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

D1.3 – Use-Cases Descriptions

Work Package	WP1 – Requirements, Reference Architecture and Use-Cases Definition
Lead Author (Org)	Marco Marchetti, Luisa Andreone (CRF)
Contributing Author(s) (Org)	Claudio Casetti, Carla Fabiana Chiasserini (POLITO) Francesca Pacella, Daniele Brevi (LINKS) Karthik Shenoy Panambur (BIBA) Shantanoo Desai (BIBA) Christina Stratigaki, Panagiotis Gouvas, Konstantinos Oikonomou, Konstantinos Theodosiou (UBI) John Kaldis (UNISYSTEMS) George Kakamoukas (K3Y) Demetris Trihinas, Zacharias Georgiou, George Pallis, Moysis Symeonides (UCY) Thomas Pusztai, Schahram Dustdar (TUW) Vasileios Psomiadis, Theodoros Toliopoulos (AUTH) Thanassis Gianetsos (DTU) Ronald Toegl (IFAT) Raphael Schermann (IFAT) Jacek Piotrowski, Wienczyslaw Plutecki (MSP) Orfeas Panagou, Alex Bensenousi (INTRA) Stefanos Venios (Suite5)
Due Date	30.12.2020
Actual Date of Submission	29.01.2021
Version	V1.0

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	16.05.2020	UNISYSTEMS, CRF, UBI	Online ToC
0.20	09.06.2020	SUITE5, UBI, UNISYSTEMS	Initial Draft & Methodologies
0.30	02.07.2020	BIBA, K3Y, LINKS, POLITO, MSP	UC additions-scenarios
0.40	25.08.2020	UNISYSTEMS, AUTH, SUITE5,	KPIs and Business Aspects
0.50	16.09.2020	BIBA, K3Y, LINKS, POLITO	2nd round of additions
0.55	07.10.2020	ALL	Technical Alignment
0.60	30.10.2020	ALL	3rd round of additions
0.70	09.11.2020	CRF,UBI,DTU,UCY,UNISYSTEMS	Restructuring
0.75	24.11.2020	ALL	1st review
0.80	15.12.2020	DTU,UCY,UBI,K3Y,UNISYSTEMS,AUTH	Alignment with Architecture
0.85	17.12.2020	ALL	Final Corrections
0.90	08.01.2021	UCY, AUTH, K3Y, BIBA, MSP	Pre-final review
0.95	18.01.2021	CRF, UBI, UNISYSTEMS	Quality Control-Typo
1.00	29.01.2021	CRF	Final Version

Table of Contents

Executive Summary *10*

1 Introduction *11*

1.1 Scope **11**

1.2 Relation to other Deliverables **11**

1.3 Document Structure **11**

2 Methodology *13*

2.1 Initial Information aggregation **13**

2.2 Alignment with Technical aspects and the RAINBOW Architecture **13**

2.3 Final Structure in steps **14**

2.3.1 Step 1: AS-IS 14
2.3.2 Step 2: Scenarios' needs from RAINBOW 14
2.3.3 Step 3: To be reference scenario 14
2.3.4 Step4: Scenario user stories 14
2.3.5 Step 5: Initial Metrics of Success 15

3 Human-Robot Collaboration in Industrial Ecosystems *16*

3.1 AS-IS scenario **16**

3.1.1 Personnel Localization and Motion Capturing service (PLMC) 19
3.1.2 Robot Motion Tracking service (RMT) 20
3.1.3 Collision Prediction and Avoidance service (CPA) 20

3.2 Scenarios' needs from RAINBOW **22**

3.3 To-be reference scenario **23**

3.4 Scenario user stories **26**

3.5 Initial Metrics of Success **28**

4 UC 2 Digital Transformation of Urban Mobility *30*

4.1 AS-IS scenario **30**

4.1.1 AHED Automatic Hazardous Events Detection 30
4.1.2 Initial configuration 31

4.2 Scenarios' needs from RAINBOW **35**

4.2.1 Smart Orchestration 35
4.2.2 Trust Enablers 36

4.3 To-be reference scenario **37**

4.3.1 Smart Orchestration 37
4.3.2 Trust Enablers 38

4.4 Scenario user stories **39**

4.5 Initial Metrics of Success **45**

5 UC 3 Power Line Surveillance via Swarm of Drones *47*

5.1 AS-IS scenario **47**

5.2 Scenarios' needs from RAINBOW 48
5.3 To-be reference scenario 51
5.4 Scenario user stories 57
5.5 Initial Metrics of Success 65
6 Conclusions 67
7 References 68

List of tables

Table 1: RAINBOW Scenario user stories Template Table 14
Table 2 RAINBOW key features as extracted from the RAINBOW Reference Architecture 15
Table 3 RAINBOW Use case Metrics of Success Template Table 15
Table 4 UC1 KPIs 28

List of figures

Figure 1: Typical human robot collaborative setup in shop floor 16
Figure 2 :Demonstrator overview before RAINBOW 17
Figure 3: Block representation of Node device 18
Figure 4: Block representation of Data Aggregator 19
Figure 5 Safety distance description 21
Figure 6: Demonstrator overview with RAINBOW 24
Figure 7: Services in Fog 25
Figure 8: Automatic Hazardous Events Detection 30
Figure 9 Connected Vehicle 32
Figure 10: Fog Node 33
Figure 11: MEC Node 34
Figure 12: Citizen App 34
Figure 13: Cloud Node 34
Figure 14: Solution Overview 37
Figure 15: AHED orchestration 38
Figure 16: Trust Enablers 39
Figure 17: Current drone system configuration 47
Figure 18: High-level Solution Overview 53
Figure 19: Communication Gateway role in drone control 54
Figure 20: GCS withdrawal, Mission Guidance Service translocation and control over drone transfer 54
Figure 21: Mission Guidance Service operation iteration 55
Figure 22: Hardware components of a GCS node 55

List of abbreviations

AI	Artificial Intelligence
AIK	Attestation Identity Keys
AODV	Ad hoc On-Demand Distance Vector
API	Application programming Interface
AR	Augmented Reality
AWS	Amazon Web Service
BABOK	Business Analysis Body of Knowledge
BVLOS	Beyond Visual Line Of Sight
CFG	Control-Flow Graph
CNC	Computer Numerical Control
CPU	Central Processing Unit
CoAP	Constrained Application Protocol
DAA	Direct Anonymous Attestation
DAG	Directed Acyclic Graph
DFG	Data Flow graph
DNS	Domain Name System
DSO	Distribution System Operators
DSR	Dynamic Source Routing
DTU	Data Terminating Unit
ECC	Elliptic-Curve Cryptography
FR	Functional Requirement
GaaS	Gaming as a Service
GCS	Ground Control Station
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HWMP	Hybrid Wireless Mesh Protocol
IDC	International Data Corporation
IoT	Internet of Things
IT	Information Technology
KPI	Key Performance Index
MAC	Media Access Control
ML	Machine Learning
MQTT	MQ Telemetry Transport or Message Queuing Telemetry Transport
OCF	Open Connectivity Foundation
OT	Operational Technology
PCR	Platform Configuration Register
QoS	Quality of Service
REST	Representational State Transfer
S-ZTP	Secure Zero Touch Provisioning
SDN	Software-Defined Networking
SHA	Secure Hash Algorithm

SLO	Service Level Objective
SME	Subject Matter Expert
SPI	Serial Peripheral Interface
TCG	Trusted Computing Group
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TPM	Trusted Platform Module
TOC	Table of Contents
TSO	Transmission System Operators
UAV	Unmanned Aerial Vehicle
VLOS	Visual Line of Sight
UDP	User Datagram Protocol
VR	Virtual reality
WiMax	Worldwide Interoperability for Microwave Access
ZB	Zettabytes

Executive Summary

The third and final deliverable of WP1, namely "D1.3: RAINBOW Use-Cases Descriptions" attempts to exemplify the three use cases (UCs) which will be used to validate the RAINBOW platform, while conversely trying to indicate ways by which the UCs will benefit through RAINBOW in achieving their business goals. The process begins by depicting the demonstrators' current status, and proceed by analysing needs, describing the "TO BE" status, illustrating specific scenarios, and particularizing on specific metrics. It should be noted that all three UCs in their respective chapters are aligned with the RAINBOW key features as extracted from the RAINBOW Reference Architecture. The present document formulates the input for all further technical design stages and serves as a reference point for all readers to understand the unique features derived from each Use Case.

1 Introduction

1.1 Scope

The scope of this document can be summarized in these main topics:

A. To document RAINBOW early demonstrator descriptions and technology challenges

B. Identify the RAINBOW supported use-cases and help analyse the needs risen up from the demonstrators

C. Describe the initial implementation scenarios of mechanisms that will be developed within the project's demonstrators

D. Link to the RAINBOW Architecture and the Requirements Analysis & Stakeholder Identification of D1.2 and D1.1 respectively

E. Illustrate the "AS-IS" and "TO BE" status

F. Provide tangible metrics of success

G. Serve as a reference point for technical stages and for deeper partner understanding of the UC scenarios

1.2 Relation to other Deliverables

The present D1.3 concludes the cycle of WP1 efforts on Requirements, Reference Architecture and Use-Cases. Despite being submitted chronologically after D1.1 "RAINBOW Stakeholders Requirements Analysis" and D1.2 "RAINBOW Reference Architecture", its first draft version was initiated early in the project, in order to also be used as an input for the aforementioned deliverables since the usecases are a cornerstone proof-of-concept for RAINBOW. Furthermore, D1.1 and D1.2 have also provided input within D1.3 during the technical alignment phase where questions such as "which parts of the RAINBOW architecture will be used" and "how does RAINBOW assist in solving problems at hand or achieving KPIs for the usecases". It should also be noted that D1.3 is a reference point throughout the project for all technical Work-Packages. For example, D6.1 will capitalize on the content of this deliverable by extending the use-case descriptions so that both qualitative and (revised) quantitative metrics are provided and linked with each demonstrator's needs from the RAINBOW Platform.

1.3 Document Structure

The structure of the present document is along these lines:

Section 1 is the introductory section

Section 2 analyses the work methodology for creation of the deliverable

Section 3 presents Use Case 1: Human-Robot Collaboration in Industrial Ecosystems

Section 4 presents Use Case 2: Digital Transformation of Urban Mobility

Section 5 presents Use Case 3: Power Line Surveillance via Swarm of Drones

Section 6 summarizes the conclusions

Section 7 includes references

2 Methodology

Depicting the 3 diverse RAINBOW use cases in a homogenous manner and addressing the potential questions related to technical implementation aspects dictated an iterative process of several steps. More specifically:

2.1 Initial Information aggregation

A series of weekly technical calls dedicated to D1.3 were held among all partners every Friday for the most part of 2020 with several iterations and close monitoring of progress. An online document with contributions by all partners in rounds was used.

In order to make sure that several questions would be addressed properly a set of "placeholder questions" were devised to make sure that all UCs were aligned. The indicative list included the following "brainstorming" questions:

A. Description
B. Setup Description
C. General Goal – KPI to be achieved
D. Target Audience
E. High-level solution overview (e.g., toolkit, SaaS, etc.)
F. Major software components under the hood and technologies (e.g., Docker containers)
G. What will RAINBOW contribute and optimize (e.g., latency, security, trust)
H. Pains
I. Early qualitative and/or quantitative KPIs
J. Dataset
K. Privacy and Security Concerns
L. Underlying Technology
M. Design Constraints
N. Benefits Expected Internally
O. Benefits Expected Externally
P. Testing/Validation Methodology
Q. Physical Topology node and edge

Having gathered input in an "organized manner", the next step was to link the UCs to technical aspects.

2.2 Alignment with Technical aspects and the RAINBOW Architecture

At the second stage, technical partners took the initial input from the previous stage and tried to answer the following:

A. Linking the input to the RAINBOW Architecture
B. Linking the input to the RAINBOW Modules and Roles
C. How RAINBOW is going to be integrated within the UCs
D. How RAINBOW will actually help the UCs fulfil their goals

This has led to the second round of restructuring of the working document

2.3 Final Structure in steps

After many iterations of discussions and workshops with all three demonstrators the final structure of the use cases description unravels as follows:

2.3.1 Step 1: AS-IS

Define as-is scenarios for each Use Case, starting from partners experience or from already existing prototypes and products.

2.3.2 Step 2: Scenarios' needs from RAINBOW

Identify issues and needs from Use Cases that RAINBOW platform can solve. This could include KPIs, goals or other optimisation targets.

2.3.3 Step 3: To be reference scenario

The obvious next step as far as the goals, metrics to be achieved presented as analytically as possible.

2.3.4 Step4: Scenario user stories

This section included Scenarios to be demonstrated written in User story (One table for User Story). The scope is to define detailed user stories with the RAINBOW key features integrated.

UserStoryNum	
User Story Confirmation	
RAINBOW Functionalities	
User Story Implementation and Workflow	

Table 1: RAINBOW Scenario user stories Template Table

The identified use cases target scenarios using RAINBOW platform were derived by highlighting the necessary key features (Table 2). The following table is used as a reference to the RAINBOW key features as extracted from the RAINBOW Reference Architecture:

Feature No.	Reference Architecture Layer	Feature
FT1		Constraint and policy editor

Feature No.	Reference Architecture Layer	Feature
FT2	Modelling	Containerized application packaging
FT3		High-level analytics query editor and job compiler
FT4	Orchestration	Application deployment over fog realms
FT5		Application lifecycle management
FT6		Underlying resource and application runtime adaptation
FT7		Fog-optimized distributed data processing
FT8	Mesh stack	Reactive routing
FT9		Adaptive monitoring
FT10		Zero-touch security fog node configuration
FT11		Fog node "smart" storage

Table 2 RAINBOW key features as extracted from the RAINBOW Reference Architecture

The support of the functionalities defined in this document will be considered as Use Cases requirements for the RAINBOW platform following development phases.

2.3.5 Step 5: Initial Metrics of Success

Qualitative Metrics are considered as Initial Metrics of Success. The Quantitative Metrics will be documented on Deliverable 6.1 "Evaluation Framework and Demonstrators Planning".
The template table used is presented below:

Id	Metric	Target Value	(M)andatory / (G)ood to Have / (O)ptional
1			
2			

Table 3 RAINBOW Use case Metrics of Success Template Table

This is an ongoing process, and it will be properly addressed within D6.1, yet the present deliverable already provides the basic guidelines.

3 Human-Robot Collaboration in Industrial Ecosystems

Reliable indoor positioning enables several innovative location-based services, because such accuracy levels essentially allow for real-time interaction between humans and cyber-physical systems. Activity recognition, machine navigation (e.g., "shelf" level), geo-fencing, and automated robotics; are among services that yield safety-critical assembly processes and logistics. For safety-critical industrial IoT, real-time indoor localization services that monitor the movement of objects and detect human worker's position with respect to the machinery (e.g., heavy-payloads robots), to prevent collisions and accidents. Specifically, the production process demands the involvement of humans and robots to assemble heavy and complex entities like car engines or power supply units, with robots assisting on carrying these heavy products for assembly. The Figure 1 shows a typical human robot collaborative setup in a shop floor.

Figure 1: Typical human robot collaborative setup in shop floor

3.1 AS-IS scenario

The demonstrator consists of a work area as represented in Figure 2 with the following installation and entities:

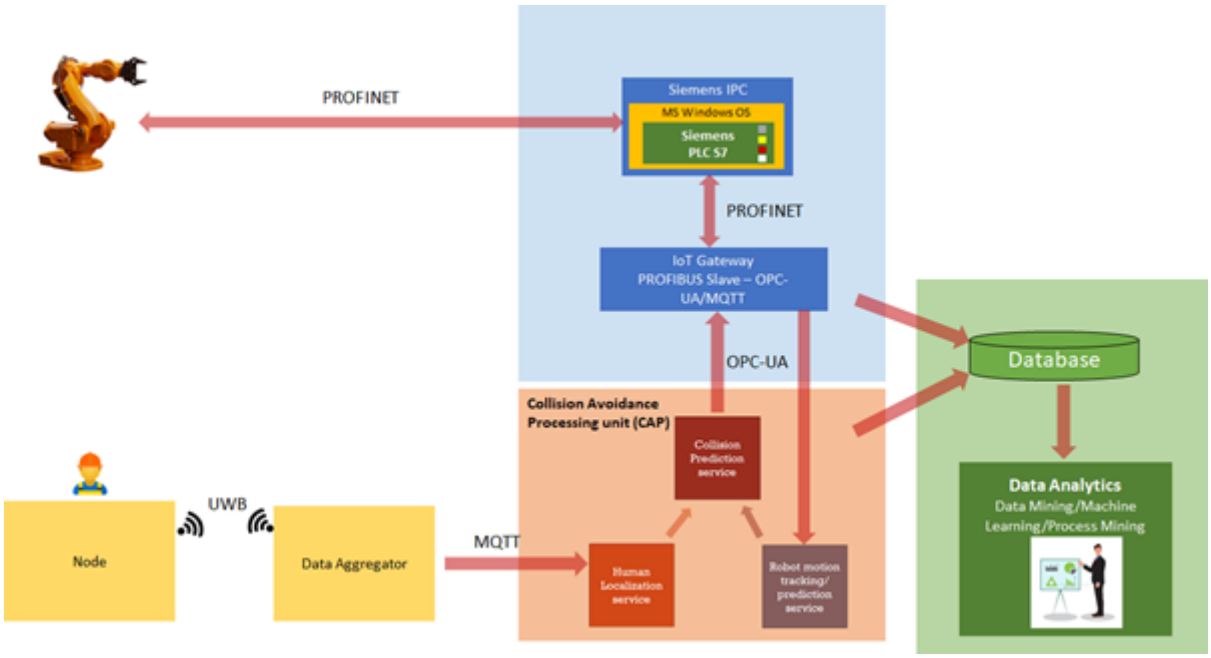


Figure 2 :Demonstrator overview before RAINBOW

- **Industrial Robotic arm:** Used for lifting heavy parts for assembly (e.g., transformer). The robotic arm can be either manually guided or be programmed to carry out particular tasks.

- **Industrial PC (IPC) and Programmable Logic Controller (PLC):** Industrial PC serves as control unit and communicates control signals to the robot arm via PLC. In the particular demonstrator, the Industrial PC will be from Siemens and runs application on Microsoft Window Operating system. Here, PROFINET field bus is used for communication.

- **IoTGateway:** IoTGateway acts as an adapter to communicate information between IPC (using Profinet) and CAP services (using MQTT, OPCUA).

- **Node:** Each Personnel in the shop floor carries a Node device unit all the time. Node device essentially provides instantaneous **3D position coordinate** and **motion dynamics** of a personnel. Each Node device is consisting of **Motion sensors, Ultra-wide band (UWB) tag/s, Microcontroller unit** and **battery** pack to support mobility as shown in Figure 3. Typically, these node devices are embedded in personnel's vest. Motion sensors captures the motion dynamics of the personnel. UWB tag/s capture 3D coordinate of personnel with respect to the stationary UWB anchors (stationary device mounted on walls, roof, or any fixed support structure). The microcontroller performs digital signal processing (filtering) on extracted data from motion sensors and tag/s before sending the processed data to **Data Aggregator** as shown in the Figure 3**.**

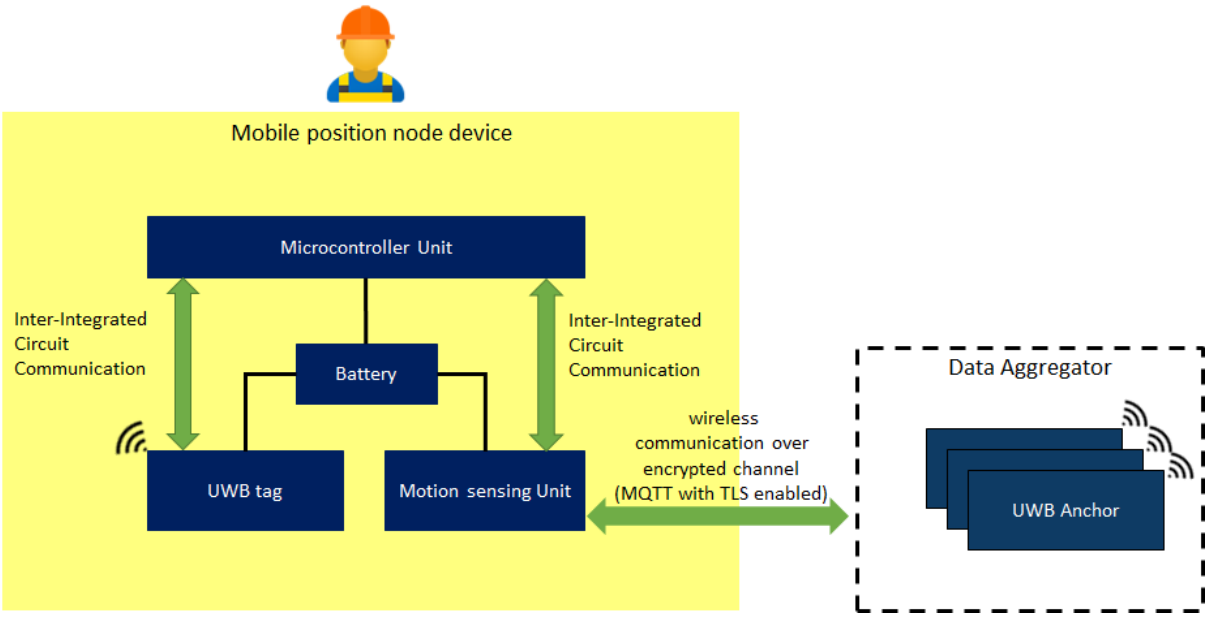


Figure 3: Block representation of Node device

- **Data Aggregator:** Primary purpose of Data aggregator is to receive telemetry data from multiple Node devices and send these received data to subscribed devices using MQTT (**Message Queue Telemetry Transport)** with TLS enabled. A work-place area can contain one or more Data Aggregators. Number of Data Aggregator is decided based upon maximum number of personnel allowed in work-place area simultaneously. Node devices uses UWB for distance ranging and telemetry, due to superior channel characteristics in noisy industrial environment. As UWB physical layer is not supported inherently by IoT Gateway or Fog devices. Thus, the need of Data aggregator as an interface between Node device and Gateway/Fog devices. The Data aggregator consists of **UWB Anchors** (distributed over work-place), **MQTT Broker** as shown in Figure 4. UWB Anchor serves two-fold purpose. First, it acts a reference beacon in distance ranging for Node devices in a work-place area. Second, it receives the telemetry messages containing motion sensing and 3D coordinates information from Node device units. These received messages are published to the local MQTT broker present in the same Data Aggregator device. MQTT broker provides the telemetry data to the MQTT client (typically running on other devices such as IoT Gateway/ Fog device) subscribed to relevant MQTT topics.

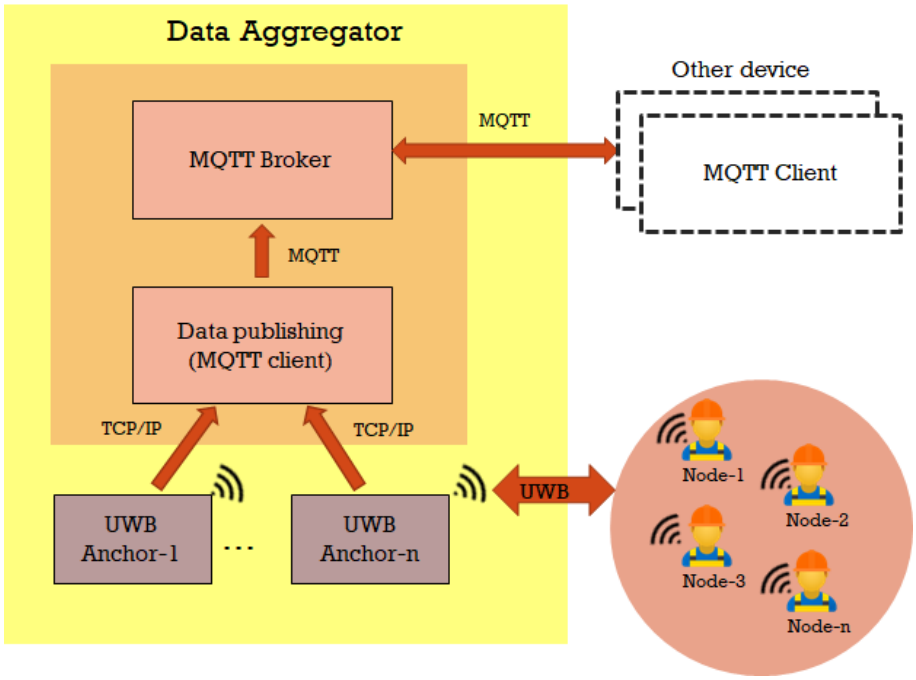


Figure 4: Block representation of Data Aggregator

- **Collision Avoidance Processing unit (CAP):** Primary purpose of CAP unit is to avoid fatal collision between robot arm and personnel. This is done by estimating probability of collision between robot arm and personnel in proximity, ahead of specified time, by taking account of current location (3D coordinates) and motion dynamics of both robot and personnel. Based on estimated probability CAP takes a decision to slow down or to stop the robot. The CAP consists of three distinct services:
    1. Personnel Localization and motion capturing service
    2. Robot motion tracking service
    3. Collision prediction and avoidance service

Each of the services and functionalities supported by them is discussed subsequently.

3.1.1 Personnel Localization and Motion Capturing service (PLMC)

This service provides optimal estimate of personnel's 3D coordinates (with respect to anchor placed in shop floor) and predicts their future motion trajectory time ahead with certain confidence level in different regions. The regions (shape and size) are determined based of likelihood.
One instance of this service is assigned to exactly one personnel in the work-place area.

This service provides following information:

- Optimal estimate of personnel's instantaneous position
- Estimate of personnel's future position/region of presence, small time ahead
- Monitor Node device QoS (Quality of Service) Parameters

The data obtained from localization sensing are often noisy and are affected by environment factors such as interference, multi-path fading etc. As behaviour of noise, interferences are unknown and are stochastic in nature. Extracting the exact measurement from noisy measurement is not feasible. Instead, it is possible to acquire an estimate of measurements at a given time. To obtain best estimate of measurement probabilistic algorithms are used such as Robust Adaptive Linear Quadratic estimator, Multi-model adaptive Kalman filters, Marginalized-Particle filter etc.

Adding to this, using probabilistic algorithm it is also feasible to predict future motion trajectories time ahead with certain confidence in each region. These algorithms are computationally intense and are required to run in hard-real time constraints. Thus, the need to run these algorithms on powerful multi-core processors.

3.1.2 Robot Motion Tracking service (RMT)

This service tracks robot arm movement and also provide future motion path at every joints. One instance of this service is assigned to exactly one Robot in the work-place area.

The service provides the following:

- Instantaneous 3D Coordinate of the robot joints
- Future motion path of robot joints ahead of time

As the Industrial PC provides instantaneous joint angles of robot arm. This service performs Forward Kinematics operations to extract 3D coordinate position of end effector. Since the Robot motion planning unit knows ahead of time about future motion attributes (like joint angles) of robot. These attributes are obtained to predict future coordinates/regions of presence.

3.1.3 Collision Prediction and Avoidance service (CPA)

This service combines the information from PLMC services and RMT services. And uses probabilistic algorithm to predict the probability of collision between a given personnel and robot in a work-place area time ahead. If possibility of collision is detected, based on likelihood, safety distance and velocities of approaching Personnel and Robot, CPA service either slows down the robot or stops the Robot by sending appropriate control signal to PLC via IPC. One instance of CPA service is assigned to a group of Robots and personnel(s) in a work-place area.

This service provides the following:

- Probability of collision between given Personnel and Robot time ahead
- Stop or slow Robot based on the likelihood, safety distance and velocities of approaching Personnel and Robot
- Calculate safety region/distance required between Robot and Personnel

Figure 5 shows a personnel walking with velocity $\mathbf{V_H(t)}$ towards a Robot which is moving its arm towards the direction of personnel at velocity $\mathbf{V_R(t)}$. The objective here is to stop the robot before the personnel exceeds fixed distance defined by cushioning constant. To attain this objective, CPA service should continuously keep track of safety distance $\mathbf{dc(t_0)}$ as per the equation in Figure 5.

Following are parameters taken into consideration to calculate safety distance:

- Human Velocity $\mathbf{V_H(t)}$: Velocity of the approaching personnel
- Robot velocity $\mathbf{V_R(t)}$: Velocity of Robot approaching personnel before stop signal is received
- Robot reaction time $\mathbf{T_R}$: Time elapsed between Stop signal generated by the CPA service to stop signal actually acting on the robot arm.
- Robot stopping velocity $\mathbf{Vs(t)}$: Velocity of robot after receiving stop signal.
- Robot stop time $\mathbf{T_{stop}}$: Time taken for robot to come to a halt after receiving stop signal.
- Cushioning constant $\mathbf{C_{dc}}$: is a constant minimum distance between personnel and human set as per safety regulation.

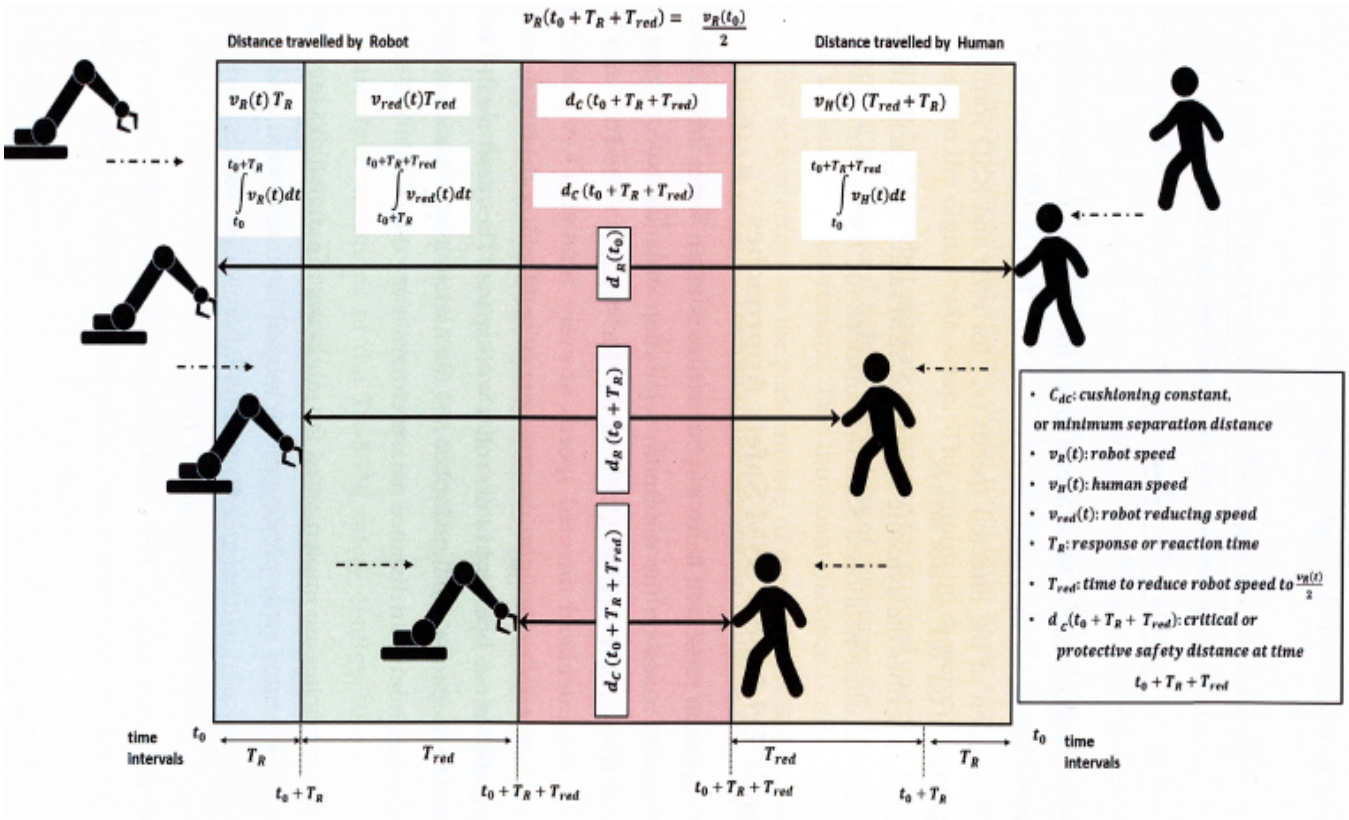


Figure 5 Safety distance description

3.2 Scenarios' needs from RAINBOW

Based on current architecture mentioned in the section 3.1 there are certain requirement/criteria that become crucial when the architecture needs to scale and cater to larger needs. The following are the high-level requirements that need to be satisfied by RAINBOW.

- **Scalability of cloud-native services and effective utilization of resource on Fog device**: In a typical industrial environment, the number of personnel and robots in a work-place area keeps changing. As PLMC, RMT, CPA service only supports fixed size group of personnel and robots, there is a need to dynamically scale instances of this service on-need basis, typically based on movement of personnel or/and change in workplace configuration of robots. Adding to this, as these services are resource intensive, an effective utilization of compute and storage resources is needed on the device (Fog) they are running. Corresponding KPI for this requirement is mentioned in Table 4 ID-1.

- **Deterministic and bounded system latency**: To predict and avoid collision between personnel and Robot, the PLMC, RMT, CPA services must be able to perform their respective processing tasks and send appropriate signals to Robots to stop in case of possible collision prediction in a deterministic timeframe with a hard upper bound. The term of this this upper bound for timeframe is 'System Latency'. System Latency can be divided into 4 parts
  1. Data Acquisition Latency
     1.1. Latency in acquiring Robot motion data
     1.2. Latency in acquiring Personnel localization and motion data.

     It's worth noting that Personnel localization and motion data (1.1) and Robot motion data (1.2) are processed in parallel.
  2. Latency in CAP (Collision Avoidance Processing unit) which includes latencies of CPA, RMT, PLMC services for predicting collision.
  3. Robot reaction time ($\mathbf{T_R}$)
  4. Robot stop time ($\mathbf{T_{stop}}$) which is fixed for a given robot from a manufacturer

  In an event of predicted collision, this system latency plays a key role in determining the "safe distance" (a minimum distance between personnel and robot to avoid collision). In the scenario when system latency is violated, it can cause unscheduled interruption in production (by stopping of robots) or in worst case can cause fatal collision between personnel and robots. Thus, there is a need to monitor the Service Level Objectives (SOL) continuously. Taking corrective or preventive measures based on policies to ensure SOLs are met with utmost importance. Further, corresponding KPI for this requirement is mentioned in Table 4(ID-1).

- **Dynamic resource provisioning at runtime:** In the scenario when a fog device in an infrastructure fails to serve many instances of PLM, RMT, CPA services due to lack of

resources (typically refers to computational, storage, network resources), there is a need to dynamically provision these resources from near-by Fog/Cloud devices (if and only if prescribed SOLs can be meet). Lack of dynamic provisioning of resources may cause unscheduled interruption in production (by stopping of robots) or in worst case can cause fatal collision between personnel and robots. Further, corresponding KPI for this requirement is mentioned in Table 4(ID-4).

- **Reliable dynamic service provisioning between Fog devices**: In a typical industrial environment, personnel move from one work-place area to other. The personnel's position is received by Aggregator device from mobile Node device mounted on the personnel through wireless communication. A major pitfall for wireless communication physical layer is that it has limited range of coverage. When a person moves from one workplace coverage area to other, the Aggregator device responsible for receiving these telemetry messages changes and exactly one Aggregator is associated to one Fog device running Collision Prediction and Avoidance services. Since the position of the personnel changes within different workplaces, the processing Fog device also changes. Thus, there is a need to reliably provision a new service instance running in Fog device within the new workplace area and then transfer the data stored in the local database of old workplace to new workplace Fog device. Once data is transferred successfully the service instance running in the Fog device of old workplace is terminated. In addition to the reliable provisioning of services and transfer of data, there is a hard upper bound on time within which this provisioning needs to complete to avoid unscheduled interruption or in worst case collision between personnel and robots. Further, corresponding KPI for this requirement is mentioned in Table 4(ID-4).

3.3 To-be reference scenario

At high-level, HRC system is a collision prediction and avoidance system between Personnel and Robots in an indoor environment. The scenario described in section 3.2 need to be addressed by RAINBOW. Following information is required continuously in a time-deterministic manner:

1. Personnel's current 3D Coordinates and motion dynamics
2. Robot's current 3D Coordinates and motion dynamics

Using above information, predictions on collision are made *a-priori*. Based on probability of collision, the collision prediction and avoidance system send control messages to slow or stop the Robot thus avoiding the collision between Personnel and Robot.

An exemplary demonstrator setup for RAINBOW is shown in Figure 6 and consists of 2 workplace areas "Area-1" and "Area-2"**.** Each of the work-place area consists of a **Robotic arm** controlled by **Industrial PC** and PLC using **PROFINET** field bus**. IoT Gateway** collects telemetry data of Robot (joint angles, velocity etc.) from Industrial PC using PROFINET and forwards this data to **RMT service** running on Fog devices using **OPC-UA**.

Additionally, if a collision is predicted by **CPA services** running in Fog device, then **control signal** is sent to the **IPC** via **IoT Gateway** for stopping or slowing the Robot.

On other hand Personnel localization and motion dynamics data from mobile **Node** device units are received by **Data Aggregator** wirelessly over UWB. The received data is published to **MQTT broker** hosted within Data Aggregator. **PLA** services running on Fog devices subscribes for this data using **MQTT Client**.

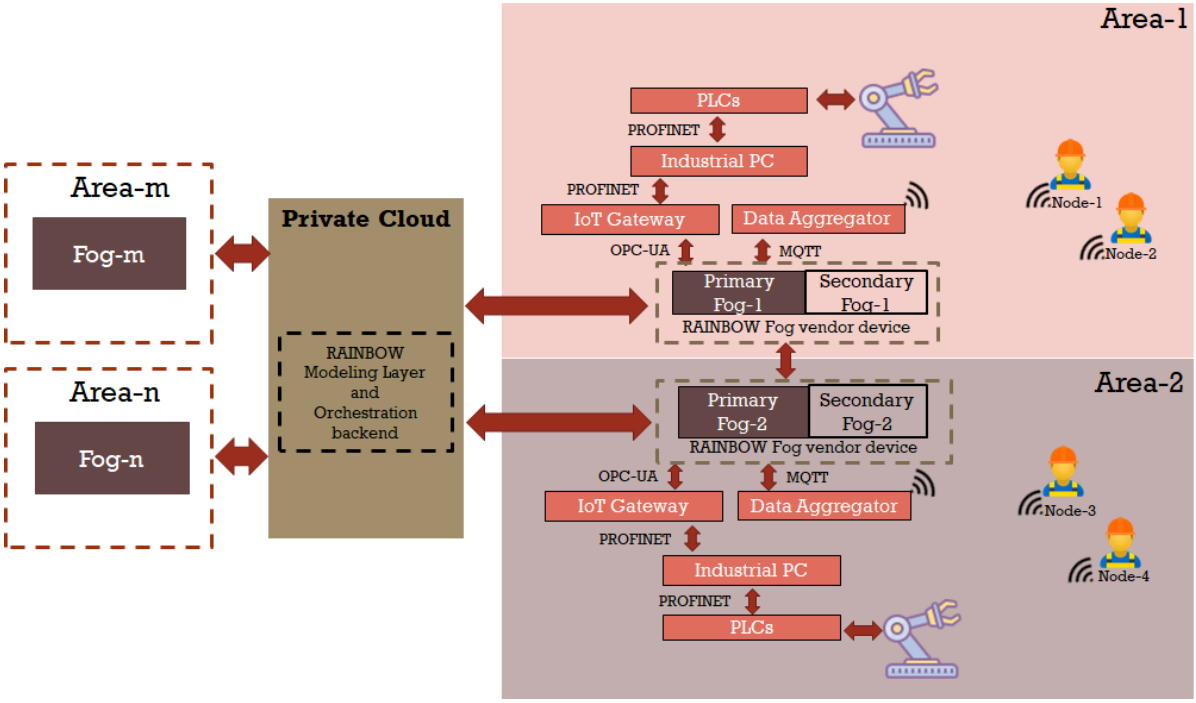


Figure 6: Demonstrator overview with RAINBOW

The Fog device is a high-performance 64-bit multi-core processor hardware with Linux OS capable of running multiple instances of each of the below services as shown in figure 7

1. Personnel Localization and Motion Capturing service (PLMC): One instance per Personnel in work-place area
2. Robot Motion Tracking service (RMT): One instance per Robot in work-place area
3. Collision Prediction and Avoidance service (CPA): One instance for group of Personnel and Robot in work-place area

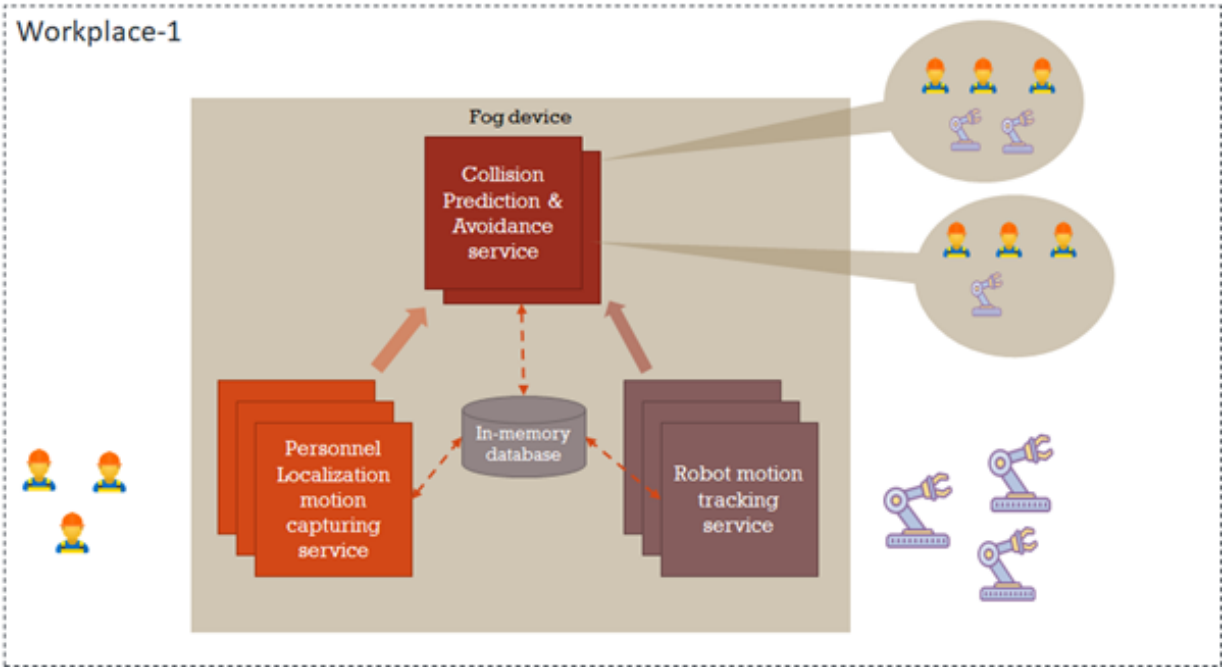


Figure 7: Services in Fog

Above service run probabilistic algorithms for estimation and prediction. The reason these services are to run in Fog device and not as cloud service is due to the requirement of short, deterministic latency between data acquisition to stopping of robot. These services run with a strict real-time constraint in processing the data and are computationally intensive.

In addition, with latency requirement, scalability, mobility, reliability, resource-sharing, secure-deployment of applications, application monitoring, distributed data management and analytics, security and data privacy are also important requirements in these applications.

Below is a high-level description that relates use case requirements with RAINBOW platform components in the proposed reference scenario.

1. **Scalability of cloud-native services and effective utilization of resource on Fog device**: To achieve this in RAINBOW framework the *Service Graph Editor* describes dependencies, Service Topology etc. of the cloud native applications. and *Policy Editor* describes both run-time and pre-deployment constraints. This information is used and evaluated by components such as *Pre-deployment constraint solver, Deployment manager, Orchestration lifecycle manager, resource Manager in Centralized Orchestration Backend and Rainbow Mesh stack* to deploy (scale-up/scale-down) instances of application in Fog device on-need basis.
2. **Reducing system latency and jitter:**
   Reduction of System latency reduces the safety distance required and hence greater collaboration with lesser unintended halts in production due to stopping

of robots. Additionally, reducing network jitter can make system more deterministic and can improve predictions results of the algorithm.
In RAINBOW, components such as *Orchestration lifecycle manager in Centralized Orchestration Backend and Rainbow Mesh stack* achieve this requirement by continuously monitoring and evaluating for network bandwidth, jitter metric and network latency against Service Level Objectives (SLOs) and provisioning resources such that system latency and jitter is reduced.

3. **Run-time application monitoring, constraint evaluation and dynamic resource provisioning:** Here performance, and health indicator metrics from deployed service can be collected continuously using *Resource Application-level Monitoring* in *Centralized Orchestration Backend* and *Multi-domain sidecar proxy in Rainbow Mesh stack*. These metrics can be evaluated with Service Level Objectives (SLOs) by *Orchestration lifecycle manager*. If SLOs are not met, suitable actions as described by service operator using *Policy Editor* are taken to resolve issues. Also, during a scenario, when a Fog device fails to handle loads (such as computation, memory etc.) RAINBOW framework will allow to dynamically provision resources from other Fog/cloud devices if SLOs are met.
4. **Reliable service and data migration between Fog devices**: In the scenario of personnel mobility, service and data migration from one Fog device to other is expected. To achieve this requirement, RAINBOW provides components such as *Orchestration lifecycle manager, resource Manager, Analytical Engine, Resource Application-level Monitoring* in *Centralized Orchestration Backend* and *Multi-domain sidecar proxy in Rainbow Mesh stack*.
5. **Data management and High-performance queries across distributed databases for data Analytics:** Each of the Fog device serves a database instance to preserve applications data and states. For continuous Analytics it needed to fetch data from databases hosted in different Fog devices distributed in the infrastructure. Also, since Fog devices have limited memory there is a need to sync database in Fog devices with central database, thus, the need of analytic engine that queries data optimally from distributed databases and syncs data with a central database periodically. To achieve this RAINBOW provides *Analytical editor* in *Modeling layer* which help to create such queries. These queries are further processed by *Analytical Engine* in *Centralized Orchestration Backend* to fetch data from distributed databases.

3.4 Scenario user stories

Scenarios to be demonstrated written in User story (One table for User Story):

HumRob.US.1	*As a Service Developer I wish to develop and deploy custom applications on the Fog Nodes via RAINBOW*
User Story Confirmation	A Service Developer can define application via comprehensible and concise templating schemes. Following are few high-level metrics that needs to be described in application templates • Deployment constraints:

	o Robot type, make supported o Max-Robot-Stop time o Processor architecture of Fog device • Operation constraints: o Max-jitter o Max-delay o Minimum-bandwidth required o Max system reaction time • Resource constraints: o Storage o CPU usage • Security constraints: o Data Integrity o Data sharing with respect other service entities • Dependencies for the application and service topology
RAINBOW Functionalities	Service Graph Editor in Modelling Layer
User Story Implementation and Workflow	
Service developer develops following services as cloud-native components and wishes to deploy along with deployment constraints. • Personnel Localization and Motion Capturing micro-service • Robot Motion Tracking micro-service • Collision Prediction and Avoidance micro-service The service developer will use Service Graph editor to describe deployment constraints, service topology and dependencies description.	

HumRob.US.2	*As a Service Provider I want to be able to describe runtime policies and control the applications running on dedicated infrastructure and obtain necessary attributes from it*
User Story Confirmation	Service provider must be able to get run-time stats, and must be able to create/modify/remove run-time constraints related to service
RAINBOW Functionalities	Policy Editor in Modelling Layer
User Story Implementation and Workflow	
Service Provider using Policy Editor provides run-time constraints such as • Max-jitter • Max-delay • Minimum bandwidth required • Max system reaction time • Storage • Dynamic sharing of resources between Fog devices in case of one of the devices lack resources. Here amount of load to move and Max system latency for moved load are described • Switch to secondary Fog device if primary Fog device fails in a work-place area. • Service up/down scaling Adding to this, Service provider views operational stats using Dashboard interface. The Service Provider should be able to create new dashboard and able to add and view application-level monitoring metrics in a graphically intuitive and interactive manner.	

HumRob.US.3	*As a Data Analyst I want to create custom monitoring metrics and obtain vital information of the Infrastructure as well as the application metrics for further analyses*
User Story Confirmation	Data Analyst must be able to describe and run queries on databases within distributed infrastructures
RAINBOW Functionalities	Analytical editor
User Story Implementation and Workflow	
Data Analyst uses Analytical editor for creating queries which are optimized on distributed database to get data necessary for high level analytics. Following are some high-level analytics performed by Data Analyst • Activity recognition and activity synchronization between Personnel and Robot • Unintended Service/Process down time • Localization, prediction accuracy, • Process optimization analysis, • Human Ergonomics analysis	

3.5 Initial Metrics of Success

Table 4 UC1 KPIs

Id	Qualitative Metrics	Target Value	(M)andatory / (G)ood to Have / (O)ptional
1	*Scalable and Secure deployment of micro-services on need basis. In the scenario of changing number of Personnel, Robots, changing workplace configurations.*	*Supported*	*M*
2	*Continuous monitoring and evaluation of QoS of applications/services running on Fog device. If constraints are not met, then actions specified in policies are to be performed.*	*Supported*	*M*
3	*No single point failure like Cluster head failure, drop in QoS, Exception in micro-service should compromise personnel safety. If all unresolved exceptions (within specified time) must stop/halt the robot.*	*Supported*	*M*
4	*Dynamic sharing of resources between Fog should be allowed*	*Supported*	*M*

Id	Qualitative Metrics	Target Value	(M)andatory / (G)ood to Have / (O)ptional
	considering service level objectives are meet. In the scenario when a Fog temporarily lack resources (may be due to overload)		
5	*Data sharing is restricted within defined boundaries (factory premises, outside factory to third-party etc) with appropriate authentication mechanism and access rights for user in different user-group.*	*Supported*	*M*
6	*All communication between devices must be secured by default*	*Supported*	*M*
7	*On-boarding new fog device must adhere to attestation policies by providing verifiable evidence on their configuration integrity and correctness.*	*Supported*	*M*
8	*Periodically Synchronize data from all distributed databases present in each of the Fog with Central database*	*Supported*	*M*
9	*Support optimized queries requiring to fetch data from Distributed database across Fog device mesh network*	*Supported*	*G*
10	*Support addition of customised high level Analytical queries*	*Supported*	*O*

4 UC 2 Digital Transformation of Urban Mobility

4.1 AS-IS scenario

The Digital Transformation of Urban Mobility Use Case aims to demonstrate how RAINBOW system will contribute on developing a real-time geo-referenced notification system for vehicles traveling in urban areas about Hazardous situation. The RAINBOW platform will also act also in the vehicle communication field, by providing a reliable and decentralized approach to safely handle exchange of messages.

4.1.1 AHED Automatic Hazardous Events Detection

The use case hinges upon a real-time geo-referenced notification system for vehicles traveling in urban areas about Hazardous situations for the city mobility network, due to any possible cause (e.g., accidents, failure of road infrastructure, animals on the road, wrong-way driving….). The notification system will be designed to collect signals issued by entities in urban areas (Vehicles, Road Side Units, and Vulnerable Users). Explicit notifications refer to those that are either triggered directly (i.e., manually) by vulnerable users (citizens), who may want to report a Hazardous situation. Automatic notifications may be triggered by on-board sensors, by Road Side Units or as a result of sensor fusion processes that may involve cars and Road Side Units all sending log data to fog, MEC (Multi-access Edge Computing) and cloud, where AI/ML algorithms can infer alert conditions that should be reported. Each alert signal will be delivered with the available geo-localization information, allowing reports to be localized in the areas where the Hazardous situation was detected.

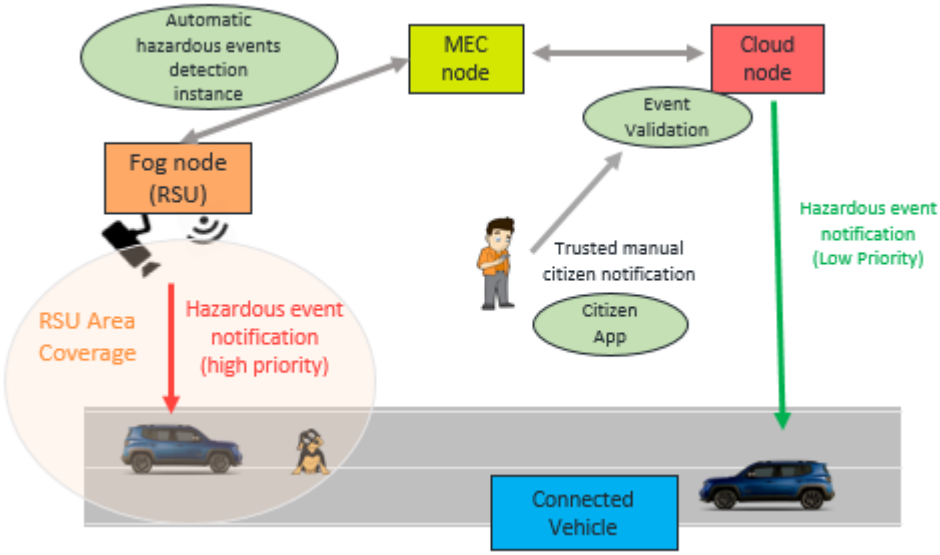


Figure 8: Automatic Hazardous Events Detection

The notification system is designed to collect signals issued by entities in urban areas (Vehicles, Road Side Units, and Vulnerable Users). There are two types of notifications:

- **Explicit notifications** that refer to those that are either triggered directly (i.e., manually) by vulnerable users (citizens), who may want to report a Hazardous situation.
- **Automatic notifications** are triggered by on-board sensors, by Road Side Units (RSU) or as the result of sensor fusion processes that may involve cars and Road Side Units all sending log data to fog and cloud, where AI/ML algorithms can infer alert conditions that should be reported. In general, the term of this software will be Automatic Hazardous Events Detection (**AHED**).

Each alert signal is delivered with the available geo-localization information, allowing reports to be localized in the areas where the Hazardous situation was detected.
In the use case definition below, the following terms are used:

- **MEC node**: based on the ETSI Multi-access Edge Computing (MEC) paradigm, is a point of computation and data storage, mostly one or two hops away from the mobile client to meet the response time constraints. It is meant to be an efficient computing device in terms of performances and power consumption.
- **Edge Device**: it is identified with all end-user devices (Vehicle equipped with communication capabilities, Smartphone…)
- **Fog Node**: it is identified with the RSU which is a small point of computation and data storage, connected directly with the road infrastructure. It is a small, lightweight and IP67 device with different power consumption profiles based on its CPU performance.
- **Cloud Node**: identified as a generic server in the cloud offering computation and storage services.

4.1.2 Initial configuration

As starting point configuration, the demonstrator consists of a road-equipped segment with an on-site installation of a Road Side Unit (RSU), as reported in Figure 8. The infrastructure is equipped with multiple IP cameras directly connected with the RSU. The video stream is also reachable by the MEC node, which communicates with the RSU through the Internet. The AHED service consists of a series of computer vision techniques and AI algorithms able to identify the presence of animals on the road, which can represent a dangerous situation for drivers. Based on this, the AHED service can run either on the RSU, or on the MEC node, where each configuration implies advantages and drawbacks with respect to the latency and accuracy constraints. Indeed, the RSU benefits of the direct connection with IP cameras, which yields a high-resolution video stream at high frame rate. Of course, the higher the resolution, the more accurate the results. At the same time, the AHED system requires high computational power to run and the RSU, to achieve this, must increase its power consumption, which may cause the overheating of the system and malfunctions in the road infrastructure. On the other hand, the MEC node does not suffer from overheating problems, since it is a powerful computer typically hosted in a place equipped with an efficient cooling system who will not have problem

running complex algorithms. Unfortunately, due to the channel latency and the available bandwidth, the node works typically with a low-quality video stream with smaller frame rate, thus resulting in less accurate results. Besides, the quality may change due to variable network conditions. Moreover, once the hazardous event has been detected, the MEC node must communicate it to the RSU, which is the one equipped with the required communication hardware able to communicate it to the incoming connected vehicles.

Node running the AHED system	Advantages	Drawbacks
Fog Node (RSU)	• Low latency • High detection accuracy	• Low computational power • High power consumption
MEC	• High computational power • Low power consumption	• High latency • Low detection accuracy

Finally, as explicit notification, a road user may spot a hazardous situation on its own. Thus, he may decide to signal it through the dedicated citizen app, collecting all manually signaled hazardous situations in the cloud node. There, an automatic or manual event validation will decide if the hazardous notification must be sent to the vehicles. If so, the notification will follow the same path as the one sent by the automatic hazardous notification system.

In the following the detail of subsystems is described.

Connected Vehicle: Vehicle equipped with a C-V2X communication device and a V2X client.

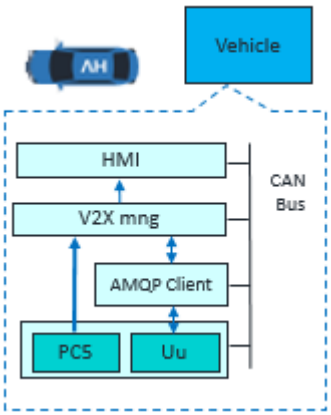


Figure 9 Connected Vehicle

The Vehicle performs data acquisition from the on-board CAN Bus and dispatches it to a cloud node using ETSI CAM messages through the Uu (and PC5) interface and the AMQP protocol. The Vehicle also receives ETSI DENM messages from Fog Nodes via PC5 or Uu interface. The messages of interest for the vehicle are the Hazardous Location Notification (DENM) messages with different cause code depending on the type of hazard.

When the vehicle receives DENM messages from the Fog Node, it parses and displays them on the HMI depending on their relevance. The main components on the Vehicle are:

- PC5 and Uu protocol stacks: each manages the UU and PC5 communication layer
- AMQP Client: it establishes a logical communication with the AMQP broker on the cloud node and it extracts and encapsulates V2X messages (CAM, DENM)
- V2X Management: it handles the transmission and reception of CAM and DENM messages.
- HMI: it displays the messages to the end user according to relevance policies.

From the privacy and security point of view, the use case will follow the PKI architectures defined by ETSI ITS and IEEE to secure all V2X communications. For privacy protection purposes, certificates have a reasonably limited validity period in order to limit the reusability in case of revocation. Furthermore, the same certificate is supposed to be used a limited number of times, for privacy reasons and in order to avoid tracking. Of course, this is a limitation for the overall efficiency of the system. Given the frequent transmission of CAM messages from vehicles, including location information, this can create a scalability problem with a centralized PKI-based approach [44].

Fog Node: RSU node with PC5 interface, equipped with one or more IP cameras and a GPU AI computer.

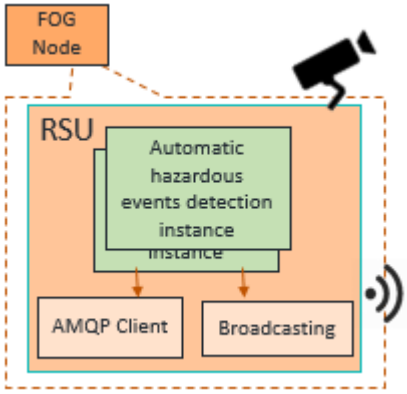


Figure 10: Fog Node

The Fog platform includes a platform equipped with IP cameras, able to perform AI computer vision tasks and to communicate with vehicles both through PC5 communication and AMQP messages (via cloud). On the computer vision side, the platform manages the video stream data gathering and forwards them to the MEC node. Moreover, it can run AI detection algorithms based on Rainbow orchestrator decisions. On the communication side, the Fog platform is in charge of broadcasting hazardous event notification through ETSI DENM messages (via PC5); it also acts as an AMQP Client, publishing messages on the hazardous notification to the cloud so they can be accessible everywhere (and not only for vehicles in the RSU coverage area). The MEC platform will receive data coming from edge vehicles in the form of ETSI CAM messages. These messages can be used to be processed together with the data coming from the cameras. The Fog Platform will also be responsible for the creation of a trusted overlay of fog and MEC nodes.

MEC Node: A node that simulates an Edge Server typically installed inside a mobile or road operator network.

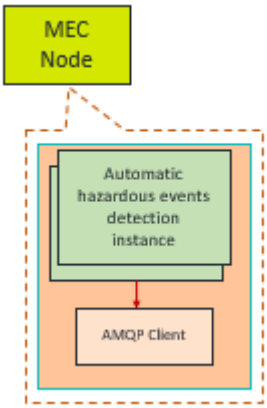


Figure 11: MEC Node

The MEC platform is typically hosted in a mobile or road operator network. The MEC node can run the AI detection algorithms thanks to the video streams received by the RSU. The MEC node can notify hazardous situations to the RSU (that will broadcast them directly to vehicles via PC5) and to the cloud node (where they can be retrieved via Uu by vehicles interested in notifications from a certain area).

Citizen App: App that allows Citizen to alert the municipality about Hazardous situations on the road

Figure 12: Citizen App

The Citizen App publishes Hazardous situations on the road on the City Cloud Node.

City Cloud Node: Aggregates contextual traffic data coming from vehicles, local traffic control centre or citizens.

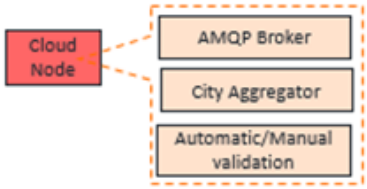


Figure 13: Cloud Node

The Cloud collects implicit alert signals issued by vehicles in urban areas on a proprietary data collection platform that will expose sanitized information to an Aggregation Platform, which will be designed to aggregate contextual traffic data coming from other

sources, such as local traffic control centers or citizens. The Aggregation Platform will expose APIs through a southbound interface toward the Fog Platform.

4.2 Scenarios' needs from RAINBOW

The Digital Transformation of Urban Mobility Use Case aims to demonstrate how the RAINBOW system can contribute to the development of the real-time geo-referenced notification system for vehicles about Hazardous situations optimizing the resources consumption and providing a reliable, decentralized approach to safely handle the exchange of messages.

4.2.1 Smart Orchestration

The challenge for the AHED system is to obtain, by means of the RAINBOW platform, the best usage configuration between MEC and Fog Node, in order to maximize the accuracy while reducing the power consumption and keeping the usage bandwidth always below a defined threshold.
In a non-hazardous situation, the RSU will monitor the road through its cameras, while sending low-resolution and low-frame-rate video streams to the MEC node, where a continuously running AHED instance will perform event detection on the received video stream. The RSU is supposed to be connected with a wireless or wired connection to the MEC node and to be powered by the electricity grid. In the standard situation the network bandwidth for connecting the RSU is enough to run the video streams and the RSU itself runs in low-power mode, leaving the most power-consuming activities on the MEC node where the energy management can be optimized.
A different orchestration of the AHED can be triggered by two events:

1) Whenever a hazardous situation is detected, the RAINBOW orchestrator will move the detection instances from the MEC node to the RSU Fog node. In this way, the algorithm will work with high resolution image frames at higher frame rate in order to accurately produce hazardous notification with no latency addition due to the network. The RSU will then broadcast the message to all vehicles in the coverage area through dedicated short-range communication (PC5) and it will also publish it on the AMQP broker placed on the Cloud node. In this way, also far away vehicles, which are not in the coverage area, will receive the message (via Uu), but with a delay based on the network speed. The RSU will run at maximum power to host the AHED instances. After the hazardous situation is ended, RAINBOW will move back the inference instances on the MEC node.

2) In case of congested network, the MEC node may receive the video streams with an exceedingly high level of latency, such that the hazardous detection event may not work properly. Since Rainbow would be able to move the inference instances back and forth, it may decide to shift the detection back to the RSU, in order to reduce the latency and wait for a better network condition to restore the detection in the MEC node.

4.2.2 Trust Enablers

Georeferenced V2X services rely on vehicles location information that can be used to generate location profiles revealing movement of their drivers. Two typical V2X messages exchanged are the ETSI-defined Cooperative Awareness Messages (CAM) and Decentralised Environmental Notification Messages (DENM). It is expected that the V2X messages are exchanged between vehicles and infrastructure, either through the 5G-Uu interface, thereby leveraging the existing cellular infrastructure, or through the PC5 interface reaching Road Side Units (RSUs). The challenge for the AHED system is to migrate, by means of the RAINBOW platform and the provided trust overlay mesh network, to a secure and authenticated (while preserving the privacy of the end devices when needed) data collection and distribution of V2X messages in order to obtain secure end-to-end architecture that enables a scalable bidirectional communication. In this context, the actual identity of the sender in many cases is not required for ensuring the trustworthiness of a transmitted message. It rather suffices to verify the origin correctness; a message has been sent by a valid V2X participant.

To obtain this goal two main RAINBOW functionalities are needed:

- The RAINBOW secure enrollment functionality, based on the designed Zero-Touch Configuration Integrity Verification enablers , for enabling the secure registration of edge devices (vehicles) and of fog devices (RSUs) based on their trustworthiness and correct state. This will also then allow for the secure key management of all communicating parties.
- The RAINBOW authenticated and privacy preserving exchange of messages through the integration of the advanced Direct Anonymous Attestation (DAA) protocol as an integral part of the offered RAINBOW routing mechanism.

The primary benefits of using the secure enrolment and symmetric crypto primitives (group-based pseudonyms as short-term anonymous credentials), over the Public Key Infrastructures (PKIs) adopted in the current V2X architectures, are in terms of security, privacy and scalability. [1]
Such decentralized approach gives also the advantage in terms of scalability due to the trust being shifted from the back-end infrastructure to vehicles that can create their own pseudonyms, protected under the DAA protocol, and used to anonymously sign all exchanged messages by leveraging Elliptic Curve Crypto (ECC) in such a way such that all recipients can verify their authenticity . This means that it is not possible for third parties to discover the vehicle identity, assuring that pseudonym resolution is not possible. This results in a simplified V2X message exchange with the infrastructure.
Another important aspect of scalability is also the performance of the computationally intensive asymmetric cryptography mechanisms currently employed. RAINBOW's security and trust models are based on the use of much more efficient symmetric crypto primitives.

4.3 To-be reference scenario

The Digital Transformation of urban mobility use case takes advantages from RAINBOW platform in offering an Automatic Hazardous Event Detection (AHED) service that can be run in different platforms towards optimizing the work performance. The main demonstrable functionalities integrated with the Rainbow platform are:

- Migration of the AHED algorithm between Road Side Unit and MEC according to some metrics (e.g., network conditions, RSU load, application-level metrics…) using Rainbow Orchestration
- V2C (Vehicle to Cloud) secure V2X connection using the Rainbow Trust Enabler

Figure 14 shows how the RAINBOW platform can be integrated in the use case architecture.

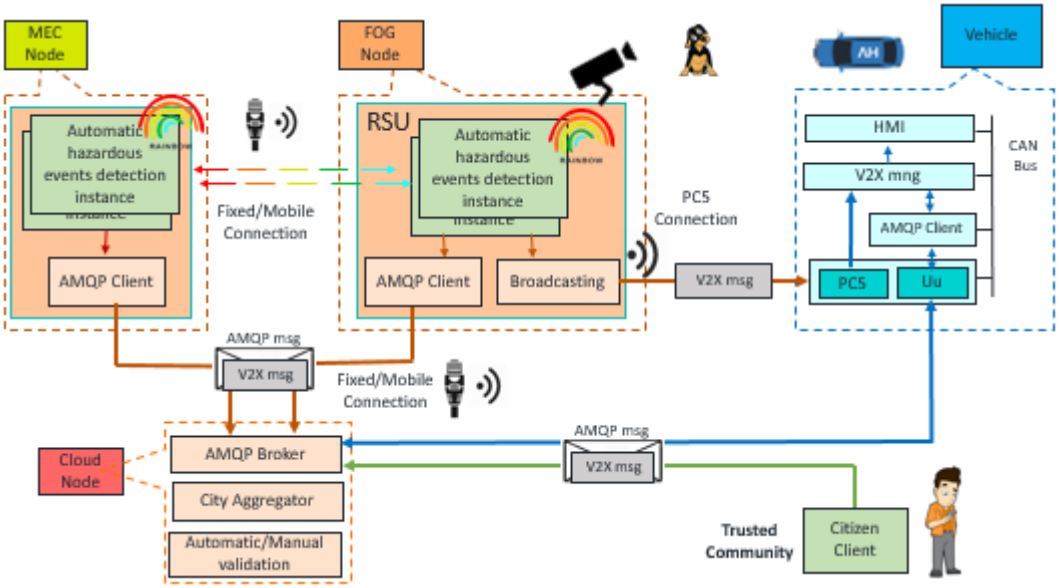


Figure 14: Solution Overview

4.3.1 Smart Orchestration

Run-time application monitoring and resource-based load shifting is achieved thanks to RAINBOW components such as the service graph editor and the policy editor.
By continuously polling metrics such as network bandwidth, network speed and software latency, the RAINBOW platform moves the AHED instance back and forth between the MEC and the fog node.

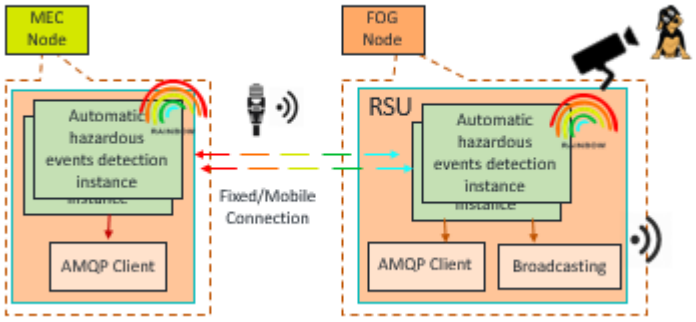


Figure 15: AHED orchestration

Monitoring data are constantly evaluated to check if SLOs defined by the service developer are met, and the AHED service will be orchestrated to satisfy all constraints. Constraints that will be monitored by the rainbow sidecar proxy:

1. Bandwidth usage
2. RSU temperature (to detect overheating)
3. Power consumption
4. Latency
5. Hazardous detection event

Both the MEC and the Fog Node will hold the RAINBOW platform. RAINBOW will continuously monitor these parameters and it will find the best place to run the AHED service so as to satisfy them.

4.3.2 Trust Enablers

As outlined in Deliverable, D2.1, some requirements must be met for the establishment and maintenance of strong guarantees of trust in a fog-based environment such as the one envisioned in the context of the AHED service. For instance, a common requirement is that each device in the network must be equipped with hardware or software support for remote attestation and/or configuration integrity verification. RAINBOW security and trust enablers provide the remote attestation mechanisms by the creation of a privacy and trust-aware service graph chains; the main goal is to enable remote authentication of trusted entities while preserving the privacy of the vehicles. The use of the security and trust enablers in the Digital Transformation of Urban Mobility Demonstrator focuses on the V2X connection between a generic edge device (On Board Unit/Citizen Device) and the Cloud. The two main enablers of functionalities involved in the use case are:

- The secure enrollment functionality, based on the attestation enablers, for when an edge node registers to the back-end service.
- The privacy-preserving exchange of messages through the integration of the DAA protocol.

Based on the D 6.3 demonstrator detailed design and on the availability of the edge devices an evaluation will be conducted, if possible, on which MEC/Fog node is more appropriate to support the security and trust enablers' demonstration.

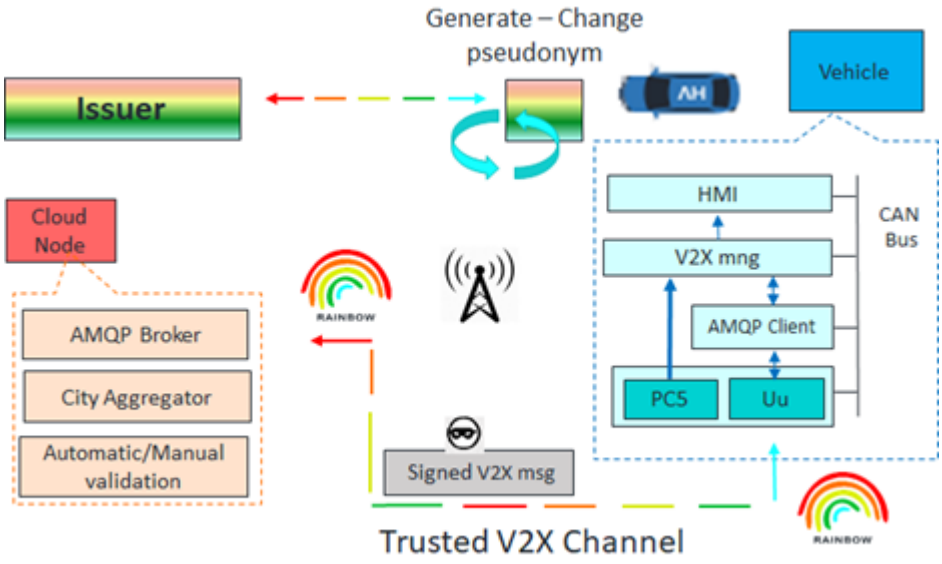
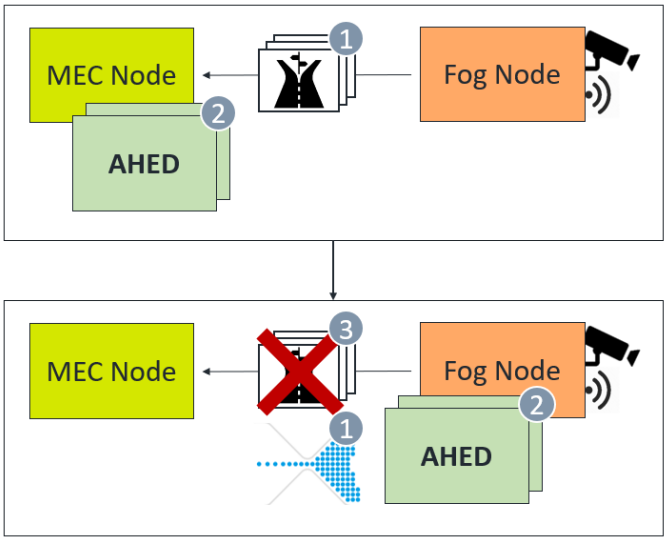


Figure 16: Trust Enablers

4.4 Scenario user stories

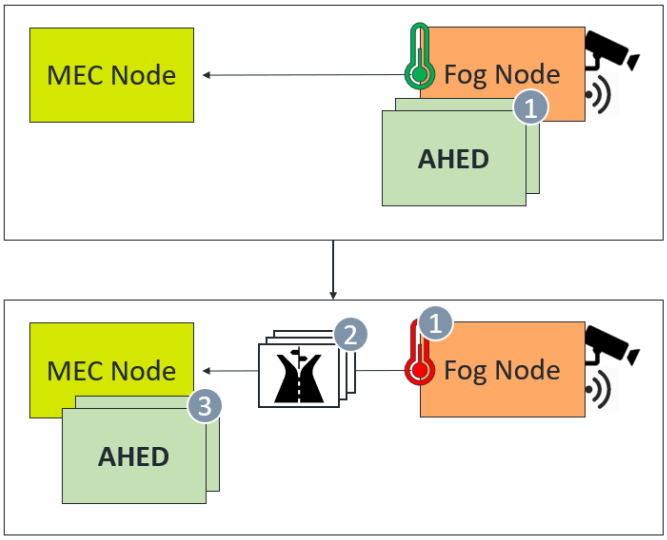
UrbanMob.US.1	As a Network Operator, I want to monitor the network load
User Story Confirmation	Control the network flow outgoing from the RSU and keep the network load below a given threshold
RAINBOW Functionalities	1. Modelling • Constraint and policy editor • Containerized application packaging 2. Orchestration • Application lifecycle management • Underlying resource and application runtime adaptation
User Story Implementation and Workflow	
Implementation	



Workflow

1. Configure the video stream to flow from the RSU toward the Edge Node
    - Low resolution
    - Low frame per second
2. Start the AHED service
    - Low accuracy
    - High latency
3. Start monitoring the bandwidth usage
    - If above the SLOs defined threshold:
        i. Move the AHED service on the RSU
        ii. Stop video stream toward the Edge node

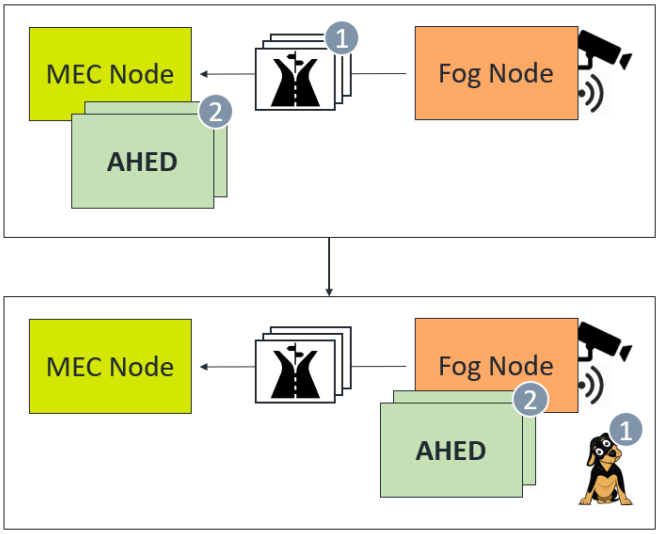
UrbanMob.US.2	As a network/road operator, I want to ensure the correct hardware working regime for reliable results
User Story Confirmation	Control the RSU temperature and ensure it is kept under the safety threshold
RAINBOW Functionalities	2. Modelling • Constraint and policy editor • Containerized application packaging 3. Orchestration • Application lifecycle management • Underlying resource and application runtime adaptation
User Story Implementation and Workflow	
Implementation	



Workflow

1. Start the AHED service on the RSU
    - High resolution
    - High frame per second
    - High accuracy
    - Low latency
2. Start monitoring the RSU temperature. If above the SLOs defined threshold
3. Configure the video stream to flow from the RSU toward the Edge Node
    - Move the AHED service on the Edge node

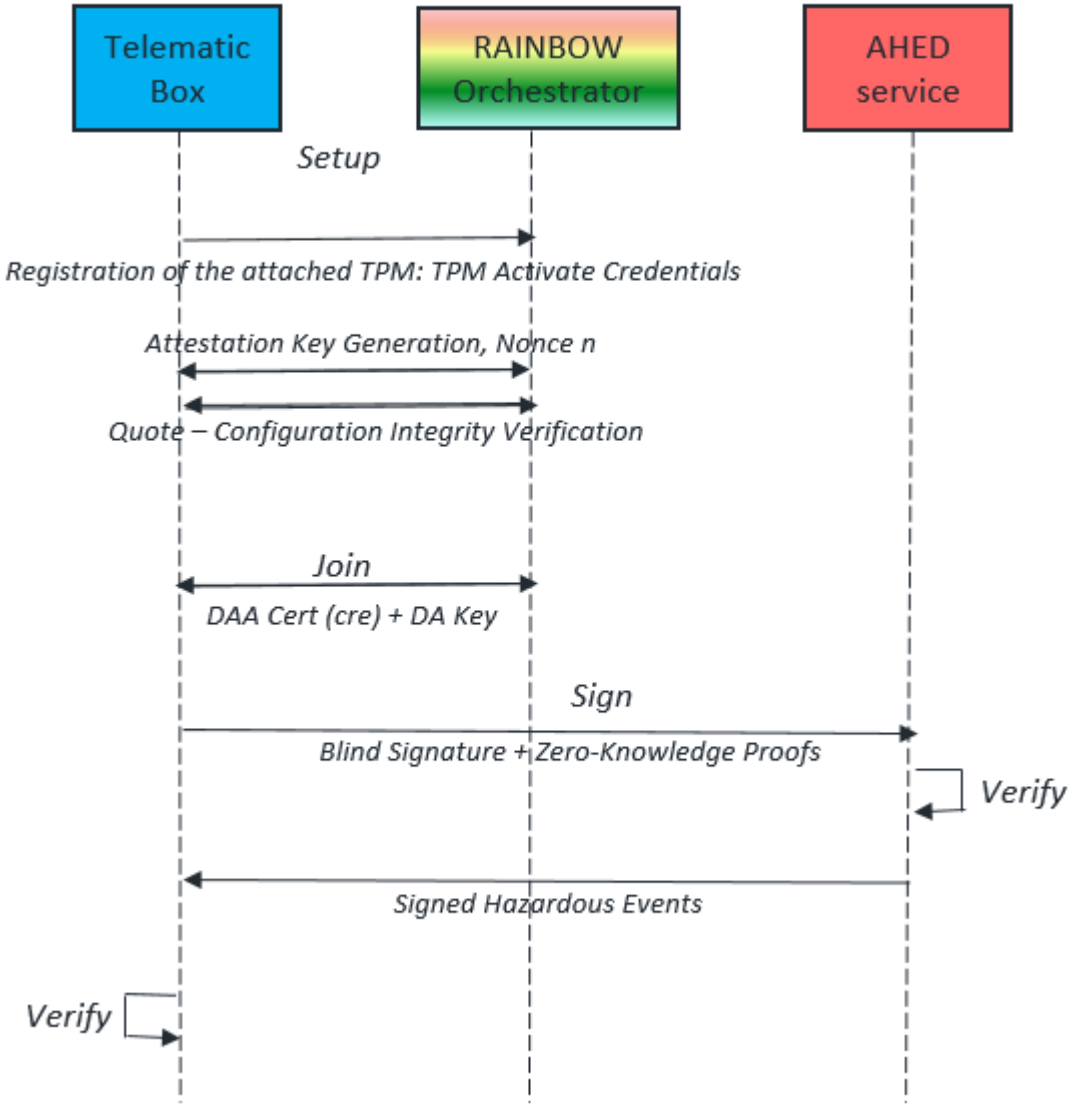
UrbanMob.US.3	As Software developer, I want to optimize the available resources and the accuracy of dangerous situation detection
User Story Confirmation	The AHED service will continuously run on the Edge Node with low resolution video stream. Whenever a hazardous event is detected, the AHED service will move on the Fog Node, to use high resolution video stream at high frame rate. At the end of the alert, the service will be brought back to the Edge Node.
RAINBOW Functionalities	3. Modelling • Constraint and policy editor • Containerized application packaging 4. Orchestration • Application lifecycle management • Underlying resource and application runtime adaptation
User Story Implementation and Workflow	
Implementation	



Workflow

1. Configure the video stream to flow from the RSU toward the Edge Node
    - High latency
    - Low accuracy
2. Start the AHED service on the Edge node. If a hazardous event is detected:
    - Move the AHED service on the RSU
        i. High accuracy
        ii. Low latency
3. If the hazardous event expires:
    - Move the AHED service on the Edge Node.

UrbanMob.US.4	As a Driver I want to access a (hazardous) info mobility service in a secure way with enhanced authentication and privacy protection
User Story Confirmation	The user's vehicle telematic box is authenticated and its correct state is verified (so as to make sure that no vulnerabilities or exploits are present) before being allowed to access the offered service or network. After being securely enrolled, the vehicle, initiates the process for establishing the necessary secrets (keys) required for the subsequent (anonymous) exchange of messages, with other neighbouring vehicles, by leveraging the underlying CJDNS dynamic mesh routing mechanism.
RAINBOW Functionalities	Zero-Touch Configuration Integrity Verification, Direct Anonymous Attestation, Anonymous Diffie-Hellman Key Exchange, RANBOW CJDNS Trust Overlay Mesh Networking
User Story Implementation and Workflow	
Implementation	



Workflow

For this scenario, the telematic box of the vehicle requesting access to AHED service needs to acquire the necessary credentials (Attestation Key, DAA Key, and CJDNS Master Key), through its attached TPM, by executing the enhanced DAA JOIN() Phase. These will then be used for the consecutive execution of the following actions: (i) **verify its correct state** to the RAINBOW Orchestrator (Attestation Key), (ii) **certify the TPM's internal DAA Key** to be used for the creation of verified (self-signed) short-term, anonymous credentials; to be used for the subsequence privacy-preserving exchange of messages, and (iii) **secure and privacy-preserving exchange of messages** to the target fog cluster through the CJDNS trust overlay mesh network

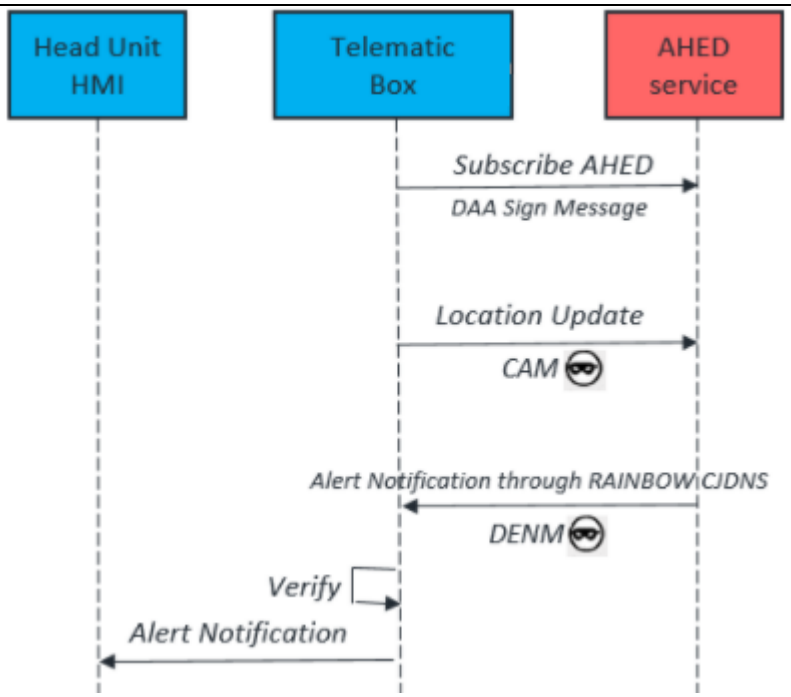
A high-level overview of these actions can be summarized as follows:

1. **SETUP:** Initial step of the workflow for the **registration** and **secure enrolment** of the vehicle's telematic box (and its attached TPM) to the RAINBOW Orchestrator. This registration process relies on the **zero-touch configuration**

integrity verification protocol, as depicted in the above figure. In this context, the RAINBOW Orchestrator acts as the verifier when executing either the Attestation-by-Quote or Attestation-by-Proof mechanism for attesting the correct state of the vehicle to be enrolled. Thus, the orchestrator sends a nonce *n* (used for freshness of the interaction) and a selection of platform configuration registers (PCRs) to attest. These arguments are then passed to the attached TPM which then constructs a quote structure comprising the current values of these PCRs (reflecting the current state of the vehicle) that are then signed and sent back to the orchestrator for verification. If this quote and signature are verified correctly, the vehicle is then enrolled to the service and is allowed to proceed to the creation of the DAA key for further participating in the privacy-preserving exchange of messages for getting access to the hazardous info mobility service.

2. **JOIN:** The now Trusted Telematic Box engages in a challenge/response protocol, with the RAINBOW Orchestrator (or the fog cluster head), so that it can create the Attestation Identity Credential (AIC) and ECC-based DAA key that will then be used for anonymously signing all the subsequent exchange of messages. This DAA key is stored and managed in the attached TPM key hierarchy.
3. **SIGN/VERIFY:** All the subsequent communication, between the vehicle and the AHED service provider, is secured through the RAINBOW DAA Sign and Verify commands. This enables **user-controlled anonymity and unlinkability** depending on the privacy conformance levels that were initially configured by the driver. In the case of anonymous communication, the TPM leverages **group-based pseudonyms** (protected under the DAA Key) for producing *blind signatures* and *verifiable credentials* on the message payload.

UrbanMob.US.5	As a Driver I want to be alerted of Hazardous situations on the road sharing my vehicle's location, while preserving privacy
User Story Confirmation	The Vehicle Head Unit alerts the driver about Hazardous situations on the road related to the vehicle's current location
RAINBOW Functionalities	RAINBOW Direct Anonymous Attestation and CJDNS trust overlay mesh network
User Story Implementation and Workflow	
Implementation	



Workflow

1. **Subscribe:** Vehicle (after the execution of US.4) sends a subscription message to the secure AHED service for receiving Hazardous location notification alters. This message is (anonymously) signed using the **DAA SIGN** command. Depending on the privacy level configured by the driver (*unconditional anonymity* or *revocable anonymity*), the attached TPM produces either a completely blind signature or a deterministic signature as part of the message payload.
2. **Publish Location:** Vehicle sends (DAA) signed CAM messages with vehicle's location to inform AHED service of its current location.
3. When a hazardous situation occurs, the AHED service sends a DENM message to the vehicle through the RAINBOW CJDNS trust overlay mesh network that is responsible for the management of dynamic routes. In the case of anonymous communication, the CJDNS routing mechanism is responsible for creating the Layer 2 message encapsulation for correctly reaching the requesting vehicle.

4.5 Initial Metrics of Success

Id	Qualitative Metrics	Target Value	(M)andatory / (G)ood to Have / (O)ptional
1	Reliable dissemination of alerts	Supported	M

2	secure enrollments based on the attestation enablers	Supported	M
3	Scalable and Secure deployment of the service based on the increasing number of edge vehicles in the area	Supported	O
4	Continuous monitoring and evaluation of the service. If constraints are not met, then actions specified in policies are to be performed.	Supported	M
5	All communication between devices must be secured by default	Supported	M
6	Support addition of customized high level Analytical queries	Supported	G
7	High frequency events managed through ETSI DENM messages	Supported	M
8	Time between the dispatching of the alert and its reception at the end users compliant with the ETSI standard	Supported	M

5 UC 3 Power Line Surveillance via Swarm of Drones

5.1 AS-IS scenario

Power line surveillance is carried out by various methods. The most commonly used one is the aerial method that involves helicopters with multiple sensors and staff on-board. Due to its high cost, power grid operators are pursuing possibilities for using drones instead, as a more cost-effective solution.
For several years, the sector of unmanned aerial systems (UAVs) has been proposing alternative methods, based on multirotor copters, equipped with RGB and IR cameras. The current state of technology and the regulations of aviation law has led to the development of methods involving the use of a flying system with the following configuration:

- **drone (multirotor)** with at least an RGB camera (often also IR camera, rarely LiDAR - heavy and ineffective), equipped with autopilot hardware that allows for autonomous operation,
- **ground control station (GCS)** – a computer running mission planning and mission control software,
- **radio link** with an antenna on a short mast located next to the GCS (for the transmission of telemetry data from the drone to the ground segment).

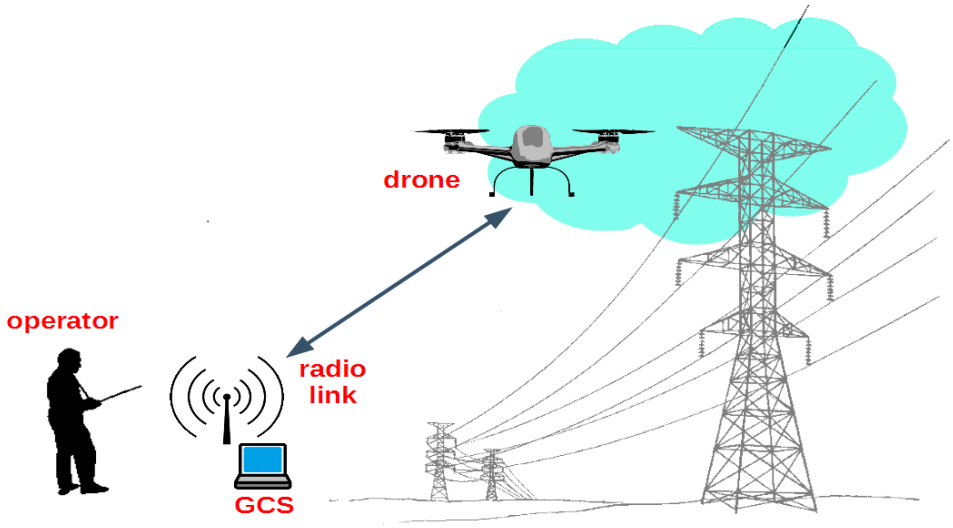


Figure 17: Current drone system configuration

An operator can plan the flight route using GCS software and upload it via radio link to the drone. Once launched, the drone will automatically navigate along this route using GPS positioning and inertial measurement unit. Apart from that, autopilot firmware is also responsible for motor drive control (maneuvering), execution of fail-safe procedures and streaming telemetry data through the radio link to the GCS. Telemetry includes,

among others, position and orientation, battery status, GPS signal quality and radio link signal strength.
The drone is operated by one pilot-operator. In some cases, there is also another person present responsible for the supervision of the camera. Based on the maps, a mission section is planned along the power line. Because of the need to maintain the flying platform in within visual line of sight (VLOS) to directly observe the drone and surrounding airspace to ensure the safety, it is usually 1 km to a maximum of 2 km.
The GCS is usually located in the middle of the planned section of the mission, which allows one to observe the multirotor flying in both directions and thus extend the distance of the mission, since the multirotor is a small aircraft and is unrecognizable against the sky after several hundred meters. During the flight, cameras collect photos and after the mission is completed, they are archived on a storage medium.
The copter is in a constant radio contact with the Ground Control Station which allows the operator to supervise the drone. In the event when the radio link is disrupted or broken, the autopilot switches to the failsafe mode and automatically steers the drone back to the starting point.
After landing, the batteries and memory cards are replaced and the new flight route can be uploaded, but the GCS station must be first moved to a new location where no data has been collected yet. The operator has to move (usually by car) by a distance of 1-2 km and reassemble the GCS. The multirotor takes off for a new mission section and the cycle repeats.
When flying subsequent sections, one system can cover a dozen, up to a maximum of twenty kilometers per day (in the summer season). Long segments of power lines must be handled by several teams and whole data acquisition lasts for several days.
Single photos are analyzed randomly in the field to check if e.g., the camera has maintained the correct focus. The entire photographic set is analyzed in the office. Single, incorrect photos are usually not a problem as the mutual coverage of the photos is kept high. If, on the other hand, there are more incorrect photos, then it is necessary to plan supplementary missions and to go to the field again to perform extra flights.
Grid operators commission inspections of thousands of kilometers of medium- and high-voltage power lines (not to mention low voltage lines) every year. They do not outsource much work to unmanned systems' operators specifically because of their low efficiency. There is no ambition to perform all such work with the use of drones today. But if their effectiveness would raise to a level where it would be possible to inspect at least several dozens of kilometers per day, it would open doors for UAVs to a part of this interesting and valuable market. This the overarching goal for introducing RAINBOW into the system.

5.2 Scenarios' needs from RAINBOW

Key challenges to overcome that would change a group of individual drones into an efficiently operating swarm are:

- **the ability to fly for longer distances than the range of the radio link** – often drones are equipped with links which range is adequate to the category of the drone and the nature of the missions, but usually its flight range is greater than

the range of the radio link, as traditionally the drone flies in the area near the operator and always returns to starting point to land. The specificity of power line inspections requires drones to fly in a straight line, which prevents them from using the full potential of their battery capacity. One solution would be to use a link with a higher power, but its larger dimensions and larger mass would result in a shorter flight time (i.e., flight range) due to the higher energy consumption by the drone's motors. The use of GCSes spread along the power line that can sequentially take over communication with the flying drone will allow it to become independent from the range of the radio link and to fly continuously without the need to return to the starting point. This in turn will raise the effectiveness of the system, since it will increase the percentage of the flight time used for data acquisition. To achieve that, GCSes need to communicate with each other over a networking infrastructure that was deployed *ad hoc* in the field, leading to potentially unstable and unreliable connections. Exactly the ones that the RAINBOW mesh networking stack was designed to deal with.

- **introducing smart route planning for individual drones in the swarm to raise the overall mission efficiency** – usually UAV operators prepare flight routes before going into the field and then they upload new flight plans before each flight. However, in the event of disturbances in the course of the mission (e.g., reduced range due to radio link loss or wind), all subsequent flight routes need to be changed. The automation of flight path planning, e.g., by **designing the master route and allowing the application to take into account the endpoint of the last route as the starting point of a new route on an ongoing basis**, will shorten the breaks between flights, reduce manual operator activities, and optimize flight path planning. This can be gained owing to several features that would help in flight planning:
    - **increasing the net use of batteries leading to longer flights** – the flight route is designed to maintain a reserve of battery energy for the event of a return flight, safe landing, or unexpected maneuvers. Usually, this reserve is set up with a large margin - with careful planning the drone could use more energy to fly and achieve a greater range. Optimization of the route planning **based on the current mission performance status and battery status monitoring** would allow to reduce a reserve of the battery to a reasonable level and to use the saved energy for the realization of the mission.
    - **minimization of overlapping between subsequent flights** – the goal of overlaps it to ensure the continuity of acquired data. They are designed manually with a safe excess. The automatic flight planning **allows for minimum overlap that can be defined based on the actual endpoint of the previous flight**, which further increases the efficiency since less time is used for duplicated work.
    - **implementation of ongoing photo control and the option of immediate completion of complementary missions** – it is assumed that

the probability of obtaining correct photos in the field is high. Single photos are randomly checked immediately after the flight. However, from time to time it happens that, despite efforts, deficiencies or incorrectly acquired photos are revealed during the control and analysis of the photographic material in the office. As a result, additional missions need to be organized to obtain photos for several or more sections of the power line again. This means the necessity to make several (sometimes more) short flights over an area of many kilometers, with the necessity to move often from place to place. The time and effort devoted to complementary flights are significant and make delays and increases the cost of the service. The implementation of automatic photo control and the **ability to define and immediately perform an additional flight for an incorrect part of the line** will greatly improve and shorten the entire service.

Smart planning requires real-time analysis of the constantly incoming status information from the entire swarm of drones, that will be used to automatically generate optimal flight plans for subsequently launched drones. The control over the drone will be automatically assigned to the GCS closest to the drone. This heavily relies on the RAINBOW functionality that allows for distributed data processing. Implementation of this feature will increase the scalability of the system beyond the human capability of coordination of multi-drone operations.

- **increasing the autonomy of the drone by minimizing human assistance** – many of the drone maintenance activities are performed by a human: route planning, route upload, controlling GPS signal status, battery level (and others), and responding to various situations during the flight. Many of these activities take time and introduce the probability of an error. Also, the operation of the system by a large team of personnel significantly increases service costs and leads to further inefficiencies due to a lack of tight coordination. It is currently not possible to introduce full autonomy of the drone system and to eliminate the human's presence due to the aviation law and the need to maintain the safety of unmanned aerial operations, but any reduction in human assistance will reduce the costs, shorten the mission time, and reduce the number of possible errors. RAINBOW functionality that allows for collecting and storage of system metrics and its analytical engine can allow for better task allocation between personnel since it will be possible to monitor a given drone from any GCS.
- **elimination of the execution of the fail-safe procedures due to interference or interruptions in the radio link caused by obstacles and terrain configurations** – any disturbance of the radio link causes the flight control system to switch to the failsafe mode directing the drone towards the place of take-off. If the radio signal reappears and the aircraft reconnects, it will turn back to continue the mission. In an extreme case, with constant interruptions of the communication, the drone returns, and lands. Even if the break in the execution of the mission is temporary, it shortens its net time. Interruptions in communication are caused by obstacles between the drone and GCS (forest, hills, buildings, etc.) or interference from other sources of electromagnetic radiation (radio and TV

stations, radio communication, radiolocation, and radio-navigation stations, etc.). Elimination or at least reduction of the effect of these interruptions, would save the flight time and help increase mission efficiency. Implementation of distributed GCSes would allow for an immediate and atomic transfer of control over a drone from one GCS to another one with a stronger radio signal, allowing for the continuation of the mission. Apart from the mesh networking, this relies on the RAINBOW functionality for the secure sharing of drone authorization keys and establishing trust relationships between GCSes, to ensure that only trusted nodes can control the drone.

- **minimizing the GCS deployment time and the need for frequent relocations** – this is the most important factor that impacts the time of current inspection missions. Power lines run across fields and forests without direct road access. The area may be inaccessible (wetlands, mountains) and a large part of the time is spent looking for convenient access to the vicinity of the power line. Often walking is necessary to reach a convenient place for the GCS location. Deployment time includes also unpacking the system, antenna setup, initialization of the software, and the radio link. Eliminating or at least reducing this time will bring a noticeable gain in system efficiency. Having a distributed GCS system means that the nodes can be deployed in parallel and then relocated one by one as the drones fly further. Smoother deployment can be achieved thanks to the RAINBOW orchestrator's functionalities. Also, the reactive routing, smart storage, and application lifecycle management will allow streamlined GCS mobility, without paying close attention to what services are running on a given GCS node.

5.3 To-be reference scenario

To overcome challenges described in the previous point there is a need for a system that will allow changing a group of independent drones into a swarm, where every drone works towards the realization of a common goal and its actions are automatically coordinated to minimize the time and effort needed to acquire data for the given power line section.

GCS deployment
Ground control stations will be deployed along the power line and connected with a mesh network provided by the RAINBOW **to form nodes of a distributed system**. The time needed to deploy a node and to establish communication can be reduced using zero-touch node configuration, containerized application packaging, automatic deployment, and application lifecycle management provided by the RAINBOW orchestrator. RAINBOW can also help with securing the system, by establishing trust relationships between the nodes, not allowing any rogue node to connect.
An alternative solution would be enabling the drones to talk to each other, but it would not solve the problem of radio link range. Any additional hardware on the board of the drone will reduce its flight time and thus reduce the efficiency of data acquisition. Additionally, GCS nodes can be equipped with directional antennas that will allow for much greater distances, whereas the drones can be equipped only with omnidirectional

antennas. Furthermore, drone flight time is very limited and they need to be shut down often for battery replacement, leading to frequent changes in system topology.

Management of drones by GCSes

Once the nodes are deployed, the operators can start deploying drones. One GCS node will be **able to control multiple drones**. Especially it can be used to launch a drone towards the next GCS, and at the same time wait for the arrival of the drone from the previous GCS. This leads to less frequent relocations. Also, the drones can spend more time on data acquisition flying along the power line, since they do not need to return to the take-off point for landing. Using RAINBOW's smart storage and distributed data processing capabilities, the **nodes can automatically generate flight routes** for the available drones from a master mission, which is the only part that is required to be defined by a human. Once the drones are launched, the nodes track the status of those individual drone missions in real-time and **coordinate their actions over the network**. Using this data, the system will be able to plan subsequent flights in such a way that will reduce the overlap between individual drone missions in comparison to the current practice, where the overlaps are planned before heading off to the field. The drones will calculate quality metrics of the acquired data in real-time and send them to the GCS, so that it can take reactive actions immediately and update planned flight routes, reducing the probability of a need for mission repetition. To keep the system cost-effective, the number of nodes must be limited and it will not be possible to cover the whole section of the power line. Instead, ground control stations must be mobile, so that when data in the given area has been collected, they can be easily relocated to another place, without disrupting the operation of other nodes. This means that the physical topology of the system will not be fixed in time, but since RAINBOW provides reactive routing, this should not be an issue.

Passing the control over drones between GCSes

Having GCS nodes spread along the power line means that a drone can be always within the radio link range of one of the nodes. By implementing a **possibility to pass control over a drone from one GCS node to another**, it will be possible to overcome legal limitations and to open the door for much more efficient use of battery capacity and longer flight times, which in turn increases data acquisition efficiency. It will also prevent interruptions in the mission when radio link interferences occur and failsafe procedures are switched on (e.g., return home), as the drone instantly will switch from one GCS (where interference occurs) to another interference-free GCS. This relies on the RAINBOW's ability to provide networking over unreliable and unstable links, as the GCS nodes are deployed in contingent terrain and are subject to varying weather conditions.

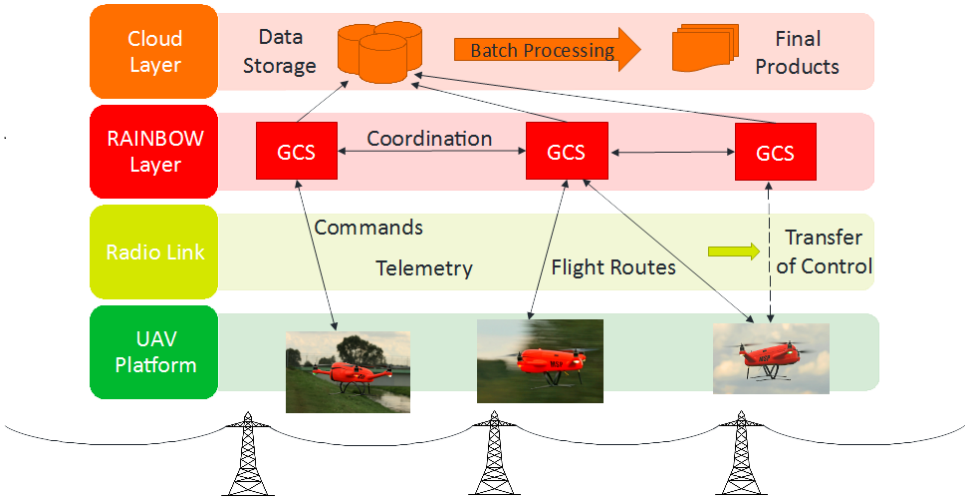


Figure 18: High-level Solution Overview

Using RAINBOW as the base of the distributed GCS means that the most of functionality pertaining distributed nature of the system is already present. It is only necessary to implement three services:

- Communication Gateway,
- Mission Guidance Service,
- GUI Service.

Communication Gateway will be a service that is responsible for communication with a drone using MAVLink protocol. It will receive mission status updates and telemetry stream from the drone and publish them in the RAINBOW data store. It will also pass commands from the Mission Guidance Service to the drone. This service needs to be deployed on every node equipped with a radio modem. It will claim the drone that is with its radio link range until the control over a drone is passed to another Communication Gateway instance or the drone lands and is shut down. Its implementation can be based on the MAVProxy project.

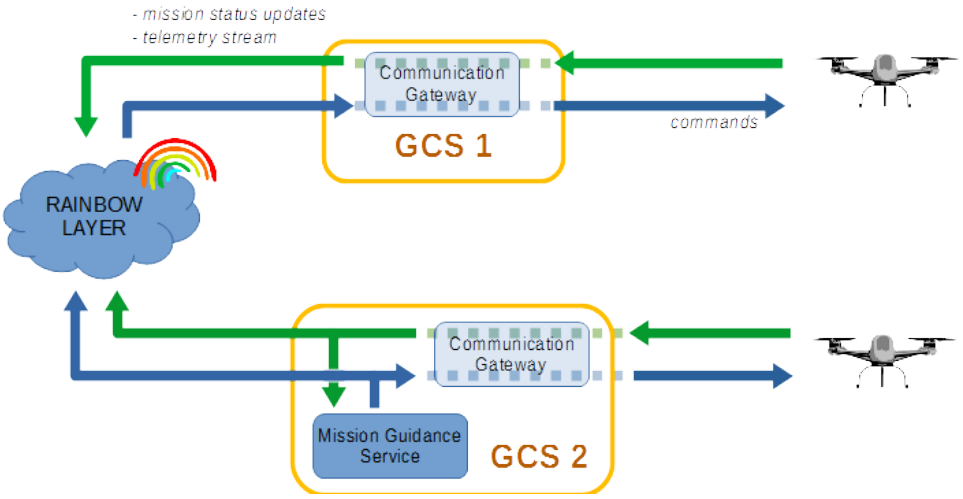


Figure 19: Communication Gateway role in drone control

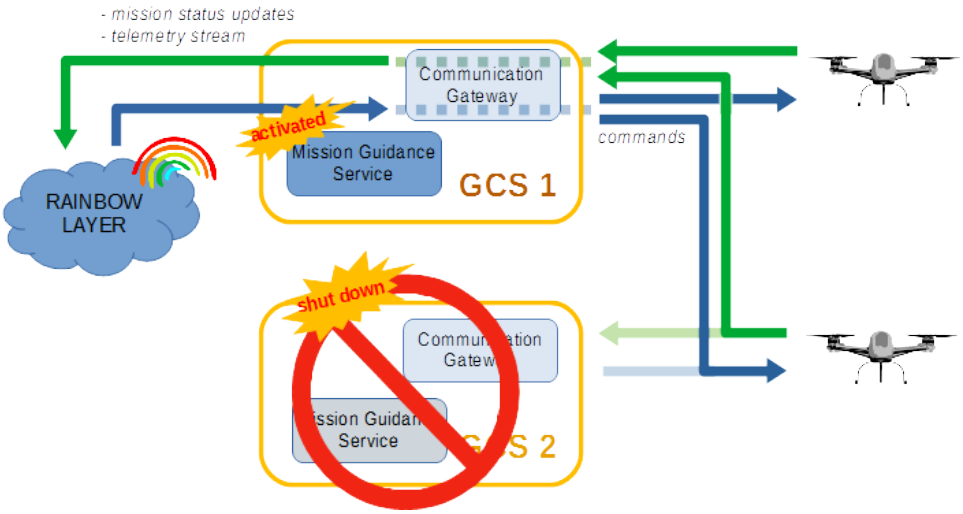


Figure 20: GCS withdrawal, Mission Guidance Service translocation and control over drone transfer

Mission Guidance Service will be responsible for dividing the master mission into flight routes that can be assigned to individual drones and issue commands to the drones. The mission division will be updated in real-time as the drones report back their progress. Since the RAINBOW orchestrator assures that there always will be exactly one active instance of this service, its implementation does not require any use of distributed algorithms – storing its state within the RAINBOW data store will suffice. This will allow it to be moved between nodes with ease, and also it will allow to shut down GCS nodes and relocate them without paying any special attention to what services are deployed in a given node. The RAINBOW orchestrator should optimize its placement in such a way that the latency to the furthest node with the Communication Gateway instance deployed is minimized.

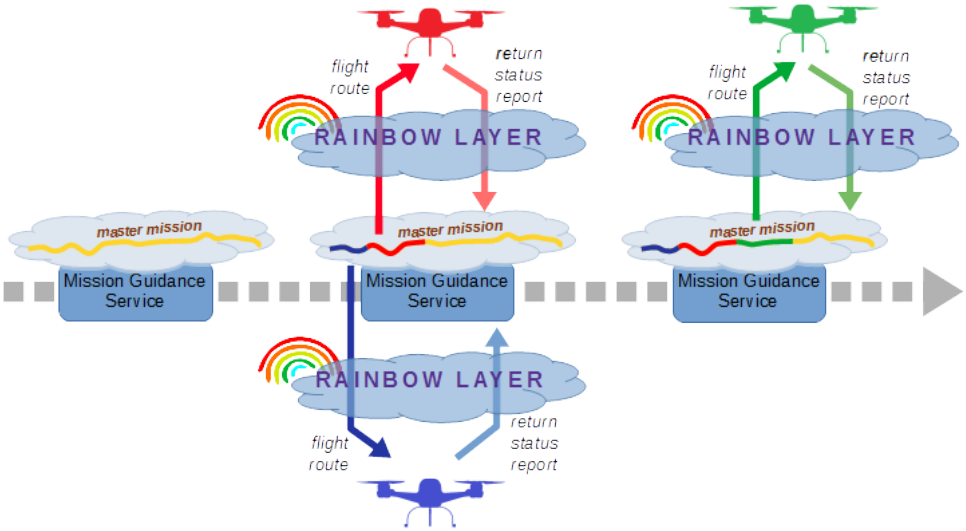


Figure 21: Mission Guidance Service operation iteration

GUI service will act as an adaptor that will allow using the existing GCS GUI software, such as QGroundControl, with RAINBOW data store. It needs to be deployed only on the computers where the GCS GUI will run.

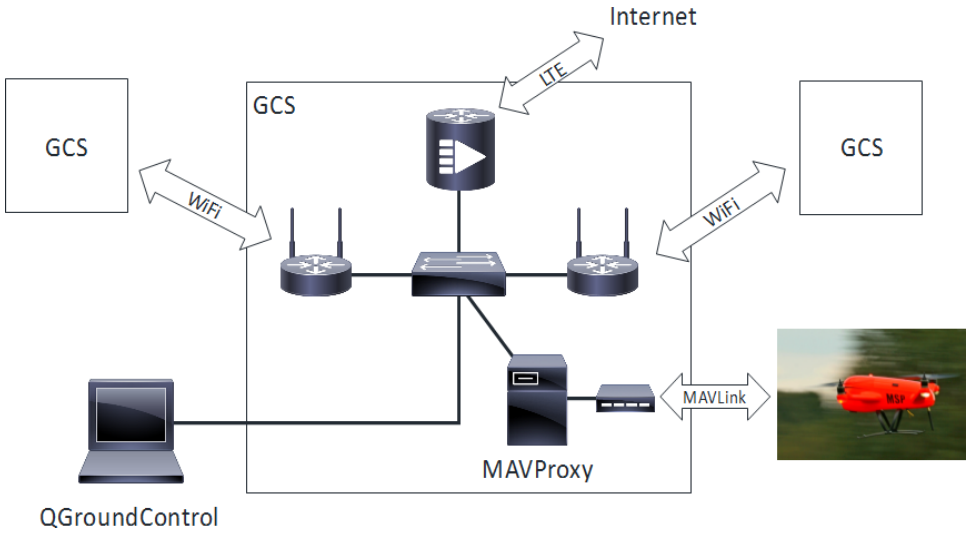


Figure 22: Hardware components of a GCS node

Since the demonstrator is a proof-of-concept implementation and not a complete product, several design constraints will be assumed to reduce effort and costs, without sacrificing its ability to validate the use of RAINBOW technology:

- a small drone with the limited computing power available onboard will be used (reduced costs),

- GCS will be able to control up to two drones at a time (this allows to use of cheaper and lighter radio link hardware),
- a drone can be controlled by one GCS at a time (reduced effort due to a smaller number of required changes to drone firmware and the communication protocol between drone and GCS),
- image quality will be assessed by checking drone orientation, speed, and acceleration at the exact moment the image was captured by the camera to estimate whether the image might be blurred or not and if the camera field of view was correctly oriented (this allows to use a smaller drone).

5.4 Scenario user stories

PLDrones.US.1	As a drone operator I want to deploy a GCS node in the field
User Story Confirmation	The node has: • established network connectivity with RAINBOW infrastructure, • established trust relationship, • been granted access to the RAINBOW data store, • detected presence of GCS service instances running on other nodes.
RAINBOW Functionalities	FT2 Containerized application packaging FT4 Application deployment over fog realms FT5 Application lifecycle management FT8 Reactive routing FT10 Zero-touch security fog node configuration FT11 Fog node "smart" storage
User Story Implementation and Workflow	
Workflow The drone operator sets up hardware components of the GCS node in a designated spot near the power line and powers them up. The node automatically connects to the RAINBOW infrastructure, established trust relationships with neighbouring nodes, and sets up mesh networking. A Communication Gateway is automatically deployed on the node and claims the radio modem. The drone operator connects a laptop and launches the GUI to verify that the node is fully operational. Implementation Considerations It should be possible to quickly deploy GCS nodes so that they can be easily relocated during the mission. This heavily depends on the RAINBOW infrastructure functionalities that allow configuration automation and auto-discovery. To ensure the safety of drone operations, only trusted GCS nodes should be allowed to connect to the system.	

PLDrones.US.2	As a drone operator I want to define a master mission through GCS GUI
User Story Confirmation	The master mission definition is available to all GCS nodes.
RAINBOW Functionalities	FT8 Reactive routing FT11 Fog node "smart" storage
User Story Implementation and Workflow	

Workflow

Drone operator defines the master mission through GCS GUI by:

- drawing trajectories along which data should be acquired,
- assigning required data quality metrics' values to those trajectories,
- and drawing keep-out zones,

or by loading its definition from a file. Then this definition is shared with other nodes via mesh network provided by the RAINBOW, so it is available to the Mission Guidance service.

Implementation Considerations

Services that constitute the GCS rely on the mesh networking provided by the RAINBOW to share data. RAINBOW's data store ensures that they can retain their state if a node is shut down. The operator needs to define a master mission so that Mission Guidance service can generate flight routes for individual drones. After that operator work is limited to monitoring and drone maintenance.

PLDrones.US.3	As a drone operator I want to deploy a new drone
User Story Confirmation	• Information about a new drone is stored in the shared state • Authorization key is securely shared between Communication Gateway services • Mission Guidance service gets notified of a new drone awaiting tasks.
RAINBOW Functionalities	FT10 Zero-touch security fog node configuration FT11 Fog node "smart" storage
User Story Implementation and Workflow	

Workflow

The operator powers one drone and uses GUI to register it in the system. The authorization key used for communication with the drone is securely stored within the RAINBOW data store. Communication Gateway service establishes communication with the drone through radio link and notifies Mission Guidance system that a new drone is available.

Implementation Considerations

A stray or hijacked drone can cause substantial damage and injure people. Therefore, it is imperative to properly secure the communication between the drone and the GCS node. MAVLink protocol allows defining a security key that will be used to authorize commands sent to the drone. Since one of the features is that control over a drone can be passed from one GCS node to another, all nodes should have access to these keys. RAINBOW offers a way to securely distribute them.

PLDrones.US.4	Mission Guidance service executes the master mission
User Story Confirmation	For each part of the mission at least one drone has reported acquisition of data that meets quality requirements.
RAINBOW Functionalities	FT1 Constraint and policy editor FT2 Containerized application packaging FT4 Application deployment over fog realms FT5 Application lifecycle management FT6 Underlying resource and application runtime adaptation FT7 Fog-optimized distributed data processing FT11 Fog node "smart" storage

User Story Implementation and Workflow

Workflow

Mission Guidance service generates flight paths from the master mission definition taking into consideration the number of available drones, their current location, and capabilities (esp. available range). These flight paths are then assigned to particular drones for execution. As the drones start to collect data, they send data quality reports back to the Mission Guidance service along with available range as it might change due to wind. The service records for which parts of the top-level mission the data with acceptable quality was acquired to rule them out from future flight paths. When a drone reports that it has failed to acquire correct data at a given point of its flight path or its range will be lower than expected, it regenerates flight paths. They are also updated when a new drone becomes available (e.g., after battery replacement).

Implementation Considerations

Mission Guidance is an event-driven service that is responsible for the optimal execution of the master mission. By utilizing functionality offered by the RAINBOW, it is possible to greatly reduce the implementation effort, since any implementation of sophisticated distributed algorithms can be avoided. For this to happen, it is necessary to:

- define a constraint that will ensure that there is exactly one instance of this service running at any given moment,
- store all of its state in the data store provided by the RAINBOW,
- provide a mechanism that ensures retention of all messages destined for the Mission Guidance while its instance is not available (e.g., it is being moved from one node to another to optimize latency or a node it runs on shuts down).

PLDrones.US.5	A drone executes a task that has been assigned to it
User Story Confirmation	Mission Guidance service has information on what parts of the task were executed successfully. Also, telemetric data has been stored for further analysis.
RAINBOW Functionalities	FT8 Reactive routing FT9 Adaptive monitoring

User Story Implementation and Workflow

Workflow

After Mission Guidance service has generated flight routes, it sends a command to the Communication Gateway service that orders it to upload a given flight route to the drone and issue a start command via the radio link. When the drone reaches the power line, it starts data acquisition. The drone calculates quality metrics for acquired data in real-time and it sends back status reports at the regular interval. Communication Gateway forwards these reports to Mission Guidance service. This service uses those reports to note which parts of the top-level mission has been completed (i.e., data of acceptable quality has been acquired). If the drone reports that has failed to acquire correct data, the Mission Guidance service decides what corrective action to take.
During the flight, the drone also streams telemetric data (battery level, speed, orientation, position, etc.). The Communication Gateway publishes this data to the RAINBOW data store, so that:

- Mission Guidance is aware of the available range of the drone (which might change in time due to the wind)
- operators can monitor drone activity as required by the law
- it can be analyzed after the mission, to optimize future missions.

Implementation Considerations

The two main ideas behind having a Mission Guidance service that generates all of the flight routes, instead of generating a flight route for a particular drone at the Communication Gateway, is that a single instance has read-write access to the state of the master mission and that it allows generating flight paths that optimize the total execution time of the master mission.
When one drone fails to finish its mission, e.g. due to excessive wind, one possible corrective action might be to extend the flight route of a different drone (that even might be already in the air). Implementing such a decision process where each Communication Gateway would be responsible for flight routes of drones within its radio link range would be much more complicated. Using the mesh networking supplied by the RAINBOW it is possible to easily share data between instances of various services, which makes the implementation of Mission Guidance service much easier.
The operators need to have real-time access to flight parameters of the drones, to be able to assess their state and take over the control in case of an emergency. RAINBOW's reactive routing and overlay network allow feeding the telemetry stream into the system even when its topology changes due to mobility of the nodes.

PLDrones.US.6	Communication Gateways agree to pass control over the drone from one to the another in atomic fashion
User Story Confirmation	Either the current Communication Gateway has ceased and another one has started to communicate with the drone or nothing has changed.
RAINBOW Functionalities	FT7 Fog-optimized distributed data processing FT8 Reactive routing

User Story Implementation and Workflow

Workflow

When Communication Gateway detects that radio link signal strength is fading, it queries shared state to locate another Communication Gateway that could take over the control. Then it sends a request to verify whether the drone is in the radio link range of that another Communication Gateway. If yes, it initiates a distributed transaction algorithm that atomically passes the control over a drone.
If the passing of control failed, either because there is no other Communication Gateway in range, or the transaction has been rolled back, Communication Gateway sends a message to Mission Guidance service to take some other corrective action.

Implementation Considerations

Due to safety and legal requirements, a drone must be in constant radio contact with the GCS system. To extend the drone's productive flight distance beyond its radio link range, Communication Gateways can control this drone in turns. It must be implemented with special care to ensure a situation where one Communication Gateway instance stops and another does not start or fails to communicate with the drone will not happen. Services can use the RAINBOW's mesh networking feature to discover each other's locations and initiate communication.

PLDrones.US.7	As a drone operator I want to monitor flight parameters of a chosen drone
User Story Confirmation	Drone operator can assess the state of a chosen drone in real time.
RAINBOW Functionalities	FT3 High-level analytics query editor and job compiler FT8 Reactive routing FT9 Adaptive monitoring

User Story Implementation and Workflow

Workflow

Drone Operator starts a GUI application that displays a map with the master mission and drone positions. He chooses one of the drones. The application displays drone's flight path on the map and its flight parameters (altitude, velocity, battery level, orientation, etc.). The operator can monitor how the parameters change in real-time to assess the drone's state. If the drone might find itself in a dangerous position, the operator has an option to override its actions.
The data is also retained for more detailed analysis after the mission.

Implementation Considerations

Drone operators are required to monitor drones in flight and step in case of an emergency. Usually, all they care about are the current values. GUI must allow the operator to view multiple parameters on one screen. A design mimicking aircraft instruments is preferred, as operators are used to it. This feature should be designed and implemented in a way that minimizes the delay of presented data. The use of RAINBOW's monitoring and reactive routing features simplifies implementation. Since the data is stored within the RAINBOW infrastructure, the operator can monitor any drone in the system, regardless of its location.

PLDrones.US.8	The drone finishes its flight and lands near an operator.
User Story Confirmation	Mission Guidance service gets notified of a drone awaiting new tasks, or the drone is removed from the system. Optionally the GCS node is moved to another location.
RAINBOW Functionalities	FT5 Application lifecycle management FT8 Reactive routing FT11 Fog node "smart" storage

User Story Implementation and Workflow

Workflow

The drone finishes its flight and lands near an operator. The operator checks via the GUI if there are any new pending flights in the nearby area.
If there are, he replaces the battery and marks the drone as ready for flight. The Mission Guidance generates a new flight route for this drone, and the drone takes off.
If all the data in the nearby area was acquired, the operator uses the GUI to find a location where he should relocate. Then he powers down the drone and the GCS node. He packs everything and heads towards the new location.

Implementation Considerations

It is more cost-effective to have a small number of GCS nodes that are mobile than deploying them throughout the whole length of the power line section at the same time. Mesh networking stack provided by the RAINBOW, especially the overlay network and reactive routing, allows implementing a system that works reliably even if its physical topology changes during its operation. The goal of the distributed GCS is to reduce the time needed to acquire data, resulting in lower costs. The GUI should provide cues for the operator where he should relocate and when. This eliminates a place for errors and inefficiencies. The process of relocation should be as easy as possible so that it does not induce delays.

5.5 Initial Metrics of Success

Id	Qualitative Metrics	Target Value	(M)andatory / (G)ood to Have / (O)ptional
1	*Scalable and Secure deployment of micro-services on need basis. In the scenario of changing number of Personnel, Robots, changing workplace configurations.*	*Supported*	*M*
2	*Continuous monitoring and evaluation of QoS of applications/services running on Fog device. If constraints are not met then actions specified in policies are to be performed.*	*Supported*	*M*
3	*No single point failure like Cluster head failure, drop in QoS, Exception in micro-service should compromise personnel safety. If all unresolved exceptions (within specified time) must stop/halt the robot.*	*Supported*	*M*
4	*Dynamic sharing of resources between Fog should be allowed considering service level objectives are meet. In the scenario when a Fog temporarily lack resources (may be due to overload)*	*Supported*	*M*
5	*Data sharing is restricted within defined boundaries (factory premises, outside factory to third-party etc) with appropriate authentication mechanism and access rights for user in different user-group.*	*Supported*	*M*
6	*All communication between devices must be secured by default*	*Supported*	*M*
7	*On-boarding new fog device must adhere to attestation policies by providing verifiable evidence on their configuration integrity and correctness.*	*Supported*	*M*

8	*Periodically Synchronize data from all distributed databases present in each of the Fog with Central database*	*Supported*	*M*
9	*Support optimized queries requiring to fetch data from Distributed database across Fog device mesh network*	*Supported*	*G*
10	*Support addition of customised high level Analytical queries*	*Supported*	*O*

6 Conclusions

The present deliverable exemplifies the 3 use cases to be employed within RAINBOW. These will be used to validate the overall RAINBOW platform, while conversely the RAINBOW platform will assist these UCs to achieve their goals both technically and business-wise. To this scope a series of calls, participatory filling of templates and linking to the RAINBOW architecture has led into an organized depiction of the various aspects that the UCs should cover. All partners contributed and it was ensured that several issues anticipated to be faced in the upcoming technical work packages were addressed. This deliverable has been linked with the RAINBOW reference Architecture (D1.2) to be used as a reference list of the requirements and functionalities that should be at least covered by the platform to be developed and that will be tested by at least one of the demonstrators.

For each of the three Use Cases the deliverable has defined the current status scenario (AS-IS) and the (TO BE) target, as well as needs, detailed user stories and qualitative/quantitative metrics.

Note that "User Stories" are provided together with the high-level description of how RAINBOW functionalities will interact with the UC infrastructure. For each Use Case main Initial Metrics of Success are also defined to verify the success of the integration between Use Cases ad RAINBOW functionalities.

The scope of this normative description of the UCs is to particularize on which are the challenges and open questions in the context of the Use Cases and how RAINBOW is going to be assisting the UCs. Afterall, this document will be used as input to the next design and implementation project's steps.

7 References

[1] I. K. Thanassis Giannetsos, "Securing V2X Communications for the Future: Can PKI Systems offer the answer?," in *14th International Conference on Availability, Reliability and Security (ARES '19). Association for Computing Machinery*, New York, NY, USA, 2019.

[2] S. Ma, C. Fan, Y. Chuang, W. Lee, S. Lee and N. Hsueh, "Using Service Dependency Graph to Analyze and Test Microservices," in *IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Tokyo, 2018.

[3] J. Whittle, J. Hutchinson and M. Rouncefield, "The State of Practice in Model-Driven Engineering," *IEEE Software,* May 2014.

[4] S. Yi, Z. Hao, Z. Qin and Q. Li, "Fog Computing: Platform and Applications," in *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, 2015.

[5] J. Thalheim, A. Rodrigues, I. E. Akkus, P. Bhatotia, R. Chen, B. Viswanath, L. Jiao and C. Fetzer, "Sieve: Actionable insights from monitored metrics in microservices.," *arXiv preprint arXiv:1709.06686,* 2017.

[6] I. Stanoi, G. Mihaila, T. Palpanas and C. Lang, "Whitewater: Distributed processing of fast streams.," *IEEE transactions on knowledge and data engineering,* 2007.

[7] Y. Shi, G. Ding, H. Wang, H. E. Roman and S. Lu, "The fog computing service for healthcare," in *2015 2nd International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech)*, 2015.

[8] Z. Shelby, K. Hartke and C. Bormann, *RFC 7252 – The Constrained Application Protocol (CoAP),* 2014.

[9] A. Rabkin, M. Arye, S. Sen, V. S. Pai and M. J. Freedman, "Aggregation and degradation in jetstream: Streaming analytics in the wide area," in *11th {USENIX} Symposium on Networked Systems Design and Implementation*, 2014.

[10] Q. Pu, G. Ananthanarayanan, P. Bodik, S. Kandula, A. Akella, P. Bahl and I. Stoica, "Low latency geo-distributed data analytics.," *ACM SIGCOMM Computer Communication Review,* 2015.

[11] P. Pietzuch, J. Ledlie, J. Shneidman, M. Roussopoulos, M. Welsh and M. Seltzer, "Network-aware operator placement for stream-processing systems.," in *22nd International Conference on Data Engineering*, 2006.

[12] P. Patel, M. I. Ali and A. Sheth, "On using the intelligent edge for IoT analytics.," *IEEE Intelligent Systems,* 2017.

[13] M. Nardelli, V. Cardellini, V. Grassi and F. L. Presti, "Efficient operator placement for distributed data stream processing applications.," *IEEE Transactions on Parallel and Distributed Systems,* 2019.

[14] A.-V. Michailidou and A. Gounaris, "Bi-objective traffic optimization in geo-distributed data flows.," *Big Data Research,* 2019.

[15] L. Liu, Z. Chang, X. Guo, S. Mao and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing.," *IEEE Internet of Things Journal,* 2017.

[16] P. Li, S. Guo, T. Miyazaki, X. Liao, H. Jin, A. Y. Zomaya and K. Wang, "Traffic-aware geo-distributed big data analytics with predictable job completion time.," *IEEE Transactions on Parallel and Distributed Systems,* 2016.

[17] B. Heintz, A. Chandra and R. K. Sitaraman, "Trading timeliness and accuracy in geo-distributed streaming analytics.," in *Proceedings of the Seventh ACM Symposium on Cloud Computing*, 2016.

[18] L. Gu, D. Zeng, S. Guo, Y. Xiang and J. Hu, "A general communication cost optimization framework for big data stream processing in geo-distributed data centers.," *IEEE Transactions on Computers,* 2015.

[19] Z. Georgiou, M. Symeonides, D. Trihinas, G. Pallis and M. D. Dikaiakos, "Streamsight: A query-driven framework for streaming analytics in edge computing.," in *2018 IEEE/ACM 11th International Conference on Utility and Cloud Computing*, 2018.

[20] J. Dizdarević, F. Carpio, A. Jukan and X. Masip-Bruin, "A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration," *ACM Comput. Surv.,* 2019.

[21] J. Carbonell, "Machine learning: paradigms and methods.," 1990.

[22] C. C. Byers, "Architectural Imperatives for Fog Computing: Use Cases, Requirements, and Architectural Techniques for Fog-Enabled IoT Networks," *IEEE Communications Magazine,* vol. 55, no. 8, 2017.

[23] C. Bormann, S. Lemay, H. Tschofenig, K. Hartke and B. Silverajan, *RFC 8323 - CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets,* 2018.

[24] F. Bonomi, R. Milito, J. Zhu and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, 2012.

[25] F. Afrati, S. Dolev, S. Sharma and J. D. Ullman, "Meta-MapReduce: A technique for reducing communication in MapReduce computations," in *Proceedings of the 17th International Symposium on Stabilization, Safety, and Security of Distributed Systems*, 2015.

[26] OpenFog Consortium Architecture Working Group, "OpenFog Reference Architecture for Fog Computing," OpenFog Consortium, 2017.

[27] "Open Connectivity Foundation," [Online]. Available: https://openconnectivity.org. [Accessed 04 2020].

[28] International Standards Organization/IEC JTC 1, *ISO/IEC 20922:2016 Information Technology - Message Queuing Telemetry Transport (MQTT) v3.1.1,* 2016.

[29] "A Guide to the Business Analysis Body of Knowledge (BABOK Guide) is a standard for the practice of business analysis created and maintained by IIBA (International Institute of Business Analysis)," [Online]. Available: https://www.iiba.org/standards-and-resources/babok/.

[30] "Lori MacVittie, Micorservices and Microsegmentation, "https://devcentral.f5.com/articles/microservices-versus-microsegmentation." 2015.".

[31] "Martin Fowler, "Microservices a definition of this new architectural term." [Online]. Available: https://martinfowler.com/articles/microservices.html.".

[32] "Eric S. Raymond, "The Art of UNIX Programming." 2013.".

[33] S. M. Fulton, "What Led Amazon to its Own Microservices Architecture," 2015.

[34] M. K. a. J. Haid, "Hardware-based Secure Identities for machines in smart factories," 2016.

[35] T. C. Group, "TCG Specification Architecture Overview Rev 1.4," 2007.

[36] J. Thones, "Microservices," *IEEE Softw., vol. 32, no. 1,* p. p. 116, Jan. 2015.

[37] D. Trihinas, A. Tryfonos, M. Dikaiakos and G. and Pallis, "Devops as a service: Pushing the boundaries of microservice adoption.," *IEEE Internet Computing,* pp. pp.65-71, 2018.

[38] N. Aaraj, A. Raghunathan, Jha and a. N. K., "Analysis and Design of a Hardware/Software Trusted Platform Module for Embedded Systems," in *ACM Transactions on Embedded Computing Systems (TECS)*, 2008.

[39] M. Strasser and H. Stamer, "Software-Based Trusted Platform Module Emulator," in *International Conference on Trusted Computing*, 2008.

[40] C. Shepherd, G. Arfaoui, I. Gurulian, R. P. Lee and a. K. Markantonakis, "Secure and Trusted Execution: Past , Present and Future A Critical Review in the Context of the Internet of Things and Cyber-Physical Systems," in *IEEE Trustcom/BigDataSE/ISPa*, 2016.

[41] Pirker, Martin, Toegl, Ronald, Hein, Daniel, Danner and Peter, A PrivacyCA for Anonymity and Trust., Vols. 101-119. 10.1007/978-3-642-00587-9_7, 2009.

[42] E. Brickell, J. Camenisch and a. L. Chen, "Direct anonymous attestation," in *11th ACM conference on Computer and communications security*, Washington DC, USA, 2004.

[43] W. Arthur, D. Challener and a. K. Goldman, "A Practical Guide to TPM 2.0," 2015.

[44] I. K. Thanassis Giannetsos, "Securing V2X Communications for the Future: Can PKI Systems offer the answer? In Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES '19). Association for Compu," in *14th International Conference on Availability, Reliability and Security (ARES '19). Association for Computing Machinery*, New York, NY, USA, 2019.