D5.1 Integration and validation plan

ENCRYPT

Date: 01/03/2023

Lead Partner: TRUSTUP

Doc. Version: 1.0

Document Control Information

Settings	Value
Document Title:	Integration and validation plan
Deliverable Identifier:	D5.1
Project Title:	ENCRYPT
Document Author(s):	Giovanni Mazzeo (**TRUSTUP**), Salvatore D'Antonio (**TRUSTUP**) Luigi Coppolino (**TRUSTUP**), Luigi Romano (**TRUSTUP**) Roberto Nardone (**TRUSTUP**), Angelos Papoutsis (**CERTH**) Giannis Lazarou (**EXUS**), James Lloyd (**EXUS**), Francesca Giampaolo (**ENG**), Paolo Roccetti (**ENG**)
Project Manager:	Giannis LAZAROU (EXUS)
Doc. Version:	1.0
Dissemination Level[1]**:**	CO
Nature:	R
Delivery Date:	01/03/2023
Due Date:	31/12/2022

Document Approver(s) and Reviewer(s):

NOTE: All Approvers are required. Records of each approver must be maintained. All Reviewers in the list are considered required unless explicitly listed as Optional.

Name [Organization]	Role	Action	Date
ENG	Paolo Roccetti	Approved	27/02/2023
CERTH	Dimitris Kavallieros	Approved	26/02/2023

[1] https://kafka.apache.org/

PU=Public, **CO**=Confidential, only for members of the consortium (including the Commission Services), **CI**=Classified, as referred to in Commission Decision 2001/844/EC

Funded by the Horizon Europe Framework Programme of the European Union under Grant Agreement nº 101070670.

Document history:

Changes to this document are summarized in the following table in reverse chronological order (latest version first).

Revision History			
Version	**Date**	**Created by**	**Short Description of Changes**
0.1	08/01/2023	Giovanni Mazzeo (TRUSTUP)	Table Of Contents (ToC)
0.2	13/01/2023	Salvatore D'Antonio (TRUSTUP)	Added Abstarct and Introduction
0.3	20/01/2023	Angelos Papoutsis (CERTH)	Specification of the front-end messages
0.4	24/01/2023	Giannis Lazarou (EXUS) James LIoyd (EXUS)	Specification of the AI-based recommendation engine messages
0.5	25/01/2023	Georgios Meditskos (AUTH)	Specification of the Knowledge Graph tool messages
0.6	02/02/2023	Francesca Giampaolo (ENG), Paolo Roccetti (ENG)	Specification of the assessment tool messages
0.7	14/02/2023	Giovanni Mazzeo (TRUSTUP)	Finalized all sections
0.8	26/2/2023	Dimitris Kavallieros (CERTH)	Review Comments & Integration in D5.1
0.9	27/2/2023	Paolo Roccetti (ENG)	Review Comments & Integration in D5.1
1.0	1/3/2023	Giannis Lazarou (EXUS)	Quality Management Approval and submission of D5.1 Final ver.

Configuration Management: Document Location

The latest version of this controlled document is stored in the ENCRYPT SharePoint Repository (ENCRYPT_D5.1.docx).

Table of Contents

Funded by the Horizon Europe Framework Programme of the European Union under Grant Agreement nº 101070670.

LIST OF FIGURES

LIST OF TABLES

Funded by the Horizon Europe Framework Programme of the European Union under Grant Agreement nº 101070670.

ACRONYMS AND ABBREVIATIONS

TEE - Trusted Execution Environment

HE – Homomorphic Encryption

Funded by the Horizon Europe Framework Programme of the European Union under Grant Agreement nº 101070670.

EXECUTIVE SUMMARY

This deliverable reports the planning activity conducted by the ENCRYPT consortium for integration and validation actions to be performed in the next period of the project's lifetime. To achieve a complete integration of the different tools belonging to ENCRYPT, it is essential to have a clear plan on how to connect the different elements developed by the consortium - both between each other and within the hosting platform. After that, we define a strategy for testing and validating their functionalities. The main purpose of this document, therefore, is to define a clear strategy to be achieved by technical partners in charge of developing the components. Our goal has been to define a common and uniform set of integration and validation guidelines to identify the infrastructures for an efficient integration and testing and to outline the candidate test cases to be executed at the component and system levels.

We first introduce the main concepts of integration and validation – with a particular focus to the case of our ENCRYPT project – and present an overview of the entire process. Afterwards, the attention moves to the different integration procedures. We describe the way of integrating specific components with other components. Finally, the document presents the validation plan. We analyse testbeds that will be used. Furthermore, we discuss the test cases to be performed and how these match with the functional and non-functional requirements specified in the context of D2.1.

Funded by the Horizon Europe Framework Programme of the European Union under Grant Agreement nº 101070670.

1. INTRODUCTION

1.1. Background

In software engineering, the integration process is a major activity of software development to create a complete platform/product. This usually is done in conjunction with a Verification/Validation activity to verify the compliance of the overall system requirements and constraints. The entire process includes the integration of subsystems, while testing interfaces and the overall system performance. Integration of composing pieces of software and validation, verifying that system meets its specifications and that it fulfils its intended purpose, are therefore intrinsic parts of the whole development process. In other words, in order to define an efficient integration plan, testing needs to be taken into account and vice versa.

There are different ways to exchange data and information as well as different integration approaches. The terms interoperability and integration are often misinterpreted analogously. According to The Open Group Architecture Framework (www.opengroup.org/togaf), there are four forms of interoperability and integration:

- At the presentation level,
- Information Integration and Interoperability,
- Application and technical Integration,
- Interoperability.

Interoperability has been defined as the ability of two or more entities to work together (inter-operate). Integration and interoperability are relevant attributes that need to be considered and further analysed. An important aspect relates to how information is exchanged among the different modules, that means the way information is exchanged and what are the main paradigms that are used. This is included in this document. Depending on the specific needs, different message exchange patterns and integration strategy are applied.

ENCRYPT is a platform made of software components completely different between each other adopting their own technologies (e.g., different programming languages, dependencies, and libraries). Each solution implements a different functionality working independently from the others. For this reason, not all the concepts of software integration make sense.

The consortium will follow an incremental approach based on both coupled and decoupled integrations where multiple development cycles take place, and these cycles are divided into smaller modules. In this approach, testing is done by joining two or more modules that are logically related. Then the other related modules are added and tested for the proper functioning. The process continues until all the modules are joined and tested successfully. At that moment the platform is complete and the integration goal set for ENCRYPT is achieved.

1.2. Purpose and Scope

As shown in Figure 1, in ENCRYPT, the integration and validation processes involve outcomes coming from two Work Packages, WP3 and WP4 producing privacy-preserving tools and supporting tools, respectively. These categories of tools will follow different integration and validation approaches due to their different nature. More precisely:

- **Intra-WP3** – These are all the core ENCRYPT solutions, that is, privacy-preserving tools (and acceleration service), which provides the protection of data-in-use following different techniques (i.e., TEE, HE, Differential Privacy). In this WP, there is a specific task (T3.4) where hybrid protections services are proposed (e.g., HE+TEE) and where integration of privacy-preserving techniques will take place. Given the type of solutions we are talking about, their combined use can only occur via a tightly coupled integration.

Funded by the Horizon Europe Framework Programme of the European Union under Grant Agreement nº 101070670.

Furthermore, task T3.5 provides the acceleration service for privacy-preserving tools. Even in this case, the integration will consist in a tightly coupled integration of the GPU-based accelerator with cryptographic schemes.

- **Intra-WP4** – WP4 comprises all the solutions that are needed to manage privacy-preserving tools. As it was already presented in D2.1, they need to communicate between each other to achieve the supporting/managing function. For example, the front-end (T4.4) must interact with the AI-based recommendation system (T4.3) to provide the requirements to follow for the configuration of privacy-preserving tools.
- **Inter-WPs (3 and 4)** – Tools belonging to WP3 must be tuned, deployed, and monitored by solutions belonging to WP4 tools. They will be leveraged via deployment tools.

The consortium defines data formats and interfaces to *Integrate* privacy preserving components with supporting tools and to integrate supporting tools with each other. Based also on the *Use Cases* scenarios, several *Test Cases* will be executed to demonstrate the compliance of ENCRYPT components to *Functional Requirements*. Such test cases will prove the correct integration of ENCRYPT tools. Whereas, a different category of test cases will be performed to validate *Non-Functional* requirements, starting from the number of *attack scenarios* defined by the consortium.

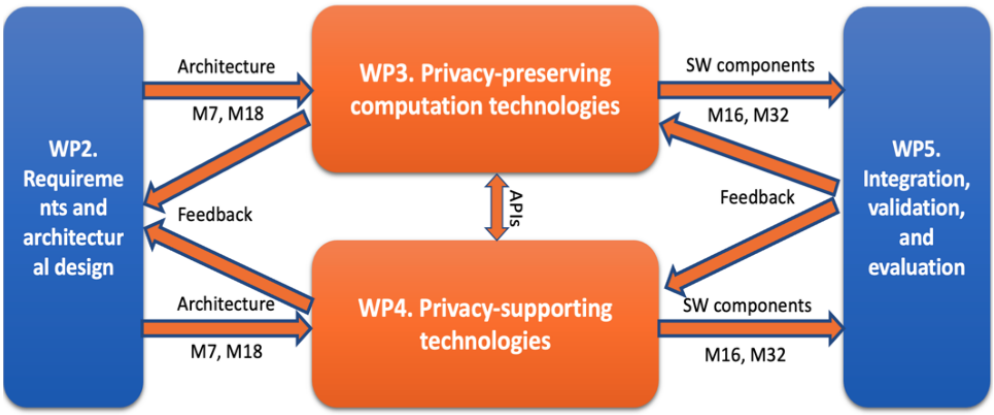

Figure 1 - Core ENCRYPT WPs and their relation (source: ENCRYPT's DOA)

1.3. Structure of the Document

The structure of the document is as follows:

- Section 2 presents the integration approach pursued in ENCRYPT. At the same time, it shows the tools, technologies, and facilities that will be used to perform the integration. Finally, it presents the message specification that will be adopted for WP4 integrations.
- Section 3 focuses on the testing activity and the validation of the ENCRYPT platform. We describe the testbeds that will be used, provide details on the categories of tests, and defined a template for reporting test case results.
- Section 4 contains the conclusions from the activity carried out until now.

2. INTEGRATION

In this section, we overview the ENCRYPT integration plan. The consortium aims at bringing together WP3 and WP4 components in larger multiple sub-systems according to the security functionalities that must be implemented, and especially provided to users in a user-friendly fashion. Outputs coming out of this section will be used to plan tests to be performed for demonstrating the correct integration during the validation activity.

2.1. Approach

Figure 2 reports a high-level view of the ENCRYPT architecture (discussed in D2.1). This architecture is useful to have an idea on how the different components interact between each other. As shown in the figure, tools belonging to WP4 will be used to tune, manage, and deploy privacy preserving tools that will execute either in the ENCRYPT platform, or in pilot infrastructures, or even in other cloud platforms that are located out of the premises of ENCRYPT partners.

The integration (and the related testing process) will be articulated in different stages:

1. Privacy-preserving tools (WP3) will be integrated between each other and tested locally on partners' premises.
2. In parallel, the AI-based recommendation system, the assessment tools, and the UI will be adapted to the established message formats that will enable their communication in ENCRYPT.
3. The ENCRYPT hosting infrastructure will be configured and integrated with the Service deployer / orchestrator and the service image repository.
4. The AI-based recommendation system, the assessment tools, and the UI will be integrated and tested in the hosting platform.
5. Finally, the two categories of components (I.e., privacy-preserving and supporting tools) will be integrated together, and the overall usage flow will be tested.

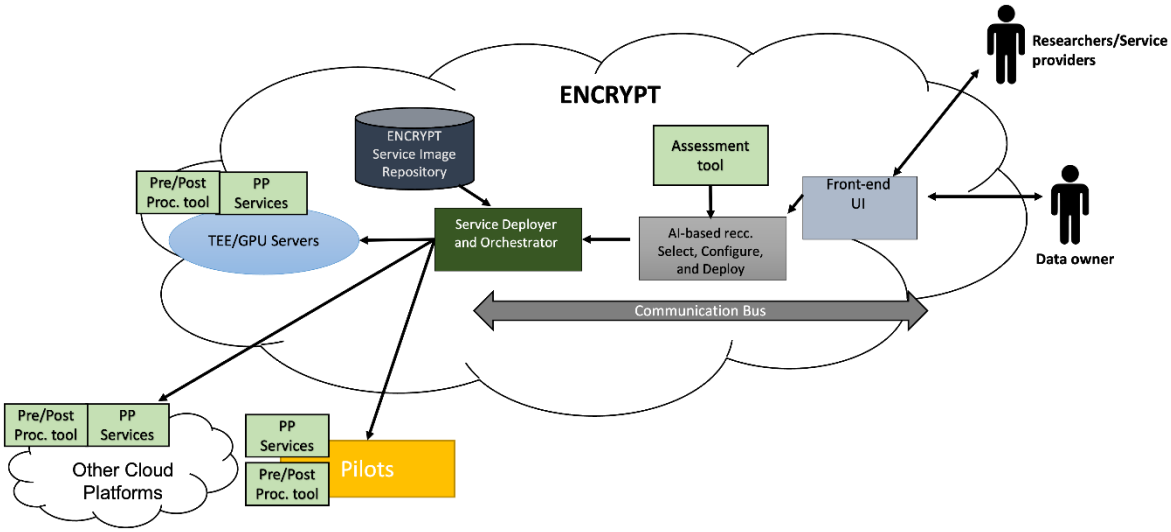

Figure 2 - ENCRYPT Deployment

2.2. Tools supporting the integration

2.2.1. *Hosting Platform*

The ENCRYPT platform will be hosted in a cloud platform. This will ease the management of OS maintenance activities over the three years of project lifetime. Among the number of cloud providers, we decided to use **Microsoft Azure Cloud (azure.microsoft.com)**. Such a decision was taken after evaluating the specifications of ENCRYPT technologies belonging to WP3. Azure Cloud offers virtual machines satisfying all the needs of the consortium:

- SGX-enabled VMs – needed by the TEE tool for the execution of sensitive data processing in secure enclaves

Funded by the Horizon Europe Framework Programme of the European Union under Grant Agreement nº 101070670.

- High performing VMs – needed by the HE tool to perform heavy computations in acceptable time
- GPU VMs – needed by the acceleration service for CPU-acceleration of privacy-preserving tools
- Managed Container Registry – needed by WP4 tools for the deployment of applications

It is important to highlight that the Azure Cloud will be used only as a reference platform. The consortium wants to avoid vendor lock-in and for this reason will leverage the Terraform (www.terraform.io) tool to set up portable Cloud deployments and being as decoupled as possible from the specific cloud provider.

2.2.2. *Deployment Solution*

As a deployment solution, we will use **Jenkins**. This is a platform for creating a Continuous Integration/Continuous Delivery (CI/CD) environment. The system offers many different tools, languages, and automation tasks to aid in pipeline creation when developing and deploying programs.

Although Jenkins requires scripting to implement some steps of the CI/CD processes, the program provides a fast and robust way to automate the software development lifecycle.

2.2.3. *Container Image Registry*

The ENCRYPT tools will be delivered by partners in the form of container images. We have also provided a repository where partners can make all the docker images accessible to the whole consortium. The resource we will use for this purpose is the **Azure Container Registry (ACR)**, i.e. a registry service used to store and manage container images and related artifacts.

2.2.4. *Interconnections*

The consortium decided to use **Azure VPN** to allow secure communications among all the resources and between developers and the platform.

The VPN service we are going to use is the **Azure Virtual Network (VNet)**. VNet enables many types of Azure resources, such as Azure VMs, to safely communicate with each other, the internet, and on-premises networks and also brings with it additional benefits of Azure's infrastructure such as scalability, availability, and isolation.

2.2.5. *Communication Bus*

To enable communication among WP4 tools, we will use a Message Broker, which provides the indirection for publish-subscribe and message queue systems. More precisely, the chosen technology is **Kafka (kafka.apache.org)**, i.e., a modern message broker system with some unconventional design choices that make it efficient and scalable in processing intensive message load. Kafka is the most popular open source distributed publish-subscribe streaming platform that can handle millions of messages per minute. The key capabilities of Kafka are:

- Publish and subscribe to streams of records
- Store streams of records in a fault tolerant way
- Process streams of records as they occur

Kafka is a distributed message broker system that runs as a cluster and the nodes are called brokers. Brokers can be leaders or replicas to provide high-availability and fault tolerance.

2.3. APIs and Message Specification

As it was already argued in Section 2.1, ENCRYPT tools belonging to WP4 will communicate between each other using a message broker. For this reason, it is important to define a message specification of these tools and therefore ease their integration. In the following, we introduce design assumptions that needs to be followed up by all project partners using messaging bus to exchange information in orderly fashion. We provide messaging format taxonomy that will be used during the project.

Before presenting technical details, the reader will benefit by checking design assumptions for this specification:

- **Publisher messages only** - the specification will contain only definition of messages that will be published on the ENCRYPT platform. It is up to the potential subscriber to consume the messages in the format provided by the publishers on the Kafka message bus.
- **Inter-components asynchronous messaging** – all messages will be published as soon as new event; result will become available meaning that in distributed ENCRYPT framework all messages become fully independent from each other emulating real conditions environment.
- **Unique topic names** – in order to avoid confusion, each component will have its own publisher topic. That way all messages will be stored on appropriate topics ready for other components to consume.
- **One topic, one message type** – to simplify message management and validation on subscriber side the publisher will push messages of the same type (schema) to a specific topic. In cases where publisher can provide different types of information, these will be required to be spread across different topics.
- **Message validation** – each component is responsible to validate received messages by using JSON Schema validators and other techniques to ensure that further processing of the message is not interrupting or stopping application or component. In case of malformed message being sent, the validation will trigger additional routines to stop internal processing, to log the event (together with the parts of the incorrect message that can safely written on the logs), and to prevent the ENCRYPT application to fail.
- **Versioning** – to follow up specification changes. This is a live document meaning that after consortium agrees on the new version of specifications, and its respective implementation timeline, the other components will update the components impmentations to the new message formats. It is important to account for specification changes and versioning is one of the best ones available. To reinforce this each message will contain the version number that will inform subscriber about the current state of component implementation and adherence to messaging specification.
- **Internal logging responsibility** – it is also assumed that all information obtained via Kafka broker and required by given component will be locally stored, logged and processed. In case of component restart it will be responsibility of the component owner to restore previous state of the application by loading already preserved data (historical data) in order to enable component to perform its functions immediately after restart.

In the following, we report the template table that we will use for specifying messages generated by WP4 tools, and then the actual formats used by each of these components.

Funded by the Horizon Europe Framework Programme of the European Union under Grant Agreement nº 101070670.

Component name	**<All ENCRYPT components supporting security tools >**
Topic name	**<component_id>_status**
Purpose	**Interrogate current status of given ENCRYPT component**
Version	**<major revision number>.<minor revision number> (e.g. 1.0)**
Frequency of publishing [messages/hour]	**Depending on ENCRYPT orchestration mechanism parameter settings not more than 300 per hour**
Response time [s]	**N/A**
Mandatory[Y/N]	**Y**
Message format	**{** **"id": "<auto_generated_guid>",** **"version": <string>,** **…** **}**
Examples	**{** **"id": "d3b81418-42da-4542-85e5-f93dafb38e95",** **"version": "1.0",** **…** **}**
Comments	**This request will have to be handled by all components.**

Table 1 - Message Specification Template

2.3.1. *Knowledge graph building tool*

Component name	**Knowledge graph building tool**
Topic name	kg_status
Purpose	To map and interlink information, such as data and schemata, in the semantic space, using Semantic Web technologies, such as RDF graphs. The generated information will be used for data validation and further analysis to support the recommendation engine
Version	1.0
Frequency of publishing [messages/hour]	As soon as the component received data, it will apply the necessary mapping mechanisms and update the underline RDF triple store, publishing relevant messages to inform the other components of the framework (the frequency depends on ENCRYPT overall orchestration mechanism)
Response time [s]	N/A
Mandatory[Y/N]	Y
Message format	{ "id": "<auto_generated_id>",

	“version”: <string>, “dataset_iri”: <URI/IRI>, “repository_id”: <string>, “sparql_endpoint”: <URL> } The complete format will be specified later, based on the requirements and the interaction with other components

Examples	{
	“id”: “d3b81418-42da-4542-85e5-f93dafb38e95”,
	“version”: “1.0”,
	“dataset_iri”: "http://encrypt.auth.gr/dataset”,
	“repository_id”: "dataset_1”,
	“sparql_endpoint”: "http://encrypt.auth.gr/repository/sparql”
	}

Comments	The purpose of the component is to generate knowledge graphs in an RDF triple store, such as GraphDB. As such, it is important to specify as soon as possible the type and the format of the data that the component will need to handle in order to develop the mapping mechanisms. The generated knowledge base will be accessible by all components, using established semantic web standards, such as http and SPARQL query endpoints.

2.3.2. *AI-based Recommendation Engine*

Component name	**AI-based Recommendation Engine**
Topic name	re_status
Purpose	Recommend appropriate privacy preserving technology and its configuration for a given scenario, with accompanying justification.
Version	1.0
Frequency of publishing	On request, will publish a messages per interaction from the user
Response time [s]	N/A
Mandatory[Y/N]	Y

Message Format	{
	“id”: <string>,
	“recommended_ppt”: <string>,
	“recommended_config”: <dict>,
	“justification”: <string>
	}

Examples	{ “id”: “d3b81418-42da-4542-85e5”, “user_id”: “1234”, “user_role”: “data owner”, “data_sensitivity”: “low”,

	“intended_use”: “light computation”,
	“location_of_computation”: “on premises”,
	“time_constraints”: “medium”,
	“computational_constraints”: “low”,
	“performance_constraints”: “>50%”;
	“cost_constraints”: 10000;
	}
	
	{
	“id”: “d3b81418-42da-4542-85e5”,
	“recommended_ppt”: “differential privacy”
	“recommended_config”: {param1: x, param2: y},
	“justification”: “Low performance constraints allow for param1 to be higher, ...”
	}

Comments	Whatever the final fields, the messages sent to and by the Recommendation Engine will be lightweight. The engine only requires information about the data and scenario, and not the data itself.

2.3.3. *Front-end*

Component name	**Front-end**
Topic name	fe_status
Purpose	To assist end-users in providing their data, visualise the results of the AI-based recommendation system and select and configure the most appropriate technologies to use in a user-friendly way.
Version	1.0
Frequency of publishing [messages/hour]	TBD
Response time [s]	N/A
Mandatory[Y/N]	Y
Message format	{ “id”: “<auto_generated_id>“, “user_id”: <string>, “user_role”: {“data owner”, “researcher”, “service provider”, “DPO”, “developer”, “project manager”}, “data_sensitivity”: {“low”, “medium”, “high”, “critical”}, “intended_use”: {“sharing”, “light computation”, “heavy computation”}, “location_of_computation”: {“on premises”, “own cloud”, “cloud”}, “time_constraints”: {“low”, “medium”, “high”}, “computational_constraints”: {“low”, “medium”, “high”}, “performance_constraints”: {“low”, “medium”, “high”}, “cost_constraints”: <float>,

	}

Examples	{
	“id”: “d4b81588-55da-4548-85f6”,
	“user_id”: “123456”,
	“user_role”: “Researcher”,
	“data_sensitivity”: “high”,
	“intended_use”: “light computation”,
	“location_of_computation”: “on premises”,
	“time_constraints”: “low”,
	“computational_constraints”: “low”,
	“performance_constraints”: “>50%”;
	“cost_constraints”: 5000
	};
	
	{
	“id”: “d4b81588-55da-4548-85f6”,
	“recommended_ppt”: “homomorphic encryption”
	“recommended_config”: {remove features x, y ,z} ,
	“justification”: “Removing these features will result in fewer computational resources, ...”
	}

Comments	Information will be presented in tiers allowing the users to pull up information, drill down into specific details when needed, and assist end-users with different backgrounds to easily analyse and adopt the privacy suggestions coming from the recommendation system.

2.3.4. *Privacy Risk Assessment Tool*

Component name	**Privacy Risk Assessment Tool**
Topic name	prat_status
Purpose	Share major privacy risks for a given scenario
Version	1.0
Frequency of publishing	On request, will publish a messages for each assessment (or re-assessment) requested by the user
Response time [s]	N/A
Mandatory[Y/N]	Y

Message format	{
	“message id”: “<auto_generated_id>“
	“scenario id”: <string>,
	“probability”: {“low”, “medium”, “high”},
	“impact”: {“low”, “medium”, “high”}
	}

Examples	{
	“id”: “e4a1519-36d3-4522-8b65”,
	“scenario_id”: “data_breach_scenario1”,
	“probability”: “low”,
	“impact”: “high”
	}

Comments	The *assessment_id* field provides subscribers to the ppa queue to link re-assessments of a given scenario across time. The exact format of *privacy_risks* field still to be detailed.

3. PLANNED TESTING ACTIVITY

The test phase is important for evaluating, in the last period of project lifetime, whether the ENCRYPT components and the integrated infrastructure satisfy specified requirements. A first important part is the definition of testbeds to be used for the execution of test cases needed to validate the requirements. In a first section the two different types of testbeds to be used – based on specific requirements – are described. Then, the template for test case report that will be adopted is briefly described. Finally, the functional and non-functional tests for the different components are analysed.

3.1. Testbed Description

Our goal is to get valuable – and closer to reality – validation results, while keeping low the time needed for the validation activity. For this reason, the consortium decided to pursue the execution of several test cases on top of two different testbeds (TB), which must be chosen based on the typology of the test case to be run. The adopted ENCRYPT deployments are the following:

- *Local Testbed (LOCALTB) - Planned release M28:* mainly used for functional evaluations, e.g., unit tests, interface tests, or regression tests. It is an important solution to reduce setup and configuration time for those tests where it makes no difference if the testbed reflects a real-like deployment. We will design a docker-compose ENCRYPT infrastructure that will be easy to deploy on a single machine instance.
- *Real-like Testbed (REALTB) - Planned release M32:* adopted for the execution of test cases that aim at demonstrating the compliance of ENCRYPT with non-functional requirements (e.g., security and performance). We will replicate the real infrastructure of the use case provider in a laboratory environment. Furthermore, we will leverage anonymous datasets coming from use case providers to replicate the same workload conditions and guarantee the respect of performance requirements.

3.2. Test Case Report Template

Before overviewing the numbers of test cases that the consortium will perform, in the following, we report the table to be adopted for the detailed description of tests.

GENERAL INFORMATION

RELATED CATEGORY	One or multiples categories related to this test case		
TEST DATE:	dd/mm/yyyy	**Testbed :**	The selected testbed
TEST DESCRIPTION:	A brief description of what functionality the case will test		
TEST CASE ID	A unique test number assigned to the test case	**Result:**	Pass/Partially Pass/Fail
TEST			

REQUIREMENT(S) TO BE TESTED:	Identify the requirements to be tested and include the requirement number coming from the deliverable
ROLES AND RESPONSIBILITIES:	Describe each project consortium partner and stakeholder involved in the test, and identify their associated responsibility for ensuring the test is executed appropriately
PROCEDURAL STEPS:	Describe the sequences of actions necessary to prepare and execute the test case. Provide detailed test procedures for each test case; explain precisely how each test case will be executed
TEST ITEMS AND FEATURES:	Identify and describe the items and features that will be exercised by the test case. Group the test cases into logically related scenarios that test related items and features. For each item or feature, a reference to its associated requirement source should be included
EXPECTED RESULTS	Describe the outcome anticipated from the test case. Specify the criteria to be used to determine whether the item has passed or failed
ENVIRONMENTAL NEEDS	
HARDWARE	Identify the qualities and configurations of the hardware required to execute the test case
SOFTWARE	Identify system and application software required to execute the test case. It specifies any software that the test case will interact with
ACTUAL RESULTS	
RESULTS ASSESSMENT:	It defines all the outputs and features required of the test case and provide expected values. While executing the test, record and describe the visually observable outputs as they occur. Produce tangible evidence of the output such as a screen print. At the conclusion, describe the actual outcome. Indicate whether the test passed or failed, and identify any discrepancies between the expected results and the actual results

3.3. Functional Tests

Functional tests represent a phase in which the individual software ENCRYPT components are combined and tested in groups or subgroups. It comes after the unit tests performed by each partner and before non-functional validation tests. The activity of integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests, and delivers as its output the integrated system ready for system testing.

We plan the following integration tests:

- **Unit Tests** – to evaluate specific components' functionalities. These are part of each component validation and out of the scope of this deliverable where we focus on the entire ENCRYPT toolset.
- **Interface Tests** – to demonstrate the correct component-component and component-platform integration
- **Regression Tests** – to ensure that functionalities of ENCRYPT still performs in the same way after changes introduced by ENCRYPT tools integration

3.4. Non-Functional Tests

This section comprises several test cases carried out to evaluate the following:

- *Security* – In this category of tests we validate both the intrinsic security of components and the security level of ENCRYPT tools. We will observe if system can preserve itself and the data it holds in situation of malicious attacks. In these test cases we will prove the compliance with requirements of security for *data-in-use*, *data-in-transit*, and *data-at-rest*. Moreover, we will demonstrate that the security solutions proposed are able to address a number of misuse cases.

- *Performance* – Tests of this category serve for demonstrating that the security solutions do not impact performance criteria and also to compare the secured and non-secured version of the framework in order to show the overhead given by ENCRYPT components. We also include in this category all the aspects that relates to *scalability*, and so those tests that show the ability of the system to scale up for increasing number of inputs.

- *Reliability* – These tests want to determine whether the software meets reliability requirements. In particular, the goal is to demonstrate that the system has the expected availability during long running execution and that the resilience mechanisms chosen work as expected.
- *KPI Validation – These tests will aim at validating the compliance of the ENCRYPT solution with the KPIs set by the consortium. More precisely:*
    - *KPI1.1: 30% improvement in the computation time over encrypted data with the ENCRYPT component for FHE;*
    - *KPI1.2: at least 4X less memory overhead for the encrypted data using FHE component;*
    - *KPI1.3: at least 10000X reduction in the bandwidth requirements with the FHE component (when trans-cipher technique is applied);*
    - *KPI1.4 reduce the amount of client decryptions in FHE deployments by 20%;*
    - *KPI1.5: at least 3 limitations mitigated (either on performance, security, or deployability) when combined solutions are used (component TEE-HE; component DP-FHE; FHE component with support for the verifiable computing;*
    - *KPI1.6: >50% improvement in performance from GPU acceleration, as compared to vanilla versions of PP technologies;*
    - *KPI1.7: 100% GDPR-compliant solutions based on the selected PP technologies*

4. CONCLUSION

In this document, we presented the approach that will be pursued by the ENCRYPT consortium for the integration and validation of the platform. We classified the categories of tools and identified two different integration approaches: tightly coupled (for WP3) and loosely coupled (for WP4). Furthermore, we presented the technologies adopted by the consortium to perform a smooth integration. Finally, we introduced the validation plan presenting the testbeds, and test cases, which will be used to demonstrate that our ENCRYPT platform meets functional and non-functional requirements.

Funded by the Horizon Europe Framework Programme of the European Union under Grant Agreement nº 101070670.