Project Title	AN OPEN, TRUSTED FOG COMPUTING PLATFORM FACILITATING THE DEPLOYMENT, ORCHESTRATION AND MANAGEMENT OF SCALABLE, HETEROGENEOUS AND SECURE IOT SERVICES AND CROSS-CLOUD APPS
Project Acronym	RAINBOW
Grant Agreement No	871403
Instrument	Research and Innovation Action
Call / Topic	H2020-ICT-2019-2020 / Cloud Computing
Start Date of Project	01/01/2020
Duration of Project	36 months

D6.8 – Validation, Performance Evaluation & Adoption Guidelines (early release)

Work Package	WP6 – Demonstrators and Performance Evaluators
Lead Author (Org)	George Amponis (K3Y)
Contributing Author(s) (Org)	Stefanos Venios , Panagiotis Dimitriou (SUITE5); Karthik Shenoy Panambur, Shantanoo Desai (BIBA); Marco Marchetti (CRF); Wieńczysław Plutecki, Jacek Piotrowski (MSP); Christina Stratigaki, Konstantinos Theodosiou (UBI); Panagiotis Radoglou-Grammatikis, Marili Zevgara, Sofia Giannakidou, Savvas Ouzounidis (K3Y); Francesca Pacella (LINKS); Raphael Schermann (IFAT); Moysis Symeonidis, Demetris Trihinas, George Pallis (UCY); Thomas Pusztai (TUW); Marco Rapelli, Claudio Casetti, Carla-Fabiana Chiasserini (POLITO); John Avramidis (INTRA)
Due Date	31.12.2021
Actual Date of Submission	22.02.2022
Version	1.0

The work described in this document has been conducted within the project RAINBOW. This project has received funding from the European Union's Horizon 2020 (H2020) research and innovation programme under the Grant Agreement no 871403. This document does not represent the opinion of the European Union, and the European Union is not responsible for any use that might be made of such content.

Dissemination Level

X	PU: Public (*on-line platform)
	PP: Restricted to other programme participants (including the Commission)
	RE: Restricted to a group specified by the consortium (including the Commission)
	CO: Confidential, only for members of the consortium (including the Commission)

Versioning and contribution history

Version	Date	Author	Notes
0.10	06.07.2021	George Amponis (K3Y)	Defined ToC
0.20	01.10.2021	Stefanos Venios (SUITE5), Panagiotis Radoglou-Grammatikis, Sofia Giannakidou (K3Y)	Modifications and additions to the ToC
0.30	08.11.2021	Karthik Shenoy Panambur, Shantanoo Desai (BIBA), Marco Marchetti (CRF), Wieńczysław Plutecki, Jacek Piotrowski (MSP), Christina Stratigaki, Konstantinos Theodosiou (UBI), Marili Zevgara (K3Y), Francesca Pacella (LINKS), Raphael Schermann (IFAT), Moysis Symeonidis, Demetris Trihinas (UCY), Thomas Pusztai (TUW), Marco Rapelli, Claudio Casetti, Carla-Fabiana Chiasserini, (POLITO), John Avramidis (INTRA)	Review of deliverable structure
0.40	09.02.2022	Karthik Shenoy Panambur, Shantanoo Desai (BIBA), Marco Marchetti (CRF), Wieńczysław Plutecki, Jacek Piotrowski (MSP)	Population of technical KPI tables
0.50	13.02.2022	Karthik Shenoy Panambur, Shantanoo Desai (BIBA), CRF, MSP, UCY, SUITE5, TUW	Provision of content for the lessons learnt section.
0.6	15.02.2022	Savvas Ouzounidis (K3Y),	Revision of the lessons learnt and the adoption guidelines section.
0.7	16.02.2022	Marili Zevgara (K3Y)	Revision of the deliverable's relationship to other RAINBOW WPs and tasks.
0.8	17.02.2022	Panatiotis Radoglou-Grammatikis (K3Y), Sofia Giannakidou (K3Y)	Added Executive Summary and Conclusions and updated section 4 (Impact). Sent for

Version	Date	Author	Notes
			internal peer review.
0.9	20.02.2022	George Amponis (K3Y), Savvas Ouzounidis (K3Y), Stefanos Venios (SUITE5)	Implemented internal peer review comments. Sent for submission to the EC
1.0	22.02.2022	George Amponis (K3Y), Stefanos Venios (SUITE5), Christina Stratigaki (UBI)	QA Review, Final version

Table of Contents

List of Tables

List of Figures

Acronyms

Term	Explanation
C-V2X	Cellular Vehicle to Everything
CLI	Command Line Interface
CPA	Collision Prediction and Avoidance
CRUD	Create, Read, Update and Delete
DoA	Description of Action
FPS	Frames per Second
GCS	Ground Control Station
GUI	Graphical User Interface
IoT	Internet of things
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
IT	Information Technology
KPI	Key performance indicator
OPEX	Operational Expenditure
PLMC	Personnel Localization Motion Capturing
RMT	Robot Motion Tracking
SLO	Service Level Objective
SME	Small-Medium enterprise
UI	User Interface

Executive Summary

This deliverable serves as the first report on the evaluation of the RAINBOW platform, based on the early stages of the operation of the platform at the premises of the three demonstrators and the test scenarios performed there. It is positioned between the first and second releases of the RAINBOW platform, as this was the proper timeslot where the RAINBOW platform was operational and the demonstrators had enough time to perform their initial testing activities which have been described as "early scenarios" and are documented in deliverables D6.2 "Human-Robot Collaboration Demonstrator - Early Release", D6.4 "Digital Transformation of Urban Mobility Demonstrator - Early Release", and D6.6 "D6.6 - Power Line Surveillance Demonstrator - Early Release".

As is discussed in the document at hand, since the first release of the RAINBOW platform in M18, demonstrators have been able to deploy the RAINBOW platform at their premises and start executing the designed test cases. Thus, they have been able to measure certain metrics that relate to the anticipated impact as foreseen in the earlier deliverables of WP6 and early results can be classified as positive for the benefits that the platform offers to the use cases selected by the project.

Although there is room for improvement, which is mostly addressing the configurability, the various usability elements of the platform, as well as certain platform limitations affecting the RAINBOW orchestrator flexibility, the collected measurements point towards benefits that have been expected and anticipated from the release of the early-stage demonstrator deliverables. The currently collected measurements refer only to the initial stages of the demonstrators' implementation and can only hint at positive or negative impacts. In order to have a more complete and objective picture, it is essential that the RAINBOW platform releases all other features which are on the backlog, and that the three demonstrators finish their implementation and testing activities due in M33 (September 2022).

The present deliverable documents the lessons learnt from the use cases, as well as some adoption guidelines, formulated with the ultimate goal of supporting wider adoption of the RAINBOW platform. Said adoption guidelines stem from the lessons learnt of the corresponding demonstrators and provide further insight into the expectations of the industrial landscape. A significant amount of work of notable importance and exceptional quality has been done by the technical partners in the context of the RAINBOW platform. The developed platform is currently capable of supporting service graph descriptions and deployment, usage of highly heterogeneous devices as nodes of the same network, runtime monitoring, historic metric storage, streaming analytics and even service-level objective (SLO)-enabled runtime QoS assessment and the elicitation of custom app-level metrics.

This deliverable is structured in a low-to-high abstraction manner: after an introduction to the deliverable, D6.8 engages in a high-level overview of the evaluation framework established in D6.1 "Evaluation Framework and Demonstrators Planning". Continuing,

this deliverable outputs data regarding the validation of the RAINBOW platform, both on a technical and a business level. Next, the deliverable demonstrates the relationship between the various demonstrator-specific KPIs, the elicited business KPIs described in D6.1 and the RAINBOW impact KPIs, as documented in the project's DoA [5]. Lastly, the deliverable extracts valuable information regarding the experience of the demonstrators with the currently available RAINBOW platform, formulating lessons learnt and various adoption guidelines for the wider applicability of the developed platform.

1 Introduction

This section provides the scope in which is the deliverable being developed along with its objectives and its relation to other tasks and deliverables of the RAINBOW project. This section also provides a "bird's eye" view of the rest of the deliverable's structure.

1.1 Scope and Objectives of the Deliverable

The purpose of this document is to provide a report on the early results of the RAINBOW demonstrators and to conduct an interim evaluation of the platform (in both technical and business perspectives), towards identifying weak and strong points of both the platform itself and the usage conducted by the demonstrators. In this context, the deliverable at hand follows the guidelines and procedures that have been devised in D6.1 - Evaluation Framework and Demonstrators Planning and aims to apply the evaluation methodology to the current release of the RAINBOW platform and to the early stages of the demonstrators. It needs to be noted that this deliverable has been compiled with the goal to provide to the consortium a holistic view on the developments of WP6 that will help the further development of the platform, the optimisation of various points and the refinement of the value proposition which can come out of the demonstrators achieving their KPIs and business impact. Lastly, a direct objective of this deliverable is to output lessons learnt from the three RAINBOW demonstrators (documented in Subsections 5.1, 5.2, and 5.3), with the ultimate goal of establishing adoption guidelines (documented in Subsection 5.4) for the wider applicability and usability of the RAINBOW platform targeting a greater industrial/research audience.

1.2 Relationship with other RAINBOW WPs, Tasks and Deliverables

D6.8 "Validation Results and Performance Evaluation – Early Release" acts as an interim document for the demonstration and evaluation phase of the project, which will be revised and delivered as D6.9 "Validation Results and Performance Evaluation – Final Release" in M36 (December 2022) of the project. As such, this deliverable is directly linked with all other deliverables of WP6, which will document the demonstrators' implementation and platform evaluation results. Moreover, D6.8also links the feedback collected from the demonstrators with the rest of the implementation WPs, based on the evaluation that will be performed in WP6. Figure 1 illustrates the relationship of the deliverable at hand to various tasks from WP1, namely Task 1.3 and the corresponding deliverable, and WP6, namely Tasks 6.1, 6.2, 6.3 and 6.4 and the three corresponding early-stage demonstrator deliverables: D6.2 "Human-Robot Collaboration Demonstrator - Early Release" [1], D6.4 "Digital Transformation of Urban Mobility Demonstrator - Early Release" [2], and D6.6 "D6.6 - Power Line Surveillance Demonstrator - Early Release" [3]. Lastly, it is important to note that since the deliverable at hand (D6.8) constitutes an evaluation of the early-stage development of the RAINBOW platform, several metrics, KPIs and further evaluation will mature and get documented in D6.9, as illustrated in the figure below.

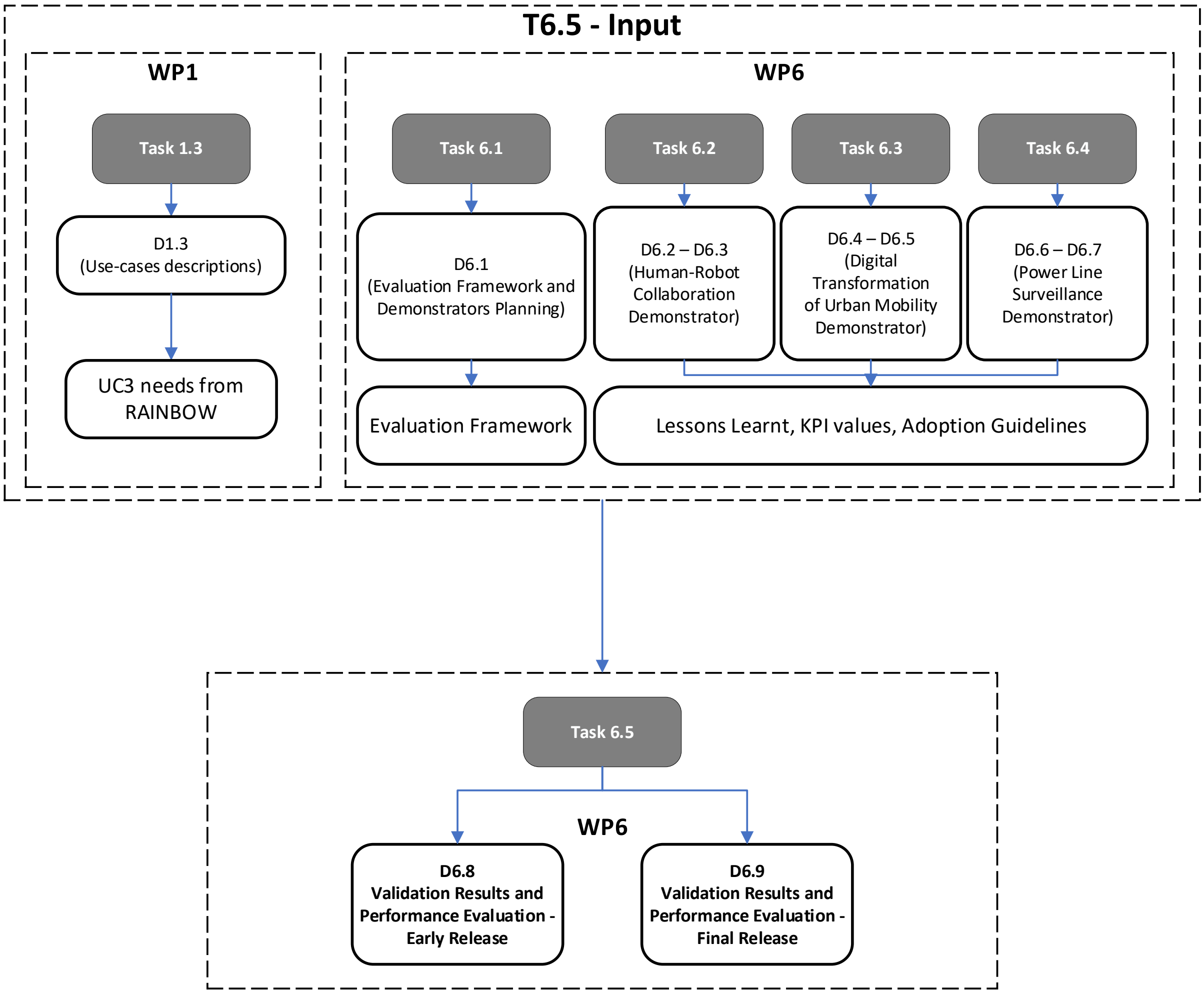

Figure 1: Inputs and outputs of this deliverable

1.3 Structure of the Deliverable

The deliverable is structured as follows:

- Section 1 “Introduction” is responsible for establishing a clear understanding of the objectives of this deliverable and the relationship it has to other RAINBOW WPs, Tasks, and Deliverables.
- Section 2 “The RAINBOW Evaluation Framework Overview” provides a short overview of the RAINBOW evaluation framework as already defined in D6.1 "Evaluation Framework and Demonstrators Planning" [4].
- Section 3 “RAINBOW Validation” serves as the main body of the deliverable, documenting the RAINBOW demonstrators’ KPI measurements.
- Section 4 “RAINBOW Impact” constitutes an endeavour to map the corresponding use cases to the impacted RAINBOW components, whilst also mapping the Impact KPIs from the RAINBOW DoA [5] to the demonstrators’ evaluation metrics.

- Section 5 "Lessons Learnt and Adoption Guidelines" extracts invaluable information on the hands-on experience of the three demonstrators with the first release of the RAINBOW platform. The three use cases provide feedback related to the available services and capabilities, their experience on what can be improved and their views on what can be considered novel in the context of each demonstrator.

2 The RAINBOW Evaluation Framework Overview

This section is dedicated to the analysis of the methodology used for the elicitation of evaluation results, along with the technicalities regarding the provision of feedback on RAINBOW's functionality. The evaluation framework will help the project to demonstrate and evaluate the benefits generated for fog and edge devices by using the RAINBOW methodology. The overall Evaluation Framework, which is briefly discussed in this section has been developed to evaluate RAINBOW from at least two different perspectives:

1. The success of the platform is directly linked to the demonstrators. Taking such a demonstrator-specific perspective in the evaluation will assure that the expectations and requirements of the demonstrator-specific stakeholders are met. This is assured by modelling different scenarios within the demonstrators and involving the related stakeholders within evaluation in each demonstrator.
2. A more generalised perspective, which will assure that expectations and requirements of non-demonstrator-specific stakeholders are evenly met, too.

The following sub-sections present the evaluation approach and the framework that will be implemented and executed during the activities of WP6. The technical verification of the RAINBOW platform is performed in the technical development work packages, while the business evaluation and demonstrator-specific testing are implemented in the context of Task 6.5, which is also outputting the deliverable at hand. The validation and verification of the platform are implemented in a two-step manner and begins in M15 of the project (March 2021). The first iteration of the RAINBOW platform is finalised in M24 (December 2021) and documented in the present deliverable with the evaluation of the early-stage demonstrators. Continuing, the developments continue and are documented in M36 (December 2022) in D6.9. Figure 2 demonstrates the overall flow of the RAINBOW evaluation through the early and advanced development and demonstrator stages.

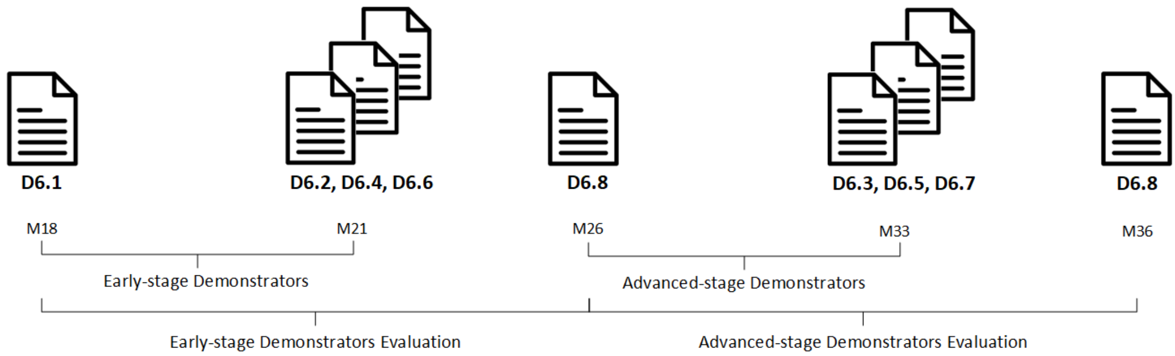

Figure 2: RAINBOW Evaluation timeline

The technical evaluation of RAINBOW also considers ISO 25010:2011 "Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models", as it encompasses a set of models that better address the evaluation of the software quality. As mentioned in D6.1, various perspectives are considered in this standard, namely (1) **Functional Suitability** (i.e., the degree to which the product provides functions that meet stated and implied needs when the product is used under specified conditions), (2) **Performance Efficiency** (i.e., the performance relative to the number of resources used under stated conditions), (3) **Compatibility** (i.e., the degree to which two or more systems or components can exchange information and/or perform their required functions while sharing the same hardware or software environment, (4) **Operability** (i.e., the degree to which the product has attributes that enable it to be understood, learned, used and attractive to the user, when used under specified conditions), (5) **Reliability** (i.e., the degree to which a system or component performs specified functions under specified conditions for a specified period), (6) **Security** (i.e., the degree of protection of information and data so that unauthorised persons or systems cannot read or modify them, and authorised persons or systems are not denied access to them), (7) **Maintainability** (i.e., the degree of effectiveness and efficiency with which the product can be modified, and lastly (8) **Portability** (i.e., the degree to which a system or component can be effectively and efficiently transferred from one hardware, software or other operational or usage environment to another).

Similarly, as mentioned the business evaluation of RAINBOW considers ISO 25010:2011 "Quality in Use" which considers the user's point of view to measure the perception of the quality of the system. This model considers (1) **Effectiveness** (i.e., the accuracy and completeness with which users achieve specified goals), (2) **Efficiency** (i.e., the resources expended concerning the accuracy and completeness with which users achieve goals), (3) **Satisfaction** (i.e., the degree to which users are satisfied with the experience of using a product in a specified context of use), (4) **Safety** (i.e., the degree to which a product or system mitigates the potential risk to economic status, human life, health, or the environment), and lastly (5) **Context coverage** (i.e., the degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in both specified contexts of use and contexts beyond those initially explicitly identified).

3 RAINBOW Validation

This section provides an overview of the technical and business validation of the RAINBOW platform, based on the metrics which have been selected in deliverable D6.1 [4] to showcase the smooth operation of the platform and its performance. As it becomes obvious from Table 2, Table 3 and Table 4, the majority of the metrics which guarantee the technical excellence of the platform are already meeting the targets set, even though there is still much development effort left to release the full feature of the platform. Furthermore, Table 1 highlights the fact that all RAINBOW components are used for each UC deployment and validation. It is important to maintain an understanding of the subsequent interconnections of the various RAINBOW use cases, and the corresponding components, as their outputs in terms of lessons learnt and adoption guidelines will be pivotal in optimising future developments of each component. With that in mind, we deduce that the technical success of the demonstrators is of utmost importance for the effective validation of the various RAINBOW components.

Table 1: RAINBOW component-use case mapping

Layer:	Component:	Use case:
Modelling	Service Graph Editor & Analytics Editor	1, 2, 3
Modelling	Policy Editor	1, 2, 3
Orchestration	Pre-deployment Constraint Solver	1, 2, 3
Orchestration	Deployment Manager	1, 2, 3
Orchestration	Orchestration Lifecycle Manager	1, 2, 3
Orchestration	Resource Manager	1, 2, 3
Orchestration	Resource and Application-level Monitoring	1, 2, 3
Mesh	Mesh Routing Protocol Stack	1, 2, 3
Mesh	Multi-domain Sidecar Proxy	1, 2, 3
Mesh	Security Enablers	1, 2, 3
Data Management & Analytics	Data Storage and Sharing	1, 2, 3
Data Management & Analytics	Analytics Engine	1, 2, 3

3.1 Technical Validation and Platform KPIs

This subsection is dedicated to the evaluation of RAINBOW's technical KPIs and its overall performance on a low (i.e., networking and computing) level. The technical validation is demonstrator-specific. As such, a subsection is dedicated to each use case respective technical validation. The RAINBOW consortium has successfully identified a set of performance-indicating evaluation metrics.

3.1.1 RAINBOW Technical Evaluation in Human-Robot Collaboration in Industrial Ecosystems

This subsection is dedicated to the technical evaluation of the RAINBOW platform in terms of specific metrics defined in D6.1. The Human-Robot Collaboration in Industrial Ecosystems use case has its own set of evaluation metrics and KPIs. As is evident, this use case is focused on low-latency and physical personnel safety for optimal human-robot cooperation.

Regarding the first technical metric of this use case "BIBA-KPI-01", as it is the output of a fully functional RAINBOW platform, it cannot be measured yet with the current maturity of the first release of the RAINBOW platform. The consortium will need to test the scalability of the application prior to the measurement of a such metric.

Similarly, regarding the second technical metric of this use case, "BIBA-KPI-02", at the start of the project this KPI was included for the use case, but after discussion with technical partners, the demonstrator considers that it seems like the RAINBOW platform may or may not provide such features. Considering this, the second technical metric of this use case is no longer relevant to the RAINBOW project; nevertheless, it is documented in the corresponding table as a point of reference.

For metrics BIBA-KPI-03 through BIBA-KPI-08, it must be noted that the SLO configuration YAML file needed to be manually updated using the CLI, as there currently exists no support from the RAINBOW editor to upload the SLO configuration YAML file. Nevertheless, the demonstrator evaluated the SLO execution by increasing the workload of the application and inspecting the number of running instances of the deployment. In conclusion, the RAINBOW platform indeed managed to perform both scale-in and scale-out actions, thus BIBA-KPI-03 to BIBA-KPI-08 are passed. As for BIBA-KPI-09, the infrastructure metrics are already visible through the dashboard for each deployment. For the user-defined metrics, they are exposed and we evaluated the existence of these metrics via manually triggered API calls directly to the RAINBOW storage since they are not accessible through the user's dashboard. In the final version of the platform application-specific metrics will be also visible through the Dashboard. Regarding BIBA-KPI-12, registering user name and password credentials was successful, while data visualization on the editor failed. The dashboard is not showing any of the graphs yet.

Table 2: RAINBOW technical achievements applicable to Demonstrator #1

KPI ID	KPI Title	Baseline Value	Value after RAINBOW	
			Expected	Obtained
BIBA-KPI-01	Deterministic System Latency for collision prediction and avoidance	System Latency is not deterministic with in the tolerance of	<u>Condition check#1</u> Stop Robot when If Metric#1 > 500ms OR Metric#2 > 200ms OR Metric#3 > 10%	Measurable in D6.9 "Validation Results and Performance Evaluation -

KPI ID	KPI Title	Baseline Value	Value after RAINBOW	
			Expected	**Obtained**
		Metric#1 > 500ms OR Metric#2 > 200ms OR Metric#3 > 10% <u>Additional information</u> *Metric#1: Network Latency between Fog Device and Gateway* *Metric#2: Network Jitter between Fog Device and Gateway* *Metric#3: Packet Loss Percentage between Fog Device and Gateway*	<u>Result</u> Condition check#1 need to be meet for the test. Type: Pass/Fail <u>Additional information</u> *Metric#1: Network Latency between Fog Device and Gateway* *Metric#2: Network Jitter between Fog Device and Gateway* *Metric#3: Packet Loss Percentage between Fog Device and Gateway*	Final Release".
BIBA-KPI-02	Reliable hand-off of data for personnel mobility scenario	Not supported.	<u>Condition check#1</u> Data migration (state variables of algorithm pertaining to moved personnel) must be updated in fog device of new work area where personnel has moved recently from fog device of previous workarea <= 1 second <u>Result</u> Condition check#1 need to be meet for the test.	Not supported in current RAINBOW release.

KPI ID	KPI Title	Baseline Value	Value after RAINBOW	
			Expected	**Obtained**
			Type: Pass/Fail	
BIBA-KPI-03	Horizontal scale-out of PLMC services	Not supported.	<u>Condition check#1</u> Maintain at least one instance of PLMC at any given time instance <u>Condition check#2</u> PLMC Service scales out horizontally If (Metric#1 > Metric#2) for 10 seconds AND Metric#3 is greater than 60% <u>Result</u> Condition check #1 **AND** Condition check #2 need to be meet for the test. Type: Pass/Fail <u>Additional information</u> *Metric#1: Message publish rate in RabbitMQ Queue Name: "generator_ personnel_rk"* *Metric#2: Message delivery rate in RabbitMQ Queue Name: "generator_ personnel_rk"* *Metric#3: CPU Utilization*	Pass
BIBA-KPI-04	Horizontal scale-in of PLMC services	Not supported.	<u>Condition check#1</u> Maintain at least one instance of PLMC at any given time instance <u>Conditional check#2</u>	Pass

KPI ID	KPI Title	Baseline Value	Value after RAINBOW	
			Expected	**Obtained**
			PLMC Service scales-in If (Metric#1 < Metric#2) for 10 seconds AND Count (PLMC instance) > 1. <u>Result</u> Condition check #1 **AND** Condition check #2 need to be meet for the test. Type: Pass/Fail <u>Additional information</u> *Metric#1: Message publish rate in RabbitMQ Queue Name: "generator_ personnel_rk"* *Metric#2: Message delivery rate in RabbitMQ Queue Name: "generator_ personnel_rk"* *Metric#3: CPU Utilization*	

BIBA-KPI-05	Horizontal scale-out of CPA services	Not supported.	<u>Condition check#1</u>
			Maintain at least one instance of CPA at any given time instance
			
			<u>Conditional check#2</u>
			If (Metric#1 > Metric#2) for 10 seconds
			OR
			If (Metric#3 > Metric#4) for 10 seconds
			AND
				Pass

KPI ID	KPI Title	Baseline Value	Value after RAINBOW	
			Expected	**Obtained**
			Metric#5 greater than 60% <u>Result</u> Condition check #1 **AND** Condition check #2 need to be meet for the test. Type: Pass/Fail <u>Additional information</u> *Metric#1: Message publish rate in RabbitMQ Queue Name: "rmt_robot_rk"* *Metric#2: Message delivery rate in RabbitMQ Queue Name: "rmt_robot_rk"* *Metric#3: Message publish rate in RabbitMQ Queue Name: "plm_walker_rk"* *Metric#4: Message delivery rate in RabbitMQ Queue Name: "plm_walker_rk"* *Metric#5: CPU Utilization*	
BIBA-KPI-06	Horizontal scale-in of CPA services	Not supported.	<u>Condition check#1</u> Maintain at least one instance of CPA at any given time instance <u>Conditional check#2</u> If (Metric#1 < Metric#2) for 10 seconds OR	Pass

KPI ID	KPI Title	Baseline Value	Value after RAINBOW	
			Expected	**Obtained**
			If (Metric#3 < Metric#4) for 10 seconds AND Count (CPA instance) > 1. Result Condition check #1 **AND** Condition check #2 need to be meet for the test. Type: Pass/Fail Additional information *Metric#1: Message publish rate in RabbitMQ Queue Name: "rmt_robot_rk"* *Metric#2: Message delivery rate in RabbitMQ Queue Name: "rmt_robot_rk"* *Metric#3: Message publish rate in RabbitMQ Queue Name: "plm_walker_rk"* *Metric#4: Message delivery rate in RabbitMQ Queue Name: "plm_walker_rk"* *Metric#5: CPU Utilization*	
BIBA-KPI-07	Horizontal scale-out of RMT services	Not supported.	Condition check#1 Maintain at least one instance of RMT at any given time instance Condition check#2	Pass

KPI ID	KPI Title	Baseline Value	Value after RAINBOW	
			Expected	**Obtained**
			RMT Service scales out horizontally If (Metric#1 > Metric#2) for 10 seconds AND Metric#3 is greater than 60% <u>Result</u> Condition check #1 **AND** Condition check #2 need to be meet for the test. Type: Pass/Fail <u>Additional information</u> *Metric#1: Message publish rate in RabbitMQ Queue Name: "generator_ robot_rk"* *Metric#2: Message delivery rate in RabbitMQ Queue Name: "generator_ robot_rk"* *Metric#3: CPU Utilization*	
BIBA-KPI-08	Horizontal scale-in of RMT services	Not supported.	<u>Condition check#1</u> Maintain at least one instance of RMT at any given time instance <u>Conditional check#2</u> RMT Service scales-in If (Metric#1 < Metric#2) for 10 seconds AND Count (RMT instance) > 1.	Pass

KPI ID	KPI Title	Baseline Value	Value after RAINBOW	
			Expected	Obtained
			<u>Result</u> Condition check #1 **AND** Condition check #2 need to be meet for the test. Type: Pass/Fail <u>Additional information</u> *Metric#1: Message publish rate in RabbitMQ Queue Name: "generator_ robot_rk"* *Metric#2: Message delivery rate in RabbitMQ Queue Name: "generator_ robot_rk"* *Metric#3: CPU Utilization*	
BIBA-KPI-09	Monitoring and evaluation of SLOs.	Not supported	<u>Condition check #1</u> Application specific metrics like message publish rates on RabbitMQ queues, memory statistics of RedisDatabase etc must be monitored <u>Condition check #2</u> Rainbow specific metrics like CPU, memory usage etc must be monitored Result: Condition check #1 and Condition check #2 must be supported Type: Pass/Fail	Pass

KPI ID	KPI Title	Baseline Value	Value after RAINBOW	
			Expected	**Obtained**
BIBA-KPI-10	Data sharing	Not supported	Condition check#1 Only Users registered using valid credentials and access permission are allowed to access data and other RAINBOW platform specific features Result: Condition check #1 and Condition check #2 must be supported Type: Pass/Fail	Pass
BIBA-KPI-11	Security and Attestation	Not supported	Condition check#1 On-boarding of new fog device must adhere to attestation policies set by the service operator. Result: Condition check #1 and must be supported Type: Pass/Fail	Measurable in D6.9 “Validation Results and Performance Evaluation - Final Release”.
BIBA-KPI-12	Data Synchronization	Not supported	Condition check#1 Periodically Synchronize data from all distributed databases present in each of the Fog with Central database Result: Condition check #1 and must be supported Type: Pass/Fail	Measurable in D6.9 “Validation Results and Performance Evaluation - Final Release”.
BIBA-KPI-13	Analytical query	Not supported	Condition check#1 Support CRUD operations for database using Analytical editor of RAINBOW platform to fetch data from	Measurable in D6.9 “Validation Results and Performance Evaluation -

KPI ID	KPI Title	Baseline Value	Value after RAINBOW	
			Expected	Obtained
			Distributed database across Fog device mesh network. Result: Condition check #1 and must be supported Type: Pass/Fail	Final Release".

3.1.2 RAINBOW Technical Evaluation in Digital Transformation of Urban Mobility

This subsection is dedicated to the technical evaluation of the RAINBOW platform in terms of specific metrics defined in D6.1. The Digital Transformation of Urban Mobility use case has its own set of evaluation metrics and KPIs. This use case is focused on low-latency and overall system reliability and node trust-enabling. AHED-KPI-01 and AHED-KPI-02 were measured, and acceptable values were obtained (and thus these KPIs are characterised as "passed"). However, since the use-case is not in its production deployment, the consortium will provide actual numbers from the final evaluation of the platform in D6.9. Regarding AHED-KPI-03, as the MEC server is simple, and there is no control on the connection between the edge node and fog node this early-stage demonstrator is not suitable for network latency measurements. Thus, it is hard to reproduce a scenario where the network latency is "under control".

Table 3: RAINBOW technical achievements applicable to Demonstrator #2

KPI ID	KPI Title	Baseline Value	Value after RAINBOW	
			Expected	Obtained
AHED-KPI-01	AHED service orchestration power consumption	Average power consumption: ~30W	Average power consumption: 15W (Estimation) Average power consumption: <30W (Expected)	Pass
AHED-KPI-02	AHED service orchestration bandwidth occupancy	fps not under control 0 < fps < 30	fps under control 10 < fps < 30	Pass
AHED-KPI-03	AHED C-V2X Alerts delivery latency	300 milliseconds between the	Less or equal to 300 ms	Measurable in D6.9 "Validation Results and

KPI ID	**KPI Title**	**Baseline Value**	**Value after RAINBOW**	
			Expected	**Obtained**
		RSE and the vehicle		Performance Evaluation - Final Release”.
AHED-KPI-04	AHED Number of C-V2X Events managed	V2X exchange messages broadcasted at a frequency between 1 and 10 Hz.	Less or equal to 10 Hz	Measurable in D6.9 “Validation Results and Performance Evaluation - Final Release”.

3.1.3 RAINBOW Technical Evaluation in Power Line Surveillance via Swarm of Drones

This subsection is dedicated to the technical evaluation of the RAINBOW platform in terms of specific metrics defined in D6.1. The Power Line Surveillance via Swarm of Drones use case has its own set of evaluation metrics and KPIs. This use case is focused on automatic deployment, node trust-enabling and extending current system capabilities on a technical level.

Regarding MSP-KPI-04, at the moment the demonstrator has not implemented the Mission Guidance service (documented in D6.6). This means that there is no automatic control over the drones. Consequently, the demonstrator can't determine the exact point where the previous drone has finished its flight. Once the Mission Guidance is functional (advanced stage demonstrator, will be documented in D6.7), the service will gather information about data acquisition progress in real-time and will be able to generate flight routes in such a way that the next drone can proceed with data acquisition from the exact location the previous drone has finished. Subsequently, since this service is not yet implemented on the demonstrator’s end, the corresponding KPI is not yet measurable.

Table 4: RAINBOW technical achievements applicable to Demonstrator #3

KPI ID	**KPI Title**	**Baseline Value**	**Value after RAINBOW**	
			Expected:	**Obtained:**
MSP-KPI-01	Time to pass the control over the drone from one GCS to another GCS	Not supported.	4 sec	3 sec
MSP-KPI-02	Increase of productive flight distance per drone	0%	50%	74,7 %
MSP-KPI-03	Reduction of data acquisition time per	0%	50%	28,2 %

KPI ID	KPI Title	Baseline Value	Value after RAINBOW	
			Expected:	Obtained:
	kilometre of power line.			
MSP-KPI-04	Reduction of overlaps between individual flight routes	0%	-75%	Measurable in D6.9 “Validation Results and Performance Evaluation - Final Release”.
MSP-KPI-05	Efficiency of battery usage for a productive phase of the drone flight	0%	55%	74,7 %

3.2 Business Validation and Platform KPIs

This subsection is dedicated to the evaluation of RAINBOW in terms of business KPIs achievement and overall user satisfaction with the functionalities of the platform. As is the case with the technical validation in the previous subsection, this investigation is demonstrator-specific and thus, a subsection is dedicated to the respective business validation of each use case.

Currently, as the available version of the RAINBOW platform is in its early stages, the business KPIs applicable to this use case, defined in D6.1 are not measurable, and will instead be documented in D6.9, the final release of this deliverable. Moreover, several of the demonstrators’ use case services are not in their final form yet, and thus several business KPIs cannot be measured due to lack of deployment maturity.

3.2.1 RAINBOW Business Evaluation in Human-Robot Collaboration in Industrial Ecosystems

This subsection elaborates on the business validation KPIs of D6.1 and is responsible for documenting its results in the Human-Robot Collaboration in Industrial Ecosystems use case. The following business KPIs will be measured in the final version of this deliverable: RAINBOW-KPI-02 “Software Delivery Cycle”, RAINBOW-KPI-03 “Security Incidents”, RAINBOW-KPI-04 “Service Availability”, RAINBOW-KPI-05 “Cost efficiency”, RAINBOW-KPI-06 “Cloud Infrastructure Costs (OPEX)”, RAINBOW-KPI-07 “Energy Consumption Costs”, RAINBOW-KPI-08 “User Satisfaction”, RAINBOW-KPI-09 “Investments for developing fog computing services”. The status of the use case and RAINBOW developments do not allow for computation of those metrics at the current stage.

Table 5: RAINBOW Business KPIs applicable to Demonstrator #1

ID	KPI	Baseline Value	Value After RAINBOW	
			Expected	Obtained
RAINBOW-KPI-01	Deployment Time	N/A	< 120 sec	90 sec (average)

3.2.2 RAINBOW Business Evaluation in Digital Transformation of Urban Mobility

This subsection elaborates on the business validation KPIs of D6.1 and is responsible for documenting its results in the Digital Transformation of Urban Mobility use case. The following business KPIs will be measured in the final version of this deliverable: RAINBOW-KPI-02 "Software Delivery Cycle", RAINBOW-KPI-03 "Security Incidents", RAINBOW-KPI-05 "Cost efficiency", RAINBOW-KPI-06 "Cloud Infrastructure Costs (OPEX)", RAINBOW-KPI-07 "Energy Consumption Costs", RAINBOW-KPI-08 "User Satisfaction". The current use case and RAINBOW development status do not allow for the computation of those metric at the current stage.

Table 6: RAINBOW Business KPIs applicable to Demonstrator #2

ID	KPI	Baseline Value	Value After RAINBOW	
			Expected	Obtained
RAINBOW-KPI-01	Deployment Time	N/A	< 1 min	20 sec (average)
RAINBOW-KPI-04	Service Availability	N/A	> 99%	100%
RAINBOW-KPI-09	Investments for developing fog computing services	N/A	< 1 PM	1.7 PM

3.2.3 RAINBOW Business Evaluation in Power Line Surveillance via Swarm of Drones

This subsection elaborates on the business validation KPIs of D6.1 and is responsible for documenting its results in the Power Line Surveillance via Swarm of Drones use case. The following business KPIs will be measured in the final version of this deliverable: RAINBOW-KPI-02 "Software Delivery Cycle", RAINBOW-KPI-03 "Security Incidents", RAINBOW-KPI-04 "Service Availability", RAINBOW-KPI-05 "Cost efficiency", RAINBOW-KPI-06 "Cloud Infrastructure Costs (OPEX)", RAINBOW-KPI-07 "Energy Consumption Costs", RAINBOW-KPI-08 "User Satisfaction", RAINBOW-KPI-09 "Investments for developing fog computing services". The status of the use case and RAINBOW developments do not allow for computation of those metrics at the current stage.

Table 7: RAINBOW Business KPIs applicable to Demonstrator #3

ID	KPI	Baseline Value	Value After RAINBOW	
			Expected	Obtained
RAINBOW-KPI-01	Deployment Time	5 min	< 2 sec	50 sec

4 RAINBOW Impact

This section summarizes the technical achievements and the high-level business-oriented achievements in the three RAINBOW use cases as a function of the metrics defined in D6.1 [4]. Furthermore, this section attempts to provide values for the impact KPIs presented in the project's DoA [5]. Method of measurement of KPIs specified in the Description of Action (DoA): KPIs from the DoA have been mapped to metrics defined in D6.1 and rendered measurable via the values collected during demonstrators' testing activities. More specifically, the various metrics from D6.1 were taken into consideration conjointly, with their values being weighted to deduce an accurate mean representative value for each respective KPI from the DoA. Due to their nature and overall definition, KPI.02 "Improved interoperability of cloud-based services used in fog and edge execution environment (compared to other state-of-the-art approaches)" and KPI.03 "Contribution to open-source initiatives and standardization groups dealing with cloud, fog and edge computing paradigms", it was deemed irrelevant to collect measurements and devise values in the context of WP6.

Furthermore, in this section, Table 9 measures values of DoA Impact KPIs, which have been mapped to previously defined business KPIs or technical metrics of the RAINBOW platform. Since many values are currently not populated in the corresponding business/technical KPI tables, the preliminary values presented in Table 9 may be potentially different from the values which will be obtained in the later stages of the project. The initial calculation method can be found in Annex I. The proper values for all the Impact KPIs shall be computed in the final release of this deliverable.

Table 8: Mapping of DoA Impact KPIs to Business/Technical KPIs from D6.1

Impact KPIs (from DoA)	Description	D6.1 Business KPIs	D6.1 Technical KPIs
KPI.1	Decrease in effort and investments for developing and managing the lifecycle of fog computing services and increase software delivery cycles speed (compared to other state-of-the-art approaches)	RAINBOW-KPI-01	Technical KPI mapping not applicable. This impact KPI is a direct function of the three RAINBOW Business KPIs described to the left.
		RAINBOW-KPI-02	
		RAINBOW-KPI-09	
KPI.4	Improved efficiency and performance of	RAINBOW-KPI-04	BIBA-KPI-12

Impact KPIs (from DoA)	Description	D6.1 Business KPIs	D6.1 Technical KPIs
	fog nodes (infrastructure) due to more efficient service development and orchestration (compared to other state-of-the-art approaches)		BIBA-KPI-13 AHED-KPI-02 AHED-KPI-03
KPI.5	Increased productivity of business applications which rely on, or can be developed based on fog computing services (compared to the current status)	RAINBOW-KPI-08	BIBA-KPI-03 BIBA-KPI-04 BIBA-KPI-05 BIBA-KPI-06 BIBA-KPI-07 BIBA-KPI-08 BIBA-KPI-09 AHED-KPI-04 MSP-KPI-02 MSP-KPI-03 MSP-KPI-04
KPI.6	Increased trust feeling of data and services relying on fog-based services and IoT infrastructure in general and improved security and privacy guarantees (compared to other state-of-the-art approaches)	RAINBOW-KPI-03	BIBA-KPI-01 BIBA-KPI-02 BIBA-KPI-10 BIBA-KPI-11 MSP-KPI-01
KPI.7	Decrease in the energy footprint of fog nodes and edge devices (compared to other state-of-the-art approaches)	RAINBOW-KPI-07	AHED-KPI-01 MSP-KPI-05

Impact KPIs (from DoA)	Description	D6.1 Business KPIs	D6.1 Technical KPIs
KPI.8	Lowering access barrier for SMEs for to the usage of advanced technologies for the development, management, security and orchestration of services, made available to SMEs	RAINBOW-KPI-05 RAINBOW-KPI-06	Technical KPI mapping not applicable. This impact KPI is a direct function of the two RAINBOW Business KPIs described to the left.

Table 9: Measuring values of DoA Impact KPIs

DoA KPIs	KPI	Target Value	Measurement
KPI.1	Decrease in effort and investments for developing and managing the lifecycle of fog computing services and increase software delivery cycles speed (compared to other state-of-the-art approaches)	20%	Preliminary value: 16.7% Final value will be computed using KPIs specified in Table 8 in D6.9 "Validation Results and Performance Evaluation - Final Release".
KPI.4	Improved efficiency and performance of fog nodes (infrastructure) due to more efficient service development and orchestration (compared to other state-of-the-art approaches)	25%	Preliminary value: 13.3% Final value will be computed using KPIs specified in Table 8 in D6.9 "Validation Results and Performance Evaluation - Final Release".
KPI.5	Increased productivity of business applications which rely on, or can be developed based on fog computing services (compared to the current status)	25%	Preliminary Value: 20% Final value will be computed using KPIs specified in Table 8 in D6.9 "Validation Results and Performance Evaluation - Final Release".
KPI.6	Increased trust feeling of data and services relying on fog-based services and IoT infrastructure in	30%	Preliminary Value: 20% Final value will be computed using KPIs specified in Table

DoA KPIs	KPI	Target Value	Measurement
	general and improved security and privacy guarantees (compared to other state-of-the-art approaches)		8 in D6.9 "Validation Results and Performance Evaluation - Final Release".
KPI.7	Decrease in the energy footprint of fog nodes and edge devices (compared to other state-of-the-art approaches)	30%	Preliminary Value: 17% Final value will be computed using KPIs specified in Table 8 in D6.9 "Validation Results and Performance Evaluation - Final Release".
KPI.8	Lowering access barrier for SMEs for to the usage of advanced technologies for the development, management, security and orchestration of services, made available to SMEs	25%	Preliminary Value: N/A Final value will be computed using KPIs specified in Table 8 in D6.9 "Validation Results and Performance Evaluation - Final Release".

5 Lessons Learnt and Adoption Guidelines

This section is dedicated to the elicitation of useful lessons learnt as well as an early version of adoption guidelines for the RAINBOW platform. The target is to support the wide adaptability of the RAINBOW platform on a European level, effectively supporting the modernisation and the increase of efficiency of the involved (industrial) stakeholders' desired outputs, considering the corresponding stakeholder requirements, as identified and documented in D1.1 [7]. Invaluable information from three diverse and cutting-edge use cases will be used as input to formulate the adoption guidelines and help align the RAINBOW platform's development with the expectations of the demonstrators' respective industries.

5.1 Lessons Learnt from Human-Robot Collaboration in Industrial Ecosystems

The first RAINBOW use case has been deployed and executed successfully, contributing with valuable information towards the technical team regarding issues that arose during the deployment of RAINBOW and their implemented solutions, the overall experience of the demonstrator, including weak and strong points, and points regarding the potential improvements which can or should be implemented in the next version of the RAINBOW platform and will be validated through the implementation of advanced demonstration of the respective use case.

Regarding issues faced by this demonstrator, it has been reported that currently, there exist no automated scripts capable of facilitating the deployment of the RAINBOW platform at the use case's infrastructure in a zero-touch manner. This means that for now, end-users need to rely on either pre-existing highly technical knowledge or need to be assisted by a more technical counterpart during the deployment process. Additionally, another issue with the RAINBOW platform reported by this use case is the fact that there currently exists no prior documentation for the respective use case partner to understand how the deployment of the components would look like on a container level; this makes it hard to notice potential issues and debug them appropriately, prior to the expression of their effects.

As for the resolution of the aforementioned deployment issues of this use case, technical partners were assigned to help with the deployment of the RAINBOW platform at the premises of the demonstrator. Moreover, in the context of this use case, technical discussions were organized to get a clearer understanding of necessary IT infrastructure and enable the effective emancipation of the end-user for greater autonomy. The technical team of the demonstrator engaged in extensive documentation of all issues faced throughout the deployment process in GitLab, which was pivotal in resolving issues and tracking their process and sharing solutions/guidelines and best practices amongst the involved partners.

When it comes to the strong Points of the RAINBOW platform, this use case validated that the RAINBOW user interface (UI) does indeed **make deployment significantly simplified**, as the end-user does not have to actually write, configure and re-edit YAML files for successful deployment. Furthermore, the RAINBOW platform has demonstrated the capability to provide service graphs that indicate interdependencies often useful to **visualise complex application deployments.**

As no deployment is impeccable, this use case identified some weak points of the RAINBOW platform. Firstly, there is currently no support for volume mounts/config maps making it **hard to upload application-specific configurations.** Additionally, it has been noted that the RAINBOW UI **does not provide sufficiently detailed logs and does not allow direct SLO configuration updates.** Lack of sufficiently detailed logs causes issues related to debugging, while **lack of direct SLO configurability** means that the end-user has to engage in such modifications manually using the command-line interface (CLI) in the respective virtual machines on which the services at hand are running. Lastly, the current version of the RAINBOW platform still requires a considerable amount of **hands-on interaction** with the servers to troubleshoot potential problems arising during deployment.

Considering the remarks above, this use case has identified several constructive improvement points for the RAINBOW platform. This use case considers that the user experience for the RAINBOW UI can be improved. Additionally, a desired functionality that would benefit the demonstrator would be to enable volume mounts for uploading application-specific configurations in a zero-overhead manner. The deployment-related documentation needs to be improved. The RAINBOW platform installation needs to be improved. The amount of CLI-based troubleshooting interaction between the end-user and the virtual machine-based infrastructure needs to be reduced. Lastly, it would be beneficial if RAINBOW could provide detailed application logs related to the UI, Query support, and its Dashboard capabilities.

Key takeaways from the lessons learnt from the first use case include the success of RAINBOW to simplify the deployment of complex applications, as well as the visualisation of such deployments via the RAINBOW UI, despite the lack of automated deployment scripts. Accordingly, in its current development status, the RAINBOW platform does not support high degrees of configurability. Besides that, the logs of the RAINBOW UI are not sufficiently detailed, and as a whole, the platform requires a substantial amount of hands-on interaction with the underlying infrastructure.

5.2 Lessons Learnt from Digital Transformation of Urban Mobility

The second RAINBOW use case has been successfully deployed and executed. This has produced valuable insights concerning the demonstrator's general experience, the onboarding experience of applications to the containerised RAINBOW environment and helped discover potential improvement points for the RAINBOW platform.

Regarding issues faced by this demonstrator, similarly to the first use case, it has been reported that the **lack of a fully automated installation script somewhat hindered the deployment of this demonstrator's application**. Nevertheless, issues of this kind are very common in the early stages of the development of such a complex platform. An additional issue reported by this RAINBOW demonstrator is the **additional complexity introduced by the need to be able to migrate a set of services between nodes with ARM64 and AMD64 processors**. Furthermore, the requirement of one service to have access to a GPU (since the Nvidia Xavier CPU is not powerful enough to properly run the object detection algorithm), led to efforts for discovering a workaround to the lack of GPU support from the underlying Kubernetes orchestrator. The last issue observed in this use case is that **the SLO controller occasionally creates too many queries to the RAINBOW analytics enabler**; his increases CPU usage up to 100% on the RAINBOW master head.

Regarding the issues which were observed in this use case, the resolution was mainly a result of extensive work and support from the technical partners, a process that flowed smoothly and continuously. The issue of making a service executable on ARM64 and AMD64 platforms was resolved through the use of multi-architecture container image manifest lists. Correspondingly, the excessive SLO queries issue was resolved through the manual deletion of the queries; this problem will be further investigated in the corresponding WP5 deliverable and corrected for the final RAINBOW release. As for the last issue, the lack of GPU support by Kubernetes was solved by using RAINBOW's capability to execute containers in privileged mode as a workaround.

When it comes to the strong points of the RAINBOW platform, this use case identified that the process of component and application creation with the RAINBOW user UI **greatly simplifies the deployment** of the application with respect to pure Kubernetes deployment, **thanks to a higher-level approach** that does not require any knowledge about Kubernetes and its technical files for successful deployment. Moreover, the RAINBOW platform provides a service graph which is a **very useful method to visualise service dependencies** and the whole deployment architecture. It is noteworthy that the demonstrator considers that RAINBOW's monitoring framework allows exposing and monitoring arbitrary, application-specific metrics. RAINBOW's SLO framework allows specifying complex SLOs that can be used in conjunction with multiple elasticity strategies to adapt deployments at runtime.

The use case identified some weak points of the RAINBOW platform which are to be used for future improvements. **The current RAINBOW deployment requires running all orchestrated containers with "network mode host"**; this may be undesired for large companies, as it creates security problems for companies ITs. Another important point is that the current version of the RAINBOW platform is the early release version and thus lacks components that facilitate error troubleshooting. As observed in the first use case, **error messages are too generic and often it has been necessary to contact the technical partners to figure out problems**. A weakness that was also observed in the first use case is that it is **not yet possible to upload application-specific configuration files**; this makes the testing phase more time-consuming. It is also mandatory for all files to contain hardcoded settings, and **container images must be rebuilt every time a configuration file is changed**. This requires access to the source code of the respective application, which is often not possible in an industry environment. As noted in the first use case as well, in this first release, the SLOs definition and manipulation is still a hidden tool, which requires the intervention of technical partners to develop them.

Considering the remarks above, this use case has identified several constructive improvement points for the RAINBOW platform. Firstly, this use case considers that external volumes binding can drastically speed up and improve the RAINBOW experience since it would allow dynamic configuration parameter changes on the fly. Additionally, this use case hints that more detailed error messages for easier debugging would be a highly desirable feature. Moreover, the host network mode should not be mandatory, since it may create security issues in production deployments. Lastly, the user interface needs to be extended for easily editing SLOs as the end-user.

Key takeaways from the lessons learnt from the second use case include the measurable simplification of deployment processes thanks to RAINBOW, despite the lack of automated deployment scripts. Nevertheless, the platform currently lacks the capability to migrate services between nodes with ARM64 and AMD64 processors. Moreover, error messages are in need of improvement. Apart from that, the second use case identified the important point that the overall deployment method is not dynamic enough, since service containers are to be rebuilt upon alteration of a configuration file.

5.3 Lessons Learnt from Power Line Surveillance via Swarm of Drones

The third RAINBOW use case was deployed and executed successfully, contributing valuable information towards the technical team related to the perceived demonstrator's experience with the RAINBOW platform, potential issues, shortcomings and positive aspects of the overall design. Lastly, as is the case with the other two use cases, potential improvements are indicated, to help better align the RAINBOW platform with the industry's needs and expectations. It must be noted that the assumptions for the early-stage UC3 tests (that were executed in a virtual environment) were selected to correspond to the real performance of the drone system that will be used in the second stage (that will be executed using real drones under real-life conditions). This is important to be noted, as the virtual demonstrator cannot accurately mimic all environmental conditions that are present in the real world (e.g., weather conditions), but rather try to approximate said metrics computationally.

Thanks to high degrees of compartmentalisation on the demonstrator's end, and timely actions on behalf of the technical partners, no blocking issues were identified in the context of this use case. However, some minor issues due to the maturity of the current (early) release of RAINBOW were found. For instance, **some deployment attempts for our service graph failed even if the underlying nodes had the proper resources**. The latter issue is easily solvable from the user perspective by resubmitting the deployment; this minor issue is to be resolved at the final version of the RAINBOW platform.

As for resolving potential issues, since no blocking issues were identified for this use case, resolution of the identified issue falls within the domain of qualitative improvements and is documented appropriately in the corresponding paragraph below.

This use case has identified a number of strong points regarding the RAINBOW platform. Tests confirmed that RAINBOW, by enabling the direct use of distributed services, **allows for more efficient operations without the efforts of developing and implementing them in order to change the existing centralized services**. Furthermore, the **usage of SLOs allows for a simpler physical topology of the system**. Thanks to the usage of SLOs, there is no need to have different types of nodes (i.e., all hardware can be uniform), so operators can focus on the application layer of their tasks without being concerned with the condition of the system and proper placement of hardware. **The orchestrator automatically assigns services to proper nodes** and makes sure that this allocation is optimal while nodes are added and removed from the system; this is pivotal to the demonstrator and showcases directly applicable added value for the demonstrator.

Nevertheless, this use case also identified some weak points of the RAINBOW platform. To begin with the weak points, currently, **the service graph editor can load but cannot edit previously saved graphs.** This adds some unnecessary overhead each time the end-user has to update the service graph, as they need to re-instantiate the entire service graph (since modifying it is not possible). Additionally, regarding component definition, the names of the interfaces exposed by the components must be unique at all times and across all components. This makes it **hard to reuse components in different service graphs** (to create two applications with one component exchanged, it is necessary to duplicate all definitions of components that consume interfaces of that exchanged component, which causes a chain reaction of changes).

Considering all the positive and negative remarks above, the demonstrator has identified several improvement points for the RAINBOW platform. The algorithm, that assigns services to nodes, needs some refinement: sometimes, **depending on the order in which services are being started during deployment, the service graph cannot be fully deployed**, because there is no node with enough resources, and sometimes, for the very same graph, services are assigned to nodes differently and there is no problem to start them all. Regarding the service graph editor, it would be **advisable to enable editing of a previously saved graph**. Currently, the graph must be edited in one go and even the slightest change requires the user to create a new graph from scratch. Lastly, for component definition, currently, it is possible to define minimal storage requirements in gigabyte increments; as Rainbow targets resource-constrained platforms, it would be more appropriate if fractional values would be allowed as well.

Key takeaways from the lessons learnt from the second use case include the significant increase in operation efficiency, while respective existing service architectures. Usage of SLOs measurably simplifies the system's topology. Importantly, the RAINBOW orchestrator is more than capable of assigning services to the proper network node. Aside from these remarks, some valuable improvement points include the possibility to modify service graphs on-the-go, and the re-usage of components in different service graphs.

5.4 Adoption Guidelines

Successes and failures coming for testing the early version of the platform at the premise of the three use cases resulted in a spectrum of valuable validation results, lessons learnt, improvement points, as well as data concerning the overall experience of the demonstrators with the current version of the RAINBOW platform. Considering the above remarks, outputs and potential improvement points, the consortium has identified some adoption guidelines. In this subsection, matters concerning implementation, operation and execution of the demonstrators are formulated as methodological adoption guidelines for the further exploitation, effective utilisation and further developments towards the advanced release of the RAINBOW platform. While it is still early for the consortium to elicit mature adoption guidelines, the set of stepping-stone guidelines defined in this deliverable will function as pivotal ground rules. As the technical partners move on with the relevant developments, more mature adoption guidelines will be generated and documented in the final version of this deliverable.

Important notes to better align RAINBOW with the stakeholders' expectations include the **automation of deployment** processes and the disengagement of human operators from the deployment and virtual infrastructure management process.

Guidelines elicited from what end users value about RAINBOW:

1. Simplification of containerised application deployments is a strong point, highlighted by all three demonstrators. End users value the fact that through RAINBOW it becomes easier to deploy highly complex services without having to deal with low-level configurations. This eliminates knowledge barriers and enables relevant stakeholders to act more independently. Future RAINBOW developments shall be aligned with this sentiment.
2. SLOs, in conjunction with the visualisation of complex architectures and deployments, is useful for the management of service graphs and interdependencies. Demonstrators valued this capability of RAINBOW, and as such, future developments should be pivoted towards this direction, as it removes undesired overhead from end-users and supports a high-level and more business-oriented approach.
3. The deployment time achieved by RAINBOW is significantly reduced in comparison to the baseline values, which is particularly valuable for demonstrators in the industry.

Considering the above remarks as metrics for the establishment of adoption guidelines, RAINBOW should push towards the direction it is already headed: **business-oriented, user-friendly, simplified deployment of complex applications, and easy-to-understand visualisations of interdependencies and application architectures.** Already, the Impact KPI #1 "Decrease in effort and investments for developing and managing the lifecycle of fog computing services and increase software delivery cycles speed (compared to other state-of-the-art approaches)" showcases a significant decrease in end-user effort for deployments, even in the early development stages. Even with only preliminary measurements and several metrics missing, deployment speed has been measurably augmented, up to fivefold in various demonstrator tests.

Guidelines elicited from what end users would like to see from RAINBOW:

4. Support for more low-level on-demand modification of application configurations, without sacrificing user-friendliness on a higher level, while minimising hands-one interaction with the underlying infrastructure.
5. Ease of the installation process and the documentation, so as to support the end-users in deploying their applications and graphs without requiring external assistance. In that context, more verbose outputs of potential error messages would reduce workload in the event of an unexpected error.
6. Furthermore, easing the process of modifying parameters on the fly is something the end-users would appreciate from RAINBOW.
7. While the existence of service graphs is valued, their modification is still not supported. To that end, end users would benefit from the capability to modify service graphs post-deployment, as that would reduce the amount of time required for modifications and would completely remove the necessity to design new graphs anew.

Considering the above remarks as additional adoption guideline establishment metrics, RAINBOW should consider targeting: **configurability, without sacrificing the zero-touch approach it currently is capable of supporting**.

6 Conclusions

The objective of the present deliverable was to document the evaluation procedure of the three demonstrators of the RAINBOW project, on a technical and business level. This deliverable aims to contribute towards the successful integration of the RAINBOW solution with demonstrator-specific applications, serving the ultimate goal of ensuring wide applicability and diversification of the project's impact. RAINBOW as a platform shows high levels of maturity, even in the early development phases. The capability to create an abstraction layer serving the end-user in the deployment, securing and management of their applications is invaluable to the industry. Upon completion of the technical developments and the resolution of use case-specific issues, all demonstrators will be able to benefit from custom application-layer metrics, "hands-free" infrastructure management and "out-of-the-box" configurability.

The RAINBOW evaluation framework has been specified as flows of validation and evaluation processes spanning from technical to business validation and further to performance evaluation on the basis of specific KPIs of RAINBOW specific functions and the whole solution. Following an iterative development and testing processes scheme, it has been ensured that stakeholders' clarifications/suggestions/changes will be addressed in the oncoming version of the RAINBOW platform, also considering the lessons learnt and the adoption guidelines outputted by the RAINBOW demonstrators. Along these lines, as planned, the list of test objectives and procedures have been refined throughout the project lifetime to better suit implementation specificities that emerge in these project stages, along with testbed specific features, environment setup/tools, etc.

In terms of technical validation of the platform, the different tests that have been conducted showcase the excellence of the platform, as the majority of the technical performance targets are already met by the system. Regarding the business value of the platform, the different users (as part of the demonstrating partners) showcased that RAINBOW has (even at the early development stages following the release of the platform) a rather positive impact on their operations and their development and deployment processes, especially in regards to deployment time and overall ease of use.

As mentioned, the results from the three RAINBOW demonstrators, as detailed in Section 3 "RAINBOW Validation", showcase a substantial level of maturity, with several optimisations being hinted. Issues concerning deployment and potential shortcomings of the platform constitute a substantial basis for the lessons the consortium has learned in regards to the integration of use case-specific elements and containerised applications in a fog environment. Moreover, the pivotal points of the demonstrators'' appreciation of the RAINBOW platform are noted, and future developments shall be direct functions thereof.

Lessons learnt indicate that while there is still room for improvement, the RAINBOW platform is headed in the right direction. Already, support for GUI-based application graph deployments is implemented, and relative functionalities have been welcomed by the demonstrators, as they help reduce overhead and increase end users' productivity. Improvements mainly concern configurability on a more technical level, as well as the addition of some useful functionalities aimed at further increasing productivity and decreasing downtime. The remarks from the three use cases are thus to be considered for future developments and will be subject to closer examination in the next release of the RAINBOW platform.

References

[1] RAINBOW Consortium (2021) Deliverable D6.2 "Human-Robot Collaboration Demonstrator - Early Release"
[2] RAINBOW Consortium (2021) Deliverable D6.4 "Digital Transformation of Urban Mobility Demonstrator - Early Release"
[3] RAINBOW Consortium (2021) Deliverable D6.6 "Power Line Surveillance Demonstrator - Early Release"
[4] RAINBOW Consortium (2021) Deliverable D6.1 "Evaluation Framework and Demonstrators Planning"
[5] RAINBOW Consortium (2020) Description of Action.
[6] RAINBOW Consortium (2021) Deliverable D1.3 "RAINBOW Use-Cases Descriptions"
[7] RAINBOW Consortium (2021) Deliverable D1.1 "RAINBOW Stakeholders Requirements Analysis"

Annex I

Impact KPI calculation formulas:

KPI.1 = x[avg(RAINBOW-KPI-01)]+ y*[avg(RAINBOW-KPI-02)] + z*[avg(RAINBOW-KPI-09)]*

$$\text{where: } x = 0.1,\ y = 0.7,\ z = 0.2$$

KPI.4 = x[avg(RAINBOW-KPI-04)] + y*(BIBA-KPI-12) + z*BIBA-KPI-13) + j*(AHED-KPI-02) +k*(AHED-KPI-03)*

$$\text{where: } x = 0.2,\ y = 0.2,\ z = 0.2,\ j = 0.2,\ k = 0.2$$

KPI.5 = x[avg(RAINBOW-KPI-08)] + y*(BIBA-KPI-03) + z*(BIBA-KPI-04) + j*(BIBA-KPI-05) +k*(BIBA-KPI-06) + p*(BIBA-KPI-07) + r*(BIBA-KPI-08) + d*(BIBA-KPI-09) + u*(AHED-KPI-04) + s*(MSP-KPI-02) + w*(MSP-KPI-03) + q*(MSP-KPI-04)*

where: x = 0.6, y = 0.3/11, z = 0.3/11, j = 0.3/11, k = 0.3/11, p = 0.3/11, r = 0.3/11, d = 0.3/11, u = 0.5/11, s = 0.5/11, w = 0.5/11, q = 0.5/11

KPI.6 = x[avg(RAINBOW-KPI-03)] + y*(BIBA-KPI-01) + z*(BIBA-KPI-02) + j*(BIBA-KPI-10) + k*(BIBA-KPI-11) +p*(MSP-KPI-01)*

where: x = 0.5, y = 0.1, z = 0.1, j = 0.1, k = 0.1, p = 0.1

KPI.7 = x[avg(RAINBOW-KPI-07)] + y*(AHED-KPI-01) + z*(MSP-KPI-05)*

where: x = 0.8, y = 0.1, z = 0.1