AI IN SECURE PRIVACY-PRESERVING COMPUTING CONTINUUM

Project Title	Artificial Intelligence in Secure PRIvacy-preserving computing coNTinuum
Project Acronym	AI-SPRINT
Project Number	101016577
Type of project	RIA - Research and Innovation action
Topics	ICT-40-2020 - Cloud Computing: towards a smart cloud computing continuum (RIA)
Starting date of Project	01 January 2021
Duration of the project	36 months
Website	www.ai-sprint-project.eu/

D5.1 - Evaluation Plan

Work Package	WP5 | Validation on use cases
Task	T5.1 | Evaluation of use cases and Monitoring
Lead author	Matteo Matteucci (POLIMI)
Contributors	Danilo Ardagna (POLIMI), Giulio Fontana (POLIMI), Danilo Ardagna, (POLIMI), Davide Cirillo (BSC), Enrico Abate-Daga (BECK)
Peer reviewers	María José Rementeria (BSC), Agatha Dabrowski (BECK), Rita Giuffrida (TRUST-IT)
Version	V1.0
Due Date	31/12/2021
Submission Date	27/12/2021

Dissemination Level

X	PU: Public
	CO: Confidential, only for members of the consortium (including the Commission)
	EU-RES. Classified Information: RESTREINT UE (Commission Decision 2005/444/EC)
	EU-CON. Classified Information: CONFIDENTIEL UE (Commission Decision 2005/444/EC)
	EU-SEC. Classified Information: SECRET UE (Commission Decision 2005/444/EC)

AI-Sprint - Artificial Intelligence in Secure PRIvacy-preserving computing coNTinuum, has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no. 101016577.

Versioning History

Revision	Date	Editors	Comments
0.0	24/10/2021	Matteo Matteucci	First Table of Content
0.1	29/11/2021	Matteo Matteucci	Written Section 1 and 2
0.2	30/11/2021	Matteo Matteucci	Revised Section 1 and 2 Written Section 2 and 3
0.3	01/12/2021	Matteo Matteucci	Written Section 4
0.4	06/12/2021	Enrico Abate-Daga	Reviewed Sections 1, 2, 3.1, 4.4 and Bibliography Written Sections related to Farming 4.0 in Section 3 and 4, as well as and 4.6
0.5	06/12/2021	María José Rementeria	Reviewed Sections 1, 2, 3.1, 4.4 and Bibliography Written Sections related to Farming 4.0 in Section 3 and 4, as well as and 4.6
0.6	12/12/2021	Matteo Matteucci	Revision of partner contributions and conclusions
0.7	13/12/2021	Davide Cirillo	Written Section 4.5.1
0.8	16/12/2021	Giulio Fontana	General review and revision; definition of use case-specific KPIs
0.9	19/12/2021	Davide Cirillo	Revised case-specific UC KIPs
0.10	16/12/2021	Stephanie Parker	Review of the deliverable by TRUST-IT
0.12	17/12/2021	Giulio Fontana	Implementation of review from TRUST-IT
0.11	19/12/2021	Matteo Matteucci	Overall revision after TRUST-IT review
0.12	20/12/2021	Agatha Dabrowski	Review of the deliverable by GREG
0.13	21/12/2021	Matteo Matteucci	Pre-final revision and formatting
1.0	27/12/2021	Matteo Matteucci	Final version

Glossary of terms

Item	Description
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
DoW	Description of Work
GQM	Goal Question Metric
TRL	Technology Readiness Level
AI/ML	Artificial Intelligence and Machine Learning
YAML	Yet Another Markup Language
POPNAS	Pareto Optimal Progressive Neural Architecture Search
TEE	Trusted Execution Environment
GPU	Graphical Processing Unit
VM	Virtual Machine
AWS	Amazon Web Services
KPI	Key Performance Indicator
RGB	Red Green and Blue

Keywords

Artificial Intelligence; Edge Computing; Computing Continuum.

Disclaimer

Executive Summary

Use cases are a joint effort involving all elements of the AI-SPRINT Consortium, i.e., end users, integrators, research partners, and the cloud provider. End users provide the actual use cases and related expertise, as well as experience in the evaluation of objectives; integrators bring their experience in implementing effective solutions; research partners provide high-level expertise and knowledge about AI-SPRINT technology; finally, the cloud provider offers the experimental infrastructure to support integration and validation activities and real-world implementation expertise.

This deliverable presents the evaluation plan of the AI-SPRINT software based solutions for its use cases. The evaluation plan is based on Basili's Goal Question Metric (GQM) method. The objective of the evaluation plan is to specify and schedule the activities that should be performed to evaluate the AI-SPRINT solution. For the evaluation itself we aim to assess the AI-SPRINT solutions with respect to the project's main objectives, the Key Performance Indicators, the elicited requirements provided by the use case providers and the objectives related to the specific technical work packages. The evaluation plan specifies what and how to evaluate as well as the overall schedule for the evaluation, applying a set of slightly revised GQM templates.

It is important to note that the main target of this evaluation is the software based AI-SPRINT solutions, thus, it is not a general evaluation of all activities and progress in the project as a whole. Moreover the focus is on the evaluation of the AI-SPRINT assets performed through the AI-SPRINT use cases and not the technical evaluation per se which is described on the deliverables related to the technical developments of AI-SPRINT assets.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Context and Objectives

This deliverable presents the plan for the evaluation and validation of the AI-SPRINT software-based solution, also referred to as AI-SPRINT assets in the following, with respect to:

1. The main objectives of the project.
2. The Key Performance Indicators (KPIs).
3. The requirements identified in WP1 and collected in a coordinated way by task T1.2.
4. The specific objectives of the technical work packages.

In this deliverable, the design of the evaluation strategy is driven by the rationale and aim of the aforementioned evaluation goals which, in general, assess whether we have met the AI-SPRINT project overall objectives, and supported the improvement of the software-based solution provided. The evaluation strategy has been derived based on state-of-the-art scientific methods and experiences from evaluation in other EU projects. Based on the selected evaluation strategy and the project road-map for delivering the AI-SPRINT tools, an evaluation plan containing goals, KPIs, methods and schedule for performing evaluations is developed in the following.

It is important to note that the main target of this evaluation is the software-based AI-SPRINT solutions, thus, it is not a general evaluation of all activities and progress in the project as a whole. Moreover, the purpose of this deliverable is to provide guidelines and a plan of the activities that will be carried out to evaluate the software-based AI-SPRINT solutions against the AI-SPRINT Uses Cases only, thereby complementing the evaluation work performed in work packages WP2, WP3, and WP4 for the evaluation of the technical requirements of the AI-SPRINT software solutions.

1.2 Structure of the document

The remainder of this document is organised as follows.

- Section 1 presents a short introduction to the context and objectives of the evaluation as well as the outline of the deliverable.
- Section 2 presents the rationale and overall evaluation strategy as well as general evaluation techniques upon which our evaluation plan is based.
- Section 3 presents the target of the evaluation, i.e., the software based AI-SPRINT solution with the set of AI-SPRINT assets.
- Section 4 presents the evaluation plan, in particular the evaluation plans associated with AI-SPRINT objectives in terms of use cases, KPIs and Technology Readiness Level.
- Section 5 provides some concluding remarks.

2. Evaluation strategy and baseline evaluation

This section presents the rationale and overall evaluation strategy as well as general evaluation techniques upon which the AI-SPRINT evaluation plan is based.

2.1 Rationale and Evaluation strategy

As a general endeavour, evaluation can be characterised by the following features [Suchman1967]:

- Evaluation is a task, which results in one or more reported outcomes.
- Evaluation is an aid for planning, and therefore the outcome is in effect an evaluation of and direction to different possible actions.
- Evaluation is goal oriented. The primary goal is to check results of actions or interventions, in order to improve the quality of the actions or to choose the best action alternative.
- Evaluation is dependent on the current knowledge of science and the methodological standards.

In our context we are interested in evaluating a software based solution, namely, the software based methods, tools and techniques of the AI-SPRINT framework, i.e., the AI-SPRINT assets. More specifically we will focus on evaluating AI-SPRINT assets quality. Software quality can be defined as "*the degree to which a product or system can be used by specific users to meet their needs to achieve specific goals with effectiveness, efficiency, freedom from risk and satisfaction in specific contexts of use*" [ISO/IEC-25010:2011]. Software quality can be evaluated with respect to different aspects: for example, functionality, reliability, usability, etc. [ISO/IEC-25010:2011].

There are many software evaluation techniques and possible evaluation strategies to be based on. Our evaluation strategy is summarised by considering a set of fundamental questions that should be taken into consideration when making an evaluation plan.

<u>Who is the evaluation for and Why do they want the evaluation?</u>

There are basically three main target audiences for our evaluation:

1. *<u>Ourselves</u>* (the AI-SPRINT project). A main purpose of our evaluation is to provide feedback internally to consortium partners. In particular, the first evaluation round (M24) is performed in order to ensure proper planning and achieve improvements of the AI-SPRINT approach when conducting our second development increment. The second (and final) evaluation round (M36) is to provide a confirmed status of our results at the end of the project's funding lifecycle[1].
2. *<u>Exploitation users</u>*. Another important target audience of the evaluation is represented by users that want to test and exploit AI-SPRINT assets (these may include both AI-SPRINT partners and third party/project external users). The evaluation will provide them with a better understanding of the status (e.g., Technology Readiness Level, available features, strengths and weaknesses, etc.) of the provided AI-SPRINT assets. Furthermore, the evaluation report is crucial background information to prepare and plan for exploitation of the AI-SPRINT assets both during and after the project (again this goes both for third party users and for our own exploitation planning with respect to the AI-SPRINT assets).
3. *<u>The Commission and the reviewers</u>*. A third target audience is the project owner, the European Commission. They want to know to what extent the goals and ambitions of the project are achieved and whether there has been a good return on investment. The evaluation of the provided software

[1] We recall here that at the time of writing, i.e., at the submission time of *D5.1 Evaluation Plan*, the project is at M12.

based tools and techniques is a valuable input in this respect. Moreover, the reviewers will evaluate the project on behalf of the commission at the project reviews, where one of the main tasks is to assess the project status. The evaluation of the software based tools and techniques is again valuable input for the general evaluation and rating of the project.

<u>What are you evaluating?</u>

The focus of evaluation is on the software based AI-SPRINT solution with respect to the requirements of the use cases. In general, aspects of functionality, usability, performance efficiency (defined according to the achievement of the individual technical work packages, i.e., WP2, WP3, and WP4 KPIs reported in the *AI-SPRINT Deliverable D7.1 1st Year Progress report*) and portability will be evaluated. The more detailed aspects and qualities to be evaluated are provided in the evaluation plan in Section 4 of this document. The main baseline to determine what characteristics and features to be evaluated are:

1. The AI-SPRINT objectives as stated in the Description of Work (DoW).
2. The AI-SPRINT defined KPIs for the technical work packages and the use cases.
3. The set of use cases requirements that are elicited in the AI-SPRINT project (available in the AI-SPRINT requirements repository).
4. The specific objectives of the technical work packages (when they impact on use cases).

<u>When do we start?</u>

Effective evaluation relies on a proper plan that identifies what to test and what questions to answer. This deliverable should provide the evaluation plan for accomplishing such an evaluation. Furthermore, the target to be evaluated (the AI-SPRINT solution) needs to be available in a stable state. According to the AI-SPRINT plan, the first increment of the AI-SPRINT solutions is expected at M18 with the first release of the AI-SPRINT integrated framework, thus, the evaluation can start at the point when this release is ready at M19. The findings for the first round of evaluation are due by M24 (AI-SPRINT project Milestone MS IV) and they will impact the planning and focus of final release of the AI-SPRINT integrated framework (AI-SPRINT project Milestone MS V, due in M30) and will provide an important input for the exploitation plans (due in M36). This timeline is reported in Figure 2.1 for reader convenience.

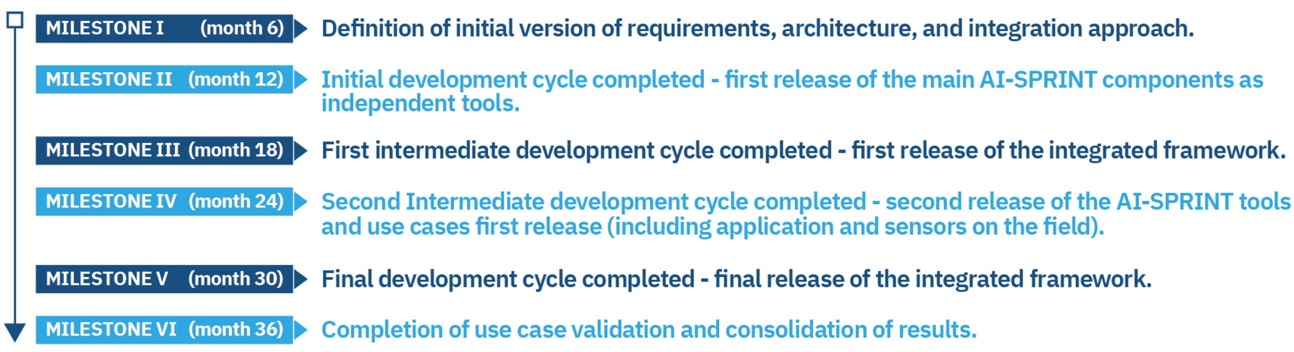

Figure 2.1 - AI-SPRINT project Milestones timeline

<u>How do we evaluate?</u>

The use case partners in the project will play the role of users evaluating the provided tools and techniques provided by the technology providers in the project. The main foundation of this evaluation is the actual development of the AI-SPRINT use cases applying the AI-SPRINT solution as well as the execution of the case study implementations using the AI-SPRINT run time platform.

When it comes to what evaluation techniques should be applied, there are a set of software evaluation techniques presented in the literature. For the purpose of evaluating the software based AI-SPRINT solution we consider it appropriate to build our evaluation based on the Goal Question Metrics (GQM) method (see Section 2.2). In addition we will provide an assessment based on the Technology Readiness Level scheme (TRL), reported in Section 2.3, for each of the AI-SPRINT exploitable assets.

What are the planned activities and resources for accomplishing the evaluation?

Task T5.1 of WP5 is the task dedicated to the evaluation of use cases. This task is about coordination of AI-SPRINT evaluation and providing the results of the evaluation of use cases. The outcomes of this task are deliverables D5.1 (this document) and *AI-SPRINT Deliverable D5.5 Final assessment report, impact analysis, lessons learned & best practice due in M36*. The latter will include, beside the final assessment reports, the impact analysis, the lessons learned and the best practices developed thanks to the use cases.

In addition, an important prerequisite to the evaluation is the elicitation of requirements that have been performed in Task T1.2: use cases and Framework Requirements Analysis. The outcome of this task has been the deliverables *D1.2 Requirements analysis*. Furthermore, the effort of designing and implementing the use cases applying the AI-SPRINT tools and platforms is an important baseline for performing the evaluation and providing the evaluation reports, this includes all tasks and deliverables of WP5.

2.2 Goal Question Metric (GQM)

The Goal Question Metric (GQM) approach [Basili1994, Van Solingen1999, Basili2009] defines a measurement model on three levels:

- *Conceptual level (Goal):* A goal is defined for an object, for a variety of reasons, with respect to various models of quality, from various points of view and relative to a particular environment. Objects of measurement are either:
    - *Products*: assets, deliverables and documents that are produced during the system life cycle; e.g., specifications, designs, programs, test suites.
    - *Processes*: software related activities normally associated with time, e.g., specifying, designing, testing, interviewing.
    - *Resources*: items used by processes in order to produce their outputs; e.g., personnel, hardware, software, office space.

  In our case the Object is a product: The software based AI-SPRINT solution with its containing set of modules and tools. Thus, in the GQM template, typically the object refers to a specific module or tool of the AI-SPRINT solution. We denote such modules and tools as *AI-SPRINT assets*. The AI-SPRINT assets are briefly presented in Section 3 of this document, for a more detailed description we point the interested reader to D*2.1 First release and evaluation of the AI-SPRINT design tools*, *D3.1 First release and evaluation of the runtime environment*, and *D3.2 First release and evaluation of the monitoring system*.
- *Operational level (question):* A set of questions is used to characterise the way the assessment/achievement of a specific goal is going to be performed based on some characterising model. Questions try to characterise the object of measurement (product, process, resource) with respect to a selected quality issue and to determine its quality from the selected viewpoint. In our case questions will be related to qualities of the software based AI-SPRINT solution such as functionality (extracted from the use cases of the AI-SPRINT requirements) and various QoS properties.
- *Quantitative level (metric):* A set of data is associated with every question in order to answer it in a quantitative way. The data can be:

- *Objective*: If they depend only on the object that is being measured and not on the viewpoint from which they are taken; e.g., the number of versions of a document, staff hours spent on a task, size of a programme.
- *Subjective*: If they depend on both the object that is being measured and the viewpoint from which they are taken; e.g., readability of a text, level of user satisfaction.

GQM prescribes a six-step process where the first three steps are about using (business) goals to drive the identification of the right metrics and the last three steps are about gathering the measurement data and making effective use of the measurement results to drive decision making and improvements. Basili [Basili 1994] described his six-step GQM process as follows:

1. Develop a set of corporate, division and project business goals and associated measurement goals for productivity and quality.
2. Generate questions (based on models) that define those goals as completely as possible in a quantifiable way.
3. Specify the measures needed to be collected to answer those questions and track process and product conformance to the goals.
4. Develop mechanisms for data collection.
5. Collect, validate and analyze the data in real time to provide feedback to projects for corrective action.
6. Analyse the data in a post mortem fashion to assess conformance to the goals and to make recommendations for future improvements.

To specify the goals GQM specific templates are applied. These templates are a structured way of specifying goals. A GQM goal specification template contains the following fields:

Field	Examples
Object of Study	AI-SPRINT abstractions, Neural Architecture Search, ...
Purpose	Characterize, understand, evaluate, predict, improve, ...
Focus	Functionality, usability, performance, ...
Stakeholders	Application developer, application provider, ...
Context factors	Prototype development, lab experiment...

Table 2.1: GQM goal specification template

With respect to the measurement goals, questions should be defined to support data interpretation towards a measurement goal. As goals are defined on an abstract level, questions are refinements of goals to a more operational level, which is more suitable for interpretation. By answering the questions, one should be able to conclude whether a goal is reached. Therefore, during question definition, checks should be performed as to whether the defined questions have the ability to support the assessment of the goal in a satisfactory way.

Once goals are refined into a list of questions, metrics should be defined that provide all the quantitative information to answer the questions in a satisfactory way. Therefore, metrics are a refinement of questions into quantitative process and/or product measurements. After all these metrics have been measured, sufficient information should be available to answer the questions. Furthermore, factors that could possibly be of influence to the outcome of the metrics should also be identified. After all, factors that directly influence metrics, also influence the answers to the questions that the metrics are related to. If the influencing factors were not to be considered during definition of a measurement programme, some conclusions or interpretations of the collected data may not be correct. These influencing factors are usually also defined as metrics.

For data collection some concerns need to be clarified such as:

- For a certain metric, who (or what, e.g., logging system) should collect the data?
- When should the data be collected?
- How can the data be collected efficiently and effectively?
- To whom (or what, e.g., some data analysis tool) should the data be delivered?

For the analysis and feedback session the main points are to interpret measurements data with respect to the goals and questions and evaluate the conformance to goals and questions, and translate the interpretation into conclusions and action points.

In our evaluation plan presented in Section 4 we apply GQM to AI-SPRINT assets. We have slightly revised the GQM template to better suit the specific evaluation of the AI-SPRINT software based solution with respect to use cases. Evaluation of functional and QoS characteristics are specified using our GQM based template.

2.3 Technology Readiness Level

As part of the evaluation of the AI-SPRINT solutions we will evaluate and assess also Technology Readiness Levels (TRLs). The TRL approach specifies the following scheme to categorise a specific object:

1. TRL 1 – basic principles observed
2. TRL 2 – technology concept formulated
3. TRL 3 – experimental proof of concept
4. TRL 4 – technology validated in lab
5. TRL 5 – technology validated in relevant environment (industrially relevant environment in the case of key enabling technologies)
6. TRL 6 – technology demonstrated in relevant environment (industrially relevant environment in the case of key enabling technologies)
7. TRL 7 – system prototype demonstration in operational environment
8. TRL 8 – system complete and qualified
9. TRL 9 – actual system proven in operational environment (competitive manufacturing in the case of key enabling technologies; or in space)

As part of the evaluation effort we will categorise each of the AI-SPRINT assets using the TRL scheme stating the TRL at the beginning of the project and at the end of the project.

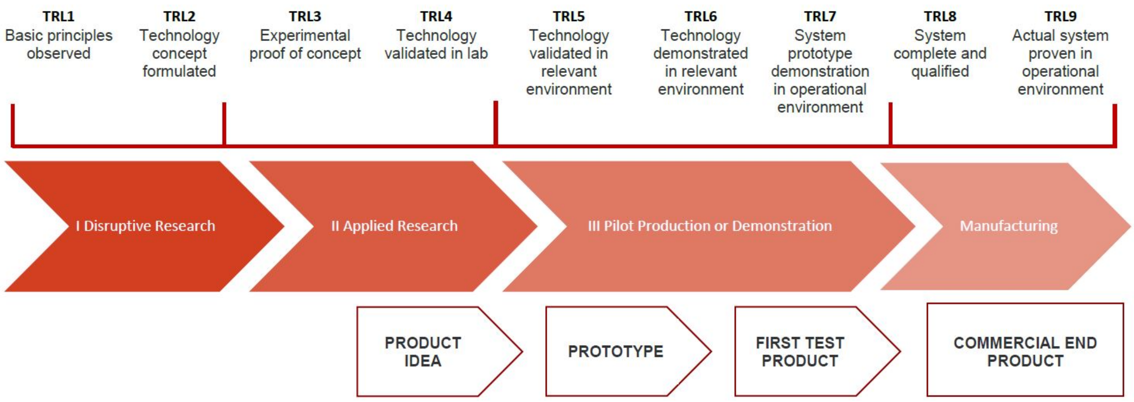

Figure 2.2 - Technological Readiness Levels (image from https://redknightconsultancy.co.uk/an-introduction-to-technology-readiness-levels-trls/)

3. AI-SPRINT Assets & Case Study Coverage

The software based AI-SPRINT solution consists of a set of assets. This section presents the overview of the AI-SPRINT assets that are subjects of the evaluation. Moreover, as evaluation in the framework of use cases in the context of this deliverable the suitability and the coverage of the various AI-SPRINT use cases with respect to evaluating the AI-SPRINT solution is discussed.

3.1 AI-SPRINT Assets

The AI-SPRINT assets are described in detail in deliverables *D2.1 First release and evaluation of the AI-SPRINT design tools*, *D3.1 First release and evaluation of the runtime environment*, *D3.2 First release and evaluation of the monitoring system*, and *D4.1 Initial release and evaluation of the security tools*.

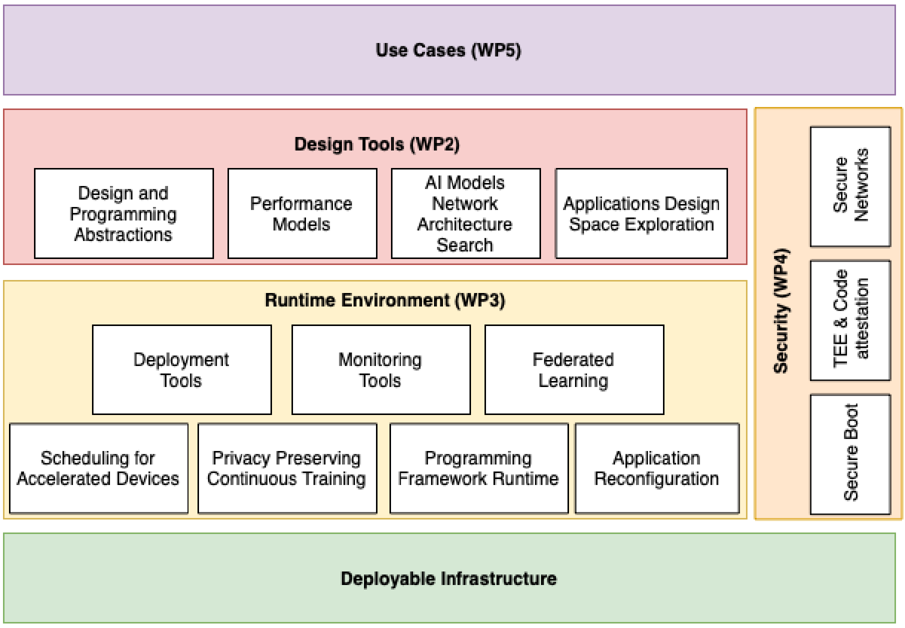

Figure 3.1 - Overall architecture of the AI-SPRINT framework highlighting AI-SPRINT tools

Figure 3.1 above summarises the AI-SPRINT Solution and the tools which are then briefly described in terms of assets in the following tables.

Design time Asset	Short Description
PyCOMPSs	PyCOMPSs provides a sequential programming model to develop AI/ML applications, hiding the complexity of the underlying infrastructure. The programming model includes the definition of the tasks and of the constraints (through Python annotations) to drive the scheduling phase at execution time. Annotations are extended to allow predicating on components performance and to specify constraints on the target deployment.
dislib (distributed machine learning library)	The Distributed Computing Library (dislib) is a Python library built on top of PyCOMPSs that provides distributed mathematical and machine learning algorithms through an easy-to-use interface. dislib abstracts Python

	developers from all the parallelization details, and allows them to build large-scale machine learning workflows in a completely sequential and effortless manner.
Quality Annotations	The module allows users to annotate application components to specify quality and security constraints and a YAML file to describe the system architecture. To this purpose, the module makes use of Python decorators, which wrap the functions of the components involved in the constraint. Furthermore, decorators compute runtime information that is communicated to the monitoring tool. An initial version of the secure policy will be automatically generated by a SCONE parser. The YAML system description will be transformed by a SPACE4AI-D parser to generate the tool initial JSON file which will be completed by the Application architect.
Performance models	Performance models are based on the aMLLibrary machine learning library initially developed within the ATMOSPHERE project and maintained and evolved by the EuroHPC LIGATE project (https://www.ligateproject.eu). This library supports feature selection, hyperparameter tuning, and model selection and is able to generate the most accurate regression model for a specific task. Using the generated regression model, the execution time of inference components or pipelines or training jobs is predicted.
AI Models Architecture Search	The AI Models Architecture Search (also named POPNAS) receives as input a set of configuration parameters and an annotated dataset, e.g., images with labels. The goal is to look for the best neural network architecture for the given classification/regression task. The algorithm searches for the best network configuration to achieve the higher accuracy in the lowest possible time by searching the Pareto front of the time-accuracy trade-off. Architectures on the Pareto front are then proposed as possible candidates to select among.
SPACE4AI-D	SPACE4AI-D tool receives resource description, performance model, performance constraints and application DAG as an input and finds the minimum cost solution for component placement and resource selection problem while guaranteeing performance requirements (namely, requirements on the maximum admissible response times of single components or sequences of components) using, in the current release, a random greedy algorithm. The output of this tool determines the optimal component placement, resource selection and the optimal number of nodes/VMs which helps the developer to find the optimal placement.
SCONE	SCONE is a runtime that is integrated into executables during the compilation process in order to run applications in Trusted Executions Environments (TEE) such as Intel SGX. Besides adding instrumentation to leverage TEEs, it also provides transparent file system encryption as well as secure communications. Applications are attested in order to verify if the code is indeed executed in an enclave of a TEE and has not been tampered with. In case the attestation succeeds, SCONE provides the applications with configuration as well as secrets such that confidential information as well as private keys will never get into human hands.

Table 3.1 - List of design-time AI-SPRINT assets

Deployment and Runtime Asset	Short Description
Infrastructure Manager (IM)	The service manages the complete deployment of virtual infrastructures or individual components within them. The status of a virtual infrastructure can be: ● Pending: launched, but still in initialisation stage. ● Running: created successfully, running, but still in the configuration stage. ● Configured: running and contextualised. ● Unconfigured: running but not correctly contextualised. ● Stopped: stopped or suspended. ● Off: shutdown or removed from the infrastructure. ● Failed: an error occurred during submission.
AI-SPRINT Federated Learning Framework	This module allows the distributed training among different peers according to the privacy level to be achieved via an adequate Federated Learning protocol (e.g., traditional or generative, with or without differential privacy constraints, centralised vs. distributed, etc.). According to the classical federated learning paradigm, each user adheres to the protocol and builds her own local model. Then, the relevant information about the model is shared with the federated party according to the chosen protocol.
rCUDA scheduler	Users of GPU services across the cluster use the rCUDA scheduler to specify the remote GPU requirements of the accelerated job (amount of available GPU memory, type of GPUs, etc) and the rCUDA scheduler tries to satisfy the request from the user by providing the set of remote GPUs that best adapts to the requirements. The rCUDA scheduler provides the set of remote GPUs to be used by the job by providing the exact value of the rCUDA environment variables to be used by the job before it is executed.
GPU Scheduler	Given the list of submitted jobs, in the form of Docker containers, together with information about their characteristics (expected execution times, collected through profiling, priorities and deadlines), and a description of the system with all the available resources, the problem addressed by the GPU Scheduler encompasses three intertwined subproblems: (i) a job scheduling problem that consists in determining which jobs to run among those available in the current time slot and assigning them to the available nodes; (ii) a capacity allocation problem that consists in selecting the most appropriate number of nodes and the best VM type for each node; (iii) a resource partitioning problem, that consists, for each selected node, in partitioning the available GPUs among the selected jobs.
Differential Privacy Component	The Differential Privacy Component has to mitigate the amount of information that can be extracted from a machine learning model via noise injection at training time. To achieve this, the threat level of deep learning models has to be defined by measuring, layer by layer, the privacy level and the amount of information leaked in case of an attack (e.g., membership inference and model inversion attacks).

PyCOMP Superscalar (COMPSs)	Service developers code their application's logic using the methods provided by the disLib library or directly following the COMPSs programming model. The COMPSs runtime will receive an external request to compute something coming either from a manual petition by an end-user or automatically triggered as a response to a change on the data or the infrastructure related to the service. Upon the reception of such a request, the runtime engine divides the application into several tasks, detects the data dependencies among them and orchestrates the execution of such tasks across all the resources within the infrastructure aiming to achieve a shorter execution time and an efficient usage of the nodes belonging to the underlying infrastructure.
Open Source Serverless Computing for Data-Processing Applications (OSCAR) Serverless Container-aware Architectures (SCAR)	With OSCAR, users upload files to a data storage back-end and this automatically triggers the execution of parallel invocations to a service responsible for processing each file. Output files are delivered into a data storage back-end for the convenience of the user. The user only specifies the Docker image and the script to be executed, inside a container created out of that image, in order to process a file that will be automatically made available to the container. The deployment of the computing infrastructure and its scalability is abstracted away from the user. SCAR supports the execution of containers out of Docker images from Docker Hub in AWS Lambda. It is Interoperable with AWS Batch to support GPU-based computing as well as interoperable with OSCAR for serverless workflows.
SPACE4AI-R	SPACE4AI-R, initially triggered by the monitoring infrastructure or run periodically, checks whether QoS local and global constraints (namely, requirements on the maximum admissible response times of single components or sequences of components, see *AI-SPRINT Deliverable D2.1 First release and evaluation of the AI-SPRINT design tools*) are satisfied. If any constraint is violated, the application manager invokes the application optimiser to determine a new optimal solution, adapting the current one to the actual load, and providing, possibly, an updated deployment description to the Infrastructure Manager (IM) server. The new optimal solution may include changing the components configuration (partition switch), and/or scaling the number of cloud VMs used to run specific components and migrating some components from edge to cloud or vice versa.
Krake	Automated scheduling and migration service for containerized applications across heterogeneous cloud platforms based on predefined label constraints and metrics.
Monitoring/Data delivery and storage	Components responsible for gathering metrics data, local buffering and sending it to InfluxDB. Via the stored data it is possible to perform the analysis of time series data describing performance parameters of other subsystems. This asset also allows sending notifications based on results of data flow analysis.

Table 3.2 - List of AI-SPRINT runtime assets

3.2 Coverage of AI-SPRINT Assets by use cases

The evaluation effort described in this deliverable refers to the development of the AI-SPRINT use cases applying the AI-SPRINT solution as well as their execution. This subsection therefore summarises the coverage of the various AI-SPRINT assets by the AI-SPRINT use cases. The actual evaluation plans related to each of the use cases are provided in Section 4.

The Epics, the User Stories and the Requirements of the three use cases were described in detail in *D1.2 Requirements analysis*. All use cases will implement solutions as outlined in the requirements and will test the validity of the AI-SPRINT assets while implementing the required functionality. In the sections below, all requirements have been matched with the AI-SPRINT assets evaluated while implementing the requirement. Note that all assets will be evaluated by at least one use case, but not necessarily by all. Also, as the use cases proceed in their development, the list of evaluated assets might change.

3.2.1 Personalized Healthcare

The requirements from the Personalised Healthcare use case are summarised in the table below.

Requirements	ID	Design	Runtime	Security
Versioning over the three stages of the UC (before, during and after the pilot study)	UC1.Req001	X	X	
Exploratory sensor data analysis and modeling	UC1.Req002		X	
Effective exchange of model parameters between the edge and the cloud	UC1.Req003	X	X	
Mobile phone app development	UC1.Req004		X	X
Hospital model parameters aggregation and update	UC1.Req005	X	X	
Federated learning performance evaluation	UC1.Req006	X	X	X
Ensuring a secure data collection and processing environment	UC1.Req007		X	X
Orchestration and resource management	UC1.Req008	X	X	
FaaS platform for model inference	UC1.Req009		X	
Monitoring and log management	UC1.Req010	X	X	

Table 3.3 - List of Requirements Personalized Healthcare

More specifically, the table below shows which AI-SPRINT asset will be used to implement which requirement and which aspect of the asset will be evaluated.

	UC1.Req001	UC1.Req002	UC1.Req003	UC1.Req004	UC1.Req005
PyCOMPSs		Initial modelling strategies.	Efficient transfer of model parameters between the edge and the cloud.	Interface between the mobile phones and the AI-SPRINT architecture.	Distributed model parameters exchange and update.
dislib		Initial modelling strategies.		Machine learning models.	
Quality Annotations	Versioning scheme reporting.		Quality and security constraints specifications.		
Performance models		Model and infrastructure performance estimation.	Model and infrastructure performance estimation.		
Infrastructure Manager (IM) / EC3					
AI-SPRINT Federated Learning Framework			Federated learning execution.		Federated learning execution.
Open Source Serverless Computing for Data-Processing Applications (OSCAR) Serverless Container-aware Architectures (SCAR)					
SPACE4AI-R					
SCONE				Secure transfer and processing of sensitive information.	
Monitoring/Data delivery and storage			Model and infrastructure performance monitoring and visualisation.		

Table 3.4 - Coverage of AI-SPRINT Solution by the Personalised Healthcare use case (1 to 5)

	UC1.Req006	UC1.Req007	UC1.Req008	UC1.Req009	UC1.Req010
PyCOMPSs			Application orchestration.		
dislib					
Quality Annotations	Quality constraints specifications.	Security constraints specifications.			Quality and security constraints specifications.
Performance models					Model performance and infrastructure performance estimation.
Infrastructure Manager (IM) / EC3			Virtual infrastructure management.		
AI-SPRINT Federated Learning Framework	Federated learning execution.				
Open Source Serverless Computing for Data-Processing Applications (OSCAR) Serverless Container-aware Architectures (SCAR)				Execute inference in a FaaS manner.	
SPACE4AI-R			Optimal resources assignment based on quality constraints.		
SCONE		Secure transfer and processing of sensitive information.			
Monitoring/Data delivery and storage	Model performance and infrastructure performance monitoring and visualisation.		Model performance and infrastructure performance monitoring and visualisation.		Model performance and infrastructure performance monitoring and visualisation.

Table 3.5 - Coverage of AI-SPRINT Solution by the Personalised Healthcare use case (6 to 10)

3.2.2 Inspection and Maintenance use case

The requirements from the Maintenance and Inspection use case are summarized in the table below.

Requirements	ID	Design	Runtime	Security
Off-line operation	UC2.Req001		X	
Model updates	UC2.Req002		X	
Operation metrics	UC2.Req003		X	
Model's performance: response time	UC2.Req004	X	X	
Cloud deployment	UC2.Req005		X	
Model's performance: throughput	UC2.Req006	X	X	
Cloud computing metrics	UC2.Req007		X	
Dynamic deployment and scaling	UC2.Req008		X	
Log collection and browsing	UC2.Req009		X	

Table 3.6 - List of Requirements Maintenance and Inspection

More specifically, the table below shows which AI-SPRINT asset will be used to implement which requirement and which aspect of the asset will be evaluated.

	UC2.Req001	UC2.Req002	UC2.Req003	UC2.Req004	UC2.Req005
PyCOMPSs					
dislib					
Quality Annotations					
Performance models				Design component placement for optimal computation performance.	
AI Models Architecture Search					
SPACE4AI-D				Design component placement for optimal computation performance.	
SCONE					
Infrastructure Manager (IM) / EC3	Edge deployment for offline operation.	Dynamic model deployment.			Cloud resources deployment.
AI-SPRINT Federated Learning Framework					
rCUDA scheduler					
GPU Scheduler					
Differential Privacy Component					
PyCOMP Superscalar (COMPSs)					
Open Source Serverless Computing for Data-Processing Applications (OSCAR) Serverless Container-aware Architectures (SCAR)	Ability to work with occasional lack of communication with the edge devices.	Dynamic model deployment.		Model deployment adjustment to meet KPIs.	Serverless computation model.
SPACE4AI-R	Ability to work with occasional lack of communication with the edge devices.	Dynamic model deployment.		Adjusting resource requirements based on model response time.	
Krake					

Monitoring/Data delivery and storage			Monitoring of models performance and infrastructure operating parameters.	Monitoring of model response times.	.

Table 3.7 - Coverage of AI-SPRINT Solution by the Inspection and Maintenance use case (1 to 5)

	UC2.Req006	UC2.Req007	UC2.Req008	UC2.Req009
PyCOMPSs				
dislib				
Quality Annotations				
Performance models				
AI Models Architecture Search				
SPACE4AI-D				
SCONE				
Infrastructure Manager (IM)			Dynamic resources deployment.	
AI-SPRINT Federated Learning Framework				
rCUDA scheduler				
GPU Scheduler				
Differential Privacy Component				
Open Source Serverless Computing for Data-Processing Applications (OSCAR) Serverless Container-aware Architectures (SCAR)	Model deployment adjustment to meet KPIs.		Model deployment adjustment to meet KPIs.	
SPACE4AI-R	Adjusting resource requirements based on current throughput.		Adjusting resource requirements based on current platform load.	
Krake				
Monitoring/Data delivery and storage	Monitoring of process throughput.	Monitoring of cloud infrastructure.		Platform operation traceability.

Table 3.8 - Coverage of AI-SPRINT Solution by the Inspection and Maintenance use case (6 to 9)

3.2.3 Farming 4.0 use case

The requirements from the Maintenance and Inspection use case are summarised in the table below.

Requirements	ID	Design	Runtime	Security
Run functions as a service	UC3.Req001	X	X	
AI-SPRINT Cloud Functions Connectors	UC3.Req002	X	X	
Software package containing the AI-SPRINT toolset	UC3.Req003		X	X
Run containers as a service	UC3.Req004	X	X	
Create custom monitoring metrics	UC3.Req005	X	X	
Synchronize configuration settings between Edge and Cloud	UC3.Req006	X	X	
Data analysis and modelling	UC3.Req007	X		
Update model version or model parameters on Edge	UC3.Req008	X	X	
Automatically upload files from Edge to Cloud based on rules	UC3.Req009	X	X	
Possibility to manage meta tags for all resources using a console or an API	UC3.Req010	X	X	
Allow moving execution of functions or containers between Cloud and Edge at runtime.	UC3.Req011	X		

Table 3.9 - List of Requirements Farming 4.0

More specifically, the two tables below show which AI-SPRINT asset will be used to implement which requirement and which aspect of the asset will be evaluated.

	UC3.Req001	UC3.Req002	UC3.Req003	UC3.Req004	UC3.Req005	UC3.Req006
PyCOMPSs	Parallel processing of data from cameras (RGB, hyperspectral, from GPS module and from tractor speed					
Infrastructure Manager (IM) / EC3		Orchestrate deployment of solution components on edge device and in multi-cloud environments	Manage deployment of new components or models	Manage deployment of containers on Edge and cloud		
Open Source Serverless Computing for Data-Processing Applications (OSCAR) Serverless Container-aware Architectures (SCAR)	Run all components in FaaS manner	Orchestrate deployment of solution components on edge device and in multi-cloud environments				Use replication of storage mechanism (integration with MinIO)
SPACE4AI-R						Adapt the runtime to fulfil performance constraints
Krake	Optimise GPU usage in model training phase					
COMP Superscalar (COMPSs)				Parallel processing of data from cameras (RGB, hyperspectral, from GPS module and from tractor speed		
SCONE			Ensure Edge device is secure			
Monitoring/Data delivery and storage					Synchronise Edge performance results with cloud	

Table 3.10- Coverage of AI-SPRINT Solution by the Farming 4.0 use case (1 to 6)

	UC3.Req007	UC3.Req008	UC3.Req009	UC3.Req0010	UC3.Req011
Quality Annotations					Define performance constraints on edge device and use at run time
AI Models Architecture Search	Development of models for grape detection, foliage volume computation and disease identification				
Infrastructure Manager (IM) / EC3					Redeploy new versions and redeploy functions based on performance requirements
rCUDA scheduler	Optimise GPU usage in model training phase				
GPU Scheduler	Optimise GPU usage in model training phase				
PyCOMP Superscalar (COMPSs)					
Open Source Serverless Computing for Data-Processing Applications (OSCAR) Serverless Container-aware Architectures (SCAR)			Use replication of storage mechanism (integration with MinIO)		Manage which functions run on the edge and which in the cloud
SPACE4AI-R					Adapt the runtime to fulfill performance constraints
Monitoring/Data delivery and storage			Use replication of storage mechanism (integration with MinIO)		

Table 3.11 - Coverage of AI-SPRINT Solution by the Farming 4.0 use case (7 to 11)

4. Evaluation Plan

This section specifies the evaluation plan. As previously mentioned the focus of this evaluation is on the AI-SPRINT assets with respect to the AI-SPRINT use cases. The main baseline to determine what to evaluate and what characteristics and features to be evaluated are:

1. The main objectives of the project.
2. The Key Performance Indicators (KPIs).
3. The requirements that were identified in WP1 and collected in a coordinated way by task T1.2.
4. The specific objectives of the technical work packages.

Furthermore, all the AI-SPRINT assets according to the Technology Readiness Level are assessed in the evaluation plan.

In the following subsections, after summarizing the evaluation schedule, the project objectives, and project KPIs directly related to the AI-SPRINT use cases, we provide the evaluation plan for each of the use cases.

4.1 Evaluation Schedule

In general, the evaluation described in this evaluation plan will be performed twice. The first evaluation will be conducted on the first implementation of the AI-SPRINT use cases as delivered in M24, and the evaluation will be performed between M20 and M24, while the second evaluation will be performed on the final AI-SPRINT use cases solution delivered in M36 and the evaluation will take place in the period M30-M36. The corresponding evaluation reports *D5.3 Initial implementation and evaluation* and *D5.4 Consolidated implementation and evaluation* will be delivered at M24 and M36 respectively. However, some scenarios and features that are not ready in the first implementation of the AI-SPRINT use case solutions will not be evaluated before the final evaluation round, thus, there will be some variations of what will be evaluated in the first and the second evaluation round. This scheduling is based on the AI-SPRINT overall plan and milestones as described in a more elaborated way in the DoW.

4.2 Evaluation of AI-SPRINT Overall Objectives

The main objective of AI-SPRINT as stated in the Description of Action is to "*define a novel framework for developing and operating AI applications, together with their data, exploiting computing continuum environments, which include resources from the edge up to the cloud. AI-SPRINT will offer novel tools for AI applications development, secure execution, easy deployment, as well as runtime management and optimization. AI-SPRINT tools will allow trading-off application performance (in terms of end-to-end latency or throughput), energy efficiency, and AI model accuracy while providing security and privacy guarantees. The AI-SPRINT framework will support AI application data protection, architecture enhancement, agile delivery, runtime optimisation, and continuous adaptation.*"

The evaluation of AI-SPRINT's main objective is performed by jointly analysing the sets of collected data from the lower level evaluations performed related to the case study implementations and WP specific goals evaluated at the WP levels. This set of collected data are the outcome of use case-specific plans that are described in the remainder of this section. The evaluation of the main AI-SPRINT objective is performed in correspondence to M24 (project Milestone MS IV) and M30 (project Milestone MS V).

4.3 Evaluation of AI-SPRINT KPIs

The strategy of AI-SPRINT to validate its features and to evaluate their performance is based on three real-world use cases: Personalised Healthcare, Inspection and Maintenance and Farming 4.0. Therefore, it is important to define use case-specific KPIs based on the expected outcomes of each use case, as defined in the Description of Work. The following table defines AI-SPRINT's use case-specific KPIs, which are directly based on the use case objectives defined in deliverable *D1.2 Requirements analysis*.

KPI ID	Relevant use case	KPI Description and expected value
KPI-UC1-1	Personalised Healthcare	Number of subjects recruited for the pilot study: >30
KPI-UC1-2	Personalised Healthcare	Number of unclassifiable recordings collected during the pilot study: <10%
KPI-UC1-3	Personalised Healthcare	Number of correctly stratified individuals based on expert evaluation: >90%
KPI-UC2-1	Inspection and Maintenance	Number of images processed during single asset inspection: >300
KPI-UC2-2	Inspection and Maintenance	Response time for quality assessment model: <2s
KPI-UC2-3	Inspection and Maintenance	Initial feedback on major damages: <5 min after inspection is completed.
KPI-UC2-4	Inspection and Maintenance	Reduction in data transferred to the cloud (compared to full dataset): 30%
KPI-UC3-1	Farming 4.0	Pollution reduction: 35%
KPI-UC3-2	Farming 4.0	Increase in good insects for the farm ecosystem: 20%
KPI-UC3-3	Farming 4.0	Prevention of new species resistant to treatment: -25%
KPI-UC3-4	Farming 4.0	Cost reduction by optimising the quantity of chemical products used for spraying: -10%

Table 4.1 - Use-Case Specific KPIs under consideration by the evaluation plan

Differently from project-level Research & Innovation KPIs, use case-specific KPIs defined by Table 4.1 can also act as a guide for development. On the other hand, the Research & Innovation KPIs defined by the DoW and listed in the following table will take a key role in guiding the final evaluation of the project's success, thus completing the set of technical KPIs of the project. The following, as they refer to technical requirements, will be taken into account by the technical work packages WP2, WP3, and WP4 so are reported here just for completeness.

KPI ID	KPI Description and expected value
KPI-RI-1	Release of AI-SPRINT framework under an open-source non-viral license.
KPI-RI-2	3 official releases of the toolchain and 750+ accesses to the software repositories (e.g., GitHub).
KPI-RI-3	Implementation of 3 use cases, together validating 100% of the AI-SPRINT toolchain.
KPI-RI-4	Productivity increment ≥ 30% (both for software and AI models development) verified on ≥ 1 use case.
KPI-RI-5	Support of at least 4 quality metrics in AI applications (e.g., latency, throughput, energy, etc.)
KPI-RI-6	Support of at least 5 cloud technologies (e.g., OpenNebula, OpenStack, Amazon AWS, and at least two EU cloud providers, other than Cloud & Heat AI-SPRINT partner).
KPI-RI-7	Support of at least 3 AI technologies (e.g., PyTorch, TensorFlow, Keras, etc.) across the whole computing continuum of cloud, edge and IoT sensors (e.g., Intel Xeon based servers with NVIDIA GPU, edge servers with embedded GPUs or TPUs, ARM processors and microcontrollers with no AI accelerators).
KPI-RI-8	Support for trusted computing for at least 2 AI technologies (e.g., TensorFlow, Keras, etc.)
KPI-RI-9	Applications operation cost reduction ≥ 20% obtained from efficient and energy-aware use of the continuum resources

Table 4.2 - Research and Innovation KPIs

Productivity increase (KPI-RI-4) is reported here in *D5.1 Evaluation Plan* as they will be measured also by comparing development and deployment time of similar production components (AI models) from at least one use case, based on past activity tracking of the product development team. Cost reductions (KPI-RI9) will be evaluated both from the perspective of the cloud end-users (obtained by the optimal and coordinated use of cloud and edge resources) and of the cloud providers (obtained through energy consumption reduction and efficient management of datacenter resources).

4.4 Evaluation Plan Template for use cases

This subsection provides the evaluation plan template applied for the detailed evaluation plans related to the AI-SPRINT use cases. The evaluation plan template is derived from model-based testing techniques, in particular use case-driven testing and the GQM method.

4.4.1 Goal template

Inspired by the GQM, the following template is applied for describing the specific evaluation goal. The KPI field indicates the related Key Performance Objectives as presented in Section 4.3, bold indicates the more targeted Key Performance Objective for this particular evaluation.

Field	Description (some examples provided)
AI-SPRINT Asset of study	<asset name>
Purpose	Characterise, understand, evaluate, predict, improve, ...
Focus	Functionality, usability, performance, ...
Stakeholders	Application Architect, Application Developer, AI-Expert, Application-Manager, Infrastructure Provider & Sysop, Application and Service Provider and then the use case-specific personas, Stroke Patient, Manager of Stroke Foundation, etc.
KPI	<pick the relevant ones from section 4.3>
Context Factor	Prototype development, lab experiment...

Table 4.3 - Goal template example

The following is an example from the Farming 4.0 use case on how to instantiate a Goal template. It is important to note that multiple assets could be used to fulfill a single goal, in which case different tables for the same goal should be used.

Goal: Yield Estimation

Field	Description (some examples provided)
AI-SPRINT Asset of study	AI Models Architecture Search
Purpose	Evaluate (for the first evaluation the main purpose is to improve our solution towards the end of the project, for the final evaluation round the main purpose is to characterise the actual asset in terms of achievable precision/recall metrics and time for training)
Focus	Characteristics: Functionality Features: ● Precision and recall of models trained with the tool. ● Time required to find suitable models.

	• Inference time of the models obtained.
Stakeholders	Application developer
KPI	KPI-UC3-1, KPI-UC3-2, KPI-UC3-3, KPI-UC3-4
Context Factor	Prototype development and lab experiment with datasets collected on the field.

Table 4.4 - Goal template example from Farming 4.0 use case

4.4.2 Data collection template

The following is the data collection template to be used for the evaluation of the Goals. It is important to note that this data refers to the evaluation of the AI-SPRINT assets according to the GQM framework and not necessarily data used for model development.

Field	Example
Who	<PARTNER>
When	Expected months of the project
How	Explain how the data to answer the question is collected
To whom	Generally this is the AI-SPRINT project or a specific partner
Data collected	<Describe the data according to the question and metrics template>

Table 4.5 - Example of question related to the Farming 4.0 use case

The following is an example from the Farming 4.0 use case on how to instantiate a Data collection template.

Field	Example
Who	POLIMI, GREG
When	From May till September 2022
How	Field campaign experiments an experimental vineyards
To whom	All data conveyed to POLIMI
Data collected	Images captured with an RGB camera and hand labeled to detect the grape clusters are run within the AI Models Architecture Search. Running time, precision, recall, and statistics about the models are extracted and reported.

Table 4.6 - Example of data collection template related to the Farming 4.0 use case

4.4.3 Questions, Metrics, and Data Collection Plan template

Questions and metrics	Description (some examples provided you can extend)
Q1	<formulation of the question>
M1.1	<formulation of (a set of) metrics>
M1.2	...
...	...
Data Collection Plan 1 (DCP1)	This entry provides the plan for: • Who is going to collect data? • When will it be done?

	● How will it be collected? ● Who will the data be delivered to? (The actual collection will be documented as prescribed in the next subsection applying the data collection template)

Table 4.7 - Question template

The following is an example of the Neural Architecture Search related to the fulfillment of one of the requirements in the Farming 4.0 use case.

Questions and metrics	Description (some examples provided you can extend)
Q1	Related to the requirement on yield estimation: can we successfully develop a model which can detect clusters of grapes from an RGB camera mounted on a harvester?
M1.1	Precision and recall of the best model developed
M1.2	Speed-up in terms of classical PNAS which is not Pareto efficient
M1.3	Complexity of the model developed and its inference time
Data Collection Plan 1 (DCP1)	Who: GREG + BECK + POLIMI When: Within M24 How: Data on the field are collected by GREG and POLIMI, Beck and POLIMI will label the data, POLIMI will run the model and collect the results To: POLIMI Data: Images captured with an RGB camera and hand labeled to detect the grape clusters are run within the AI Models Architecture Search. Running time, precision, recall, and statistics about the models are extracted and reported.

Table 4.8 - Example of question related to the Farming 4.0 use case

4.5 AI-SPRINT use cases Evaluation Plans

In the following for each use case we report the Evaluation plan according to the GQM approach. In particular the relevant questions for the evaluation of the specific goals, i.e., the AI-SPRINT assets, are stated explicitly and the means for verification is described.

4.5.1 Personalized Health use case - Goals, questions and metrics

Goal: Secure and privacy-preserving distributed machine learning with sensitive data

AI-SPRINT Asset	dislib
Purpose	Evaluate (the main purpose is to characterise the actual asset in terms of distributed machine learning capabilities)
Focus	Characteristics: Functionality Features: ● Implemented on top of PyCOMPSs programming model ● Execution of machine learning models in distributed platforms ● Ready to use as a library

Stakeholders	AI expert, Application architect
KPI	KPI-UC1-1, KPI-UC1-2, KPI-UC1-3
Context Factor	Prototype deployment and application in pilot study
Questions and metrics	
UC1-Q1	Can distributed machine learning be effectively used for the AI-SPRINT pilot study?
UC1-M1.1	Run-time performance compared to baseline
Data Collection Plan	Who: BSC When: Within M30 How: Run-time performance will be evaluated throughout the pilot study

Table 4.9 - Question UC1-Q1 related to the Personalized Health use case

AI-SPRINT Asset	AI-SPRINT Federated Learning Framework
Purpose	Evaluate (the main purpose is to characterise the actual asset in terms of ability to train a robust machine learning model without sharing data)
Focus	Characteristics: Functionality Features: ● Privacy preservation ● Distributed federated users ● Not confined to one location
Stakeholders	AI expert, Application architect
KPI	KPI-UC1-1, KPI-UC1-2, KPI-UC1-3
Context Factor	Prototype deployment and application in pilot study
Questions and metrics	
UC1-Q2	How are federated learning typical issues (e.g., heterogeneous data distribution, unstable communication, etc.) addressed?
UC1-M2.1	Evaluation of performance constraints
Data Collection Plan	Who: POLIMI & BSC When: Within M30 How: Performance of federated learning will be evaluated throughout the pilot study

Table 4.10 - Question UC1-Q2 related to the Personalized Health use case

AI-SPRINT Asset	SCONE
Purpose	Evaluate (the main purpose is to characterise the actual asset in terms of secure data collection and processing environment)
Focus	Characteristics: Functionality Features: ● Secure transfer and processing of highly sensitive information ● Encryption and secure communication
Stakeholders	Application developer

KPI	KPI-UC1-1, KPI-UC1-2, KPI-UC1-3
Context Factor	Prototype deployment and application in pilot study
Questions and metrics	
UC1-Q3	Is data transfer and processing occurring in a secure and encrypted fashion?
UC1-M3.1	Evaluation of security constraints
Data Collection Plan	Who: TUD When: Within M18 How: Evaluation of the level of security prior to the execution of the pilot study

Table 4.11 - Question UC1-Q3 related to the Personalized Health use case

Goal: Cloud and serverless computing functionalities for model training and inference

AI-SPRINT Asset	IM/EC3
Purpose	Evaluate (the main purpose is to characterise the actual asset in terms of ability to provide virtual infrastructure and deploy of elastic virtual clusters)
Focus	Characteristics: Functionality Features: ● Computing resource provision ● Multi-cloud capabilities ● Adaptability and responsiveness
Stakeholders	Application developer, Infrastructure provider
KPI	KPI-UC1-1, KPI-UC1-2, KPI-UC1-3
Context Factor	Prototype deployment and application in pilot study
Questions and metrics	
UC1-Q4	Are the resource management tools able to select and utilise the most appropriate sites to run jobs in a distributed manner?
UC1-M4.1	Evaluation of automatic deployment and configuration efficiency
Data Collection Plan	Who: UPV When: Within M30 How: Performance evaluation of the virtual cluster management throughout the pilot study

Table 4.12 - Question UC1-Q4 related to the Personalized Health use case

AI-SPRINT Asset	Open Source Serverless Computing for Data-Processing Applications (OSCAR)
Purpose	Evaluate (the main purpose is to characterise the actual asset in terms of FaaS functionalities for model inference)
Focus	Characteristics: Functionality Features: ● Functions as a Service (FaaS) ● Automatic triggering of parallel executions

	• High scalability
Stakeholders	Application manager, Service provider
KPI	KPI-UC1-1, KPI-UC1-2, KPI-UC1-3
Context Factor	Prototype deployment and application in pilot study
Questions and metrics	
UC1-Q5	Can the pre-trained models be run to make inference on-demand efficiently?
UC1-M5.1	Evaluation of the efficiency of event-driven serverless functionalities
Data Collection Plan	Who: UPV When: Within M30 How: Performance evaluation of the inference on-demand using the FaaS framework throughout the pilot study

Table 4.13 - Question UC1-Q5 related to the Personalized Health use case

AI-SPRINT Asset	**PyCOMP Superscalar (COMPSs)**
Purpose	Evaluate (the main purpose is to characterise the actual asset in terms of orchestration of task execution across multiple resources of the distributed infrastructure)
Focus	Characteristics: Functionality Features: • Eased development of parallel applications for distributed infrastructures • Automatic scheduling and execution • Exploitation of parallelism at task level
Stakeholders	Application architect, Infrastructure provider
KPI	KPI-UC1-1, KPI-UC1-2, KPI-UC1-3
Context Factor	Prototype deployment and application in pilot study
Questions and metrics	**PyCOMPSs Superscalar**
UC1-Q6	Can PyCOMPS achieve short execution times and efficient usage of the nodes during the pilot study?
UC1-M6.1	Evaluation of the efficiency of PyCOMPSs functionalities
Data Collection Plan	Who: BSC When: Within M30 How: Performance evaluation of PyCOMPSs throughout the pilot study

Table 4.14 - Question UC1-Q6 related to the Personalized Health use case

Goal: Performance assessment of the use case architecture

AI-SPRINT Asset	**Quality annotations**
Purpose	Validate (the main purpose is to validate the actual asset in terms of specification of quality and security constraints)
Focus	Characteristics: Ease of Use Features:

	● User-defined annotations about the application components ● Runtime information communicated to the monitoring tool ● Information processed by SPACE4AI-D
Stakeholders	Application architect
KPI	KPI-UC1-1, KPI-UC1-2, KPI-UC1-3
Context Factor	Prototype deployment and application in pilot study
Questions and metrics	
UC1-Q7	Which quality annotations are required for the pilot study?
UC1-M7.1	Definition of the quality annotations
Data Collection Plan	Who: POLIMI & BSC When: Within M18 How: Identification of the system descriptors and constraints prior to the pilot study

Table 4.15 - Question UC1-Q7 related to the Personalized Health use case

AI-SPRINT Asset	Performance models
Purpose	Evaluate (the main purpose is to characterise the actual asset in terms of prediction of execution time of required processes of training and inference)
Focus	Characteristics: Functionality Features: ● Accurate regression of execution times ● Resource needs estimation ● Information processed by SPACE4AI-D
Stakeholders	Application architect
KPI	KPI-UC1-1, KPI-UC1-2, KPI-UC1-3
Context Factor	Prototype deployment and application in pilot study
Questions and metrics	
UC1-Q8	Are the models for execution time prediction more accurate than the baseline?
UC1-M8.1	Model performance evaluation
Data Collection Plan	Who: POLIMI When: Within M30 How: Application of standard model performance metrics throughout the pilot study

Table 4.16 - Question UC1-Q8 related to the Personalized Health use case

AI-SPRINT Asset	SPACE4AI-R
Purpose	Evaluate (the main purpose is to characterise the actual asset in terms of adaptability of infrastructure management when the application quality constraints are violated)
Focus	Characteristics: Functionality Features: ● Continuous evaluation of constraint satisfaction.

	• Automatic application optimization (e.g., components migration, updated deployment description, etc.).
Stakeholders	Application architect
KPI	KPI-UC1-1, KPI-UC1-2, KPI-UC1-3
Context Factor	Prototype deployment and application in pilot study
Questions and metrics	
UC1-Q9	Are the optimal solutions identified by SPACE4AI-R mitigating the quality constraint violation?
UC1-M9.1	Evaluation of the performance of the proposed optimal solutions
Data Collection Plan	Who: POLIMI When: Within M30 How: Performance evaluation of the proposed optimal solutions throughout the pilot study

Table 4.17 - Question UC1-Q9 related to the Personalized Health use case

4.5.2 Inspection and Maintenance use case - Goals, questions and metrics

Goal: Effective data acquisition in the field using GPU equipped UAV

AI-SPRINT Asset	Infrastructure Manager (IM)
Purpose	Evaluate
Focus	Functionality Features: • time needed to deploy/update application • possibility to deploy edge components
Stakeholders	Application architect, application manager
KPI	KPI-UC2-1, KPI-UC2-2, KPI-UC2-3, KPI-UC2-4
Context Factor	Prototype deployment
Questions and metrics	
UC2-Q1	Can we efficiently deploy a data processing pipeline to cloud and edge resources?
UC2-M1.1	Time required to deploy on edge component (run-time of deployment process)
UC2-M1.2	Time required to deploy cloud components (run-time of deployment process)
Data Collection Plan	Who: AF When: M24-M34 How: Time of deployment from scratch will be measured in lab conditions. Update deployments will be tested both in lab and in field setup.

Table 4.18 - Question UC2-Q1 related to the Inspection and Maintenance use case

AI-SPRINT Asset	SPACE4AI-R
Purpose	Evaluate
Focus	Characteristics: Functionality / usability

	Features: ● Ease of configuration ● Speed of reaction
Stakeholders	Application manager, sysop
KPI	KPI-UC2-1, KPI-UC2-2, KPI-UC2-3, KPI-UC2-4
Context Factor	Prototype redeployment and rescaling based on simulated performance conditions.
Questions and metrics	
UC2-Q2	Can we trigger a rescaling of the image processing pipeline based on defined performance parameters?
UC2-M2.1	Time to react to performance event
UC2-M2.2	Ease of configuration
Data Collection Plan	Who: AF When: M27-M34 How: performance events are simulated in a lab-environment to trigger redeployment or rescaling of application. Actual behaviour in field condition will be monitored during final tests.

Table 4.19 - Question UC2-Q2 related to the Inspection and Maintenance use case

AI-SPRINT Asset	OSCAR/SCAR
Purpose	Evaluate
Focus	Characteristics: Functionality / usability Features: ● Ease of configuration ● Speed of reaction ● Speed of migration
Stakeholders	Application manager, sysop
KPI	KPI-UC2-1, KPI-UC2-2, KPI-UC2-3, KPI-UC2-4
Context Factor	Prototype redeployment and rescaling based on simulated performance conditions.
Questions and metrics	
UC2-Q3	Can we trigger a rescaling of the image processing pipeline based on defined performance parameters?
UC2-M3.1	Time to react to performance event
UC2-M3.2	Ease of configuration
Data Collection Plan	Who: AF When: M27-M34 How: performance events are simulated in a lab-environment to trigger redeployment or rescaling of application. Actual behaviour in field condition will be monitored during final tests.

Table 4.20 - Question UC2-Q3 related to the Inspection and Maintenance use case

AI-SPRINT Asset	Monitoring/Data storage
Purpose	Validate (check availability of historical metric and log data, validate reporting capabilities)
Focus	Characteristics: Functionality / usability Features: ● Reporting capabilities (dashboarding, complex queries) ● Efficiency of data reporting on big datasets
Stakeholders	Application developer, application manager, data analyst
KPI	KPI-UC2-1, KPI-UC2-2, KPI-UC2-3, KPI-UC2-4
Context Factor	Prototype development and lab experiment with datasets collected during tests.
Questions and metrics	
UC2-Q4	Can we easily create dashboards based on collected data?
UC2-Q5	Can we search historical logs easily and quickly?
UC2-Q6	Can we report efficiently on big datasets ?
UC2-M4.1	Time to search collected logs
UC2-M4.2	Time to create and deploy dashboards
Data Collection Plan	Who: AF When: Within M24-M34 How: Logs are collected by edge devices running in a lab-environment and in the field.

Table 4.21 - Questions UC2-Q4, UC2-Q5, UC2-Q6 related to the Inspection and Maintenance use case

AI-SPRINT Asset	Monitoring/Data delivery
Purpose	Validate (possibility to measure performance of edge application)
Focus	Characteristics: Functionality / usability Features: ● Possibility to collect data on disconnected devices and to mirror it when collection is established ● Possibility to trigger event on edge device based on performance metrics ● Ease of creation of custom metrics ● Complexity of use of the interface for the monitoring
Stakeholders	Application Developer, Application Manager
KPI	KPI-UC2-1, KPI-UC2-2, KPI-UC2-3, KPI-UC2-4
Context Factor	Prototype development and lab experiment with datasets collected on the field.
Questions and metrics	
UC2-Q7	Do the metrics gathered on edge devices (UAV) while there is no network connection are correctly populated to the monitoring infrastructure when the connection is restored?

UC2-M7.1	Time required to synchronise edge nodes that are out of sync.
Data Collection Plan	Who: AF When: Within M24-M32 How: Connection issues will be simulated In lab-environment.

Table 4.22 - Question UC2-Q7 related to the Inspection and Maintenance use case

Goal: Efficient data analysis and report preparation in the cloud

AI-SPRINT Asset	Infrastructure Manager (IM)
Purpose	Evaluate
Focus	Functionality Features: ● time needed to deploy/update application in the cloud ● downtime required to update
Stakeholders	Application architect, application manager
KPI	KPI-UC2-1, KPI-UC2-2, KPI-UC2-3, KPI-UC2-4
Context Factor	Prototype deployment
Questions and metrics	
UC2-Q8	Can we efficiently re-deploy data processing pipeline elements in the cloud?
UC2-M8.1	Time required to deploy cloud components.
UC2-M8.2	Downtime required to update infrastructure.
Data Collection Plan	Who: AF When: M24-M34 How: Cloud application will be deployed and updated during lab tests.

Table 4.23 - Question UC2-Q8 related to the Inspection and Maintenance use case

AI-SPRINT Asset	Open Source Serverless Computing for Data-Processing Applications (OSCAR)
Purpose	Evaluate
Focus	Characteristics: Functionality Features: ● number of concurrent functions running in the cloud ● overhead time to execute function ● possibility to deploy in multi-provider setting (AWS, Azure, private cloud)
Stakeholders	Application Architect, Application Manager
KPI	KPI-UC2-1, KPI-UC2-2, KPI-UC2-3, KPI-UC2-4
Context Factor	Prototype deployment
Questions and metrics	**Open Source Serverless Computing for Data-Processing Applications (OSCAR)**
UC2-Q9	Can we run several models in parallel in a FaaS manner in the cloud?

UC2-M9.1	Number of concurrent inferences
UC2-M9.2	Overhead time to spin up a function
UC2-M9.3	Response times
UC2-M9.4	Throughput
Data Collection Plan	Who: AF When: Within M27-M34 How: Blade images will be fed to the application running in the cloud.

Table 4.24 - Question UC2-Q9 related to the Inspection and Maintenance use case

AI-SPRINT Asset	Monitoring/Data storage
Purpose	Validate (check availability of historical metric and log data, validate reporting capabilities)
Focus	Characteristics: Functionality / usability Features: ● Possibility to analyse performance bottlenecks / performance parameters per component or function ● Log traceability - issues in the infrastructure should be possible to track back to its origin
Stakeholders	Application developer, application manager
KPI	KPI-UC2-1, KPI-UC2-2, KPI-UC2-3, KPI-UC2-4
Context Factor	Prototype development and lab experiments with cloud resources.
Questions and metrics	
UC2-Q10	Can we easily track origins of the issue?
UC2-Q11	Can we identify bottlenecks by measuring performance of each component individually?
UC2-Q12	Can we report efficiently on big datasets
UC2-M10.1	Time to search collected logs
UC2-M10.2	Number of tags that can be assign to datapoint of the metric
Data Collection Plan	Who: AF When: Within M24-M34 How: Logs are collected during cloud application tests.

Table 4.25 - Questions UC2-Q10, UC2-Q11, and UC2-Q12 related to the Inspection and Maintenance use case

AI-SPRINT Asset	Monitoring/Data delivery
Purpose	Validate (possibility to measure performance of cloud application)
Focus	Characteristics: Functionality / usability Features: ● Possibility to efficient collect data from massive, multithreaded processing
Stakeholders	Application Developer, Application Manager

KPI	KPI-UC2-1, KPI-UC2-2, KPI-UC2-3, KPI-UC2-4
Context Factor	Prototype development and lab experiments with wind blade images processing.
Questions and metrics	**Monitoring/Data delivery**
UC2-Q13	Can monitor infrastructure support load from concurrent operations.
UC2-M13.1	Log and metric ingest throughput.
Data Collection Plan	Who: AF When: Within M24-M32 How: Data will be collected during cloud component performance tests.

Table 4.26 - Question UC2-Q13 related to the Inspection and Maintenance use case

4.5.3 Farming 4.0 use case - Goals, questions and metrics

Goal: Adaptive Treatment Management

(Development of a tool which integrates different models and orchestrates their deployment in multi-cloud and edge-cloud environments)

AI-SPRINT Asset	Infrastructure Manager (IM)
Purpose	Evaluate (asset will be evaluated at least twice in terms of stability, operability and ease of use)
Focus	Characteristics: Functionality Features: ● Time needed to deploy/update application ● Possibility to deploy edge components ● Possibility to deploy in multi-provider setting (AWS, Azure, private cloud)
Stakeholders	Application architect, application manager
KPI	KPI-UC3-1, KPI-UC3-2, KPI-UC3-3, KPI-UC3-4
Context Factor	Prototype deployment
Questions and metrics	
UC3-Q1	Can we successfully deploy the adaptive farming application on edge and cloud?
UC3-M1.1	Complexity of deployment file compared with commercially available provider-specific tools (e.g. Terraform)
UC3-M1.2	Time required to deploy on edge component (run-time of deployment process)
Data Collection Plan	Who: BECK When: Within M30 How: Deployment time will be measured in lab and in field conditions (e.g., edge component will be on a vineyard). Qualitative measures will be assessed in the lab-environment by deploying the application on infrastructures from different cloud providers.

Table 4.27 - Question UC3-Q1 related to the Farming 4.0 use case

AI-SPRINT Asset	Open Source Serverless Computing for Data-Processing Applications (OSCAR)
Purpose	Evaluate (asset will be evaluated at least twice in terms of the stability, operability and performance)
Focus	Characteristics: Functionality Features: ● Number of concurrent functions running on the edge device. ● Overhead time to execute function. ● Possibility to deploy in a multi-provider setting (AWS, Azure, private cloud).
Stakeholders	Application Architect, Application Manager
KPI	KPI-UC3-1, KPI-UC3-2, KPI-UC3-3, KPI-UC3-4
Context Factor	Prototype deployment
Questions and metrics	**Open Source Serverless Computing for Data-Processing Applications (OSCAR)**
UC3-Q2	Can we run several models in parallel in a FaaS manner on edge devices?
UC3-M2.1	Number of concurrent inferences
UC3-M2.2	Overhead time to spin up a function
Data Collection Plan 1 (DCP1)	Who: BECK, UPV When: Within M24 How: Images collected on field will serve as input for the application running on the edge in a lab-environment.

Table 4.28 - Question UC3-Q2 related to the Farming 4.0 use case

AI-SPRINT Asset	SPACE4AI-R
Purpose	Evaluate (asset will be evaluated at least twice in terms of the stability, operability and performance)
Focus	Characteristics: Functionality / usability Features: ● Ease of configuration ● Speed of reaction
Stakeholders	Application manager, sysop
KPI	KPI-UC3-1, KPI-UC3-2, KPI-UC3-3, KPI-UC3-4
Context Factor	Prototype redeployment and rescaling based on simulated performance conditions.
Questions and metrics	
UC3-Q3	Can we trigger a rescaling of the farming application based on defined performance parameters?
UC3-M3.1	Time to react to performance event
UC3-M3.2	Ease of configuration
Data Collection Plan	Who: BECK When: Within M34

	How: performance events are simulated in a lab-environment to trigger redeployment or rescaling of application.

Table 4.29 - Question UC3-Q3 related to the Farming 4.0 use case

AI-SPRINT Asset	**Monitoring/Data storage**
Purpose	Validate (check historical data in the Monitoring/Data storage to inspect the execution of detection models for yield estimation, disease detection, execution time of edge components, amount of used chemicals)
Focus	Characteristics: Functionality / usability Features: ● Insert and retrieve historical data from the repository ● Complexity of use of the interface for the monitoring
Stakeholders	Application developer, application manager, factory owner
KPI	KPI-UC3-1, KPI-UC3-2, KPI-UC3-3, KPI-UC3-4
Context Factor	Prototype development and lab experiment with datasets collected on the field.
Questions and metrics	
UC3-Q4	Can we easily create dashboards based on collected data?
UC3-Q5	Can we search historical logs easily and quickly?
UC3-M4.1	Time to search collected logs
UC3-M4.2	Time to create and deploy dashboards
Data Collection Plan	Who: BECK When: Within M32 How: Logs are collected by edge devices running in a lab-environment and in a vineyard. Data is being mirrored to the cloud.

Table 4.30 - Questions UC3-Q4 and UC3-Q5 related to the Farming 4.0 use case

AI-SPRINT Asset	**Monitoring/Data delivery**
Purpose	Validate (possibility to measure performance of edge application)
Focus	Characteristics: Functionality / usability Features: ● Possibility to collect data on disconnected devices and to mirror it when collection is established ● Possibility to trigger event on edge device based on performance metrics ● Ease of creation of custom metrics ● Complexity of use of the interface for the monitoring
Stakeholders	Application Developer, Application Manager
KPI	KPI-UC3-1, KPI-UC3-2, KPI-UC3-3, KPI-UC3-4
Context Factor	Prototype development and lab experiment with datasets collected on the field.
Questions and metrics	

UC3-Q6	Can the edge part of the farming application react based on performance metrics collected on the edge while the application has no connection?
UC3-Q7	Can data be easily mirrored to the cloud?
UC3-M6.1	Ease of management on edge devices
UC3-M6.2	Time lapse between performance event and reaction to event
Data Collection Plan	Who: BECK When: Within M32 How: Performance events will be simulated In lab-environment.

Table 4.31 - Questions UC3-Q6 and UC3-Q7 related to the Farming 4.0 use case

Goal: Yield Estimation and Disease Detection

(Development of models for grape detection and counting in RGB images and for identification of diseases using hyperspectral cameras)

AI-SPRINT Asset	AI Models Architecture Search
Purpose	Evaluate (for the first evaluation the main purpose is to improve our solution towards the end of the project, for the final evaluation round the main purpose is to characterize the actual asset in terms of achievable precision/recall metrics and time for training)
Focus	Characteristics: Functionality Features: ● precision and recall of models trained with the tool ● time required to find suitable models ● inference time of the models obtained
Stakeholders	Application developer
KPI	No specific KPI as this is an additional feature that will be explored during the development of the Use Case
Context Factor	Prototype development and lab experiment with datasets collected on the field.
Questions and metrics	
UC3-Q8	Can we successfully develop a model which can detect clusters of grapes from an RGB camera mounted on a harvester?
UC3-M8.1	Precision and recall of the best model developed
UC3-M8.2	Speed-up in terms of classical PNAS which is not Pareto efficient
UC3-M8.3	Complexity of the model developed and its inference time
Data Collection Plan	Who: GREG + BECK + POLIMI When: Within M24 How: Data on the field are collected by GREG and POLIMI, BECK and POLIMI will label the data, POLIMI will run the model and collect the results To: POLIMI Data: Images captured with an RGB camera and hand labeled to detect the grape clusters are run within the AI Models Architecture Search. Running time, precision, recall, and statistics about the models are extracted and reported.

Table 4.32 - Question UC3-Q8 related to the Farming 4.0 use case

AI-SPRINT Asset	GPU Scheduler
Purpose	Evaluate (for the first evaluation the main purpose is to improve our solution towards the end of the project, for the final evaluation round the main purpose is to characterize the actual asset in terms of achievable usage optimization of GPU resources)
Focus	Characteristics: Functionality Features: ● Time and GPU-resource savings when training models
Stakeholders	AI-Expert
KPI	No specific KPI as this is an additional feature that will be explored during the development of the Use Case
Context Factor	Prototype development and lab experiment with datasets collected on the field.
Questions and metrics	
UC3-Q9	Can we achieve the required model quality with less costs?
UC3-Q10	Can we optimise the utilisation of available GPU resources?
UC3-M9.1	Time savings in model training
UC3-M9.2	Achieved GPU utilisation
Data Collection Plan	Who: POLIMI When: Within M30 How: Models will be trained using POLIMI GPU infrastructure with data collected on fields.

Table 4.33 - Questions UC3-Q9 and UC3-Q10 related to the Farming 4.0 use case

AI-SPRINT Asset	rCUDA
Purpose	Evaluate (for the first evaluation the main purpose is to improve our solution towards the end of the project, for the final evaluation round the main purpose is to characterize the actual asset in terms of achievable usage optimization of GPU resources)
Focus	Characteristics: Functionality Features: ● Time and GPU-resource savings when training models
Stakeholders	AI-Expert
KPI	No specific KPI as this is an additional feature that will be explored during the development of the Use Case
Context Factor	Prototype development and lab experiment with datasets collected on the field.
Questions and metrics	
UC3-Q11	Can we achieve the required model quality with less costs?
UC3-Q12	Can we optimise the utilisation of available GPU resources?
UC3-M11.1	Time savings in model training

UC3-M11.2	Achieved GPU utilisation
Data Collection Plan 1 (DCP1)	Who: UPV When: Within M30 How: Models will be trained using POLIMI GPU infrastructure with data collected on fields.

Table 4.34 - Questions UC3-Q11 and UC3-Q12 related to the Farming 4.0 use case

AI-SPRINT Asset	Krake
Purpose	Evaluate (for the first evaluation the main purpose is to improve our solution towards the end of the project, for the final evaluation round the main purpose is to characterize the actual asset in terms of achievable usage optimization of GPU resources)
Focus	Characteristics: Functionality Features: ● Time and GPU-resource savings when training models
Stakeholders	AI-Expert
KPI	No specific KPI as this is an additional feature that will be explored during the development of the Use Case
Context Factor	Prototype development and lab experiment with datasets collected on the field.
Questions and metrics	
UC3-Q13	Can we achieve the required model quality with less costs?
UC3-Q14	Can we optimise the utilisation of available GPU resources?
UC3-M13.1	Time savings in model training
UC3-M13.2	Achieved GPU utilisation
Data Collection Plan	Who: C&H When: Within M30 How: Models will be trained using POLIMI GPU infrastructure with data collected on fields.

Table 4.35 - Questions UC3-Q13 and UC3-Q14 related to the Farming 4.0 use case

Goal: Sensor-driven PWM Spraying

(Compute the optimal amount of chemicals required for phytosanitary treatment)

AI-SPRINT Asset	PyCOMPSs and dislib
Purpose	Evaluate (for the first evaluation the main purpose is to improve our solution towards the end of the project, for the final evaluation round a main purpose is to characterize the actual asset in terms of time to evaluate all inferences needed on edge)
Focus	Characteristics: Functionality Features: ● Number of parallel inferences

	• Time to complete all tasks needed to compute required quantity of chemicals
Stakeholders	Application developer
KPI	KPI-UC3-1, KPI-UC3-2, KPI-UC3-3, KPI-UC3-4
Context Factor	Prototype development and lab experiment with datasets collected on the field for first evaluation. Field experiment at the end of the project.
Questions and metrics	**PyCOMPSs and dislib**
UC3-Q15	Can we achieve the required model quality with less costs?
UC3-Q16	Can we optimise the utilisation of available GPU resources?
UC3-M15.1	Time savings in model training
UC3-M15.2	Achieved GPU utilisation
Data Collection Plan	Who: POLIMI & BSC When: Within M30 How: Models will be trained using POLIMI/BSC GPU infrastructure with data collected on fields.

Table 4.36 - Questions UC3-Q15 and UC3-Q16 related to the Farming 4.0 use case

AI-SPRINT Asset	Quality Annotation
Purpose	Evaluate (for the first evaluation the main purpose is to improve our solution towards the end of the project, the final evaluation round purpose is to characterize the actual asset in terms of features and ease of use)
Focus	Characteristics: Functionality Features: • Ease of completion of YAML file • Supported provider-agnostic features
Stakeholders	Application architect, application developer
KPI	KPI-UC3-1, KPI-UC3-2, KPI-UC3-3, KPI-UC3-4
Context Factor	Prototype development and lab experiment.
Questions and metrics	
UC3-Q17	Can we set the required quality parameters and constraints in the YAML file?
UC3-M17.1	Time savings in provider-agnostic yaml file vs. provider-specific YAML (Terraform)
Data Collection Plan	Who: BECK When: Within M30 How: Write and modify YAMLl in lab-environment, compare time required with time required to write and maintain more provider-specific YAML formats.

Table 4.37 - Question UC3-Q17 related to the Farming 4.0 use case

4.6 Evaluation with Respect to TRL

As described in the proposal document, AI-SPRINT assets are expected to reach a TRL between 5 and 8 at the end of the project. More specifically, the design & programming abstractions (PyCOMPSs), continuous deployment (IM), programming framework runtime (OSCAR), and TEE components (SCONE), which start from solutions already used in production environments, will have a TRL of 7 or 8. The remaining components will have a TRL of 5 as they require the development of new research. The three use cases were selected because they provide real-space environments, it is not an explicit goal of the project to bring them (within the project time span) to the full maturity needed for a validation into a production operational environment at system level. Some components of the use cases will achieve TRL of 6, others will achieve a lower TRL (for example, in the Farming use case, the adaptive treatment management should achieve TRL of 6, whereas the yield estimation and disease detection will achieve at most a TRL of 3).

The above cited TRL levels will not be considered as a limit: on the contrary, AI-SPRINT commits to push its outcomes as far as possible along the TRL scale, thus facilitating their exploitation after the end of the project. The goal TRL for the individual assets is summarized in the table below.

AI-SPRINT Asset	Initial TRL	Target TRL
PyCOMPSs	6	7
dislib (distributed machine learning library)	6	7
Quality Annotations	3	5
Performance models	3	5
AI Models Architecture Search	3	5
SPACE4AI-D	3	5
SCONE	7	8
Infrastructure Manager (IM)	8	8
AI-SPRINT Federated Learning Framework	1	5
rCUDA scheduler	6	7
GPU Scheduler	3	5
Differential Privacy Component	1	5
Open Source Serverless Computing for Data-Processing Applications (OSCAR) Serverless Container-aware Architectures (SCAR)	7	7
SPACE4AI-R	1	5
Krake	5	7
Monitoring/Data delivery and Data storage	3	7

Table 4.38 - Goal TRL of AI-SPRINT Assets

In order to assess the TRL level of each assets, the checklists compiled as part of the "Human Brain Project" will be used (the project was co-founded by the EU):
https://sos-ch-dk-2.exo.io/public-website-production/filer_public/34/8f/348f734a-9401-4ad6-a0ec-c8eeb3cca358/hbp_trl_assessment_guide_public.pdf

5. Conclusion

This deliverable describes the evaluation plan including the schedule and specific tasks to be performed. Thus, it forms the foundation for the evaluation reports D5.3 and D5.4 due at M24 and M36 respectively. The evaluation plan prescribes to both evaluate the high level objectives of the project as well as the detailed evaluation of the various AI-SPRINT assets in terms of functionality and QoS. The focus of the evaluation reported in this deliverable is related to the AI-SPRINT use cases.

References

[Suchman1967] E. A. Suchman. Evaluation Research: Principles and practice in public service and social action programs. New York: Russell Sage Foundation. 1967.

[ISO/IEC-25010:2011] ISO/IEC 25010:2011 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models.
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=35733

[Basili1994] V. R. Basili, G. Caldiera, H. Dieter Rombach. Goal Question Metric Paradigm. In John J. Marciniak, editor, Encyclopedia of Software Engineering, volume 1, pages 528-532. John Wiley & Sons, 1994.

[Solingen1999] R. Van Solingen, E. Berghout. The Goal/Question/Metric Method. McGraw-Hill Education. ISBN 0-07-709553-7. 1999.

[Basili2009] V.R Basili, J. Heidrich, M. Lindvall, J. Münch, C.B. Seaman, M. Regardie, A. Trendowicz. Determining the impact of business strategies using principles from goal-oriented measurement. Business Services: Konzepte, Technologien, Anwendungen. Internationale Tagung Wirtschaftsinformatik. Books OCG. Vienna, Austria: Österreichische Computer Gesellschaft. ISBN 978-3-85403-246-5. 2009.