Real and synthetic scenarios generated for the development, training, virtual testing and validation of CCAM systems

SYNERGIES

D8.1 Evaluation Framework methodology

Document Type	Document, Report
Document Number	D8.1
Primary Author(s)	Riccardo Berta | University of Genoa (UNIGE)
Document Version / Status	1.0 | Final
Distribution Level	PU (public)

Project Acronym	SYNERGIES
Project Website	www.synergies-ccam.eu
Project Coordinator	IDIADA AUTOMOTIVE TECHNOLOGY SA
Grant Agreement Number	101146542
Date of latest version of Annex I against which the assessment will be made	2024-05-27

Funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

CONTRIBUTORS

Name	Organization	Name	Organization
Riccardo Berta	UNIGE	José Rodriguez	ERTICO
Luca Lazzaroni	UNIGE		
Francesco Bellotti	UNIGE		
Hadise Rojhan	UNIGE		
Matteo Fresta	UNIGE		
Mario Marchesoni	UNIGE		
Francesco Guastavino	UNIGE		
Antonio Bruto da Costa	UoW		

FORMAL REVIEWERS

Name	Organization	Date
Onur Yildirim	TUE	2025-05-23
Jordi Pont	Applus+ IDIADA	2025-05-29

DOCUMENT HISTORY

Revision	Date	Author | Organization	Description
0.1	2025-02-12	Riccardo Berta | UNIGE	ToC completed
0.2	2025-03-04	Luca Lazzaroni | UNIGE Hadise Rojhan | UNIGE	SoA section completed
0.3	2025-03-20	Luca Lazzaroni | UNIGE Hadise Rojhan | UNIGE	KPI card description completed
0.4	2025-04-07	Mario Marchesoni | UNIGE	Revised chapter on the Usability KPIs
0.5	2025-04-08	Riccardo Berta | UNIGE	Revised chapter on the SoA in User Experience
0.6	2025-04-08	Matteo Fresta | UNIGE Antonio Bruto da Costa | UoW	KPI Manager section completed
0.7	2025-04-17	Francesco Guastavino | UNIGE	KPI Manager revised
0.8	2025-04-24	Matteo Fresta | UNIGE Jose Rodriguez | ERTICO	User Manager section completed
0.9	2025-05-02	Riccardo Berta | UNIGE	Requirements referenced revised
1.0	2025-05-08	Mario Marchesoni | UNIGE	Final revision before internal review
1.0-rev	2025-05-16	Onur Yildirim | TUE	Internal review
1.0	2025-05-21	Francesco Guastavino | UNIGE	Final Version

TABLE OF CONTENTS

1 EXECUTIVE SUMMARY 7

2 INTRODUCTION 8

2.1 Scope of the Document 8
2.2 Objectives of the Evaluation Framework 8
2.3 Approach and tools 9
2.4 Structure of the Document 11

3 STATE OF THE ART ANALYSIS 12

3.1 Usability Evaluation 12

3.1.1 Usability metrics 13
3.1.2 Usability Key Performance Indicators (KPIs) 14

3.2 Technical Assessment 15

3.2.1 Technical Assessment Attributes 16
3.2.2 Technical Assessment Key Performance Indicators (KPIs) 17

3.3 Interoperability Assessment 18

3.3.1 Evaluation Attributes 19
3.3.2 Interoperability KPIs 19

4 DEFINITION OF KPIS AND EVALUATION METHODOLOGY 21

4.1 Methodology for KPI Definition and Selection 21

4.1.1 KPI Card Definition 22

4.2 Usability Assessment KPIs 22

4.2.1 KPI_01 Customer Satisfaction Score (CSAT) 22
4.2.2 KPI_02 System Usability Scale (SUS) 23
4.2.3 KPI_03 Net Promoter Score (NPS) 24
4.2.4 KPI_04 Task completion rate (TCR) 25
4.2.5 KPI_05 Error Rate (ER) 26
4.2.6 KPI_06 Trust & Compliance Assurance (TCA) 27
4.2.7 KPI_07 Cross- Browser and API Compatibility (CBAC) 28
4.2.8 Usability Assessment KPIs Summary 29

4.3 Technical assessment KPIs 29

4.3.1 KPI_08 System Responsiveness (SR) 29
4.3.2 KPI_09 Extensibility (EXT) 30
4.3.3 KPI_10 Security Evaluation (SE) 31
4.3.4 KPI_11 System Resilience (SRE) 32
4.3.5 Technical Assessment KPIs Summary 33

4.4 Interoperability KPIs 33

4.4.1 KPI_12 API Intent Assessment (AIA) 33
4.4.2 KPI_13 Semantic Interoperability of API (SIA) 34
4.4.3 Interoperability Assessment KPIs Summary 35

4.5 Data Collection Techniques and procedures 35

5 KPI MANAGER 38

5.1 Overview 38
5.2 Architecture and Technology Stack 39
5.3 Customization 40
5.4 Functionality 42
5.5 Deployment 45

6 USER MANAGEMENT 46

6.1 User Roles and Permissions 46
6.2 User Authentication and Security 47
6.3 Data Privacy 48
6.4 Logging and Audit Trails 49

7 CONCLUSIONS AND FUTURE WORK 50

7.1.1 Findings 50
7.1.2 Next Steps 50

8 REFERENCES 52

A. ABBREVIATIONS AND DEFINITIONS 55

LIST OF FIGURES

Figure 1 The black-box approach 10
Figure 2 The KPI Manager 11
Figure 3 Data collection procedure 37
Figure 4 KPI Manager Architecture 39
Figure 5 Data sources 40
Figure 6 Measurify entities 41
Figure 7 KPI Manager implementation 42
Figure 8 Manual creation of a Feature entity 43
Figure 9 JSON file for Feature entity 43
Figure 10 Questionnaire uploading phase 44
Figure 11 Score graph 44

LIST OF TABLES

Table 1 Usability Assessment KPIs 29
Table 2 Technical Assessment KPIs 33
Table 3 Interoperability Assessment KPIs 35
Table 4 Role-based access control model 47

1 EXECUTIVE SUMMARY

This deliverable presents the design and implementation of the **SYNERGIES platform's Evaluation Framework**, a structured methodology developed **to assess how effectively the platform meets user needs and technical requirements** in terms of **usability**, **technical quality**, and **interoperability**.

The main goal of this deliverable is to establish a **comprehensive and standardized framework** for evaluating the SYNERGIES platform. By defining **measurable indicators** and implementing **dedicated assessment tools**, the framework ensures that the platform evolves in alignment with the expectations of its stakeholders, researchers, developers, and users operating in the domain of Cooperative, Connected and Automated Mobility (CCAM). This evaluation process is essential to validate the platform's **robustness**, **user experience**, and **capacity for seamless integration** across diverse data sources and systems.

Among the primary achievements is the development of a detailed set of **Key Performance Indicators (KPIs)** covering three core evaluation domains: **usability**, **technical assessment**, and **interoperability**. Each KPI is defined using a **standardized format** that includes measurement methodology, requirement references, and performance targets aligned with project objectives. Notable examples include the **Task Completion Rate (TCR)**, which measures how effectively users can achieve their goals on the platform, and **Cross-Browser/API Compatibility (CBAC)**, which assesses the platform's ability to operate reliably across different environments and third-party systems.

A major milestone in this task was the creation of the **KPI Manager**, a cloud-based platform that facilitates the **structured collection, storage, visualization, and analysis of KPI data**. Built on the open-source **Measurify** framework, the tool integrates both manual inputs (e.g., questionnaires and expert checklists) and automated metrics (e.g., from performance monitors or fault injection tools), enabling continuous and transparent evaluation of the SYNERGIES platform over time.

The framework also incorporates **diverse evaluation techniques**, including surveys, expert assessments, automated testing, and real-time monitoring, ensuring a holistic view of the platform's strengths and areas for improvement. The methodology is directly linked to stakeholder requirements defined in WP2, providing traceability between project needs and system validation results. One key challenge addressed in this context was the complexity of integrating heterogeneous evaluation data (ranging from automated logs to subjective user feedback) into a coherent and traceable assessment pipeline.

Stakeholders will be **actively involved** through structured surveys, usability walkthroughs, and dedicated workshops, ensuring their feedback is continuously integrated into the evaluation cycle.

This deliverable thus lays a solid foundation for ongoing evaluation activities across the SYNERGIES project. It supports iterative improvement of the platform while reinforcing user trust, regulatory compliance, and cross-system interoperability.

Keywords: Key Performance Indicators, KPI Manager, Platform Evaluation, Usability, Technical Assessment, Interoperability

2 INTRODUCTION

This chapter provides an overview of the **evaluation framework** developed for the SYNERGIES platform, detailing the methodology, objectives, and tools used to assess its performance, user satisfaction, and technical compliance, while aligning with the broader SYNERGIES objectives and requirements outlined in WP2. It also introduces the approach taken in defining the tool and the key methodologies for gathering the data necessary for the assessment. Finally, the chapter offers an overview of the document structure to ensure clarity and ease of navigation.

2.1 Scope of the Document

The deliverable presents the **evaluation framework** designed to assess the SYNERGIES platform's performance and requirements fulfilment through a systematic data collection. The SYNERGIES platform is a federated European driving scenario dataspace, supporting researchers, developers, and other stakeholders in transportation and mobility domains. The platform allows users to interact with multiple, inhomogeneous scenario databases as if they were a single, cohesive repository.

The evaluation framework includes a structured methodology, and an online tool specifically developed to enable efficient implementation of the assessment process. This tool, referred to as the KPI Manager, is a web-based application that supports data collection, visualization, and analysis of evaluation metrics gathered through surveys, automated tests, and performance monitoring systems. By gathering feedback and analysing the platform's technical and usability performance, this framework ensures that the SYNERGIES platform evolves in line with user needs and expectations. The framework focuses on key dimensions such as system latency, scenario query success rate, user satisfaction, and compliance with FAIR (Findability, Accessibility, Interoperability, and Reusability) data principles, which are reflected in the set of KPIs.

The deliverable provides a detailed description of the assessment methodology and its deployment mechanism.

The evaluation framework is built upon and directly aligned with the **requirements** outlined in D2.1 Stakeholder High-Level Requirements. Throughout the framework's description, every design choice is explicitly linked to the specific requirement it addresses, ensuring a clear and consistent rationale for each decision. The references to requirements provided are not intended to cover the full set of the platform requirements but focus only on those related to the objectives of the Evaluation Framework, specifically usability, technical assessment, and interoperability. Other requirements, such as those related to data governance, scenario generation, and representativeness will be addressed through complementary work packages (e.g., D3.2 Assessment of available data from external sources, D6.1 Tools for governance, sustainability and representativeness), ensuring a **holistic evaluation across the project**.

In this way, the framework aligns with SYNERGIES broader objectives of providing a robust, user-centred, and interoperable platform for interacting with driving scenarios. By establishing a systematic approach to performance assessment, this framework contributes to the project's goals of ensuring usability, technical soundness, and seamless interoperability with various databases and formats.

2.2 Objectives of the Evaluation Framework

The evaluation framework aims to establish a structured methodology for assessing the performance of the SYNERGIES platform. This methodology is designed to systematically measure how well the platform meets technical requirements and user expectations. By integrating both qualitative and quantitative approaches, the framework ensures a comprehensive evaluation process that provides actionable insights for continuous improvement.

A key focus of the framework is the assessment of three core aspects: **usability**, **technical quality**, and **interoperability**. Usability evaluation considers how efficiently and effectively users can interact with the platform, assessing factors such as ease of use, user satisfaction, and accessibility. The technical assessment examines system functionalities, including latency, scalability, robustness, and security, ensuring that the platform meets high performance and reliability standards. Interoperability focuses on the platform's ability to exchange and integrate driving scenarios across different databases and formats, facilitating seamless data sharing and system compatibility.

To ensure objective performance measurement, the framework defines and manages **Key Performance Indicators** (KPIs) that provide quantifiable metrics for evaluating each aspect. These KPIs were derived through a combination of stakeholder engagement (through workshops conducted in WP8 with OEM partners) and alignment with recognized best practices in platform usability and software quality assessment. This dual approach ensures that the indicators are both context-relevant and methodologically sound. These KPIs enable continuous monitoring, allowing stakeholders to track platform progress over time. The goals of KPI management include setting measurable targets, facilitating data-driven decision-making, and ensuring transparency in performance assessment. Each KPI is linked to a target threshold or range, defined using literature survey, to enable meaningful assessment and track success over time. By aligning KPIs with user requirements and system functionalities, the framework provides a solid foundation for improving the SYNERGIES platform in a structured and efficient manner. KPI data is reviewed at regular intervals, aligned with the major release cycles of the SYNERGIES platform already identified (feedback on first platform release M20, and feedback on second platform release M26) and integrated into an iterative development process. This feedback loop ensures that evaluation outcomes are directly used to guide platform improvements and refinements.

2.3 Approach and tools

The evaluation framework employs a **black-box approach** (see Figure 1) to assess system performance from an external perspective. This methodology enables evaluators to analyse how the SYNERGIES platform functions without delving into its internal mechanisms. By focusing on input-output behaviours and user interactions, this approach provides an objective measure of the platform's effectiveness, usability, and interoperability. Inputs include standardized scenario queries and scenario uploads issued through the platform interface or API; outputs are assessed in terms of result accuracy, response time, and compliance with expected data formats. Tests cover a variety of use cases, such as querying federated databases for scenario metadata, uploading new scenarios, or retrieving scenarios across formats, to ensure consistent and correct system behaviour.

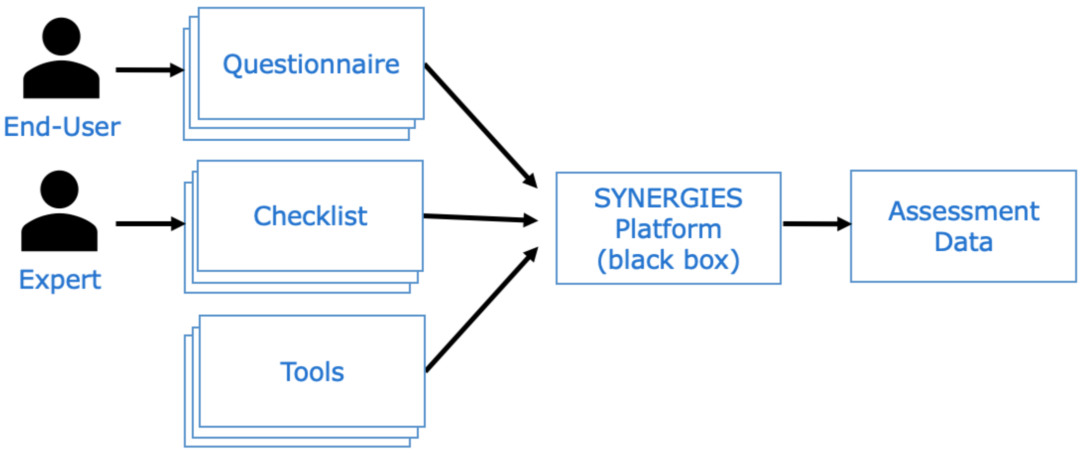

Figure 1 The black-box approach

Figure 1 illustrates the three primary approaches used for data collection in the platform assessment:

- To gather qualitative insights from users, **structured interviews** are conducted with key stakeholders, including researchers, developers, and end-users. These interviews help identify user expectations, challenges, and areas for improvement. The structured format ensures consistency in data collection, allowing for comparative analysis across different user groups and scenarios. The specific **feedback collection process** (including the number of planned interviews, their duration, thematic focus, and the methods for conducting and analysing them) will be defined as part of Task T8.2.
- Compliance with predefined requirements is verified using **checklists**. These checklists serve as systematic validation tools, enabling experts to assess whether the platform meets specified usability, technical, and interoperability criteria. By methodically evaluating each requirement, checklists ensure a comprehensive and standardized assessment process.
- In addition to qualitative methods, **automated tools** are employed for performance monitoring. These tools track key metrics such as system latency, availability, error rates, and scalability under different conditions. By leveraging automation, the framework ensures real-time performance tracking and facilitates prompt identification of potential issues. The integration of automated monitoring with user feedback mechanisms enhances the reliability and depth of the evaluation process.

These three approaches are selected to provide a comprehensive evaluation by integrating qualitative user insights, systematic compliance verification, and continuous performance tracking. This balanced methodology combines subjective user feedback with objective validation and real-time monitoring, ensuring a thorough assessment of the system's usability, reliability, and overall effectiveness.

To streamline the collection and management of KPI data, we have designed and implemented a **KPI Manager** (see Figure 2), a software tool that integrates both manual data entry via a graphical user interface (GUI) and automated data input through an application programming interface (API). This dual approach ensures flexibility in the data collection and processing methods. The tool centralizes the collected KPI measurements in a dedicated database, the **KPI Store**, making this valuable data accessible to the SYNERGIES platform and to project partners.

Access to the KPI Manager is role-based, with differentiated permissions for administrators, contributors, and viewers to ensure data integrity and controlled access. Manual data entries (such as those from questionnaires or expert checklists) are subject to verification procedures, which include validation against schemas (called “features” in the Measurify framework) and logging of submission to ensure accountability.

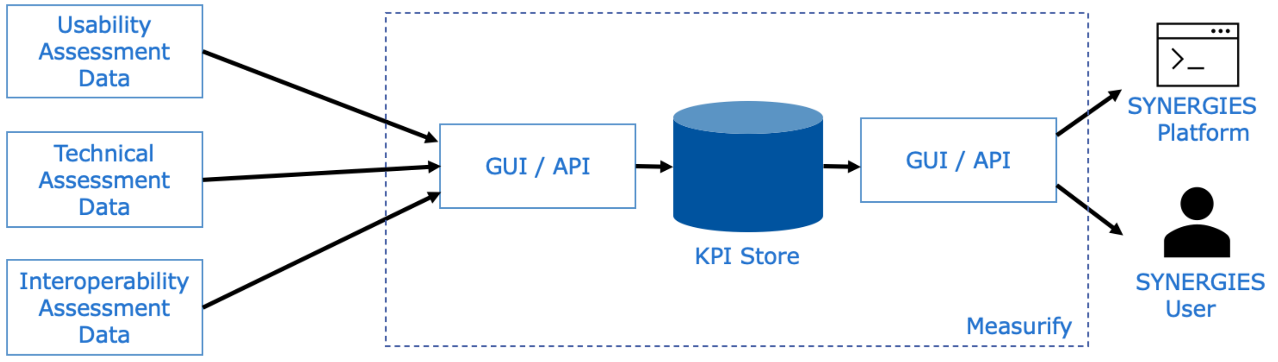

Figure 2 The KPI Manager

By combining black-box assessment, structured interviews, checklists, automated monitoring tools and the KPI Manager tools, the evaluation framework provides a **holistic approach** to assessing the SYNERGIES platform. This multi-faceted methodology ensures that performance evaluation remains objective, comprehensive, and aligned with user needs and technical requirements.

2.4 Structure of the Document

The document is structured as follows. The **Introduction** defines the scope, objectives, and approach of the evaluation framework. The **State-of-the-Art Analysis** reviews existing methodologies and best practices for evaluating usability, technical performance, and interoperability. The **Definition of KPIs and Evaluation Methodology** chapter details the process of defining KPIs, along with data collection techniques and measurement procedures. The **KPI Manager** chapter presents the cloud-based tool for KPI collection, visualization, and export, describing its architecture, implementation, and deployment. The **User Management** chapter outlines user roles, access levels, authentication mechanisms, and data privacy measures. Finally, the **Conclusions and Future Work** chapter summarizes the main findings and discusses the next steps for integrating the evaluation framework into future activities.

3 STATE OF THE ART ANALYSIS

This chapter provides an overview of existing methodologies, standards, and best practices related to the evaluation of usability, technical performance, and interoperability. It serves as a foundation for defining the evaluation framework by identifying established techniques and benchmarks used in similar platforms.

3.1 Usability Evaluation

Usability is a fundamental component of **user experience (UX) design**, referring to how **effectively**, **efficiently**, and **satisfactorily** a system (such as a website, software, or digital tool) enables users to achieve their intended goals [1]. As a key attribute of interactive systems, usability determines how intuitively users can navigate, interact with, and utilize a product's features. According to the International Organization for Standardization (**ISO 9241-11**), usability is defined as "the extent to which a system, product, or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use". This widely accepted definition emphasizes that usability is not an isolated characteristic but an integral part of the overall user experience, influenced by system design, user expectations, and contextual factors [2]. In the context of SYNERGIES, usability evaluation focuses on how intuitively researchers, developers, and policy makers can interact with the platform's core functionalities, such as querying distributed driving scenario databases, uploading scenario metadata, or navigating the user interface to configure search parameters.

Usability encompasses several critical dimensions that shape the quality of user interaction, including [3]:

- **Learnability**: how easily new users begin using the system and understand its basic functions with minimal instruction [4].
- **Efficiency**: how quickly experienced users can perform common tasks once they have learned the system [4].
- **Effectiveness**: the degree to which the system reliably supports users in achieving their goals without error.
- **Memorability**: how easily occasional users can return to the system and reestablish proficiency after a period of non-use,
- **Satisfaction**: the comfort and positive attitudes users develop through interaction with the system.
- **Understandability**: how clearly the system conveys the meaning and purpose of its functions and outputs, supporting mental models and reducing cognitive load.
- **Operability**: how easily users can control the system through input mechanisms such as buttons, menus, or forms.
- **Visual Design and Aesthetics**: the system's attractiveness, which affects users' first impressions and emotional engagement.
- **Ease of Use**: a high-level perception of how intuitive and non-frustrating the system feels in practice, often reflecting a combination of learnability, operability, and satisfaction.
- **Error Tolerance and Recovery**: the system's ability to prevent user errors and support effective correction when errors occur.

In SYNERGIES, these dimensions are assessed through a mix of structured usability walkthroughs, think-aloud protocols, and post-task questionnaires, applied during interactions with the platform's main user interface. Metrics such as Task Completion Rate (TCR), average time-on-task, and self-reported satisfaction scores are collected and analysed using the KPI Manager.

Beyond these aspects, usability plays a central role in determining a product's success. It enhances system performance and user engagement for several key reasons:

- It directly impacts product marketing and business growth, as user-friendly systems strengthen **brand credibility** and improve the **customer journey** [2].
- It boosts user satisfaction, reducing frustration and **increasing loyalty** and **positive referrals** [1].
- It contributes to system **popularity** and user **retention** by fostering enjoyable and seamless experiences [5].
- It improves **productivity** by streamlining workflows and minimizing unnecessary actions [5].
- It identifies **areas for continuous improvement** through testing and user feedback [6].
- It supports platform **sustainability** by easing the learning curve for new users and reducing maintenance needs.
- It enables the development of intuitive, **user-centred designs** that fulfil user expectations [6].

For SYNERGIES, improving usability not only benefits individual users but also supports the platform's broader goal of fostering data sharing, cross-border research, and long-term adoption across diverse mobility stakeholders.

Summarizing, usability is essential not only for making systems functional but for ensuring that user interactions are efficient, effective, and satisfying, leading to greater user engagement, system success, and lasting positive impact for both users and service providers.

3.1.1 Usability metrics

Usability metrics [4] are essential tools for evaluating the attributes previously defined (e.g. efficiency, satisfaction, effectiveness, learnability, etc.) [7]. The **measurable characteristics** (both quantitative and qualitative) that form the foundation for defining and applying usability metrics are:

- **Time-on-task** and **task completion time**, as well as the **rate of user errors**, which are primary indicators of efficiency and are considered **core metrics** in the SYNERGIES evaluation due to their direct relevance to platform performance.
- **Task success rate**, **number of steps to completion**, and **frequency of backtracking actions**, which are used to evaluate **effectiveness**. These are **prioritized metrics** in SYNERGIES, particularly in tasks involving scenario search and upload workflows.
- **Direct user feedback** via Likert-scale surveys, used to measure **satisfaction**. These are integrated into SYNERGIES' structured feedback collection and will be gathered at key release milestones
- **Number of attempts per action**, **need for assistance**, and **error frequency**, often used to gauge **learnability**, are treated as **secondary metrics**, employed selectively in onboarding sessions or early usability tests.

- **Memorability**, evaluated by comparing user performance before and after a period of inactivity, is acknowledged **but considered a lower-priority metric** due to the practical constraints of longitudinal testing within the project timeline.

Selecting an appropriate evaluation method is crucial for obtaining accurate and meaningful results. **Controlled observation** is one of the primary methods, involving direct observation of users performing tasks in a structured environment, enabling precise measurement of task completion rates, performance times, and error rates**. Surveys and questionnaires** are also widely used to gather user feedback on satisfaction, perceived ease of use, and overall experience. In SYNERGIES, these two methods form the backbone of usability data collection. Secondary methods, such as think-aloud protocols, and interviews, offer deeper qualitative insights, particularly for complex systems, though they are employed less frequently compared to observation and surveys [8].

The exact **frequency** of usability evaluations, the **number of users involved**, and the **testing settings** (e.g., in-lab vs. remote, pre-release vs. post-deployment) will be defined in Task T8.2, in alignment with the platform's development milestones and stakeholder availability.

3.1.2 Usability Key Performance Indicators (KPIs)

In addition to defining usability attributes and evaluation methods, establishing **Key Performance Indicators (KPIs)** is essential for objectively measuring the success of usability efforts. KPIs offer **quantifiable insights** into how effectively users interact with a system and whether the SYNERGIES platform meets its intended goals. By systematically tracking these indicators, developers can identify strengths, uncover usability issues, and implement targeted improvements to enhance the overall user experience. The following KPIs are particularly important for evaluating usability in digital platforms:

- The **Customer Satisfaction Score** (high priority) measures overall user satisfaction based on survey responses [9]. Typically collected using **Likert-scale questions (e.g., 1–5 or 1–10 ratings)**, this KPI captures how users perceive their experience with the platform. The CSAT is computed as the percentage of users who select positive ratings (e.g., 4 or 5 out of 5) divided by the total number of respondents. High CSAT scores indicate positive interactions and satisfaction, while lower scores highlight areas where user needs and expectations are not fully met. Tracking this value helps developers prioritize enhancements that boost engagement, loyalty, and retention.
- The **System Usability Scale** (medium priority) is a standardized and widely accepted tool for assessing system usability [10]. It employs a **10-item questionnaire** where users rate various aspects of the system's ease of use, design, and functionality on a **5-point Likert scale**. The raw score is converted to 0-100 scale using a specific scoring formula. Higher scores indicate better usability. SUS provides a simple yet reliable measure that is applicable across a range of digital products, including websites, mobile applications, and software platforms.
- The **Task Completion Rate** (high priority) measures the percentage of tasks that users successfully complete during their interaction with the platform [11]. It serves as an indicator of the system effectiveness, TCR is calculated as the number of successfully completed tasks divided by the total number of attempted tasks, expressed as a percentage. A high value suggests that users can navigate the system efficiently and complete tasks with minimal difficulty, whereas a low value signals potential usability barriers such as unclear instructions or poor navigation design.

- The **Error Rate** (medium priority) captures the frequency of user errors during platform interactions [11]. Errors may include failed task attempts, incorrect data entries, or mis-navigations. ER is typically measured as the number of errors per task or per user session and may be complemented with severity scores assigned by evaluators. Monitoring the error rate helps identify design weaknesses or usability flaws that disrupt user workflows. A lower error rate indicates a smoother, more intuitive system, while a higher error rate highlights areas requiring immediate improvement to reduce user frustration and enhance confidence.
- The **Trust and Compliance Assurance** (low priority) measures the extent to which users trust the platform regarding data security, privacy, and regulatory compliance [12]. As concerns about data protection continue to grow, trust has become a critical driver of platform adoption and engagement. It will be measured through user trust questionnaires (e.g., confidence in data handling, perceived transparency), along with compliance checklists scored against standards such as GDPR. Key elements evaluated under this KPI include the clarity of privacy policies, transparency in data handling, implementation of security measures like SSL certificates and two-factor authentication, and adherence to industry standards such as GDPR, ISO 27001, or HIPAA. Strong performance in this area boosts user confidence and strengthens long-term relationships.
- The **Cross-Browser and API Compatibility** (high priority) assesses the platform's ability to deliver consistent performance across different browsers, devices, and third-party integrations [9]. Essential elements of this KPI include maintaining consistent layout, functionality, and responsiveness. CBAC will be measured by conducting functional regression tests across environments (e.g., Chrome, Firefox, Safari, Edge on multiple OS), and validating API responses using automated test suites for format correctness and latency thresholds. Poor compatibility can lead to user frustration and abandonment, making this KPI critical for delivering a seamless and accessible user experience [13].

3.2 Technical Assessment

Technical assessment is a critical process in the design, development, and maintenance of digital platforms, serving as a cornerstone for **ensuring robust, efficient, and secure systems**. It encompasses the systematic evaluation of a platform's architecture, performance, scalability, and overall operational integrity. In the case of the SYNERGIES platform, the technical assessment focuses on evaluating backend performance, data integration pipelines, API responsiveness, and system robustness under realistic usage conditions. Within the disciplines of software engineering and digital systems design, technical assessments help align the platform's technological foundations with industry standards, business goals, and user expectations. For SYNERGIES, this alignment is essential to ensure the platform can support seamless access to heterogeneous scenario databases, maintain high availability across distributed nodes, and enable secure and compliant data exchange.

By identifying **technical strengths**, **uncovering vulnerabilities**, and **revealing potential performance bottlenecks**, technical assessment enables data-driven decision-making throughout the software lifecycle. As SYNERGIES evolves through iterative releases, technical assessments are conducted at key milestones to verify performance targets, guide infrastructure scaling decisions, and validate software component integration. This process is particularly relevant for online platforms that must support high availability, multi-device compatibility, and secure data exchange under dynamic workloads. Applications of technical assessments are broad, ranging from APIs and web services to cloud-based and mobile systems, all of which benefit from continuous evaluation and refinement of their underlying infrastructure [14]. The SYNERGIES architecture includes RESTful APIs and a federated search engine, all of which are subject to continuous performance and interoperability testing as part of the technical evaluation framework.

The importance of technical assessment lies in its capacity to provide **clear, actionable insights** that directly impact a platform's functionality, maintainability, and growth potential. key benefits include:

- **Identifying Strengths and Weaknesses**. Technical assessment reveals architectural soundness, optimized code structures, and well-implemented features, while also exposing inefficiencies, technical debt, or integration issues that could hinder performance or increase development costs [14].
- **Enhancing Performance and Operational Efficiency**. Performance directly affects both user experience and system throughput. Assessments help pinpoint bottlenecks (such as slow database queries or inefficient algorithms) thus enabling targeted optimization that improves responsiveness and efficiency [11].
- **Supporting Scalability and Growth Readiness**. With increasing user loads and feature complexity, systems must scale effectively. Assessments ensure that the architecture and infrastructure can accommodate demand without degradation, laying the groundwork for cost-effective expansion [15].
- **Reducing Technical Debt and Maintenance Overhead**. Regular assessments highlight outdated practices, undocumented components, and poor modularization. Addressing these issues reduces future development effort, minimizes bugs, and facilitates sustainable code evolution.
- **Strengthening Security and Compliance**. Technical assessments include the evaluation of authentication methods, data encryption, and software dependencies to identify security flaws. Proactively resolving these issues is crucial for regulatory compliance and user trust [16].
- **Improving Interoperability and Compatibility**. Platforms rarely operate in isolation. Assessments ensure integration with third-party APIs, devices, browsers, and operating systems is seamless and reliable, reducing user friction and support needs [17].
- **Boosting User Experience and Reliability**. A technically robust system ensures low downtime, quick load times, and consistent behaviours, elements that directly influence user satisfaction and engagement [11].

3.2.1 Technical Assessment Attributes

A comprehensive technical assessment examines **multiple attributes** that collectively determine the system's **robustness**, **flexibility**, and **performance**:

- **Functionality**. The extent to which the system performs its intended tasks correctly and completely, meeting business and technical requirements [15].
- **Performance**. Measured through speed, responsiveness, and stability [15]. Subcomponents include:
    - **Response Time**: Time taken to process and return outputs.
    - **Scalability**: Ability to handle increasing load without performance loss.
    - **Stability**: Consistency under prolonged or peak usage.
    - **Efficiency**: Resource utilization, including CPU, memory, and network bandwidth.

- **Security**. Resistance to threats, including data breaches and unauthorized access. Assessment includes encryption, access control, vulnerability scanning, and compliance with standards such as GDPR or ISO 27001 [16].
- **Reliability**. The system's ability to operate continuously and correctly over time, particularly under stress or during failure scenarios [15].
- **Maintainability**. Ease of updates, debugging, and feature enhancement. Influenced by code modularity, documentation quality, and development practices [15].
- **Compatibility**. Consistency across platforms, browsers, and environments. A key factor for user accessibility and integration readiness [17]
- **Testability**. The ease with which system components can be verified through unit, integration, and regression testing. Often determined by architecture and code structure.

3.2.2 Technical Assessment Key Performance Indicators (KPIs)

Technical assessment KPIs offer measurable indicators of a platform's performance, adaptability, and robustness. Among the most critical, we will consider:

- **System responsiveness** [18] (high priority) reflects how efficiently a platform processes and returns results to user or application requests. It is primarily influenced by three aspects: **response time**, **concurrency**, and **scalability**. Response time represents the delay between a request (e.g., a SQL query) and its result. Shorter response times improve overall system performance, enhance user satisfaction, ensure operational efficiency in transaction-heavy systems, and optimize resource utilization. **Concurrency** [19] (the simultaneous execution of multiple operations) is essential for maintaining data integrity [20] and availability when multiple users or services access the system in parallel. Effective concurrency control prevents interference, enables parallel processing, and ensures system reliability. **Scalability** measures the system's capacity to handle increasing user loads or data volumes while maintaining stable performance [21]. It is vital for sustaining growth, avoiding system failures, and supporting business continuity with efficient resource use. This KPI is essential in SYNERGIES and directly feeds into continuous monitoring through the KPI Manager.
- **Extensibility** [22, 23] (medium priority) denotes a system's ability to evolve through the addition of new features without requiring significant changes to its core structure. It supports customization, simplifies integration with external tools, and ensures long-term adaptability to shifting user needs or technological advancements. In SYNERGIES, extensibility is addressed during architectural reviews and feature update planning but is assessed qualitatively rather than through automated metrics.
- **Security** [16] (high priority) evaluation focuses on protecting data from unauthorized access and ensuring regulatory compliance. Key aspects include data **encryption**, **authentication** mechanisms, and **access controls**. A secure platform builds user trust, meets legal requirements (e.g., GDPR), and mitigates risks such as data breaches or service disruption, thereby safeguarding digital assets.
- **System resilience** [24] (high priority) refers to the platform's ability to remain operational during and after failures. It encompasses both **fault tolerance** (maintaining continuous functionality despite errors) and **recovery** mechanisms that restore normal operations rapidly. Resilient systems minimize downtime, maintain service reliability, and ensure long-term infrastructure health, especially in cloud environments. In SYNERGIES, resilience is

tested during fault injection campaigns and monitored using platform health-check endpoints.

Together, these KPIs provide a comprehensive framework for evaluating the technical soundness of online platforms, supporting informed decisions for improvement and sustainable growth. Among them, system responsiveness and resilience are prioritized as core indicators due to their direct impact on performance, user satisfaction, and platform reliability in a federated, multi-stakeholder environment.

3.3 Interoperability Assessment

Interoperability is the capability of systems, platforms, or components to **work together** and **exchange information** seamlessly, despite differences in programming language, interface design, or execution environment. In modern digital ecosystems, where services often rely on heterogeneous technologies and third-party integrations, interoperability has become essential to ensure **scalable, sustainable, and efficient system cooperation** [25]. It represents a **high-level form of reusability**, enabling systems to access external resources and services even when their native access mechanisms are incompatible [26].

In the SYNERGIES platform, interoperability is foundational: the platform is designed to unify access to distributed and heterogeneous driving scenario databases, each potentially using different data formats, metadata schemas, APIs, and query protocols. Interoperability assessments evaluate the extent to which systems can **effectively exchange data, collaborate across diverse infrastructures, and support consistent interactions**. These evaluations focus on critical dimensions such as data exchange quality, communication speed, protocol compatibility, and adherence to common standards [27]. The process involves analysing APIs, data formats, middleware solutions, and security mechanisms to determine if existing system configurations support robust and meaningful interactions. For SYNERGIES, this includes validating conformance to the internal data model, assessing REST API endpoints exposed by the platform, and ensuring that results from remote sources are normalized into the platform's unified schema for querying and visualization.

By identifying **integration barriers and gaps**, interoperability assessments provide a roadmap for improving system alignment and establishing structured pathways toward integration maturity. These insights are vital for designing more cohesive digital services and reducing system friction, especially in environments involving legacy systems, cloud platforms, and cross-organizational operations [28]. In SYNERGIES, this process helps guide both the onboarding of new data providers and the refinement of data harmonization procedures, ensuring that each node contributes usable, queryable, and compliant content to the federated dataspace.

In complex IT environments, interoperability challenges frequently emerge from **incompatible technologies**, **inconsistent data representations**, or **divergent interface standards** [25]. Assessments help isolate these issues by systematically analysing communication failures, revealing where systems cannot interact effectively. Common interoperability obstacles include:

- **Data format mismatches** (e.g., ISO 8601 vs. Unix epoch for timestamps).
- **API inconsistencies** (e.g., method naming conventions, parameter types, or response structures.
- **Protocol divergences** (e.g., systems using HTTP versus MQTT or WebSocket).

In SYNERGIES, these challenges were encountered when integrating scenario sources with different metadata models and when adapting third-party scenario repositories to support the unified SYNERGIES API. These issues are **often subtle** yet critical and pinpointing them enables targeted fixes without requiring costly architectural overhauls. Interoperability assessments also allow for improved integration by employing strategies such as middleware for protocol bridging, adapters for legacy systems, or schema alignment to harmonize data fields and formats.

The interoperability evaluation in SYNERGIES includes automated validation of API responses, conformance testing against specifications, and cross-node scenario retrieval tests to confirm consistency. In addition, they support **compliance with standards and regulations** (e.g., GDPR) and offer structured inputs for developing **interoperability maturity models**, which guide long-term system evolution and architectural cohesion.

3.3.1 Evaluation Attributes

Effective interoperability assessments are based on key system attributes and levels of interaction. These qualities serve as the foundation for evaluation criteria and metric design [25]:

- **Adaptive Behaviour**. Ability to dynamically adjust to new data formats, protocols, or services without requiring manual reconfiguration.
- **Dynamic Connection**. Capacity to establish runtime interactions, crucial in service-oriented or cloud-native environments.
- **Standardization**. Use of common formats (e.g., JSON, XML), protocols (e.g., REST, SOAP), and authentication methods (e.g., OAuth, OpenID) to reduce integration complexity [30].
- **Availability and Compatibility**. Degree to which systems remain consistently accessible and interoperable, even under software version changes or load variations.
- **Alignment with Organizational Goals**. Ensuring that system interactions support business workflows, regulations, and operational contexts.

Interoperability is assessed across four main levels:

- **Technical level**. Ensures connectivity and message exchange through compatible protocols, hardware, APIs, and secure channels.
- **Syntactic level**. Focuses on consistent data structure using schema validation and standardized message formats.
- **Semantic level**. Validates shared understanding of data meaning through ontologies, vocabularies, and clearly documented data models.
- **Organizational level**. Aligns roles, responsibilities, and policies across systems to support cross-functional workflows and long-term strategic goals.

3.3.2 Interoperability KPIs

KPIs offer measurable ways to evaluate how well systems can interact. For the SYNERGIES platform, the following two KPIs are prioritized:

The **API Intent Assessment** (high priority) KPI evaluates whether an API functions as specified in its documentation and design logic. In SYNERGIES, this KPI is assessed using automated API contract validation tools, API conformance checks, and regular endpoint test scripts. It measures the API's ability to:

- Deliver expected responses with correct status codes and data,
- Properly validate input and provide informative error messages,
- Offer standardized endpoints, naming conventions, and versioning,

- Support reliable integration for external clients.

APIs that are well-documented, consistent, and predictable reduce the likelihood of integration failures and contribute directly to interoperability success [28].

The **Semantic Interoperability** (medium priority) KPI assesses whether the exchanged data is not only syntactically correct but also semantically consistent. It verifies that:

- Both systems use shared vocabularies or ontologies.
- Data models are aligned and clearly documented.
- The intended use and meaning of each data element are mutually understood.

Misalignment at the semantic level can lead to significant operational errors despite successful data transmission. Ensuring semantic clarity is therefore essential for dependable system interactions [25].

While both KPIs are important, API Intent Assessment is prioritized as it offers automated, scalable, and actionable insights critical for interoperability within the SYNERGIES platform. Semantic Interoperability, though equally foundational, is monitored more strategically during integration phases and reviewed during major updates.

Interoperability assessments are essential for evaluating how systems exchange and interpret data across technological and organizational boundaries. They help identify misalignments in protocols, data formats, and API behaviours, and guide corrective actions that enable scalable and future-proof integration. By leveraging standardization, adaptive architectures, and structured KPIs, organizations can reduce integration complexity, maintain compliance, and build platforms that are resilient, flexible, and ready for continuous evolution in increasingly connected digital ecosystems.

4 DEFINITION OF KPIS AND EVALUATION METHODOLOGY

This chapter outlines the process of defining and selecting KPIs to assess the SYNERGIES platform's usability, technical performance, and interoperability. Each KPI is carefully crafted to align with the platform's specific requirements and goals, following a standardized **KPI Card format**. This ensures consistency across the evaluation process, facilitating a clear and structured measurement framework. The chapter also delves into the comprehensive evaluation methodology, detailing the data collection techniques, measurement procedures, and analytical methods employed to guarantee an accurate, objective, and reliable assessment of the platform's performance. By establishing these procedures, the chapter aims to ensure that the evaluation process remains both transparent and effective, providing actionable insights for continuous platform improvement.

4.1 Methodology for KPI Definition and Selection

The selection of Key Performance Indicators (KPIs) for the SYNERGIES platform is a critical process designed to ensure that the evaluation framework accurately measures the platform's performance and effectiveness. The primary criteria for KPI selection are:

- **Relevance to platform objectives**: KPIs are chosen to directly align with the overarching objectives of the SYNERGIES platform. This ensures that the evaluation focuses on aspects that are most critical to the platform's success, such as usability, technical performance, and interoperability.
- **Linkage to platform requirements**: each KPI is explicitly linked to specific requirements defined in WP2. This linkage ensures that the evaluation framework assesses whether the platform fulfils its intended functions and meets the needs of its stakeholders, providing a clear rationale for each KPI's inclusion.
- **Considering best practices**: the selection of KPIs should also considers alignment with the SoA in evaluation methodologies. This ensures that the evaluation is not only relevant to the project's specific goals but also grounded in established and effective approaches.

A KPI Card format is employed to provide a structured and standardized approach to documenting and communicating KPIs. The use of KPI cards offers several benefits:

- **Clarity and Consistency**: the format ensures that each KPI is clearly defined, consistently measured, and easily understood by all stakeholders.
- **Comprehensive documentation**: each card provides detailed information on the KPI's name, description, measurement methodology, requirement references, target value, and data schema. This comprehensive documentation supports a thorough and transparent evaluation process.
- **Alignment with project goals**: by linking each KPI to specific requirements in WP2, the KPI card format ensures that the evaluation is directly aligned with the broader goals and objectives of the SYNERGIES platform.

The KPI selection criteria and the KPI card format are essential components of the evaluation framework, ensuring that the assessment of the SYNERGIES platform is relevant, structured, and contributes to the platform's continuous improvement.

4.1.1 KPI Card Definition

The KPI Card format is designed to provide a **structured and standardized way to document and communicate KPIs** for the evaluation of the SYNERGIES platform. Each card includes the following elements:

- **Name**: a concise and clear label for the KPI, ensuring easy identification and reference.
- **Description**: this section provides a detailed definition of the KPI, explaining its significance and relevance to the overall evaluation. It highlights what the KPI measures and why it is important in assessing the platform's performance, usability, or interoperability.
- **Measurement Methodology**: here, the process for collecting and analysing data is outlined. This includes the tools, techniques, and procedures used to gather data, as well as the analytical methods employed to derive meaningful insights from the collected information.
- **Requirement References**: this section links the KPI to the relevant requirements defined in WP2. It ensures that each KPI is tied to specific project objectives and requirements, facilitating alignment with the broader goals of the SYNERGIES platform.
- **Target Value**: the value or performance range for the KPI is specified here. This is an acceptable performance threshold/range, providing clear benchmarks for assessing whether the platform meets or exceeds predefined criteria.

Together, these components of the KPI Card create a comprehensive framework for monitoring, evaluating, and improving the SYNERGIES platform based on clear, measurable objectives.

4.2 Usability Assessment KPIs

This chapter defines the Usability Assessment KPIs established to evaluate how effectively the SYNERGIES platform meets user expectations in terms of satisfaction, loyalty, task efficiency, error minimization, trust, and compatibility. Each KPI provides a focused measurement approach, allowing the project team to monitor, analyse, and improve critical aspects of the platform experience through targeted surveys, structured evaluations, and systematic performance testing.

4.2.1 KPI_01 Customer Satisfaction Score (CSAT)

Description: The Customer Satisfaction Score (CSAT) evaluates user satisfaction with specific features or interactions on the platform, such as scenario search, visualization tools, or data export. Collected via a targeted questionnaire, this KPI assesses how well the SYNERGIES aligns with user needs and expectations.

Measurement Methodology: The CSAT is measured using a structured questionnaire designed to capture user feedback on their platform experience. The survey is brief and includes key questions to effectively assess satisfaction, accessible via the following link:

https://forms.office.com/e/ZtRb2mx4zk

The CSAT KPI is calculated as:

$$CSAT\ (\%) = (\text{Number of satisfied users, ratings 4 or 5}) / (\text{Total responses}) \times 100$$

Microsoft Forms is used to create and distribute the questionnaire, offering an easy interface for survey creation (including support for conditional logic and customizable questions), sharing, and automatic chart generation. Data can be automatically exported to the KPI Manager for deeper analysis, exploiting the Power Automate tool.

To enrich quantitative scores with **qualitative insights**, the survey includes **open-ended questions** where users can explain the reasons for their rating. This enables the identification of common pain points, unmet needs, or praised features. **User segmentation** (e.g., by stakeholder group such as OEMs, developers, or researchers) is supported to allow differentiated analysis and targeted follow-up actions.

Timing of survey administration is also defined to ensure contextual accuracy:

- For in-person events (e.g. conferences), participants will be invited to complete the survey immediately after engaging with specific platform features, using digital devices or paper forms.
- For remote users, the questionnaire will be distributed via email or through the SYNERGIES collaboration platform (based on Zulip), with timing coordinated to follow major user sessions or interaction milestones (e.g., scenario upload, API usage, feature testing).

Requirement References: This KPI captures the overall perception of the platform's usefulness, interface clarity, and user experience by directly measuring satisfaction with core functionalities like scenario search, visualization, and export. It is a high-level indicator of how well the platform supports diverse stakeholders (e.g., OEMs, technical services) in their operational needs (Req_48), and indirectly reflects on aspects such as data format usability (Req_13) and input quality perception (Req_15).

Target Value: In most industries, a CSAT score of 80% or higher indicates that customers are generally satisfied with the service. Anything above 90% is considered excellent, though reaching and maintaining this level consistently can be challenging. Targeting a CSAT score of 85% is a strong and achievable goal for the SYNERGIES platforms, signifying a high level of user satisfaction while allowing room for continuous improvement.

4.2.2 KPI_02 System Usability Scale (SUS)

Description: The System Usability Scale (SUS) evaluates the platform's usability, focusing on ease of use, efficiency, and the overall user experience. As a widely recognized metric, SUS ensures the platform aligns with user needs, minimizes friction, and offers an intuitive interface.

Measurement Methodology: The assessment focuses on measuring the ease of use, efficiency, and effectiveness of the platform for users. This measurement is realized through a standardized questionnaire-based method, accessible via the following link:

https://forms.office.com/e/MhetwWv2Rh

As discussed for CSAT, Microsoft Forms is used to create and distribute the questionnaire. The questionnaire consists of a set of questions designed to assess system usability. Each question is rated on a 1-5 Likert scale, where 1 represents "Strongly Disagree" and 5 represents "Strongly Agree". Odd-numbered questions are positive statements, while even-numbered questions are negative statements to reduce response bias. The SUS KPI is calculated in a standard way as:

SUS Score = (sum of adjusted scores) × 2.5

Where we adjust each response as follows: subtract 1 from odd-numbered question scores, subtract even-numbered scores from 5.

To ensure that SUS responses are **grounded in recent and meaningful platform use**, participants are asked to complete a **predefined set of core tasks** (e.g., scenario search, metadata editing, cross-node query) before filling out the questionnaire. This task set ensures that users have **sufficient exposure** to the relevant features being evaluated.

To address the potential **limited diagnostic power** of SUS, the questionnaire includes **open-ended fields** inviting users to comment on what they found most usable or most confusing. This hybrid approach provides both a standardized score and actionable qualitative insights.

Requirement References: SUS quantifies perceived usability using a standardized 10-question Likert-scale form, targeting aspects such as interface intuitiveness, navigation consistency, and ease of learning. It addresses fundamental usability expectations (Req_48) and indirectly validates design choices related to input/output consistency (Req_08, Req_13) that support efficient scenario interaction across heterogeneous sources.

Target Value: A good target for the SUS score ranges from 68 to 75, with scores above 80 indicating excellent usability. For the SYNERGIES platform, we define a target of 75 to reflect high usability.

4.2.3 KPI_03 Net Promoter Score (NPS)

Description: The Net Promoter Score (NPS) measures user loyalty and their likelihood to recommend the platform. A high NPS score often reflects positive user experiences, driven by strong usability and satisfaction. The measurement is done using the same questionnaire developed for the SUS KPI. By combining both metrics in a single survey, we aim to gain insights that not only highlight the platform's usability strengths but also demonstrate how these strengths impact overall satisfaction and user loyalty. This integrated approach ensures the platform effectively meets user needs, fostering long-term engagement and trust.

Measurement Methodology: The NPS section of the questionnaire asks the user to rate, on a Likert scale from 0 to 10, how likely they are to recommend the platform to others. Based on participant responses, NPS KPI will be categorized into three groups: "promoters" (9-10) are highly satisfied and loyal users who are likely to recommend the platform; "passives" (7-8) are satisfied but neutral users who are not likely to promote the platform and are excluded from the NPS calculation but considered potential promoters. The NPS score is calculated as:

NPS = (% Promoters) – (% Detractors)

To enable **meaningful trend analysis**, NPS will be measured at multiple stages in the user lifecycle: after onboarding, to capture first impressions; after 1 month of use, to evaluate sustained satisfaction; and following major platform updates, to assess the impact of feature improvements or interface changes. This recurring measurement approach allows us to **track changes in user sentiment** and identify the effects of specific interventions.

To better understand user loyalty, NPS data will be **segmented** by stakeholder type (e.g., data contributors, scenario builders, policy makers, developers). This enables targeted analysis of group-specific pain points or unmet needs and informs role-specific platform improvements.

Requirement References: NPS measures users' willingness to recommend the platform, offering insight into long-term satisfaction and perceived value. By capturing loyalty beyond immediate interactions, it complements CSAT and SUS to reflect whether the platform meets real-world user expectations (Req_48) and supports a positive collaborative environment for scenario database stakeholders.

Target Value: A good target for the SUS score ranges from 68 to 75, with scores above 80 indicating excellent usability. For NPS, a score above 50 is considered strong, with scores above 70 considered outstanding. For the SYNERGIES platform, we define a SUS target of 75 to reflect high usability, and a NPS target of 60 to have a solid user loyalty and satisfaction, crucial for long-term engagement and growth.

4.2.4 KPI_04 Task completion rate (TCR)

Description: The Task Completion Rate (TCR) measures the effectiveness of a platform by calculating the percentage of users who successfully complete a specific task, such as signing up, downloading a scenario, or accessing a feature. These tasks should reflect key user goals and interactions with the platform. This KPI offers valuable insights into usability, user experience, and any barriers that may impede task completion. TCR is crucial for identifying friction points and optimizing workflows to improve efficiency and overall platform performance.

Measurement Methodology: TCR is measured by tracking users' success in completing predefined tasks on the platform. First, the user is instructed about the task he should complete. Then, we use a screen recording tool to capture user interactions, which are later analysed to determine whether the task was completed successfully based on predefined success criteria (e.g., completing the registration or downloading a specific scenario). To ensure that task definitions remain relevant and representative, the list of core tasks will be finalized after the platform's deployment and will evolve alongside feature releases.

We defined several task completion categories, described by a weight: "Complete Success" (weight=1) occurs when the user completes the task smoothly and without issues; "Success with Minor Issues" (weight=0.7) refers to tasks completed with some difficulty or hesitation; "Success with Major Issues" (weight=0.3) describes tasks completed only after significant struggle or with external guidance.; and "Failure" (weight=0) indicates that the user was unable to complete the task. These categories allow for a more nuanced understanding of the user experience beyond binary outcomes. The TCR KPI is calculated as a weighted average, reflecting not just whether users completed tasks, but how well they did so:

TCR = [(number of Complete Successes × 1) + (number of Successes with Minor Issues × 0.7) + (number of Successes with Major Issues × 0.3) + (number of Failures × 0)] / total number of task attempts

We conduct remote and in-person testing to capture a wide range of user interactions. During both testing methods, the supervisor must remain silent and patient, allowing the user to either complete the task independently or decide to abandon it. A minimum of 10 users with varying experience levels will participate in the testing.

To increase external validity and mitigate the influence of artificial test conditions, we plan to complement these sessions with telemetry data from real-world usage, once platform logging features are in place.

TCR will be tracked at multiple stages of platform maturity, such as after onboarding, after major releases, or before public demonstrations. This repeated measurement supports longitudinal tracking and allows the evaluation of usability trends over time.

Additionally, TCR results will be segmented by user profile, including stakeholder type (e.g., data providers, scenario consumers, policy stakeholders) and experience level (novice vs. expert). This differentiation helps identify specific workflows that require refinement for targeted user groups.

Requirement References: TCR assesses how effectively users achieve key goals such as scenario search, filtering, or export. It directly reflects operational usability and the success of interface workflows (Req_48). High completion rates indicate that standardized outputs (Req_13) and search tools (Req_23) are well-implemented and usable, making TCR a critical usability and functional effectiveness indicator.

Target Value: A good target value for TCR KPI is typically above 80%. This indicates that most users can complete tasks effectively, even if some experience minor issues. In our case, since TCR is calculated as a weighted score that accounts for varying levels of difficulty in task completion, a target of 85% or higher is reasonable. This reflects a high level of usability while still allowing for occasional minor challenges.

4.2.5 KPI_05 Error Rate (ER)

Description: The Error Rate (ER) KPI measures the frequency of errors users encounter while interacting with the platform, such as failed tasks, crashes, or incorrect outputs. It quantifies usability issues that frustrate users and impede task completion, offering insights into potential technical flaws or design inefficiencies. A low Error Rate indicates a smooth, intuitive user experience, signalling that the platform functions effectively. Conversely, a high Error Rate highlights critical areas that need improvement. ER also correlates with user retention and satisfaction: when the Error Rate is low, users are more likely to trust and continue using the platform.

Measurement Methodology: The ER KPI measurement involves several steps. First, we define error types, including "task failures" (when users can't complete tasks due to platform issues), "system crashes" (unexpected shutdowns or freezes), "incorrect outputs" (faulty or incomplete data), and "user errors" (mistakes due to unclear instructions). Data collection includes tracking user interaction logs and system logs, analysing them offline to identify potential errors, crashes, and technical failures. Once data is gathered, errors are categorized using automated tools like error reporting software or session replay tools. The ER KPI is then calculated as the number of errors per unit of time spent using the platform, offering a time-based perspective on performance:

$$ER = (\text{Number of Errors} / \text{Total Time of Usage}) * 100$$

Errors should also be classified by severity (e.g., critical, major, or minor) to prioritize areas that require immediate attention. A representative sample of user interactions must be tracked to ensure the data is reflective of various devices, browsers, and use cases. Once the Error Rate is calculated, it will be tracked over time to identify trends or recurring issues. Contextual analysis should be conducted to understand when and why errors occur, helping to determine whether they are due to usability flaws or technical issues.

The purpose of this KPI is to evaluate the platform's reliability, it is crucial for identifying design flaws, technical issues, or unclear instructions that may hinder successful task completion.

Requirement References: This KPI tracks both user-facing and system-level errors (e.g., failed scenario exports, search crashes), providing quantitative evidence of usability bottlenecks and technical reliability issues. It supports debugging of data interface usability (Req_15), error-prone filters (Req_23), and overall robustness of the platform under typical use conditions (Req_48).

Target Value: For a server-side application, where uptime and performance are crucial but not mission-critical, a good ER over time target would be 1 error per 1000-2000 server hours for non-critical errors, and 1 critical error per 5000-10,000 server hours. This would ensure that the application remains stable and reliable without overloading the team with minor issues, while still addressing critical failures in a timely manner.

4.2.6 KPI_06 Trust & Compliance Assurance (TCA)

Description: The Trust & Compliance Assurance (TCA) KPI measures user confidence in the platform's ability to protect data, adhere to regulations (e.g., GDPR), and maintain ethical standards. It evaluates user perceptions of security, transparency, and regulatory alignment, ensuring that the platform meets both legal obligations and user expectations. This KPI is crucial for maintaining a strong reputation and fostering trust with users. High trust scores are linked to increased user engagement, loyalty, and reduced churn, while gaps in compliance can lead to legal risks and damage to the platform's credibility.

Measurement Methodology: To assess user trust in the platform's security and compliance practices, we will employ an Audit Compliance methodology to ensure the platform adheres to data protection regulations and meets industry security standards. This also includes evaluating the platform's transparency in data handling and security protocols. A detailed checklist is used to track the implementation and effectiveness of these compliance measures. The checklist covers tasks such as verifying up-to-date certifications, conducting regular security audits, monitoring incidents and their resolutions, and analysing user feedback on privacy and security. The checklist is implemented through Microsoft Forms and can be accessed at this link:

https://forms.office.com/e/bUniw1y4a4

To score the TCA KPI we use a weighted system based on four key components: Data Security (0.4), Regulatory Compliance (0.3), Transparency (0.2) and Ethical practices (0.1), totalling a possible 100 points:

*TCA= (Data Security Score * 0.4) + (Regulatory Compliance Score * 0.3) + (Transparency Score * 0.2) + (Ethical Practices Score * 0.1)*

Each component is scored on a scale from 0 to 100 based on the evidence gathered from the checklist.

To better understand how trust evolves over time, trust assessments will be conducted at multiple stages of the user journey, including: before initial platform use (baseline trust), immediately after key interactions, such as submitting personal data, and after security-related incidents or major platform changes. This pre/post comparison helps determine whether trust is being built, maintained, or diminished as users interact with the platform.

Requirement References: TCA evaluates perceived and actual adherence to data protection (GDPR), security best practices, and ethical standards. It aligns directly with privacy- and security-related requirements (Req_09, Req_49, Req_50, Req_51) and supports the credibility of the platform in scenarios requiring sensitive or regulatory-relevant data (e.g., those derived from ISMR reports or user uploads).

Target Value: Research industry benchmarks for similar KPIs. often aim for scores above 85. This indicates strong adherence to data protection regulations, robust security measures, and high user trust. For the SYNERGIES platform we aim for at least 90 compliances with the proposed checklist to ensure strong alignment with important regulations and to build user trust.

4.2.7 KPI_07 Cross- Browser and API Compatibility (CBAC)

Description: The Cross-Browser and API Compatibility (CBAC) KPI measures the platform's ability to maintain consistent performance across different web browsers (e.g., Chrome, Firefox, Safari, Edge) and to seamlessly integrate with external APIs. Ensuring cross-browser compatibility provides a uniform user experience, while API compatibility ensures reliable data exchange, effective error handling, and adherence to specifications. The KPI is a key predictor of user satisfaction and platform reliability. High compatibility means users encounter fewer errors, enjoy faster load times, and experience consistent functionality. This drives user engagement, builds trust, and enhances overall platform credibility.

Measurement Methodology: To assess CBAC KPI, we use a comprehensive testing approach that includes a detailed checklist implemented using Microsoft Forms, which can be accessed at this link:

https://forms.office.com/e/NRkjFHeUnh

We test the platform on major web browsers and their latest versions. The criteria for evaluation will include performance, layout, and functionality across these browsers. Automated testing tools such as Selenium is utilized to simulate user interactions and identify discrepancies. The platform's APIs will be tested with various client applications and systems. The criteria will ensure APIs adhere to standards like REST. API testing tools such as Postman will be used to validate endpoints, data formats, and response times.

The scoring method involves calculating scores for the two categories, Cross-browser compatibility (0.6 weight) and API compatibility (0.4 weight), by assigning each checklist item 2 points for full compliance, 1 point for partial compliance, and 0 points for non-compliance, computing each category score as a percentage of the achieved points over the maximum possible points, and then determining the final CBAC KPI by applying the weighted percentages:

CBAC = [(Total achieved points in Cross-browser compatibility / Maximum points in Cross-browser compatibility) × 0.6 + (Total achieved points in API compatibility / Maximum points in API compatibility) × 0.4] × 100

Requirement References: CBAP ensures the platform behaves consistently across different browsers and external system integrations. This reflects the quality of user experience across technical environments (Req_41), compatibility with open data standards (Req_12, Req_54), and ability to operate in federated settings (Req_43). It also supports stakeholder collaboration by ensuring that interfaces remain accessible regardless of technological context.

Target Value: The CBAC score can be interpreted as excellent (85–100), good (70–84), moderate (50–69), or poor (below 50), reflecting the overall compatibility status and guiding further improvements. The target value for the SYNERGIES platform is set at 90.

4.2.8 Usability Assessment KPIs Summary

The following Table 1 summarize the defined KPI with a brief description, the list of requirements reference, and the established target values:

KPI Name	Brief Description	Requirements Reference	Target Value
Customer Satisfaction Score (CSAT)	Measures user satisfaction with platform features and interactions.	Req_13, Req_15, Req_48	85%
System Usability Scale (SUS)	Evaluates ease of use, efficiency, and intuitiveness of the platform.	Req_08, Req_13, Req_48	75
Net Promoter Score (NPS)	Assesses user loyalty and likelihood to recommend the platform.	Req_48	60
Task Completion Rate (TCR)	Measures the percentage of users successfully completing tasks.	Req_13, Req_23, Req_48	85%
Error Rate (ER)	Quantifies the frequency of errors encountered by users.	Req_15, Req_23, Req_48	1 error per 1000-2000 server hours (non-critical), 1 per 5000-10,000 server hours (critical)
Trust & Compliance Assurance (TCA)	Evaluates user confidence in data security, transparency, and compliance.	Req_09, Req_49, Req_50, Req_51	90% compliance
Cross-Browser and API Compatibility (CBAC)	Assesses consistent platform performance across browsers and APIs.	Req_12, Req_41, Req_43, Req_54	90

Table 1 Usability Assessment KPIs

4.3 Technical assessment KPIs

This chapter defines the Technical Assessment KPIs established to evaluate the SYNERGIES platform's performance, robustness, scalability, and responsiveness under varying operational conditions. Each KPI offers a structured measurement framework that enables the project team to assess and optimize key technical dimensions of the platform through load testing, fault tolerance analysis, system monitoring, and benchmarking against defined performance thresholds.

4.3.1 KPI_08 System Responsiveness (SR)

Description: This KPI measures the platform's ability to process and respond to user requests efficiently under varying workloads, such as concurrent scenario searches or real-time simulations. It is essential for assessing performance across a broad spectrum of use cases, from small research tasks to complex workflows. This KPI comprises three core metrics: Response Time, which captures the duration required by the system to return query results; Concurrency, indicating the platform's ability to execute multiple processes in parallel; and Scalability, reflecting the platform's capacity to handle increased data volumes and user traffic without performance degradation.

Latency is also monitored as a subcomponent of response time. In this context, we define latency as the internal system processing time, excluding network transmission delays and client rendering. Response Time, by contrast, captures end-to-end duration from request initiation to response delivery, including all intermediate components.

Measurement Methodology: The KPI evaluation is conducted under simulated load conditions to assess the platform's stability and speed. Apache JMeter is used to generate virtual users executing various tasks such as scenario searches and data exports, with incremental increases in concurrency. Metrics collected during tests include Average Response Time (duration of request completion), Throughput (requests processed per second), and Error Rate (percentage of failed requests). Maximum Response Time is calculated as the total sum of individual request durations divided by the number of requests, and Standard Deviation to assess performance variance across requests).

These distribution metrics are critical for evaluating the platform's performance stability and ensuring that occasional outliers do not severely impact user experience.

Requirement References: SR monitors how quickly the platform processes scenario searches and API requests under load, addressing concerns related to latency, throughput, and concurrency. It is essential for assessing performance under virtual test environments or batch scenario operations (e.g., Req_05, Req_40), and supports stress testing in scalable use cases (e.g., for Req_32, Req_27).

Target Value: The target values for this KPI are defined to ensure a responsive and reliable user experience: a query response time of less than 1000 milliseconds is set to guarantee that users receive timely feedback during interactions; latency is kept below 200 milliseconds to minimize delays in real-time operations and maintain fluid system performance; and an error rate threshold of 10% is acceptable for minor failures (e.g., due to external service delays), however, the target error rate for production-quality performance is set to <1%, based on industry benchmarks for API reliability.

4.3.2 KPI_09 Extensibility (EXT)

Description: This KPI measures the platform's ability to handle growth in users, data, and functionalities without performance degradation or requiring major redesigns. It assesses the platform's long-term scalability, adaptability, and structural robustness, ensuring that it can evolve and expand while maintaining a high level of technical performance and maintainability.

Measurement Methodology: The extensibility of the platform is evaluated using static code analysis via SonarQube. This methodology allows for checking software quality indicators across the codebase. SonarQube provides automated, scalable analysis and produces detailed metrics reports related to code complexity (based on cyclomatic complexity, which reflects the number of independent execution paths through a function; lower values generally indicate simpler and more maintainable code), maintainability, duplication, test coverage, and technical debt (quantified as the estimated time required to fix code quality issues and bring the codebase up to standard). These metrics are critical to understanding how easily the system can be modified, extended, or integrated with new features or components.

Requirement References: This KPI assesses the platform's architectural ability to accommodate new scenario formats, evaluation modules, or AI tools without major rework. It supports maintainability and long-term evolution (Req_38), fosters modular integration (Req_03, Req_20), and helps accommodate future co-simulation or adversarial scenario generation tools (Req_31, Req_25).

Target Value: To ensure system extensibility, specific benchmarks have been defined: code complexity should remain below 10 per function to promote clarity and ease of maintenance; the maintainability index should be as high as possible, reflecting lower long-term maintenance costs and simpler code evolution; duplicated code should be kept below 10% to avoid redundancy and minimize the risk of future errors; code coverage should exceed 50%, ensuring that core functionalities are adequately tested; and technical debt should be kept to a minimum, indicating reduced rework effort and greater adaptability for future development.

4.3.3 KPI_10 Security Evaluation (SE)

Description: This KPI measures the platform's ability to protect data and privacy while preventing unauthorized access. It evaluates how effectively the platform mitigates security threats, safeguards sensitive information, and maintains user trust through adherence to robust security practices.

Measurement Methodology: To assess the platform's security posture, a detailed technician checklist will be employed, it can be found at the following link:

https://forms.office.com/e/ijAhYbg5pg

This checklist is designed to evaluate compliance with security best practices, including mechanisms for authentication, authorization, data protection, and error handling. The evaluation will be performed by a qualified research member to ensure accuracy and objectivity. The Security Score is calculated as the percentage of completed checklist items relative to the total, using the formula:

$$SS = (\text{Number of completed checklist items} / \text{Total number of checklist items}) \times 100$$

To ensure that security is maintained over time, the evaluation is conducted at three critical stages: before each major release, to validate that no new security regressions are introduced.; and after any change in authentication or data handling modules, to verify the continued compliance of sensitive components,

Requirement References: SE measures the actual implementation of secure access control, encrypted communication, and role-based data handling policies. It supports safe operations within distributed systems (Req_18), secure cloud deployments (Req_51), and auditability of platform modifications for regulatory reporting (Req_56). It also contributes to legal compliance verification (Req_50).

Target Value: To meet acceptable security standards, the system must achieve a checklist completion score above 70%. This is defined as a minimum operational threshold, ensuring that most critical controls are implemented and verified. While 70% is acceptable for pre-deployment readiness, the long-term goal is to exceed 85%, aligning with best-practice benchmarks for research platforms handling sensitive data.

4.3.4 KPI_11 System Resilience (SRE)

Description: This KPI measures the platform's ability to endure and recover from failures or disruptions, ensuring continuous operation and minimizing downtime. It assesses how well the system performs under stress conditions such as hardware failures, network interruptions, or unexpected traffic surges, providing insights into its robustness and fault tolerance.

Measurement Methodology: The system's resilience is evaluated through controlled failure simulations using tools like Chaos Monke. we randomly terminate components (e.g., virtual machines or microservices) in a production-like environment, while the platform manages continuous delivery and automated recovery processes. Failures are injected across a representative subset of platform components, including API gateways, processing microservices, and storage backends, selected to reflect critical user-facing functions. The test coverage evolves over time to include new components introduced in future releases.

Tests are executed on a recurring basis (at minimum once per release or infrastructure changes) to ensure resilience is consistently validated across evolving system configurations.

Key performance metrics collected include Mean Time to Recovery (MTTR), Error Rate, System Uptime, and Latency Increase. The following formulas are used:

$$MTTR = \text{Total Downtime} / \text{Total Number of Failures}$$

$$\text{Error Rate} = (\text{Failed Requests} / \text{Total Requests}) \times 100$$

$$\text{System Uptime} = (\text{Total Uptime} / \text{Total Test Time}) \times 100$$

$$\text{Latency Increase} = (\text{Latency After Failure} - \text{Latency Before}) / \text{Latency Before}$$

Requirement References: SRE evaluates the platform's ability to remain operational during service interruptions and recover from failures. It addresses needs for robust operations during scenario ingestion, simulation, or export (Req_07), and supports traceability and fault tolerance in deployments with in-service monitoring needs (Req_45). It also supports versioning and rollback features (Req_39).

Target Value: To demonstrate sufficient resilience and maintain operational continuity under failure scenarios, the platform must meet the following performance thresholds during controlled stress testing: a Mean Time to Recovery (MTTR) of less than 10 seconds to ensure rapid fault recovery and minimal disruption; an Error Rate below 10% to confirm that most requests are successfully handled even during failure conditions; a System Uptime greater than 70% to validate that the platform remains operational for the majority of the test duration; and a Latency Increase of less than 10% compared to baseline to ensure that responsiveness remains within acceptable limits under stress. These targets collectively ensure that the platform is robust, fault-tolerant, and capable of sustaining reliable service continuity in the face of unexpected disruptions.

4.3.5 Technical Assessment KPIs Summary

The following Table 2 summarize the defined KPI with a brief description, the list of requirements reference, and the established target values:

KPI Name	Brief Description	Requirements Reference	Target Value
System Responsiveness (SR)	Measures the platform's ability to efficiently respond to user requests under varying workloads, ensuring performance across different use cases.	Req_05, Req_27, Req_32, Req_40	Query Response Time < 1000 ms; Latency < 200 ms; Error Rate < 10%
Extensibility (EXT)	Evaluates the platform's capacity to handle growth and change without redesign, focusing on scalability, maintainability, and code quality.	Req_03, Req_20, Req_25, Req_31, Req_38	Code Complexity < 10; Duplicated Code < 10%; Code Coverage > 50%; Minimized Technical Debt
Security Evaluation (SE)	Assesses how well the platform protects sensitive data, prevents unauthorized access, and complies with security best practices.	Req_18, Req_51, Req_50, Req_56	Security Score > 70%
System Resilience (SRE)	Evaluates the platform's ability to recover from failures and disruptions while maintaining availability and performance.	Req_07, Req_39, Req_45	MTTR < 10s; Error Rate < 10%; Uptime > 70%; Latency Increase < 10%

Table 2 Technical Assessment KPIs

4.4 Interoperability KPIs

This chapter defines the Interoperability KPIs established to evaluate the SYNERGIES platform's ability to exchange, integrate, and interpret data consistently across diverse systems and services. These KPIs are designed to ensure that the platform's APIs not only conform to structural standards but also behave according to defined business logic and support meaningful semantic exchange. Each KPI in this category provides a structured assessment of how effectively the platform enables cross-system communication, functional coherence, and data fidelity. Through checklist-based evaluations and real-world scenario testing, these KPIs ensure that the platform upholds interoperability principles essential for federated architectures, distributed collaboration, and long-term scalability in heterogeneous environments.

4.4.1 KPI_12 API Intent Assessment (AIA)

Description: This KPI measures whether the platform's API behaves as intended, beyond structural correctness, by validating its adherence to expected functional behaviour, business logic, state management, and security policies. It evaluates whether the API implementation not only conforms to defined specifications but also performs reliably and consistently in real-world usage scenarios.

Measurement Methodology: The evaluation is carried out through a structured technician checklist designed to validate the API's compliance with its intended functionality. It can be found at the following link:

https://forms.office.com/e/RGqhBVnVqF

Key assessment areas include correct handling of data and state, compliance with business logic rules, error response behaviour, and proper enforcement of security constraints. The checklist includes both static inspection and dynamic testing of API endpoints using representative scenarios. Each item on the checklist is marked as complete when the corresponding behaviour is successfully verified. The overall performance is quantified using the following formula:

API Intent Score = (Number of completed checklist items / Total number of checklist items) × 100

The checklist will be reviewed and completed by qualified research personnel with knowledge of the API specification and platform architecture, ensuring accurate and consistent evaluation.

Requirement References: AIA checks whether APIs deliver responses consistent with their specifications, enforcing correct logic, validation, and error handling. It supports the reliability of federated database integration (Req_43) and ensures consistent behaviour during scenario manipulation or export workflows, vital in regulatory use cases or batch operations.

Target Value: A baseline score of 60% is defined as the minimum acceptable threshold only for early development or integration phases, where selected endpoints or partial workflows are under active implementation and validation. However, for stable releases or APIs used in regulatory or production-critical workflows, the required target is raised to at least 85%. This indicates that the API behaves consistently with its documented purpose and meets the functional expectations of the platform.

4.4.2 KPI_13 Semantic Interoperability of API (SIA)

Description: This KPI evaluates the platform's ability to ensure that different systems can correctly interpret, and process exchanged data, even when information is presented in varying formats. It measures whether the API supports consistent, meaningful, and machine-readable data exchange that enables seamless integration and communication across distributed tools and services.

Measurement Methodology: To assess the API's semantic interoperability, a structured technician checklist will be employed, it can be found at the following link:

https://forms.office.com/e/2WtHU0L3Uw

This checklist examines the API's design, data representation practices, schema consistency, and compliance with interoperability standards. The goal is to determine whether data exchanged through the API retains its intended meaning across different client systems. Both static interface analysis and scenario-based testing are used to verify correct interpretation and handling of structured data.

A score is calculated using the following formula:

Semantic Interoperability Score = (Number of completed checklist items / Total number of checklist items) × 100

Evaluation will be conducted by qualified research staff familiar with API design and semantic modelling. Their role is to verify that data representations and interfaces align with interoperability goals.

Requirement References: SIA focuses on whether the APIs and data formats used by the platform maintain semantic integrity across heterogeneous systems. This supports harmonized integration (Req_01), standardized scenario descriptions for cross-country use (Req_21), community reuse and alignment (Req_58), and ensures data meaning remains stable across platform updates (Req_39).

Target Value: A Semantic Interoperability Score above 70% is required to demonstrate acceptable conformance. This indicates that the API can sustain semantic integrity across diverse environments and use cases.

4.4.3 Interoperability Assessment KPIs Summary

The following Table 3 summarize the defined KPI with a brief description, the list of requirements reference, and the established target values:

KPI Name	Brief Description	Requirements Reference	Target Value
API Intent Assessment (AIA)	Measures whether the API behaves according to its intended functional, logical, and security design, beyond mere structural validity.	Req_43	API Intent Score ≥ 60%
Semantic Interoperability of API (SIA)	Evaluates whether data exchanged via the API is semantically meaningful and correctly interpreted across different systems and formats.	Req_01, Req_21, Req_58, Req_39	Semantic Interoperability Score > 70%

Table 3 Interoperability Assessment KPIs

4.5 Data Collection Techniques and procedures

Accurate and structured data collection is essential for evaluating the platform. Data is collected through a combination of complementary techniques, each addressing specific aspects of system evaluation.

One fundamental technique is **logging analysis**. This involves gathering and inspecting system-generated logs that record user activities, system events, and resource usage during operation. Through log analysis, it becomes possible to identify performance bottlenecks, detect failures, and measure system responsiveness. Since logs are collected automatically in the background, this technique does not interfere with the user experience and enables continuous monitoring over extended periods. In the context of CCAM, driving-related logs such as scenario search patterns, federated node responses, and simulation executions are particularly critical. These allow developers to evaluate system behaviour under real-world-like CCAM workloads.

Another key method for data collection is **user feedback**, obtained through surveys, interviews, and questionnaires. Surveys and questionnaires provide structured forms for users to express their opinions about system usability, performance, and satisfaction. Interviews, on the other hand, allow for more open and in-depth discussions, uncovering nuances that may not emerge from standard questionnaires. This direct input from users is invaluable for identifying areas of improvement and aligning system development with user needs. For CCAM systems, early feedback from domain-specific stakeholders such as scenario providers, simulation users, and regulatory actors is vital to ensure interoperability, traceability, and ease of use across the ecosystem.

To systematically evaluate system functionality, experts rely on **checklists**. These checklists list predefined criteria that the system must meet and guide evaluators in assessing compliance. They ensure consistency in evaluations across different experts and instances and facilitate the identification of functional gaps or defects.

Automated testing tools complete the range of data collection techniques. These tools simulate user behaviour, run test cases, and automatically record results. Automated testing not only verifies system functionality but also measures reliability, scalability, and robustness under different scenarios, providing an objective and repeatable means of evaluation. Where appropriate, such tools are configured for batch evaluation of large datasets and can operate under CI/CD pipelines. In future stages, integration with streaming analytics platforms may allow real-time assessment of KPIs as data is ingested.

The comprehensive collection and management of data necessitate a dedicated tool capable of supporting all the described techniques and procedures (see KPI Manager chapter). The KPI Manager is not a passive repository but an extensible computation engine. It supports modular plugins to compute specific KPIs based on structured input, such as log parsers, scoring functions for questionnaires, and validators for checklist compliance. It ensures traceability by logging contributor identities, and time-stamping every data input. All KPI evaluations are reproducible and auditable. This tool must facilitate interaction with end-users and experts, offering interfaces for completing questionnaires and checklists. It must automatically collect, store, and organize user responses. It should integrate with system logs, parsing them, and support visualization and analysis of performance data. Furthermore, it must collect automated tests their results and consolidate all collected data into a unified KPI database. To ensure scalability, the KPI Manager supports asynchronous batch processing of large data volumes and is designed for deployment on cloud infrastructure using containerized services (e.g., via Docker/Kubernetes).

Privacy and data protection are embedded throughout the evaluation pipeline. GDPR-compliant practices are enforced, including consent forms for user-facing questionnaires and interviews, pseudonymization of log data, and role-based access to sensitive performance or feedback data. Interviews undergo ethical review where applicable, and raw data is never exposed without proper anonymization or authorization.

Figure 3 represents the workflow for data collection and KPI management:

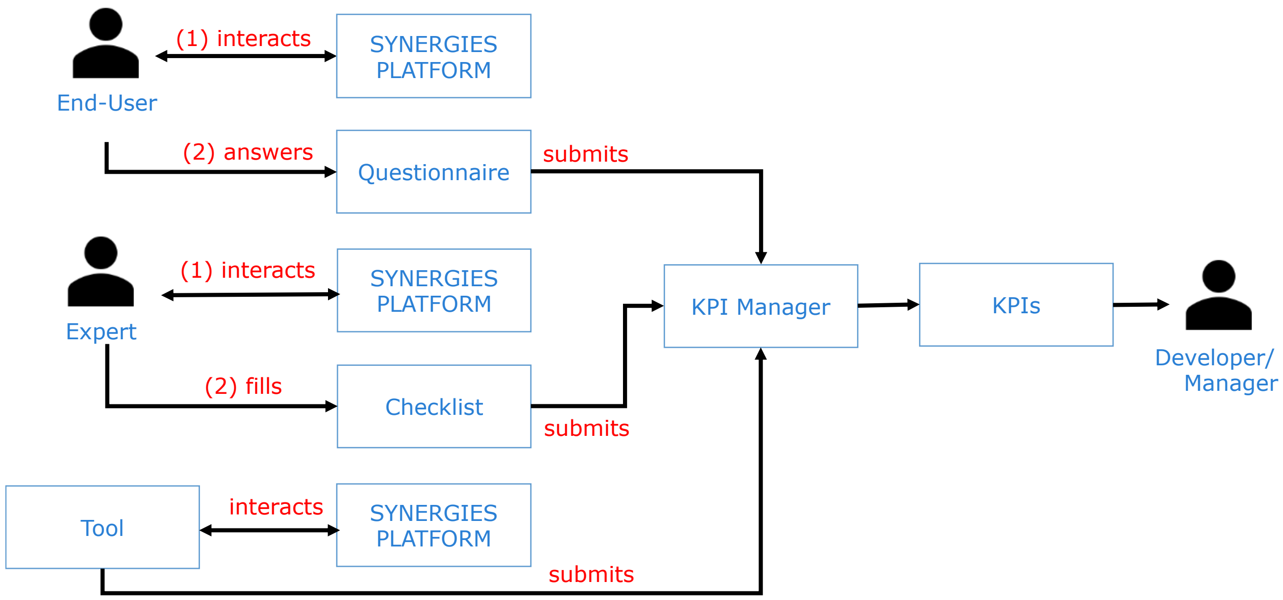

Figure 3 Data collection procedure

First, the **end-user interacts with the platform.** As part of this interaction, the user is **prompted to answer a questionnaire**. After answering, the questionnaire is **submitted to the KPI Manager**, where the data is processed for KPI evaluation. Similarly, **experts interact with the platform** and **fills out a checklist** evaluating system functionality. This **checklist is also submitted to the KPI Manager** for processing. In parallel, **automated tools interact with the platform**, performing system operations and **generating log data or test results**. This data is likewise submitted to the KPI Manager. The **KPI Manager consolidates all incoming data**, computes the relevant KPIs, and delivers these KPIs to developers and managers. Each KPI computation is timestamped, versioned, and stored in an auditable form to support future certification or regulatory reporting. This structured process ensures that all aspects of system performance, functionality, and user satisfaction are captured systematically and are available for informed decision-making.

5 KPI MANAGER

This chapter describes the KPI Manager, a cloud-based tool designed for the efficient collection, visualization, and export of KPIs. It provides an overview of the system's architecture, the implementation using the Measure framework, and the technologies used. Additionally, it details the deployment strategy.

5.1 Overview

The KPI Manager is a cloud-based solution designed to facilitate the **storage**, **management**, and **analysis** of KPIs. Its primary objective is to provide developers and managers with an efficient, scalable, and user-friendly platform for monitoring system performance through a centralized interface. To ensure scalability, high availability, and ease of integration with external services, the system is built on a **cloud-based architecture**. This approach not only guarantees seamless remote access but also supports future expansions without requiring major infrastructure modifications.

The selection of Measurify as the underlying platform for KPI collection and management in SYNERGIES is grounded in both **technical merit** and **proven effectiveness** in prior EU research initiatives. Measurify has demonstrated scalability, modularity, and ease of integration in real-world deployments and has been successfully adopted in EU-funded CCAM projects such as **L3Pilot** and **Hi-Drive**, where it served as the central tool for structured KPI acquisition and performance monitoring.

In particular, L3Pilot employed Measurify to monitor key usability and technical indicators related to automated driving experiences, handling both subjective data (e.g., user questionnaires) and objective sensor-derived metrics across a large, heterogeneous data ecosystem. Similarly, in Hi-Drive, Measurify supported the evaluation of complex, multi-actor connected vehicle scenarios by managing high-frequency inputs and providing reproducible, traceable KPI reporting. These projects have demonstrated that Measurify is capable of operating in demanding CCAM contexts, aligning well with the requirements of SYNERGIES.

Beyond functional performance, the adoption of Measurify reflects a strategic decision **to reuse and extend technological assets developed in previous EU research**, maximizing the return on public investment and ensuring continuity in evaluation methodologies across related projects. This reuse accelerates development, reduces integration risk, and fosters interoperability with other CCAM platforms adhering to similar evaluation frameworks.

For SYNERGIES, Measurify provides a robust foundation to:

- Integrate user-facing questionnaires, expert checklists, and automated tests;
- Process and store logs and performance metrics;
- Compute complex KPIs across multiple categories (usability, technical, interoperability);
- Ensure data provenance, traceability, and GDPR compliance.

Thus, Measurify is not only a technically sound choice but also a strategically aligned and field-proven platform that reinforces the continuity, efficiency, and interoperability goals of the SYNERGIES Evaluation Framework.

At the core of the KPI Manager lies the **Measurify framework** (https://github.com/measurify), an open-source platform offering a flexible and structured environment for KPI management. Measurify allows users to define, store, and retrieve KPI data through standardized APIs, and provides capabilities for data visualization, export in multiple formats, and automated analysis, streamlining the entire KPI lifecycle.

5.2 Architecture and Technology Stack

The KPI Manager is built on the Measurify architecture, which adopts a cloud-based design focused on scalability, maintainability, security, and seamless communication with external systems through standardized APIs.

The system follows a **service-oriented architecture**, where the database, API interface, and graphical user interface are decoupled into distinct components. This modular structure ensures that data collection, transmission, processing, storage, and visualization are handled by specialized services, enhancing system robustness, flexibility, and ease of maintenance.

Figure 4 shows a high-level view of the system based on the Measurify framework:

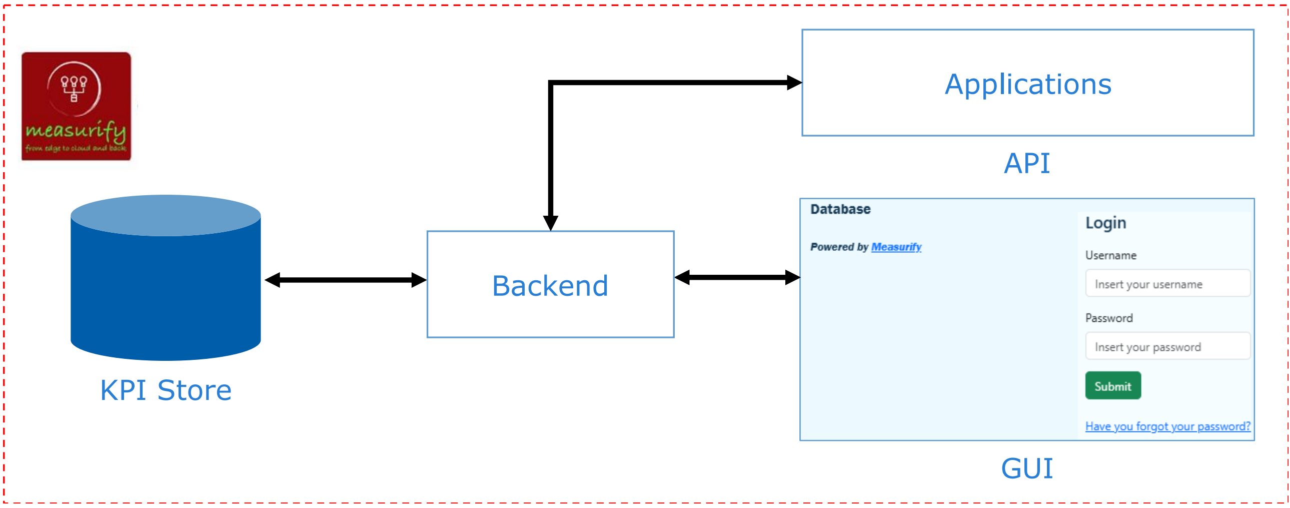

Figure 4 KPI Manager Architecture

The architecture consists of the following core components:

- **KPI Store**: the system leverages **MongoDB**, a widely adopted NoSQL database renowned for its flexibility, high performance, and scalability. Its document-oriented model is particularly well-suited for storing structured KPI data, especially in scenarios involving dynamic schemas (since KPI definition can change over time to adapt to the Platform evolution) and real-time requirements. MongoDB provides high throughput for both read and write operations, along with native support for time-based queries and indexing. Designed for seamless integration with cloud environments, MongoDB is optimized to efficiently manage a large volume of incoming resources, each containing relatively small amounts of data. This makes it especially ideal for use cases that involve frequent, lightweight updates to KPI records.
- **Backend**: this software layer is implemented in **Node.js** acts as the backend interface for receiving, validating, and organizing KPI data before storing it in the KPI Store. Measurify enforces a structured model for defining and managing measures, ensuring consistency, flexibility, and seamless integration with external systems. The backend emphasizes security through data validation, authentication, and controlled access to sensitive endpoints. It exposes a **RESTful Application Programming Interface (API)** that supports secure transmission, ingestion, and retrieval of KPI data. Designed for interoperability, the API uses standard HTTPS methods and structured data formats, enabling easy integration with third-party tools, dashboards, and automated pipelines across a wide range of platforms.

- **Graphical User Interface (GUI)**: this interface is developed using **React**, a JavaScript library for building responsive and dynamic web applications. It offers users intuitive access to the KPI Manager's functionalities, including data submission, real-time monitoring, data visualization, and configuration of KPI sources. As a web-based application, the customized GUI enables interactive exploration of KPIs through filters, charts, tables, and dashboards. Designed with usability and responsiveness in mind, it delivers a seamless experience across different devices and facilitates direct interaction with the underlying APIs for efficient data querying and management.

This architectural approach ensures that the KPI Manager remains flexible and optimized, enhancing high level of performance, security and interoperability.

5.3 Customization

The Measurify framework is designed to receive data from multiple sources in various formats, including CSV, XLSX, JSON, as well as manually entered data (see Figure 5):

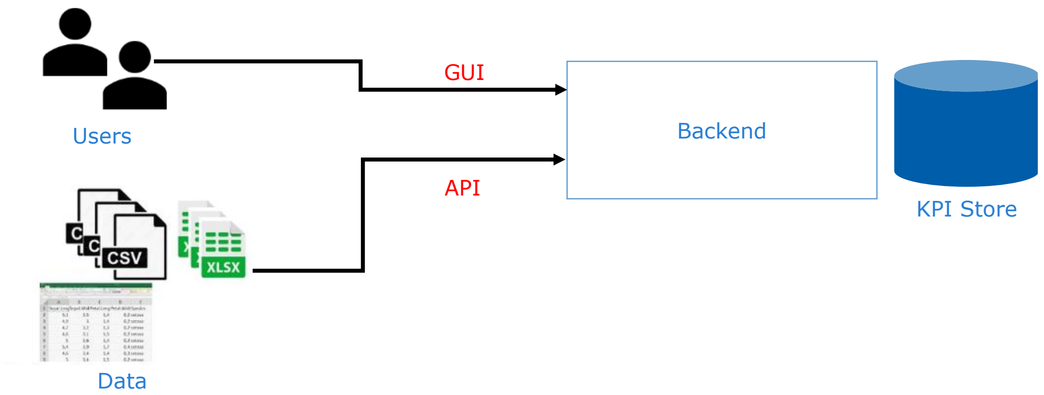

Figure 5 Data sources

This flexibility allows the platform to seamlessly adapt to heterogeneous environments and support a broad range of use cases. To achieve this, Measurify provides a robust data model based on well-defined software entities, ensuring efficient and consistent management of data tailored to KPI management tasks. Figure 6 illustrates this data model, emphasizing the concept of a **Measurement**, an abstraction of a data point that can represent either a simple numerical value or a more complex structure enriched with contextual information, such as how, when, and where the data was collected.

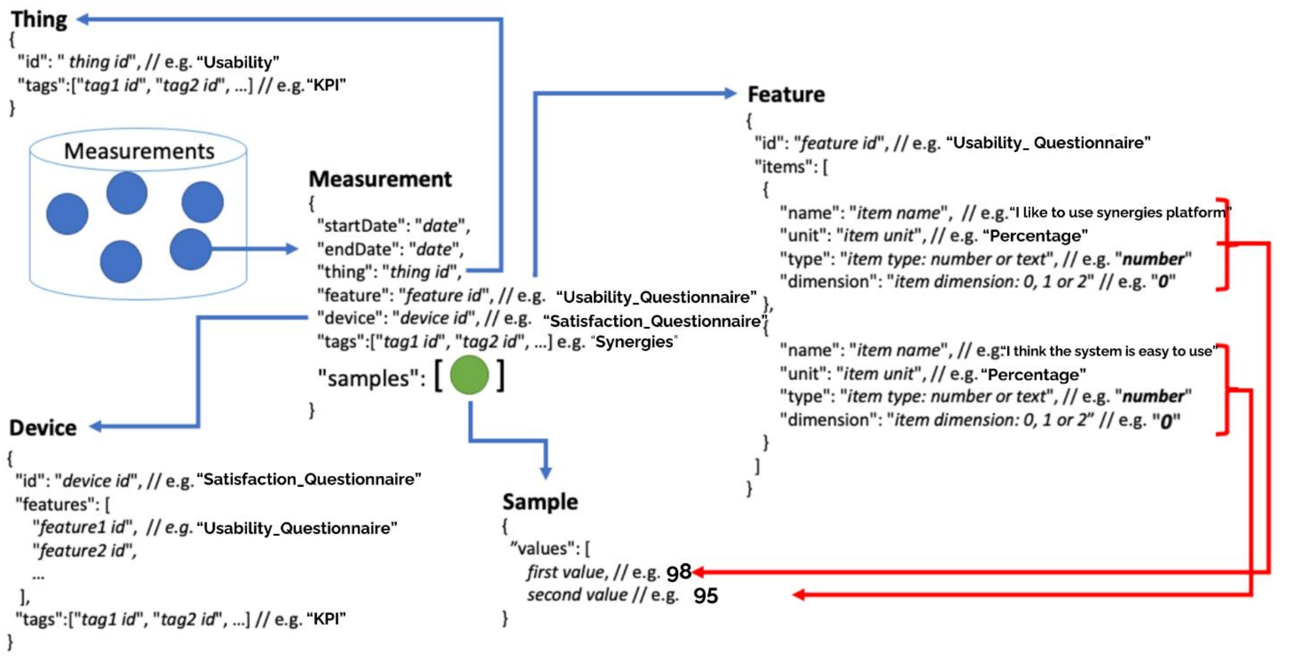

Figure 6 Measurify entities

The foundation of the Measurify data model is the entity **Thing**, which defines the object or concept to which a measurement refers. In the KPI Manager implementation, a Thing enables the categorization of KPIs into distinct domains (such the Usability, the Technical Assessment, or the Interoperability Assessment) providing semantic grouping and improving the clarity of KPI organization. This categorization allows stakeholders to navigate large sets of KPIs more intuitively and supports domain-specific analysis.

Each Measurement in the model is linked to a **Feature**, which describes the expected structure and semantics of the associated data. A Feature acts as a template, containing one or more Items. Each Item specifies critical attributes such as:

- The name of the KPI (e.g., "System Response Time")
- The unit of the stored datapoint (e.g., percentage, seconds)
- The data type (e.g., numerical, textual)
- The cardinality, indicating whether it represents a single value or a list of values.

This structured approach ensures that all collected data adheres to predefined validation rules, enhancing the consistency, reliability, and interpretability of the KPIs across different use cases.

The **Device** entity represents the specific source through which the data is collected, such as a user questionnaire, a monitoring script, or an external third-party tool. By associating each device with specific features, the system maintains complete traceability of the data's origin. This traceability is crucial for validating measurements, analysing source-specific variations, and ensuring accountability during audits or evaluations.

The **Sample** entity holds the actual recorded values, for example, values such as “98” or “95” as shown in Figure 6. A Sample is embedded within a Measurement object, which also contains metadata including:

- The timestamp of data collection
- References to the associated Thing, Feature, and Device.

This design ensures that every recorded value is fully contextualized within its measurement environment, both in terms of its domain (Thing) and its source (Device).

Additionally, all entities (Things, Features, Devices, and Measurements) can be annotated with one or more **Tags**, which serve as flexible metadata labels that provide further contextual information, supporting advanced querying, filtering, and grouping of KPIs based on project-specific or domain-specific criteria.

Figure 7 illustrates the two **main logical flows** within the system: **configuration** and **measurement ingestion**. The configuration flow involves setting up Features based on predefined KPI structures, ensuring that all incoming data conforms to expected formats and validation rules. The measurement ingestion flow represents the continuous stream of KPI data collected from users and tools, which is then stored, processed, and made available for real-time monitoring and analysis.

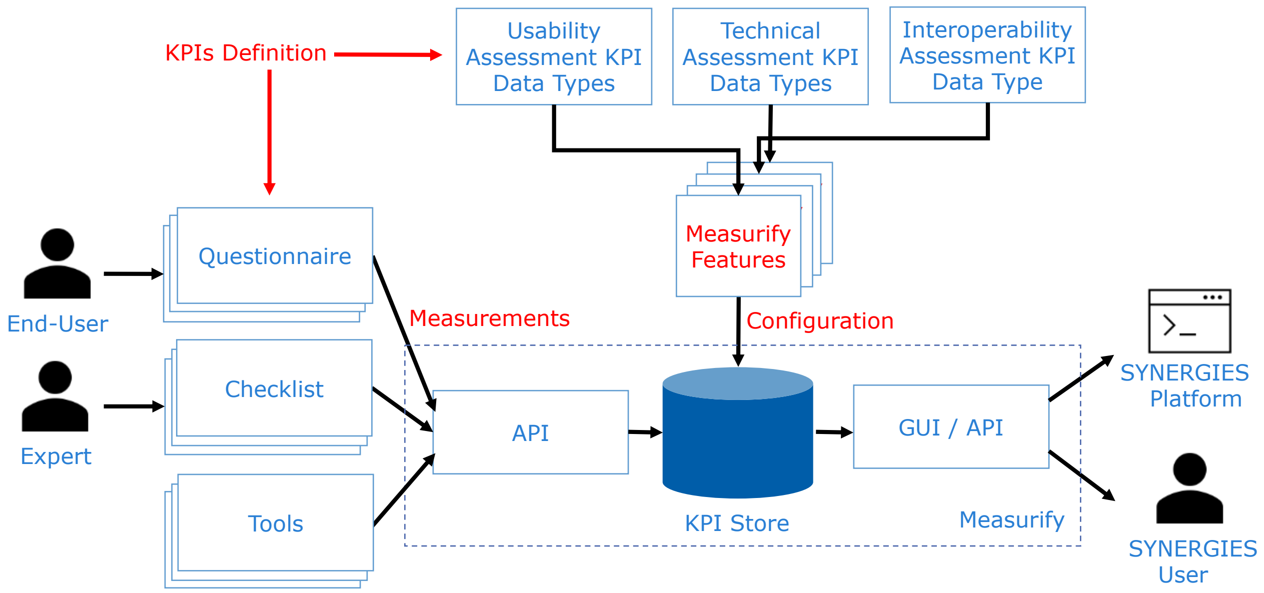

Figure 7 KPI Manager implementation

In summary, the Measurify entities act as a comprehensive container, bringing together the data values, their semantic definitions, their sources, and their temporal context. This structured approach enables consistent, traceable, and meaningful management of KPIs, making the system highly adaptable to diverse operational and analytical needs.

5.4 Functionality

The operation of the KPI Manager starts with the definition of the core entities that structure each KPI, including things, features, and devices. These entities can be configured either through a **dedicated graphical interface** that allows manual creation and editing (as shown in the manual feature creation example in Figure 8), or by **uploading a JSON file** containing the complete specification (as illustrated by the feature JSON file example in Figure 9)

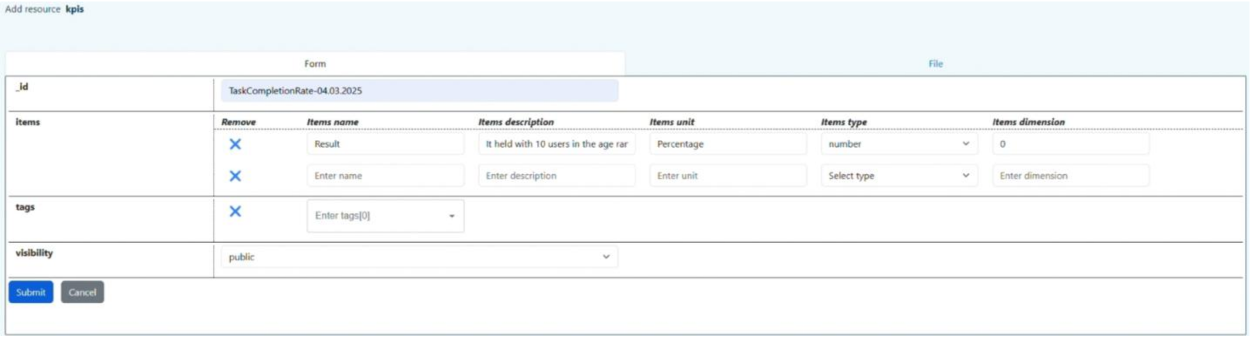

Figure 8 Manual creation of a Feature entity


```json
{
    "_id": "SUS-04.03.2025",
    "items": [
        {
            "name": "I like to use synergies platform frequently",
            "description" : "It shows satisfaction of user",
            "unit": "Percentage",
            "type": "number",
            "dimension": 0
        },
        {
            "name": "I found the system unnecessarily complex",
            "description" : "It shows dissatisfaction of user",
            "unit": "Percentage",
            "type": "number",
            "dimension": 0
        },
        {
            "name": "I thought the system was easy to use",
            "description" : "It shows satisfaction of user",
            "unit": "Percentage",
            "type": "number",
            "dimension": 0
        },
        {
            "name": "I think that I would need the support of a technical person to be able to use this system",
            "description" : "It shows dissatisfaction of user",
            "unit": "Percentage",
            "type": "number",
            "dimension": 0
        },...
```


Figure 9 JSON file for Feature entity

Once the entities are defined, the system enables data collection through both **automated and manual methods**. Automated collection is managed via the RESTful API endpoints, allowing seamless integration with external systems and services. Manual input is supported through the frontend interface, where users can directly enter or modify KPI values as needed.

Furthermore, the system supports the **upload of CSV files**, such as questionnaires (as shown in Figure 10). In this case, the platform automatically maps the CSV columns to the corresponding KPI resources based on the predefined structure of features and items, streamlining the import process and minimizing the risk of mismatches or data loss.

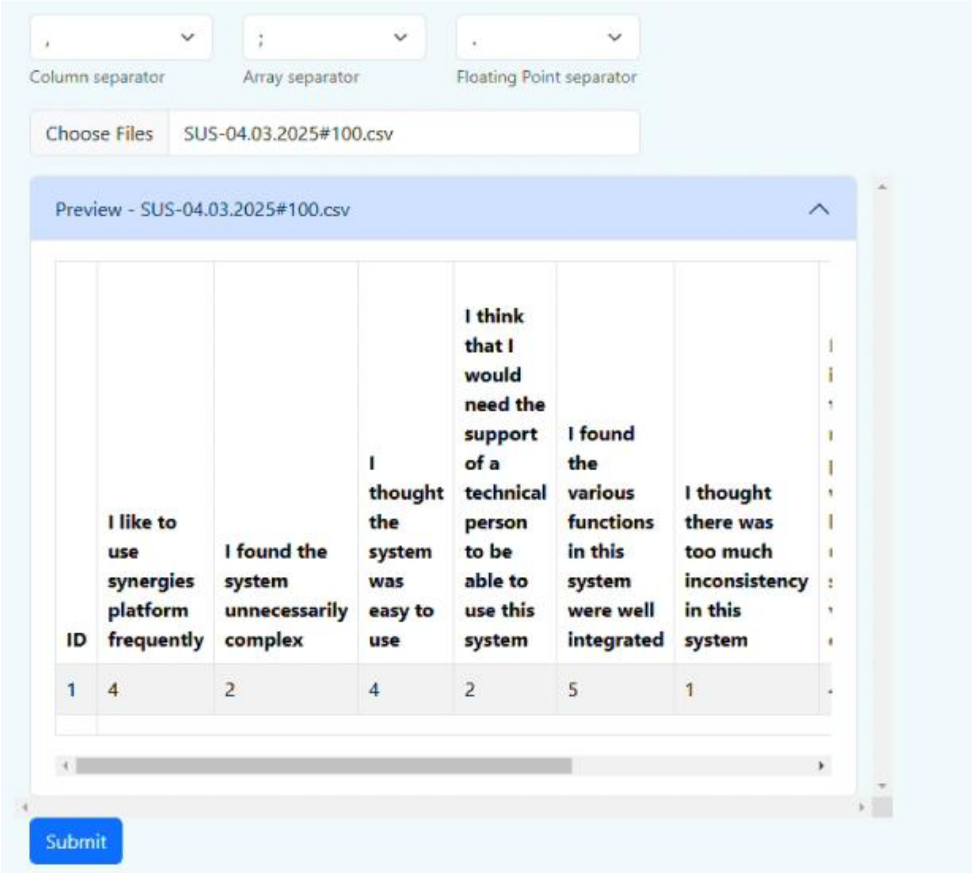

Figure 10 Questionnaire uploading phase

For data visualization, the platform provides an **interactive dashboard** that presents real-time KPI data through dynamic charts (as shown in Figure 11) and tables. Users can monitor trends, filter results by tags or time ranges, and quickly identify anomalies or changes in performance, ensuring effective and immediate insights into system behaviour.

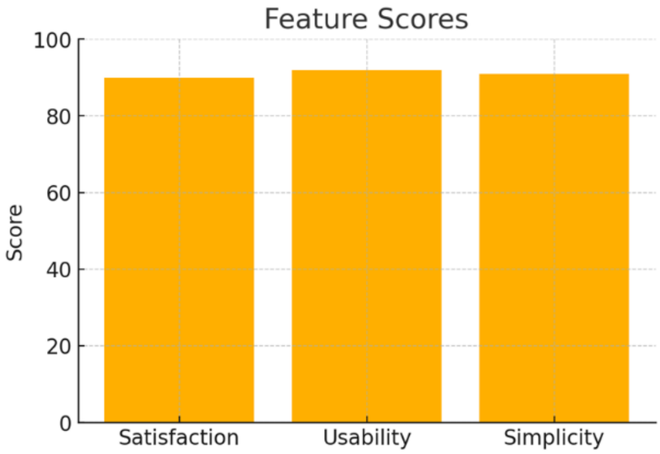

Figure 11 Score graph

The system also supports **API-based integration**, enabling seamless connection with third-party services, analytics platforms, and external databases for automated data exchange. Additionally, KPI data can be **exported in multiple formats**, including CSV, JSON, and PDF, to facilitate reporting, offline analysis, and easy sharing with managers or stakeholders.

5.5 Deployment

The deployment of the KPI Manager requires an active online server infrastructure, hosted in the cloud to guarantee high availability, scalability, and remote access. The system is deployed on **Amazon Web Services (AWS)**, utilizing EC2 virtual machines for both **testing** and **production environments**.

For the test environment, a lightweight a **small T2 instance** is used, equipped with 1 vCPU, 2 GB of RAM, and running Ubuntu Server 24.04 LTS. This configuration is suitable for development activities, functional testing, and interface validation, offering a cost-effective and flexible solution during the early stages of deployment.

The production environment relies on a more powerful **medium T2 instance**, providing 2 vCPUs, 4 GB of RAM, and the same operating system, Ubuntu Server 24.04 LTS. This setup ensures the necessary performance and stability for real-time KPI ingestion, processing, and dashboard access, supporting the system's typical operational workload.

To facilitate accessibility and maintain a clear separation between testing and production, user-friendly DNS names are employed. The test environment is accessible at synergies-test.kpi-manager.org while the production deployment is available at synergies.kpi-manager.org .

Operational procedures include **periodic system updates**, **scheduled backups** of the MongoDB database, and continuous automated monitoring of CPU and memory usage.

Backups of the database are performed automatically on a daily basis at 02:00 AM CET. These backups include both the production KPI data and the associated metadata (e.g., definitions of features, things, and devices). Backup files are stored in encrypted Amazon S3 buckets with versioning enabled to support point-in-time recovery. Retention policies ensure that daily backups are stored for 30 days, with one weekly snapshot retained for six months to support long-term recovery and rollback scenarios.

The monitoring system is configured to track key infrastructure metrics, including CPU utilization, memory consumption, disk I/O, and uptime status. Thresholds are set to trigger alerts when CPU usage exceeds 80% for more than 5 minutes, or when memory usage surpasses 75%, indicating potential performance bottlenecks. Alerts are managed through AWS CloudWatch and routed via email notifications to the system administrators.

Additionally, **system logs** (including API access logs, error logs, and audit logs) are archived daily to Amazon S3 buckets. These logs are retained for at least 12 months and serve multiple purposes: performance trend analysis, failure diagnosis, GDPR-compliant access audits, and long-term compliance reporting.

Usage statistics (such as the number of KPI submissions, API call frequency, and feature-specific load) are also periodically summarized and stored, enabling future analytics or infrastructure planning.

6 USER MANAGEMENT

This chapter details the user management system within the KPI Manager, outlining the different user roles, access levels, and authentication mechanisms. It ensures secure and efficient interaction with the platform while maintaining data integrity and privacy.

6.1 User Roles and Permissions

The KPI Manager incorporates a comprehensive **Role-Based Access Control (RBAC)** system designed to **regulate user interactions** based on **explicitly defined roles** and associated **permissions**. This approach ensures that **each user can access only those resources and functionalities necessary for their responsibilities**, thereby enhancing system security, promoting accountability, and reducing the risk of accidental or unauthorized modifications to critical data.

The RBAC model in the KPI Manager is structured around **three distinct user roles**, each representing a different level of access and responsibility within the platform:

- **Administrator**. Administrators possess the **highest level of access** within the system. They are responsible for the **overall configuration**, **governance**, and **maintenance** of the KPI Manager. Their permissions include the ability to **create**, **modify**, and **delete** all core entities such as KPI definitions, and recorded data. Administrators can also **manage user accounts** by creating, updating, or revoking user roles. Additionally, they can **configure global system settings**, **access and audit complete data logs**, and generate or customize advanced reports. This role is intended for **platform maintainers** or system integrators who oversee the full lifecycle and integrity of the KPI infrastructure.
- **Analyst**. The Analyst role is tailored for users who are primarily involved in **monitoring**, **evaluating**, and **reporting** on SYNERGIES platform performance and usability metrics. Analysts have permission to **browse KPI definitions**, **upload new measurements**, **view historical data trends**, and **generate reports** through built-in visualization tools and dashboards. However, they are restricted from altering any system configurations, managing user roles, or deleting data. This separation of duties helps ensure that analytical tasks can be carried out without compromising the structure or reliability of the KPI dataset.
- **Viewer**. Viewers are granted **read-only access** to the KPI Manager. This role is suitable for **stakeholders** who need visibility into system performance and trends but are **not directly involved in data entry or analysis**. Viewers can **navigate** dashboards, **apply filters** to explore specific KPIs, and **export available data** for offline review or presentation purposes. They cannot modify, upload, or delete any data, nor can they access administrative or analytical functions. This restricted access ensures the integrity of the data while still enabling transparency and information sharing.

To enforce these role-specific permissions, the RBAC framework is implemented consistently across all interaction layers of the KPI Manager, including both the **graphical user interface** and the **application programming interface** (API). Each user's role is assigned at the time of account creation or role update, and their access rights are programmatically checked at runtime before executing any requested action. This **prevents privilege escalation**, ensures **compliance with organizational access policies**, and **supports auditability** by clearly defining and limiting the scope of each user's capabilities.

The RBAC system also **facilitates scalability and maintainability**. As the platform evolves or is extended to accommodate new types of users or use cases, additional roles can be defined and integrated without disrupting the existing access control logic. This modularity supports long-term flexibility while maintaining a secure and controlled operational environment.

Table 4 outlines the RBAC model implemented in the KPI Manager platform, detailing the specific permissions associated with each user role.

Functionality	Administrator	Analyst	Viewer
User management	✓	✗	✗
System configuration	✓	✗	✗
Entity creation/editing (e.g., Features)	✓	✗	✗
KPI deletion	✓	✗	✗
Upload KPI data	✓	✓	✗
Dashboard navigation	✓	✓	✓
Report generation/export	✓	✓	✓

Table 4 Role-based access control model

6.2 User Authentication and Security

To ensure **robust** and **secure** access control to the KPI Manager platform, the system incorporates a comprehensive **authentication and security framework** designed to protect user credentials, preserve session integrity, and prevent unauthorized access to sensitive data, operations, and administrative functionalities.

The **authentication mechanism** is based on a role-bound username and password system, where each registered user is **uniquely identified** through a designated login credential. During the onboarding process, users are assigned a system role (e.g., Administrator, Analyst, Viewer) and provided with a temporary password and unique username. On first login, the user is immediately required to change the password to one that complies with established **security policies**. These policies mandate a minimum password length of eight characters, including at least one uppercase letter, one lowercase letter, one numeric digit, and one special character. This approach enforces **password complexity** to resist brute-force attacks and unauthorized credential guessing.

Each user account is also linked to a **personal email address** which serves as a **verified recovery channel** for password reset requests. This allows secure identity validation in the event of forgotten credentials or account recovery procedures. All email-based recovery flows are subject to **rate limiting** and **token-based validation** to prevent abuse or spoofing.

User credentials are never stored or transmitted in plain text. Instead, all login interactions occur over **HTTPS-secured channels** to protect data in transit. Submitted passwords are hashed using the **SHA-256 cryptographic hashing algorithm** before being validated against encrypted records stored in the backend database. The system does not retain raw passwords at any stage, thereby ensuring protection even in the unlikely event of a data breach.

Upon successful authentication, the platform issues a **secure JSON Web Token (JWT)**, which is valid for a rolling session period of 30 minutes. This token acts as a temporary proof of authentication and is securely stored on the client side (e.g., in HTTP-only cookies or session storage). Every subsequent API call or client-server interaction includes the token in its header, allowing the platform to validate the user's identity and permissions **without re-authentication**, thus ensuring both usability and session security.

The JWT includes encoded claims regarding the user's role and session context, and it is **cryptographically signed** to prevent tampering. If the token is expired or invalid, access is immediately revoked, and the user must re-authenticate to continue interacting with the system.

This multi-layered authentication and session management framework ensures that only authorized users can access the system, that user interactions are trusted and traceable, and that sensitive operations and data remain protected throughout the entire lifecycle of a session within the KPI Manager platform.

6.3 Data Privacy

The KPI Manager platform integrates a comprehensive **data privacy framework** to ensure that all personal and sensitive data managed within the system is **protected** from unauthorized access, misuse, or exposure. These privacy controls are implemented across all stages of data collection, storage, processing, and transmission, aligning with both technical best practices and the legal requirements of applicable data protection regulations, most notably the **General Data Protection Regulation (GDPR).**

To **minimize the exposure of sensitive information**, the KPI Manager enforces a **least privilege access model**, ensuring that users can only access the data strictly necessary for the performance of their role. Access to personally identifiable information (PII) and confidential operational data is restricted based on role-based access control (RBAC), as described in Section 6.1, and further governed by security policies defined at the application and API levels.

Data fields containing personal or sensitive information are protected by **fine-grained access controls**, ensuring that such data is only visible to users with explicit clearance. Audit trails are used to monitor all accesses to protected datasets, thereby supporting accountability and transparency in line with data protection principles.

All data transmitted between client applications and the KPI Manager backend is **encrypted** using **Transport Layer Security (TLS)** protocols to prevent interception or tampering during transit. Furthermore, sensitive data stored in the system (such as user credentials, email addresses, or access tokens) is protected through **encryption at rest**, using industry-standard algorithms and secure key management practices.

Additionally, any temporary or cached data is stored in encrypted memory structures and purged regularly to minimize the risk of data leakage through residual artifacts.

To comply with GDPR's data minimization and storage limitation principles, the KPI Manager supports configurable **data retention policies** that define the lifecycle of stored data. Personal data that is no longer necessary for the purposes for which it was collected is automatically flagged for anonymization or deletion.

Users may exercise their **data subject rights** (e.g., right to access, rectification, or erasure of personal data) by submitting formal requests through authorized channels. These requests are logged and fulfilled within the legally mandated timeframes.

All personal data collected through the platform is processed based on a valid legal basis, such as **user consent** or **legitimate interest**. Where applicable, the system includes **explicit consent mechanisms** for collecting and using personal data, with clear documentation on how the data will be used. Consent records are retained for audit purposes and can be withdrawn by users at any time through account settings or data privacy request forms.

6.4 Logging and Audit Trails

To strengthen platform security, promote operational accountability, and ensure alignment with regulatory and organizational compliance requirements, the KPI Manager implements a robust and comprehensive **logging and audit system**. This subsystem is designed to **capture**, **store**, and **protect detailed records of all critical user interactions and system-level operations**, providing a transparent, traceable history of activity within the platform.

The **activity logging mechanism** continuously monitors and records a broad range of user-driven and system-triggered events. These include, but are not limited to, **authentication** attempts (both successful and failed logins), **access to KPI data**, **creation and upload of new measurements**, **modification or deletion of entities** (such as KPIs), **role-based permission changes**, and **system configuration updates**. For each logged event, the system captures a rich set of metadata, including:

- **User identity** (associated username and role)
- **Timestamp** (exact time of the event using coordinated UTC)
- **Client details** (IP address and user-agent string, if applicable)
- **Action type** (e.g., READ, WRITE, DELETE, CONFIGURE)
- **Affected resources** (e.g., specific KPI, entity ID, or API endpoint involved)
- **Status outcome** (e.g., success, failure, denied)

These logs serve as a **critical foundation for auditability**, enabling system administrators and security personnel to review, trace, and analyse activities across the platform. The availability of detailed audit trails supports **incident response**, **anomaly detection**, and **forensic analysis** in the event of unexpected behaviour, misuse, or a suspected security breach.

To prevent tampering and maintain the **integrity and confidentiality of audit records**, all logs are securely stored in a protected environment with **access control restrictions**, ensuring that only authorized personnel (typically Administrators) may view or export them. Additionally, the logs are **write-once**, meaning once an entry is written, it cannot be altered or deleted without triggering a formal integrity violation alert. In high-assurance configurations, the system may also support **log encryption** and **signing** using cryptographic hashes to further reinforce the non-repudiation and authenticity of stored entries.

The audit system is designed for **long-term retention**, ensuring historical data is preserved to meet internal governance policies as well as external compliance frameworks such as GDPR. Log data can be exported periodically or integrated with external **Security Information and Event Management (SIEM)** solutions for centralized monitoring and cross-platform threat correlation.

In summary, the KPI Manager's logging and auditing infrastructure not only enhances operational transparency and trust but also plays a central role in proactive security monitoring, post-event investigation, and compliance verification. By offering full traceability of user activity and system changes, it upholds core principles of accountability, data integrity, and responsible digital stewardship.

7 CONCLUSIONS AND FUTURE WORK

This chapter summarizes the main contributions and findings achieved through the definition and implementation of the evaluation framework for the SYNERGIES platform. It also outlines the planned next steps to ensure that the framework remains a dynamic and effective tool for performance monitoring, iterative improvement, and cross-work-package integration.

7.1.1 Findings

The deliverable has introduced a comprehensive **Evaluation Framework** tailored to assess the SYNERGIES platform across three fundamental dimensions: **usability**, **technical performance**, and **interoperability**. The framework is grounded in a structured methodology based on **Key Performance Indicators (KPIs)**, each clearly linked to stakeholder requirements defined in WP2. These KPIs provide objective, traceable, and repeatable measures of platform quality and maturity.

A total of **13 KPIs** were defined, organized into three categories:

- **Usability Assessment KPIs** focus on user satisfaction, task success, error rates, and trust.
- **Technical Assessment KPIs** evaluate responsiveness, extensibility, resilience, and security.
- **Interoperability KPIs** assess API behaviour and semantic consistency across systems.

Each KPI is documented through a standardized **KPI Card** format, which includes the KPI's purpose, measurement methodology, performance targets, and requirement references. This consistent structure ensures clarity and comparability while facilitating traceability to project objectives.

A key achievement of the task is the design and deployment of the **KPI Manager**, a cloud-based software tool built on the open-source Measurify framework. The tool supports the **full KPI lifecycle**: data collection (both manual and automated), secure storage, real-time visualization, and data export. Its modular design enables easy integration with questionnaires, expert checklists, and monitoring tools, making it a flexible and extensible solution for both current and future evaluation needs. The KPI Manager also features a robust user management system, fine-grained access controls, and strong compliance with data privacy and security standards, including GDPR.

Together, the framework and the KPI Manager form a complete and operational infrastructure for ongoing performance monitoring and quality assurance within the SYNERGIES project.

7.1.2 Next Steps

In the upcoming phases of the project, the evaluation framework and KPI Manager will be actively integrated into the broader Work Package 8 (WP8) activities. Specifically:

- **Deployment and Testing**: The KPI Manager will be rolled out to real users and developers for structured feedback collection, continuous monitoring, and compliance verification during pilot activities and iterative platform development cycles.
- **Cross-WP Integration**: The framework will be used to assess specific functionalities implemented in other work packages, such as federated database access, and marketplace ensuring that all components meet shared performance and usability criteria.

- **Feedback Collection Process**: A dedicated feedback collection process will be designed and implemented to gather direct insights, comments, and suggestions from end users. This process will include digital surveys, structured interviews, and usage analytics, enabling the project team to capture the user experience and identify usability or functionality gaps that may not be revealed through automated metrics alone.
- **Framework Refinement**: Based on the data collected and feedback received during the initial deployment phase, the evaluation framework will be refined and expanded. This may include introducing new KPIs, updating target values, adapting measurement methods to evolving platform features, and improving data collection procedures for scalability.
- **Sustainability and Reuse**: The KPI Manager and its underlying framework are designed to be reusable beyond the scope of SYNERGIES, with the potential to be adopted by other research projects or initiatives requiring similar evaluation capabilities. Future work will include exploring how the tool can be generalized, modularized, and published as an open-source resource.

By establishing a structured, data-driven evaluation approach complemented by user-centric feedback mechanisms, this task ensures that the SYNERGIES platform can evolve based on evidence, remain responsive to user needs, and deliver high-quality, interoperable services throughout its lifecycle.

8 REFERENCES

[1] G. E. M. Yudantha and T. Tranggono, "Usability Analysis of the Gudang Madu Sumatera Website for Individual Customers using Nielsen's Attributes of Usability," J. Appl. Sci. Eng. Technol. Educ., vol. 6, no. 1, pp. 34–42, Jun. 2024, doi: 10.35877/454RI.asci2734.

[2] N. A. Nik Ahmad and N. S. Hasni, "ISO 9241-11 and SUS Measurement for Usability Assessment of Dropshipping Sales Management Application," in Proceedings of the 2021 10th International Conference on Software and Computer Applications, in ICSCA '21. New York, NY, USA: Association for Computing Machinery, Jul. 2021, pp. 70–74. doi: 10.1145/3457784.3457794.

[3] S. Habib, T. K. Ahsyar, M. Afdal, F. N. Salisah, and S. Syaifullah, "Enhancing Website Usability by Utilizing Heuristic Evaluation and User Feedback for Better User Experience," J. Inf. Syst. Res. JOSH, vol. 4, no. 4, Art. no. 4, Jul. 2023, doi: 10.47065/josh.v4i4.3706.

[4] B. Rahmawati, A. Wibowo, and S. Fitrianingrum, "System Usability Scale (SUS) As An Analysis Method For Official Website," Telematika, vol. 21, p. 173, Jun. 2024, doi: 10.31315/telematika.v21i2.12918.

[5] M. A. F. Kamil, "User Experience Analysis of LinkedIn Social Media Using Usability Metric for User Experience (UMUX)," JIEET J. Inf. Eng. Educ. Technol., vol. 7, no. 2, pp. 78–82, Dec. 2023, doi: 10.26740/jieet.v7n2.p78-82.

[6] S. Firmenich, A. Garrido, J. Grigera, J. M. Rivero, and G. Rossi, "Usability improvement through A/B testing and refactoring," Softw. Qual. J., vol. 27, no. 1, pp. 203–240, Mar. 2019, doi: 10.1007/s11219-018-9413-y.

[7] P. Weichbroth, "Usability of Mobile Applications: A Systematic Literature Study," IEEE Access, vol. 8, pp. 55563–55577, 2020, doi: 10.1109/ACCESS.2020.2981892.

[8] S. S. Salleh, "Knowledge Representation of Supply and Demand Through Data Visualization: Assessing Usability Using Dual Method," Int. J. Acad. Res. Econ. Manag. Sci., vol. 13, no. 3, pp. 9–20, Jul. 2024.

[9] D. Singh, N. K. Pandey, V. Gupta, M. Prajapati, and R. Senapati, "Beyond Textual Analysis: Framework for CSAT Score Prediction with Speech and Text Emotion Features," in 2024 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI), Oct. 2024, pp. 1–6. doi: 10.1109/CVMI61877.2024.10782234.

[10] R. A. Grier, A. Bangor, P. Kortum, and S. C. Peres, "The System Usability Scale: Beyond Standard Usability Testing," Proc. Hum. Factors Ergon. Soc. Annu. Meet., vol. 57, no. 1, pp. 187–191, Sep. 2013, doi: 10.1177/1541931213571042.

[11] N. A. N. Ahmad, N. I. M. Hamid, and A. M. Lokman, "Performing Usability Evaluation on Multi-Platform Based Application for Efficiency, Effectiveness, and Satisfaction Enhancement," Int. J. Interact. Mob. Technol. IJIM, vol. 15, no. 10, Art. no. 10, May 2021, doi: 10.3991/ijim.v15i10.20429.

[12] C. Labadie and C. Legner, Understanding Data Protection Regulations from a Data Management Perspective: A Capability-Based Approach to EU-GDPR. 2019.

[13] A. Popov, J. Bilokin, T. Solianyk, and K. Vasylchenko, "Development of the system to provide cross-browser compatibility of web application," in 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), May 2018, pp. 117–122. doi: 10.1109/DESSERT.2018.8409111.

[14] R. D. Maknuna, A. Mukhadis, and S. Pathmantara, “Effectiveness of Use of Technical Skill Assessment Instruments to Increase Web Programming Competency,” presented at the 1st International Conference on Vocational Education And Training (ICOVET 2017), Atlantis Press, Nov. 2017, pp. 43–51. doi: 10.2991/icovet-17.2017.9.

[15] A. Uzzaman, M. M. I. Jim, N. Nishat, and J. Nahar, “OPTIMIZING SQL DATABASES FOR BIG DATA WORKLOADS: TECHNIQUES AND BEST PRACTICES,” Acad. J. Bus. Adm. Innov. Sustain., vol. 4, no. 3, Art. no. 3, Jun. 2024, doi: 10.69593/ajbais.v4i3.78.

[16] U. Kishnani and S. Das, “Dual-Technique Privacy & Security Analysis for E-Commerce Websites Through Automated and Manual Implementation,” Oct. 19, 2024, arXiv: arXiv:2410.14960. doi: 10.48550/arXiv.2410.14960.

[17] R. S. P. Maciel, P. H. D. Valle, K. S. Santos, and E. Y. Nakagawa, “Systems Interoperability Types: A Tertiary Study,” ACM Comput Surv, vol. 56, no. 10, p. 254:1-254:37, Jun. 2024, doi: 10.1145/3659098.

[18] H. Tm, U. K, M. Shafiulla, and Dadapeer, “An Overview of SQL Optimization Techniques for Enhanced Query Performance,” in 2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Apr. 2023, pp. 1–5. doi: 10.1109/ICDCECE57866.2023.10151265.

[19] P. Kahoro, C. Wanjiru, and N. Karumba, Concurrency Control In Distributed Databases. 2014. doi: 10.13140/2.1.3321.1520.

[20] J. Helt, A. Sharma, D. J. Abadi, W. Lloyd, and J. M. Faleiro, “C5: cloned concurrency control that always keeps up,” Proc VLDB Endow, vol. 16, no. 1, pp. 1–14, Sep. 2022, doi: 10.14778/3561261.3561262.

[21] I. Gorton and V. Teja Rayavarapu, “Foundations of Scalable Software Architectures,” in 2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C), Mar. 2022, pp. 233–236. doi: 10.1109/ICSA-C54293.2022.00052.

[22] M. Sicho, X. Liu, D. Svozil, and G. Van Westen, “GenUI: interactive and extensible open source software platform for de novo molecular generation and cheminformatics,” J. Cheminformatics, vol. 13, Sep. 2021, doi: 10.1186/s13321-021-00550-y.

[23] M. A. Maruf, A. Azim, N. Auluck, and M. Sahi, “FeaMod: Enhancing Modularity, Adaptability and Code Reuse in Embedded Software Development,” in 2024 IEEE International Conference on Information Reuse and Integration for Data Science (IRI), Aug. 2024, pp. 246–251. doi: 10.1109/IRI62200.2024.00058.

[24] R. Jhawar, V. Piuri, and M. Santambrogio, “Fault Tolerance Management in Cloud Computing: A System-Level Perspective,” IEEE Syst. J., vol. 7, no. 2, pp. 288–297, Jun. 2013, doi: 10.1109/JSYST.2012.2221934.

[25] R. C. Motta, K. M. de Oliveira, and G. H. Travassos, “A conceptual perspective on interoperability in context-aware software systems,” Inf. Softw. Technol., vol. 114, pp. 231–257, Oct. 2019, doi: 10.1016/j.infsof.2019.07.001.

[26] N. Haile and J. Altmann, “Evaluating investments in portability and interoperability between software service platforms,” Future Gener. Comput. Syst., vol. 78, pp. 224–241, Jan. 2018, doi: 10.1016/j.future.2017.04.040.

[27] R. Rezaei, T. K. Chiew, S. P. Lee, and Z. Shams Aliee, “Interoperability evaluation models: A systematic review,” Comput. Ind., vol. 65, no. 1, pp. 1–23, Jan. 2014, doi: 10.1016/j.compind.2013.09.001.

[28] N. Haile and J. Altmann, “Evaluating investments in portability and interoperability between software service platforms,” Future Gener. Comput. Syst., vol. 78, pp. 224–241, Jan. 2018, doi: 10.1016/j.future.2017.04.040.

[29] A. Bröring et al., “Enabling IoT Ecosystems through Platform Interoperability,” IEEE Softw., vol. 34, no. 1, pp. 54–61, Jan. 2017, doi: 10.1109/MS.2017.2.

[30] C. I. De Gaetani, M. Mert, and F. Migliaccio, “Interoperability Analyses of BIM Platforms for Construction Management,” Appl. Sci., vol. 10, no. 13, Art. no. 13, Jan. 2020, doi: 10.3390/app10134

A. ABBREVIATIONS AND DEFINITIONS

Term	Definition
API	Application Programming Interface
CBAC	Cross-Browser and API Compatibility
CSAT	Customer Satisfaction Score
ETSI	European Telecommunications Standards Institute
FAIR	Findable, Accessible, Interoperable, Reusable
FRM	Functional Reference Model
GDPR	General Data Protection Regulation
GSR	General Safety Regulation
GUI	Graphical User Interface
HiL	Hardware-in-the-loop
HMI	Human-Machine Interface
ITS	Intelligent Transport Systems
IVS	In-Vehicle System
JRC	Joint Research Centre
KM	Knowledge Management
KPI	Key Performance Indicator
NFR	Non-Functional Requirement
NPS	Net Promoter Score
RBAC	Role-Based Access Control
SiL	Software-in-the-loop
SoA	State of the Art
SUS	System Usability Scale
SUT	System Under Test
TRL	Technology Readiness Level
UNECE	United Nations Economic Commission for Europe
ViL	Vehicle-in-the-loop
WP	Work Package