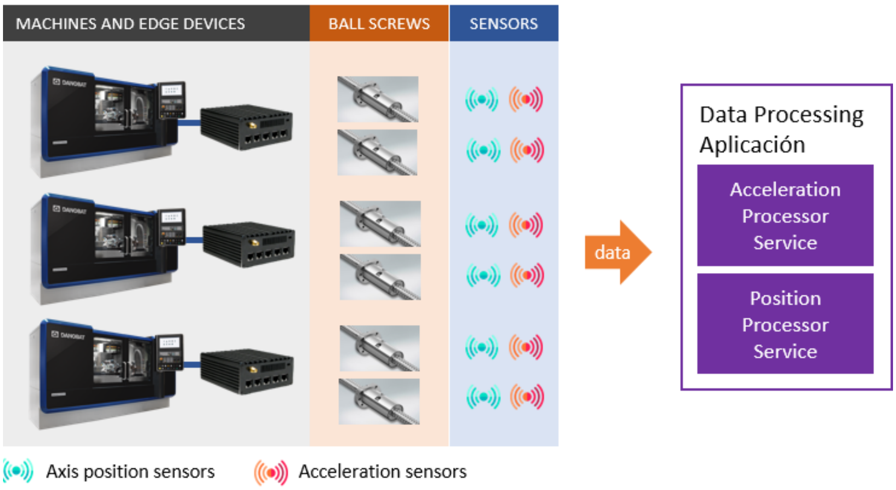

Figure 4: Data Processing Application to analyse real-time signals from ball-screw sensors.

In addition, to obtain data from real machines at IDEKO's facilities, a test bench has been built with two sensorized ball screws (X and Y axis), simulating data from machines in a real-production scenario. The generated data will be sent to SERRANO to be analysed in order to detect anomalies through the applications/services (Data Processing Application) developed internally.

The data generated in the machine is limited since it is generally stored at the edge, and the resources at the edge are limited, so data requests are made on demand. For this reason, this UC will leverage the SERRANO platform to change the state-of-the-art approach and to perform the ball-screws' status assessment without stopping the machine. Hence, the SERRANO platform will enable the transition from the current on-demand analysis due to lack of resources at the edge to real time data analysis. Furthermore, the new approach will support processing high data volume at the edge-cloud continuum, using any resource available to keep availability and performing anomaly detection over a non-controlled machine.

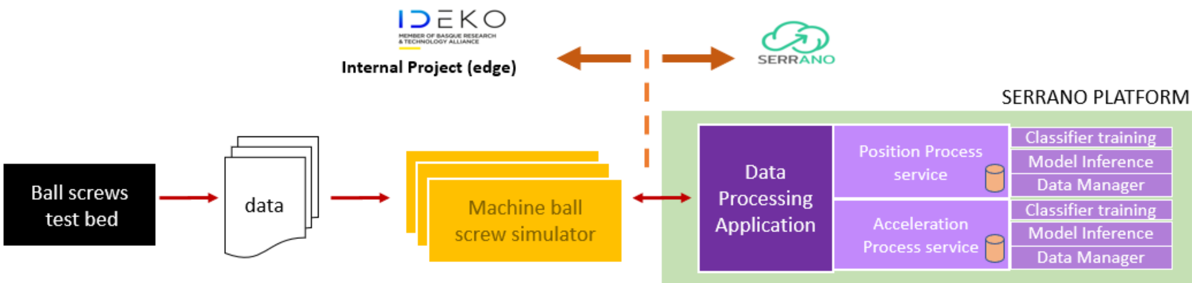

Figure 5: Data from ball screw to Data Processing Application services sequence simulating data from machines in a real scenario.

5.2 Integration

For an effective integration with the SERRANO project, both services of the use case application have been adapted and divided into three different microservices (Model Inference, Data Manager, Classifier Training). The following image shows a first integration view approach of the data processing application (inside the circle) developed at IDEKO to detect anomalies in the ball screw, with the integration of some of the components available in SERRANO to achieve the use case requirements and goals.

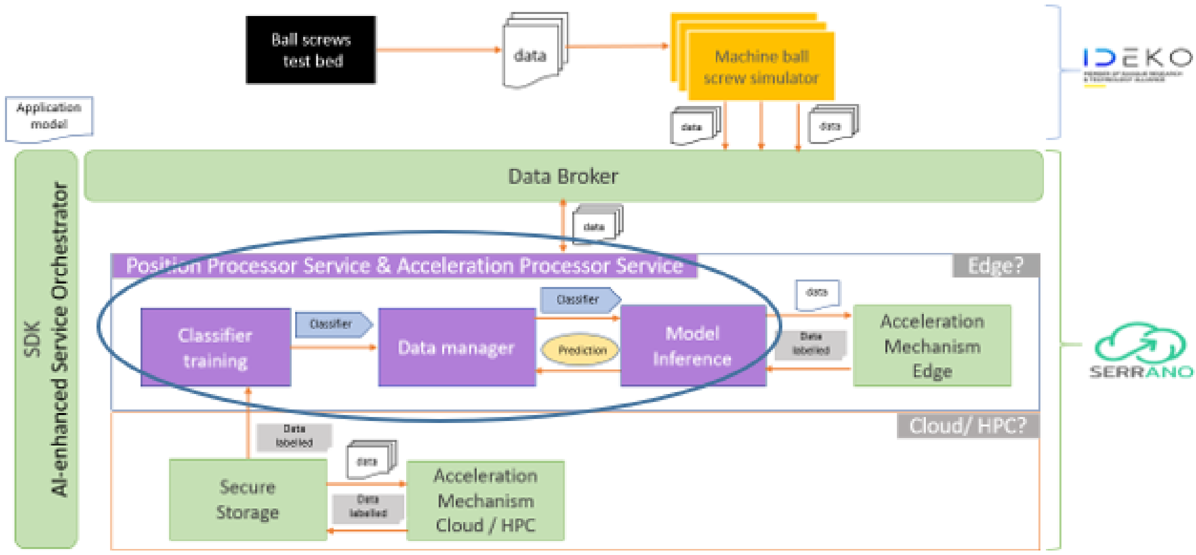

Figure 6: Draft - First integration view approach integration with SERRANO components.

These anomaly detection services/microservices (Position and Acceleration Processor Services) will be deployed on the SERRANO platform as appropriate containerized applications.

To facilitate the transparent deployment and management of the services/microservices, the appropriate description of the execution requirements should be provided to the AI-enhanced Service Orchestrator using the Application Model. The SERRANO orchestration mechanisms (AI-enhanced Service Orchestrator and Resource Orchestrator) will then translate these requirements into SERRANO infrastructure-specific runtime parameters and map them to the available resources (Resource Optimization Toolkit).

The machine streaming data will be integrated with the SERRANO platform through the Data Broker component, which will provide an interface based on the MQTT protocol to facilitate the publication and consumption of the data generated from the simulated machines' ball screws to use case applications/services and other SERRANO components.

In addition, the developed anomaly detection services will leverage the available interfaces of the SERRANO acceleration mechanisms in edge/cloud and HPC to enable their access to the SERRANO-enhanced computational resources. These resources will provide better performance and optimization of the kernels used (e.g., DTW, KMeans, KNN, FFT) by the Model Inference and Classifier Training services. Moreover, the Secure Storage component interface will be used to store (S3-compatible) the last N streaming data received through the Data Broker. This way, the required data will be stored and accessible by all SERRANO components and the use case services.

The idea is to reduce the classifier training time and the time needed to make a new prediction through the streaming data. This will enable the early detection of possible imminent failures of the ball screw, eliminating also their occurrence. In addition, it will provide greater control of the health status of the ball screw in real time. Since the current techniques and resources available at the edge cannot support the above operations, the SERRANO platform is needed.

5.3 Evaluation

As previously mentioned, two services are being developed for anomaly detection within the Data Processing Application depending on the data sensor (position or acceleration) from which the data comes. The initial version of the first service, "Position Process Service", is being tested through an initial set of unit tests while the first integration tests are also being defined. These tests are explained in detail in the next section.

The second service, "Acceleration Processor Service", is currently being defined. Both services will be updated and integrated into their final form by the end of the project.

Although the two services will be similar in their definition and integration, the integration tests detailed below are focused on the first service, the "Position Process service".

5.3.1 Testing

The testing and evaluation methodology will be based on the procedures for developing and releasing the software components of the SERRANO platform defined and detailed in Section 6 of deliverable 6.3 "The SERRANO Integrated platform".

SERRANO adopts and applies Continuous Integration and Continuous Delivery/Deployment (CI/CD) practices to set up a standardized process for developing and releasing the software components of the SERRANO platform. The typical CI/CD steps include committing the developed anomaly detection services (Position and Acceleration) source code to GitLab, which will trigger the Jenkins server to build the Docker image. Finally, utilizing the container images, the generated containers will be deployed to the K8s clusters of the integration and operational environments.

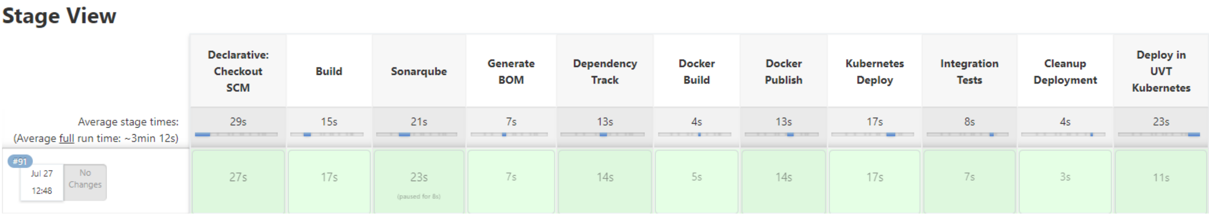

Figure 7: CI/CD stage view steps.

In order to test the correct performance of the microservices created for the Position Process Service, a set of unit tests has been created inside each code repository (Gitlab). Inside "tests" folder, everything needed to check the performance of the microservice can be found.

Microservices communicate with each other using a MQTT broker. Because of this, a broker is needed to perform the unit tests. To test every feature of the microservices, testing data is published into specific topics of the broker in order to check if the microservice is working as expected. The result of each run, based on the published data, is known in advance, so it is easy to check if the microservice has worked as expected.

With every commit to the GitLab code repository, a Jenkins pipeline is triggered using a webhook. The Jenkins agent that executes the pipeline (which is nothing more than a set of containers) has a local MQTT broker and a Python image to perform the developed unit tests.

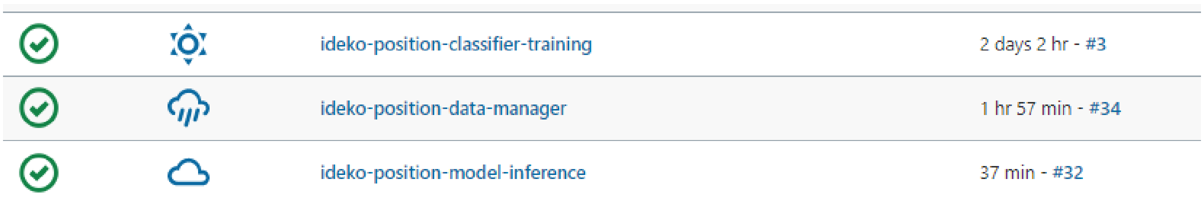

Figure 8: Status of the tests performed on the three microservices from the Jenkins server.

The microservices must be subscribed to the local MQTT broker to perform the unit tests. However, when we build the container image that would be deployed in the production environment, they have to subscribe to the respective data broker service that the SERRANO platform would provide. Hence, every microservice is developed to support a configurable working mode (test, dev, and prod) where it is specified to which MQTT broker it will subscribe to.

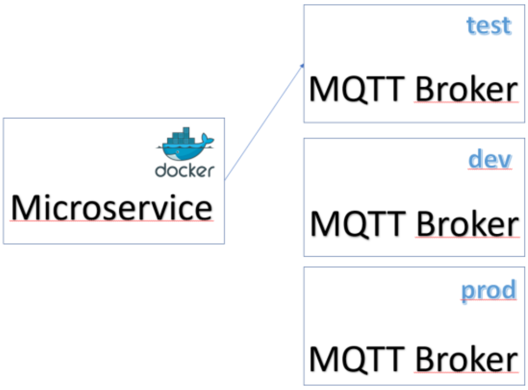

Figure 9: Different deployment mode according to its stage.

These tests are performed on the **Unit tests** stage of the Jenkins Pipeline. The following example shows a unit test set performed by every microservice. It uses the Python *unittest*[4] library as a build automation tool for building and running all the unit test classes.

```
stage('Unit tests') {
        steps {
            container('python') {
                sh 'python tests/setup.py test'
                sh 'python src/subscriber.py &'
                sh 'python -m unittest discover tests'
                sh 'python tests/setup.py prod'
            }
        }
    }
```

Figure 10: Jenkins pipeline – Unit test stage.

If any of the unit tests fail, the Jenkins pipeline stops. There are 12 unit tests in total to test every feature of the three microservice that integrate the Position Process Service.

[4] Python unittest framework: https://docs.python.org/3/library/unittest.html

```
+ python -m unittest discover -s tests -v
test_column_manage (test_file_manager.TestFileManager) ... ok
test_create_add_file (test_file_manager.TestFileManager) ... ok
test_create_file (test_file_manager.TestFileManager) ... ok
test_add_value (test_prediction_manager.TestPredictionManager) ... ok
test_buffer (test_prediction_manager.TestPredictionManager) ... ok
test_create_file (test_prediction_manager.TestPredictionManager) ... ok
test_prediction_file_delete (test_prediction_manager.TestPredictionManager) ... ok
test_prediction_file_delete_retrain (test_prediction_manager.TestPredictionManager) ... ok
test_publish (test_prediction_manager.TestPredictionManager) ... ok

----------------------------------------------------------------------
Ran 9 tests in 18.134s

OK
```

Figure 11: Output of unit test in data-manager microservice.

Once the correct performance of a microservice has been verified, its integration with the rest of the components that it will use within the SERRANO platform must be tested. These types of tests are called integration tests. To date, it has only been possible to test the integration with the Data Broker (MQTT) component and Secure Storage. As the project progresses, it will be tested with the rest of the necessary components described in section 5.2 Integration (e.g. Acceleration Mechanisms).

So as to test the integration of the microservice with the Data Broker, a unit test is reused, but this time the working mode of the microservice is set up to dev mode. This test is performed on the **Integration tests** stage of the Jenkins Pipeline.

```
      stage('Integration Tests') {
            when {
                environment name: 'DEPLOY', value: 'true'
            }
            steps {
                container('python') {
                    sh 'sleep 4m'
                    sh 'python tests/setup.py dev'
                    sh 'python tests/test_model_inference.py
TestModelInference.test_publish'
                }
            }
        }
```

Figure 12: Jenkins pipeline – Integration test stage.

```
+ python -m unittest discover -s tests/integration -v
test_data_broker (test_integration.TestClassifierTrainingIntegration) ... ok
test_secure_storage (test_integration.TestClassifierTrainingIntegration) ... ok

----------------------------------------------------------------------
Ran 2 tests in 338.335s

OK
```

Figure 13: Output of integration test in classifier-training microservice.

5.3.2 Methodology

The methodology for evaluating the technical results of the SERRANO projects has been described in detail in D6.2 through 5 KPIs.

These KPIs were selected based on the factors that affect the detection of anomalies in critical components such as the ball screw, taking into account the current state-of-the-art techniques, which base their techniques on stopping the machine to execute controlled and measured movements and compare them with previous measurements. These state-of-the-art techniques obtain a low knowledge of component health and decrease machine availability.

Success Criteria will be improved due to leveraging the SERRANO platform to change the state-of-the-art approach and to perform the ball-screws' status assessment without stopping the machine, moving to real time health assessment. Reaching deep knowledge of component health and increasing machine availability.

Table 5: UC3 technical success criteria of Anomaly detection in Manufacturing.

KPI	Success criterion	Estimated target value
Transition to real-time anomaly detection	Transition from on-demand to real time data analysis for anomaly detection to reduce machine stoppages	Transition successful
Anticipation of failures	Anticipate failures comparing to current state-of-the-art techniques	Anticipation successful
Anomaly detection accuracy increase	Increase of anomaly detection accuracy (avoiding nuisance alerts and false positives/negatives)	Increase by 35% or more
Rate of streaming data processing	Being able to quickly process large amounts of streaming data	20MB/s or more

Increased availability of machine	Increase of machine availability	2% or more, measured in a monthly basis

5.4 Results and outcome

Both technical and business results will be presented in more detail in deliverable D6.8 "Final version of Business, end user and technical evaluation", when the deployment of the proposed anomaly detection services are fully integrated into the SERRANO platform. In the following sections, the expected results are presented, but they have not been validated yet.

5.4.1 Technical results

The technical results will focus on transitioning from on-demand to real-time anomaly detection to reduce machine stoppages. The data generated in the machine is limited since it is generally stored at the edge, and the resources at the edge are limited, so data requests are made on demand. With the SERRANO platform, it will be possible to deploy multiple instances of the developed applications/services to detect anomalies, utilizing mechanisms that optimally orchestrate data and computationally demanding processes in the edge and cloud.

In addition, it will be able to anticipate failures (anomalies) compared to current state-of-the-art techniques through the provision of efficient services for predicting and detecting anomalies in manufacturing components using algorithms/kernels with greater accuracy and precision. Therefore, it will be possible to achieve faster failure anticipation compared to the current state-of-the-art techniques that perform offline analysis for predicting and detecting anomalies due to the lack of resources.

5.4.2 Business results

IDEKO is a research center specialized in manufacturing and industrial production technologies. Providing companies with differentiating technological solutions to enhance their competitiveness. In the last years, IDEKO has been intensively researching and developing solutions to digitalize machines and industrial plants. IDEKO and DANOBATGROUP, our strategic client, count on a vast amount of data from the machines and face the challenge of developing high-added value functionalities and services based on them. Therefore, data processing and applied artificial intelligence are core business fields for IDEKO.

To leverage the data generated by the machines, IDEKO is constantly requested to provide added value services in three areas of interest: (i) machine condition, (ii) machining process, and (iii) production. Combining the skills of engineering, software and data analysis IDEKO is intensively working on artificial intelligence applications to provide intelligence and autonomy to the machines. The objective is to build and scale AI solutions that will be exploitable in the market using the machine IoT platforms without memory and processing speed constraints. For this to succeed, IDEKO is working heavily in data-intensive applications and architecture research, and SERRANO directly impacts these critical areas of interest.

SERRANO will provide a scalable architecture to deploy AI services for anomaly detection, deploying and AI solutions that will be exploitable in the market using the machine IoT platforms without the constraints of memory and processing speed resources.

5.4.3 End user feedback

It is still early to get a proper and detailed feedback (challenges, advantages, …) on the integration with SERRANO. This will be presented in the D6.8 "Final version of business, end user and technical evaluation" (M36), providing feedback and insights from the experience of the integration with the SERRANO platform.

6 Summary

D6.4 is the first of the two reports that are scheduled to present the evaluation outcomes of tasks 6.3, 6.4, and 6.5, from the technical, business, and end user perspectives. Since most of the effort from the beginning of WP6 (M13) until the preparation of this deliverable was focused on adapting and extending the use case applications to leverage the SERRANO platform advancements, this report mainly describes the successful initial deployment and execution of unit and integration tests for the use case applications into the initial release of the SERRANO platform.

The Secure Storage (Task 6.3) use case has completed the planning phase on how the first use case will be evaluated. The first major integration point with the SERRANO Secure Storage Service has been implemented, along with the associated tests. In the next period, the UC will focus on finishing the integration and making the storage service a core part of the SERRANO platform, including also its key features as part of the SERRANO SDK. Once this is complete, the evaluation of the KPIs will be performed.

The FinTech (Task 6.4) use case will demonstrate the Dynamic Portfolio Optimization application on the SERRANO platform. The DPO application and the corresponding APIs have not been fully integrated into the SERRANO platform. In the following period, the activities within this task will focus on completing the developments and the full integration with the SERRANO platform, along with the corresponding evaluation.

For the Anomaly Detection (Task 6.5), SERRANO will provide a scalable architecture to deploy AI services for anomaly detection that will be exploitable in the market, without the constraints of the resources' memory/storage and processing capacity. In the next period, the UC will focus on fully integrating the application components with the SERRANO platform and performing the evaluation activities.

According to the project development plan, the technological developments in the project use cases will be completed at M27 and reported on deliverable D6.5 "Final version of the use cases technological developments" (M27). Moreover, the complete version of the SERRANO platform will be available at M31. Therefore, the actual and detailed performance and production-level evaluation of the three use cases applications will be described in D6.8 "Final version of business, end user and technical evaluation" (M36).