This project has received funding from "HORIZON 2020" the European Union's Framework Programme for research, technological development and demonstration under grant agreement no 645220

Road-, Air- and Water-based Future Internet Experimentation

Project Acronym:	RAWFIE		
Contract Number:	645220		
Starting date:	Jan 1st 2015	Ending date:	Dec 31st, 2018

Deliverable Number and Title	D3.1 - Specification & Analysis of RAWFIE Components Requirements (a)		
Confidentiality	PU	Deliverable type[1]	R
Deliverable File	RAWFIE_D_3_1_final	Date	2015-03-30
Approval Status[2]	For review	Version	1.000
Contact Person	Nikolaos Pringouris	Organization	HAI
Phone	+302262046572	E-Mail	PRIGGOURIS.Nikolaos@haicorp.com

[1] Deliverable type: P(Prototype), R (Report), O (Other)

[2] Approval Status: WP leader, 1st Reviewer, 2nd Reviewer, Advisory Board

AUTHORS TABLE

Name	Company	E-Mail
Nikolaos Pringouris	HAI	PRIGGOURIS.Nikolaos@haicorp.com
Kiriakos Georgouleas	HAI	GEORGOULEAS.Kiriakos@haicorp.com
Bastien Boulnois	PEGASE	bastien.boulnois@pole-pegase.com
Patrick Brausewetter	FRAUNHOFER	patrick.brausewetter@ivi.fraunhofer.de
Anestis Trypitsidis	AVIONTEK	rtd-projects2@epsilon.gr
Kostas Kolomvatsos	UOA	kostasks@di.uoa.gr
Sotirios Glykofrydis	HMoD	sglikofridis@dideap.mil.gr
Stathes Hadjiefthymiades	UOA	shadj@di.uoa.gr
Blerina Lika	UOA	b.lika@di.uoa.gr
Alexandre Sousa	MST	alex@oceanscan-mst.com
Miquel Cantero	ROBOTNIK	mcantero@robotnik.es
Philippe Dallemagne	CSEM	philippe.dallemagne@csem.ch
Lionel Blondé	HES-SO	lionel.blonde@hesge.ch

REVIEWERS TABLE

Name	Company	E-Mail
Savvas Chatzichristofis	CERTH	schatzic@iti.gr
Alan Thodey	EPSILON BULGARIA	Alan.thodey@epsilon-bulgaria.com

DISTRIBUTION

Name / Role	Company	Level of confidentiality[3]	Type of deliverable
All		PU	R

CHANGE HISTORY

Version	Date	Reason for Change	Pages/Sections Affected

[3] Deliverable Distribution: PU (Public, can be distributed to everyone), CO (Confidential, for use by consortium members only), RE (Restricted, available to a group specified by the Project Advisory Board).

0.001	2015-01-19	First Document Issue with Introduction & ToC.	All
0.002	2015-01-22	Examples of concrete requirements cards included	4.3
0.003	2015-01-26	1st draft list of scenarios	2.3
0.004	2015-01-29	More scenarios added. Allocation of work performed	All
0.005	2015-01-31	ToC and classification of requirements stabilized	4,5
0.006	2015-02-12		4.3
0.007	2015-02-13	Update list of Platform Requirements, Enhance scenario 1 description	2.3.1, 4.3, 5.1
0.008	2015-02-16	Update list of Testbed Requirements	4.4, 5.2
0.009	2015-02-16	Updated list of Platform Requirements	4.3
0.010	2015-02-17	Updated info for UxV & UGV specific models	3.1
0.011	2015-02-20	Deliverable adapted to RAWFIE official template	All
0.012	2015-03-04	Requirements section updated, added section 7.1	5,5,7
0.013	2015-03-06	Scenario section Updates, Stakeholder section update, UxV characteristics & constraints updates, Minor updates to Requirements section, References and Abbreviations added	1.2, 3.2, 3.3, 4, 5.3, 6,1, 9
0.014	2015-03-16	Scenarios updated, Platform & Testbed requirements section updated, UxV constraints section updated, minor updates and corrections to all other sections	All
0.015	2015-03-23	Final Version for review process	All
1.000	2015-03-30	Address reviewers comments	All

Abstract:
The deliverable provides a first version of the RAWFIE user and system requirements that will form the basis for the definition of the Platform and Testbeds architecture and detailed design. In keeping with the overall project workflow, the requirements captured and synthesized here will be given to WP4 – Platform Design as input for the first development cycle.

Within the deliverable, a number of use cases are presented. A use case is in most cases a real world scenario that aims to serve as source for identifying user specific requirements, capabilities and corresponding experiments that need to be supported by the RAWFIE Platform.

The requirements identified in this first iteration are divided in 2 broad categories (a) Platform and (b) Testbed (including UxV resource specific) requirements and are also prioritized base on their significance.

The document also includes a section related to constraints and limitations pertaining to UxV systems and the testbed areas based on facility providers' feedback.

Keywords: requirements, scenario, experiment, constraints standards & regulations, functional & non-functional

Part II: Table of Contents

List of Figures

List of Tables

Part III: Executive Summary

The deliverable provides a 1st version of the RAWFIE requirements including a number of representative scenarios that could form the basis for the platform design and development activities in subsequent WPs. The present document is the first in a series of three requirements analysis documents that are to be delivered in the beginning of each RAWFIE iteration cycle.

Part IV: Main Section

1 Introduction

1.1 Scope of Deliverable

The purpose of this document, "D3.1 **Specification & Analysis of RAWFIE Components Requirements**", is to identify requirements from a users and high level system perspective in order to build the RAWFIE Platform and the federation of underlying UxV testbed facilities. It is the first deliverable in a cycle of three which will all focus on identifying requirements for the various RAWFIE components. This version of the deliverable besides providing a high level requirement analysis of the RAWFIE system (performing a logical high level division of them between platform and testbed related), it also includes a number of use cases serving as reference for the capabilities that need to be supported by the RAWFIE platform in general and the experimental description language more specifically.

This document is structured as follows:

- Chapter 2 starts with a general overview of the various phases of requirement analysis-management process and conclude with the suggested methodology to be adopted for RAWFIE project while it also proposes the template and formalizations guidelines to be used during requirement writing.
- Chapter 3 explains the purpose of the project, briefly presents its various stakeholders and focus on the description of the scenarios (use cases) and the experiments derived from them. A total of 6 scenarios were identified.
- Chapter 4 presents general information and possible limitations pertaining to the testbed facilities as well as constraints and limitations of the UxV resources (both general and model specific). Information on regulation and standards for each UxV domain is also included wherever available.
- Chapter 5 focuses on the presentation of functional requirements divided in 2 broad categories (Platform & Testbed)
- Chapter 6 focuses on the presentation of non-functional requirements divided in 2 broad categories (Platform & Testbed).
- Chapter 7 presents various issues that should be taken into consideration during project development (with focus in EDL and data analysis)
- Chapter 8 provides a summary of the work performed in the present deliverable and sets the target for the next iterations

1.2 Abbreviations

Abbreviation	Meaning
AHRS	Attitude and Heading Reference System

AGL	Above Ground Level
AP	Access Point
AT	Aerial Testbed
AUV	Autonomous Underwater Vehicle
B-VLOS	Beyond Visual Line Of Sight
CAA	Civil Aviation Authority
CAO	Cognitive-based Adaptive Optimization
CBNR	Chemical Biological Nuclear Radiological
CEP	Circular Error Probability
CPU	Central Processing Unit
DETEC	Department of the Environment, Transport, Energy and Communication
DGCA	Directorate General of Civil Aviation
DoW	Description of Work
EASA	European Aviation Safety Agency
ECC	Error Correction Code
EDL	Experiment Description Language
EU	European Union
E-VLOS	Extended Visual Line Of Sight
FIRE	Future Internet Research & Experimentation
FOCA	Federal Office of Civil Aviation
FPS	Frames Per Second
FPV	First Person View
GAA	German Aviation Act
GIS	Geographical Information System
GNSS	Global Navigation Satellite System
GPIO	General Purpose Input/Output
GPS	Global Positioning System
HD	High Definition
HW	Hardware
IAA	Irish Aviation Authority
IaaS	Infrastructure as a Service
IFR	Instrument Flight Rules
IDE	Integrated Development Environment
IP	Internet Protocol
ISO	International Standards Organization
JSON	JavaScript Object Notation
KPI	Key Performance Indicators
LBL	Long Baseline
MEMS	MicroElectroMechanical System
MM	Monitoring Manager
MSO	Multi Swarm Optimization
MT	Maritime Testbed
NF	Non Functional
OEDL	OMF EDL
OMF	cOntrol and Management Framework
OS	Operating System
OTA	Over The Air
P2P	Point to Point

PSO	Particle Swarm Optimization
PTZ	Pan Tilt Zoom
RC	Radio Controller
RE	Requirement Engineering
RIA	Research and Innovation Action
ROS	Robot Operating System
ROV	Remotely Operated Vehicle
RPA	Remotely Piloted Aircraft
RPAS	Remotely Piloted Aircraft System
RPS	Remotely Piloted Station
SaaS	Software as a Service
SQL	Simple Query Language
TM	Testbed Manager
TMS	Testbed Manager Suite
TP	Testbed Proxy
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UI	User Interface
USB	Universal Serial Bus
USV	Unmanned Surface Vehicle
UxV	Unmanned System (of any type)
VFR	Visual Flight Rules
VLL	Very Low Level flight, below 150m above ground level
VLOS	Visual Line of Sight
VT	Vehicular Testbed
XML	Extensible Markup Language

Table 1: Abbreviations

2 Methodology

2.1 General

The process of developing requirements involves four (4) generic activities, namely: requirements elicitation, requirements analysis, requirements specification, and requirements verification & validation. In practice this process is tailored based on the domain or organization where it is applied.

Classically, requirements elicitation involves identifying stakeholders, their needs and constraints, and methodically extracting requirements. The outcome of elicitation is a set of raw requirements which are likely to be incomplete, ambiguous, inconsistent, and which are largely documented in natural language (or with the use of graphical notations that may or may not be augmented with natural language). Subsequently, the raw requirements are analyzed to resolve the ambiguities, inconsistencies, and conflicts. Requirements analysis is considered a critical step to the success of a systems or software project. It should lead to requirements that are well documented, measurable, testable and traceable to subsequent outputs (i.e. architecture design documents) of the project. Requirements should be defined to a level of detail sufficient for the system design that will follow.

During post analysis, a requirements specification (document) is created, that contains the analyzed requirements documented in some acceptable form e.g. natural language (as is common in the state of the practice), formal or graphical notations. The requirements specification then undergoes validation, where it is confirmed that the specification is an accurate, correct and complete statement of the stakeholders' needs. The goal is to produce a requirements specification document of only those requirements which have been accepted.

In fact, this progression of activities described above, abstractly represents the requirements development process, the outcome of which is a requirements specification that exhibits a set of desired quality properties e.g. those recommended by the various IEEE standards.

This progression of activities may be linearly ordered, in a waterfall like process [1], as shown in Figure 1. Here the activities are serially ordered such that requirements elicitation and analysis is followed by specification, and preceded by a feasibility study. Subsequent to specification, requirements validation occurs, and the outcome of the process is a requirements document containing a model of the system to be built, user and system requirements.

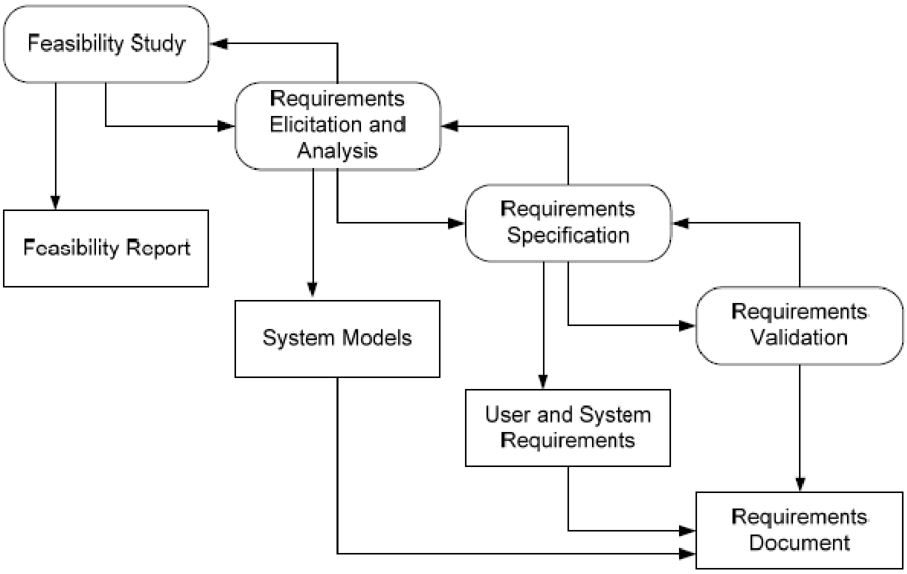


Figure 1: Waterfall model of RE process [1]

In practice the process may not follow such a strict linear progression; rather, elicitation is observed often to intertwine and iterate with analysis, specification and validation. Besides these, some supporting activities exist for requirements development. Briefly, they include requirements discovery, classification, prioritization and negotiation. Requirements discovery is performed, in part, during elicitation where interaction with stakeholders occurs to clarify user needs and discover requirements arising from the domain. During classification of requirements, coherency among requirements is achieved by organizing them according to the identified classification categories. Subsequently, prioritization and negotiation of requirements assists in identifying and resolving requirements conflicts.

In many cases, the creation of a complete requirement specification before moving to the next phases of system or software design is a pretty difficult task due to the inherent system complexity or the fact that certain constraints or issues will not become evident until development activities start or even a first version of the system is put in operation. In such situations acceptable best practices mandate the combination of linear and iterative systems development methodologies, with the primary objective of each being to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process. Such processes are described as incremental or evolutionary.

RAWFIE project prescribes the use of 3 iterations during platform development therefore it seems that an evolutionary development life cycle process is well suited. During each iteration, a requirement specification is provided that can be used for the delivery or evolution of certain operational capabilities of the final product which can be put to operation. Feedback will be provided that may lead to modify or extend requirements in subsequent iterations. Each delivery in this model represents a full development cycle, including requirements analysis. The deliveries may overlap, as shown in Figure 2 or one delivery may be completed before the next is begun. The product of each requirements analysis phase is an addition or improvement to the product(s) of the requirements analysis phase of the previous delivery. Similarly, the implementation portions of each delivery may add to, or upgrade, products of earlier deliveries. With this understanding, each delivery may be looked at as a small example of a baseline management life cycle, with a development process and time span small enough to minimize the problems discussed above.

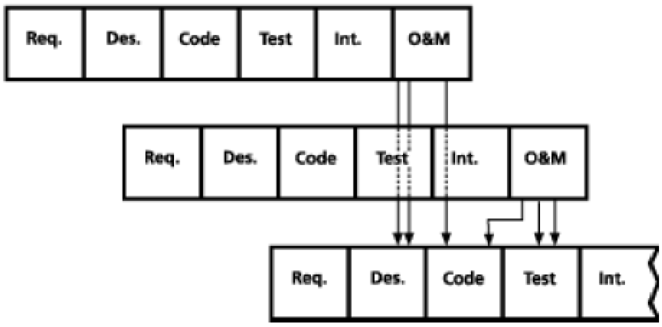
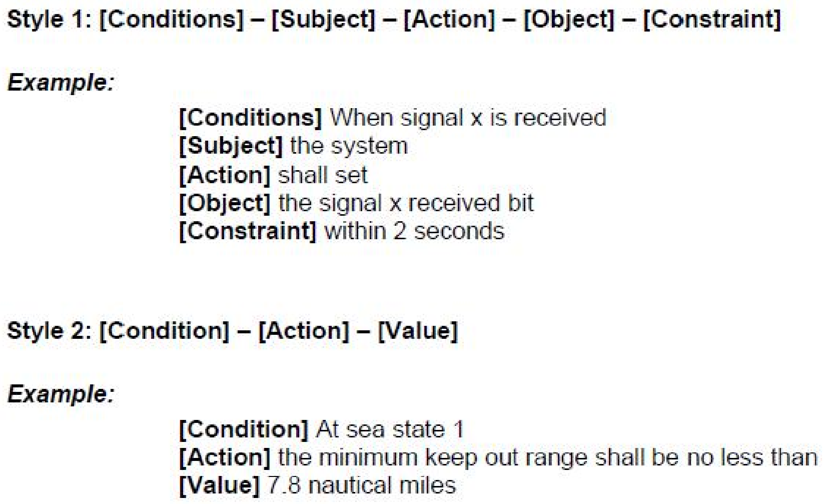


Figure 2: Simplified schema of Evolutionary Development model

In modern requirement analysis, appropriate requirement management tools are often employed that facilitate:

- The definition of requirements using certain patterns or wordings (often based on predefined vocabulary or a domain specific ontology). This enables the creation of well-formed requirements (semi-formal or formal) which can be further supported by formal languages. The advantage is obvious. Requirements become clearer to the potential reviewer; tools can apply for validating their correctness and consistency while identification of inconsistencies, ambiguities, missing requirements, and requirements conflicts can be facilitated. Figure 3 below presents an example styles for writing requirements based according to ISO WD 29148:2011 standard [2].



Style 1: [Conditions] – [Subject] – [Action] – [Object] – [Constraint]

Example:

[Conditions] When signal x is received
[Subject] the system
[Action] shall set
[Object] the signal x received bit
[Constraint] within 2 seconds

Style 2: [Condition] – [Action] – [Value]

Example:

[Condition] At sea state 1
[Action] the minimum keep out range shall be no less than
[Value] 7.8 nautical miles

Figure 3: examples of requirements statement construction styles recommended by ISO WD 29148:2011

The standard recommends the usage of the words 'shall' and 'should' to be used to express whether a requirement is mandatory or not, and exhorts the avoidance of vague or ambiguous terms and subjective language.

- The linking and traceability of requirements to design and validation documents like architecture design, detailed design, test cases etc. This makes much easier to detect requirements that were overseen, not fulfilled or not tested/verified.

2.2 Requirement Analysis methodology in RAWFIE

In RAWFIE although we do not impose the use of a proper Requirement Management tool or strict formalization of requirements we do propose to follow some general principles and adopt a common template for gathering and eliciting requirements from all the partners involved in this project. The VOLERE requirements specification template [3] has been chosen as a start point, adapting it to the particularities and needs of RAWFIE.

Each of the partners involved in the project follows their own processes in the requirements definition phase of their activities. It was important to propose a common way to formalize the requirements that is easy to use and adapted to the needs of the project. In this context, VOLERE is a straightforward methodology that does not require a complex analysis to be applied. Furthermore, it guarantees the participation of all relevant actors, who are further involved in the design and development that have to fulfill the requirements defined.

The adapted methodology used by RAWFIE, will allow the identification and formalization of unambiguous requirements while it will ease the subsequent assessment and validation processes, during Architecture Definition, Development and Evaluation. It is important to make use of a common methodology to gather, classify and assess the requirements a priori. The management of the requirements depends on this common methodology, providing the means to trace the identification, definition, assessment, formalization and if necessary improvement of the requirements gathered.

Moreover, requirements should be the key to evaluate the entire project at the end of the development phase. A set of well-defined and unambiguous requirements is needed, not only as input for any further specification and development, but also as part of the evaluation framework.

VOLERE defines the gathering process and the shell to register the requirements, classified in 27 categories in 5 main groups:

Project drivers, the business-related forces. For example, the purpose of the project is a project driver, as are all of the stakeholders - each for different reasons.

Project constraints, restrictions on how the product must be designed. For example, it might have to be implemented in the hand-held device being given to major customers, or it might have to use certain existing servers and desktop computers, or any other hardware, software, or business practice.

Functional requirements, the fundamental or essential subject matter of the product. They describe what the product has to do or what processing actions it is to take.

Non-functional requirements, the properties that the functions must have, such as performance, security and usability. Do not be deterred by the unfortunate type name (we use it because it is the most common way of referring to these types of requirements) - these requirements should be considered of equal important as the functional requirements for the product’s success.

Project issues, the conditions under which the project will be done. The reason for including them as part of the requirements is to present a coherent picture of all factors that contribute to the success or failure of the project and to illustrate how managers can use requirements as input when managing a project.

VOLERE methodology is a universal way to describe requirements. In the 1st version of the requirements deliverable, we expect to focus on user requirements more from user needs perspective rather than system requirements. Of course it is not possible to present only user view on the RAWFIE project because users are not aware of many details, constraints and other assumptions. Therefore, for the 1st version of the Requirement Analysis document we will deliver a list of requirements consisting mainly of user requirements prescribed by a set of potential scenarios as well as of high level functional and non-functional requirements derived from DoW, consortium expertise or review of work performed in other FIRE experimental platforms. The requirements will be grouped in 2 major classes based on RAWFIE envisaged architecture:

- Platform Requirements
- Testbed Requirements

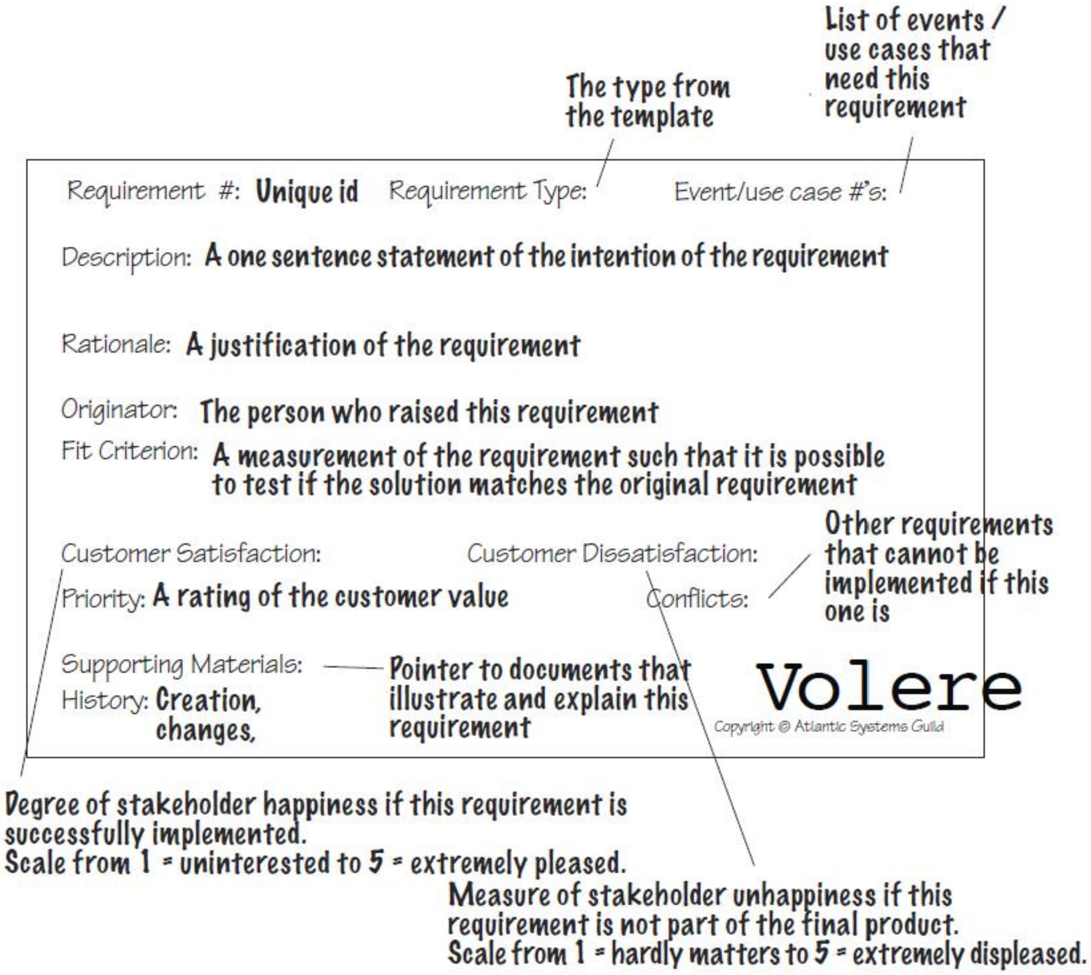
The first class includes requirements related to the server side part of RAWFIE that is middleware, front-end and data tier while the second class includes any kind of requirement that has to do with the local testbed nodes and the UxV platforms. For each of the above categories a further sub grouping may occur while requirement analysis and architecture definition proceeds in details and more functionalities, capabilities and/or modules are identified.

The VOLERE template prescribes description of a particular requirement with following characteristic:

- *Requirement Numbering*. Give each requirement a unique identifier to make it traceable throughout the development process. The numbering scheme suggested in the requirement shell is as follows:
    - Requirement # is the next unique requirement number
    - Requirement Type is the section number from the template for this type of requirement. The inclusion of the section number is not absolutely necessary because we do have a unique requirement id. However it serves as a reminder of what this requirement relates to and helps to remind why the requirement is considered important. Also the ability to compare requirements of the same type makes it easier to identify contradictions and duplications.
- *Event/use case #* is the identifier of a business event or use case that contains this requirement. There might be several Event/use case #'s for one requirement because the same requirement might relate to a number of events. The terms event and use case are already widely used in the systems development world.
- *Customer Value* is a measure of how much your client cares about each requirement. Customer should grade each requirement for Customer Satisfaction or Customer Dissatisfaction. The point of having a satisfaction and a dissatisfaction rating is that it guides clients to think of the requirement from two different perspectives, and helps to uncover what they care about most deeply.
- *Dependencies* keep track of other requirements that have an impact on this requirement. If the dependency exists because requirements use the same information, then use of standard naming conventions and definitions will implement this dependency.

Other dependencies exist because a solution to this requirement has a positive or negative effect on solutions to other requirements. Some requirements, especially project drivers and project constraints, have an impact on all the other requirements.

- *Conflicts* keep track of other requirements that disagree with this one.



The type from the template

List of events / use cases that need this requirement

Requirement #: Unique id Requirement Type: Event/use case #'s:

Description: A one sentence statement of the intention of the requirement

Rationale: A justification of the requirement

Originator: The person who raised this requirement

Fit Criterion: A measurement of the requirement such that it is possible to test if the solution matches the original requirement

Customer Satisfaction: Customer Dissatisfaction:

Priority: A rating of the customer value Conflicts:

Other requirements that cannot be implemented if this one is

Supporting Materials: Pointer to documents that illustrate and explain this requirement

History: Creation, changes,

Volere

Copyright © Atlantic Systems Guild

Degree of stakeholder happiness if this requirement is successfully implemented.
Scale from 1 = uninterested to 5 = extremely pleased.

Measure of stakeholder unhappiness if this requirement is not part of the final product.
Scale from 1 = hardly matters to 5 = extremely displeased.

Figure 4 Requirement card template based on Volere (extract from Volere Template Edition 13 — August 2007)

For the purpose of RAWFIE a more simplified card is proposed compare to the one presented in Figure 4 . This is presented in Table 2 below.

Id:	{ClassId}-{XXX}	**Type:**	follow categorization proposed in Volere template (see Table 3)	**Importance (priority):**	LOW, MEDIUM, HIGH	**Source:**	Requirement origin e.g.: Consortium Know-how, members, law regulation, standards etc.	**Ver:**	1		
Title:		Requirement title/name (1 sentence)									
Description:		More detailed description of particular requirement (textual form only). If Requirement title is sufficient enough to understand the requirement, this filed can remain empty.									
Additional Info (comments):		Any additional info to better clarify or illustrate concepts (pictures may be possible).									
Related Scenario(s)											

Table 2: Exemplary Requirement Card used in this Deliverable

Functional	Functional	FUNC
	Data	DATA
Non-functional:	Look and Feel Requirements	L&F
	Usability Requirements	USE
	Performance Requirements	PERF
	Operational - Environmental Requirements	ENV
	Maintainability and Support Requirements	SUP
	Security & safety Requirements	SEC
	Other	OTH

Table 3 List of Requirements Types

2.3 Definitions

Throughout this document and to enable better formalization of requirements, the following wording is encouraged to be used during definition of requirements:

“Shall” statements are binding requirements. They describe something that is mandatory. If a requirement uses “shall”, then that requirement must be satisfied without fail. Non-compliance is not allowed. Failure to comply with one single 'shall' is sufficient reason to reject the entire product

“Should” is weaker. It can be regarded as a non-mandatory provision. It describes something that might not be satisfied in the final product, but that is desirable enough that any non-compliance shall be *explicitly* justified. Any use of 'should' should be examined carefully, as it probably means that something is not being stated clearly. If a 'should' can be replaced by a 'shall', or can be discarded entirely, so much the better.

“May” statements are also non-mandatory provisions. It grants permission to do something, and makes only a weak statement. It does not mean that it is possible to do it, only that you have permission to do it. In a user requirements document it shall only appear rarely, if ever. It is more appropriate to the detailed design, where it may be used to define the behaviour of the product.

“Will” statements are non-mandatory, either they imply intent on design constraints or they are future tense.

Moreover, requirements are classified in two (2) broad categories as **functional** and **non-functional**. Functional requirements include also data related requirements. Non functional ones include requirements related to performance, usability, safety & security etc. A full list is provided in Table 3 including the abbreviations to be used within the requirement card.

3 Project Drivers

3.1 The Purpose of the Project

The purpose of the RAWFIE project is to create a federation of different network testbeds that will work together to make their resources available for experimentation under a common framework. RAWFIE will integrate multiple testbeds for experimenting in vehicular (road), aerial and maritime environments. The following 3 distinct types of testbeds are foreseen:

- A Vehicular Testbed (VT) will deal with Unmanned Ground Vehicles (UGVs)
- An Aerial Testbed (AT) will deal with Unmanned Aerial Vehicles (UAVs)
- A Maritime Testbed (MT) will deal with Unmanned Surface Vehicles (USVs) or Autonomous Underwater Vehicles (AUV)

A potential testbed facility, provided it has the necessary infrastructure, may also host different types of UxVs acting i.e. both as VT and AT.

The basic idea behind the RAWFIE effort is the automated, remote operation of a large number of robotic devices (UGVs, UAVs, USVs) for the purpose of assessing the performance of different technologies in the networking, surveillance/sensing and mobile/autonomic application domains.

3.2 The Client and Other Stakeholders

RAWFIE is a Research and Innovation Action (**RIA**) that falls under EU **FIRE** initiative (Future Internet Research and Experimentation) aiming at creating an open research environment which facilitates strategic research and development of new Future Internet concepts, giving researchers the tools they need to conduct large-scale experiments on new paradigms. At the moment and until sustainability plan becomes available; there is no specific client in the sense of buyer or acquirer of the platform. In the future it will be necessary to build a cost model for the federation, by identifying and quantifying all costs involved in setting up, maintaining, developing and managing the different facilities that are part of the federation taking into account national or regional funding as well as EC funding and industry funding. This will also strongly depend on the specifics of the experimentation facilities, the experimenter communities and the maturity of the technologies investigated. From a platform usage perspective, the actual clients should be considered the various experimenters.

Main actors – stakeholders of the RAWFIE federation are:

- *Experimenters:* users who belong to the RAWFIE federation if they have credentials which are acknowledged by the federation partners. They are the ones that perform the experiments and can review the possible outcome.

- *RAWFIE Platform Administrator:* administrator of RAWFIE middleware framework. Middleware is owned by the RAWFIE consortium
- *Facility Providers:*
    - *Testbed Operators:* owners of UxV testbeds responsible for their operation and maintenance
    - *UxV providers :* suppliers or operators of UxVs resources

The above classes are the primary participants in a federation – it would be difficult (though not impossible) for a federation to operate without all of these three stakeholder types. However from a broader point of view, RAWFIE will indirectly support or more precisely interact with additional types of stakeholders that are:

- *Suppliers of goods and services:* either developers of facilities or suppliers of infrastructure and services) provide tools to operate and monitor the facilities, supporting the experimentation lifecycle process. The tools to be used can be community open source tools or other software provided by system integrators or any software developer (e.g. an SME or even a facility provider). This category includes equipment manufacturers developing and providing devices, sensor manufacturers, IT equipment, etc. to the facilities
- *End-users:* are the potential target users of the experiment object and thus, in the longer term, they are the most benefited players by the federation and its activity. End users can be citizens, residential or business users who will benefit from the experiment outcomes once they are consolidated and ready to be operational in a production environment
- *EU policy makers and funding bodies (EU, national, regio*nal or local).

Figure 5 below depicts the various RAWFIE's stakeholders and the interactions – relationships among them as prescribed in RAWFIE DoW.

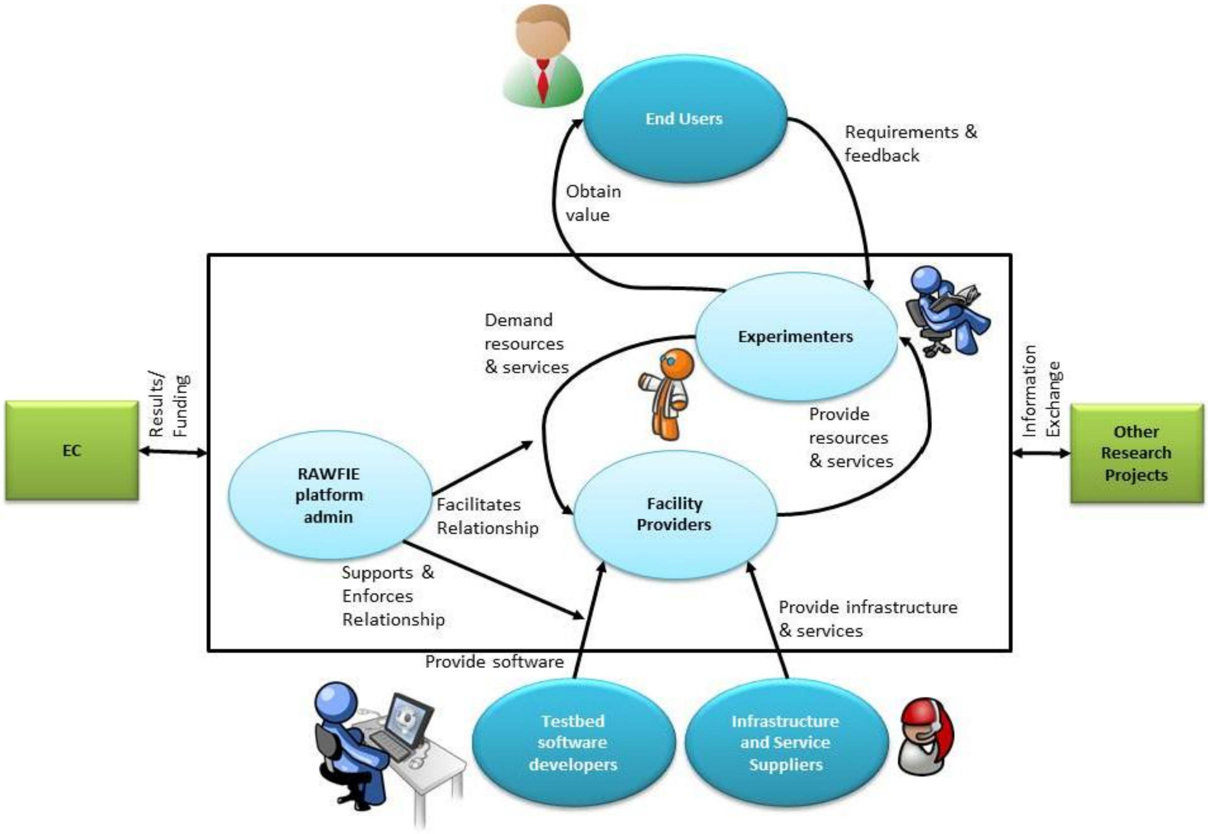


Figure 5: RAWFIE Stakeholders

3.3 Users of the Product

In this section, a list of representative scenarios-use cases aiming to provide an insight on the potential usage of the experimental platform is presented. These scenarios will drive requirement analysis and the identification of requirements listed in subsequent sections. For each scenario a general description is provided outlining the general concept and potential end users, followed by a more detailed analysis that elaborates on possible experimental aspects and steps that need to be performed. A simplified Use Case diagram is provided for each scenario. The scenarios descriptions will be revised and possibly enhanced in subsequent versions of the document.

3.3.1 Scenario 1 – Environmental Monitoring of Water Canals

Overview/Rationale

In this use case RAWFIE platform will be used to mobilize resources that can collaborate for the purpose of environmental monitoring of water canals and gather of information that can be used for assessing quality of the water and structural integrity of canal walls. End user of this use case is a water company, responsible for administering and monitoring multiple “islands” of water canals used for irrigation or drinking purposes. On a periodic or ad hoc basis the company wants to:

- Gather measurements regarding the concentration of grass and/or sediment on the bottom of water canals
- Detect cracks in the canal's wall structure

In this way, the company can proceed with appropriate actions in case a problem is detected. Indeed, the collected information should be reported in company's premises in order to be analyzed (offline) and subsequent procedures for "cleaning" the canal are initiated if deemed necessary.

Picture

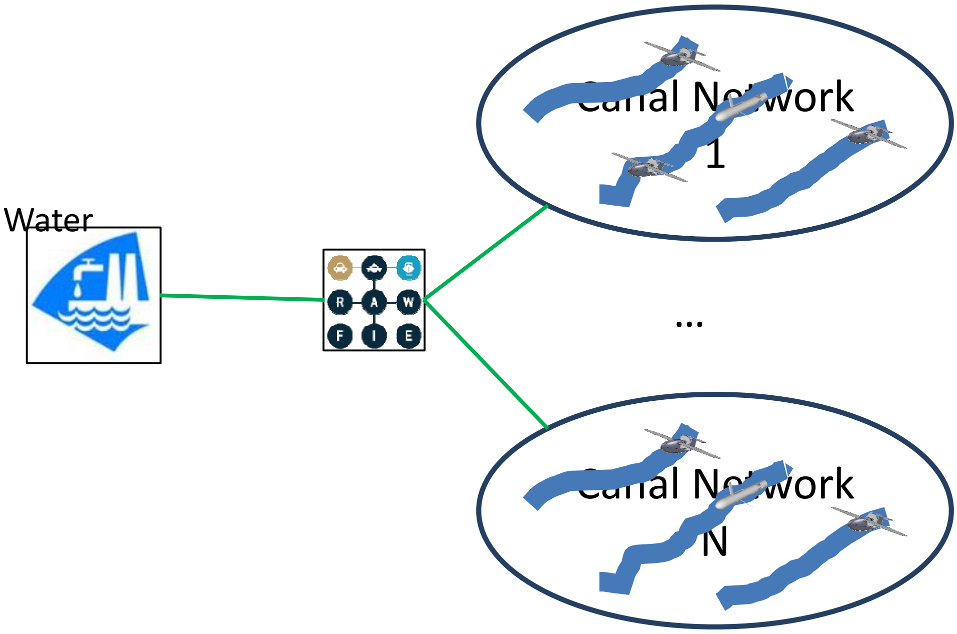


Figure 6: Environmental Monitoring of Water Canals scenario

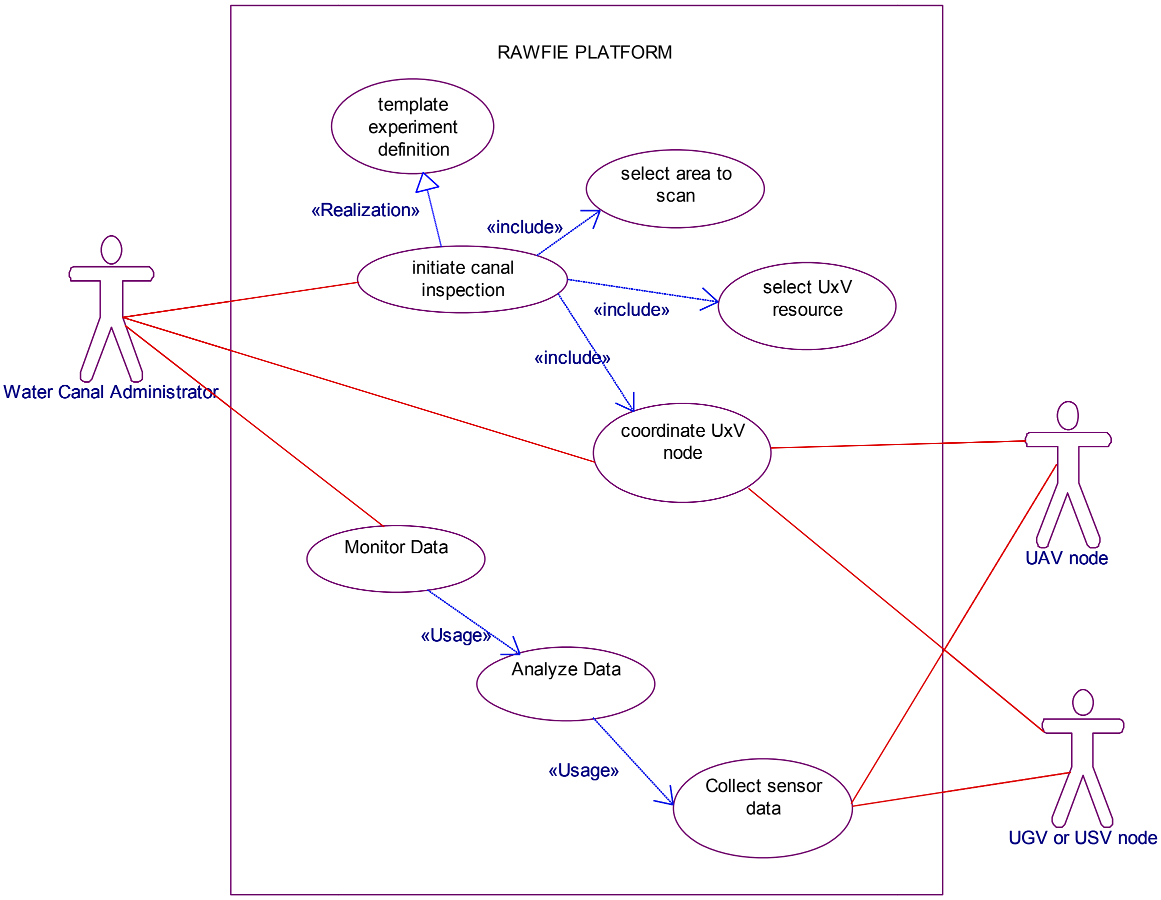


Figure 7: Use Case diagram for scenario 1

Analytic Description

The potential environments of this scenario are canals that need recurring monitoring. The objective is to manage and monitor multiple networks simultaneously.

In the context of the present use case, RAWFIE platform facilities can be utilized to execute a series of experiments in order to assess and identify the optimum way to utilize available UxVs resources for the purpose of examining the canals of a "network" for pollution or structural integrity problem.

Different UxV can be use, whose type and variety are related to the objectives and mission requirements:

- Fixed wing UAV: in order to inspect rapidly a large area. The objective is to determine by photogrammetry potential problem areas

- Rotary wing UAV: in order to inspect precisely the problem area. The objective is to confirm and quantify the problem out of the water. Only the above water areas can be analyzed.
- USV (or AUV): in order to inspect precisely the underwater problem area. The objective is to confirm and quantify the problem in the water by bathymetric measurement.
- UGV: in order to inspect precisely bank areas unreachable or not easily visible by an UAV.

In this use case, the control of the quality of the structure and condition of the banks, and the identification of pollution by algae and sediment deposition is done first by an UAV. This one makes a first large mapping of the situation in order to identify problem areas. Then UGV and/or USV (AUV) may operate in these areas identified as suspicious in order to perform more precise observations and do a better assessment of the situation.

The objective is to achieve an annual campaign (or two) depending on seasonal parameters.

For aerial work, spectral imaging sensor will be used; for the most accurate maritime labor, sound sensors will be used; for ground work, the study is still ongoing.

Therefore, at this stage the expected data to be collected will be spectral maps, aerial photos and acoustic maps;

The RAWFIE platform must allow managing the campaign on a given network. That is to control the drones, to interact with them and possibly redirect their movement plan, and to be able receive and analyze the gathered sensor data.

In addition, the platform must learn, i.e. integrate use cases or results and be able to reuse or complement to other campaigns and other networks.

3.3.2 Scenario 2 – Border Surveillance or Perimeter protection of large areas

Overview/Rationale

In this use case RAWFIE platform will be used to mobilize resources that can collaborate for the purpose of border, infrastructure or sensitive area monitoring and gather information that can be used for assessing a potential threat and take urgent action to protect the area or borders from invention or asymmetric threats.

The potential environments of this scenario are land/sea borders or a camp/ infrastructure, environments that need constant monitoring.

Potential end users for this scenario are:

- Owners of Critical Infrastructures (i.e. energy production facilities, water treatment facilities
- Airports-Ports-Central cargo railway stations
- Forest Protection Organizations
- Border Security Units

Picture(s)

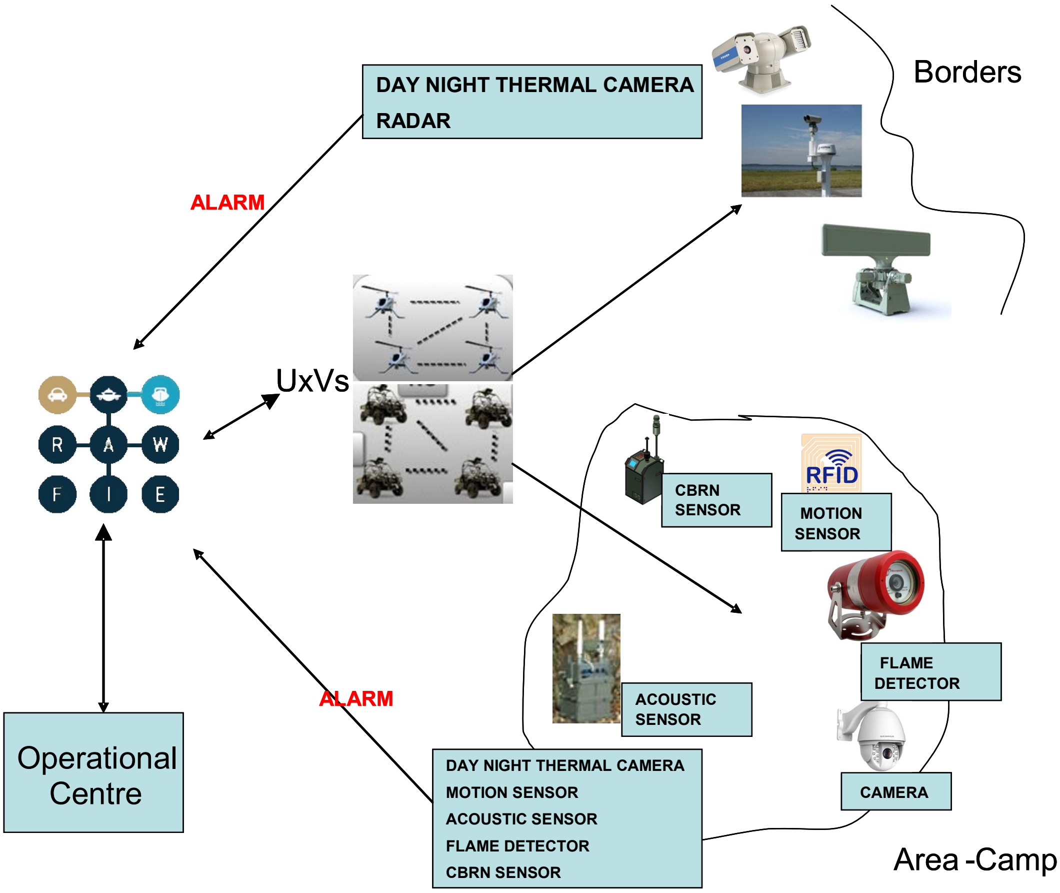


Figure 8: Border Surveillance or Perimeter protection of large areas scenario

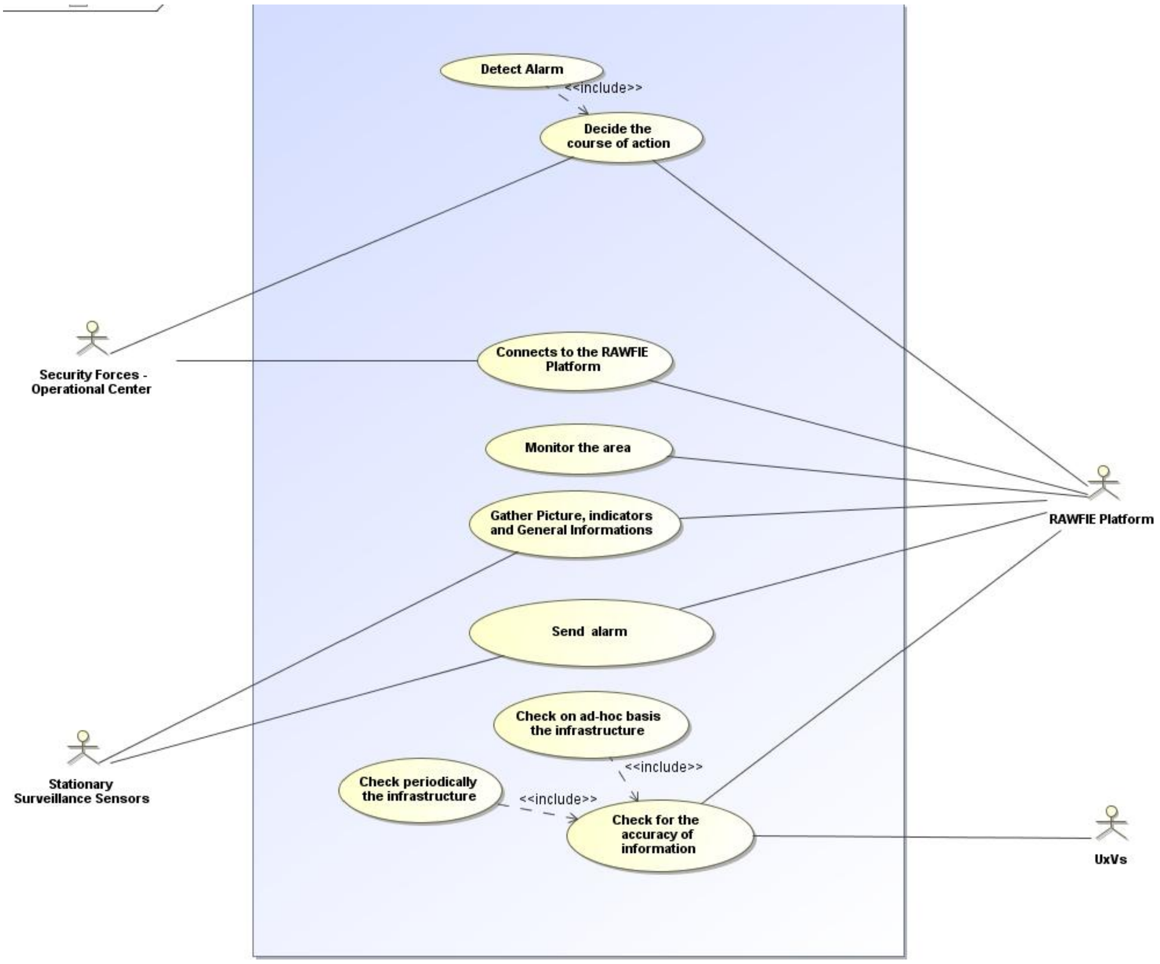

Detect Alarm
<<include>>
Decide the course of action
Security Forces - Operational Center
Connects to the RAWFIE Platform
Monitor the area
Gather Picture, indicators and General Informations
RAWFIE Platform
Send alarm
Check on ad-hoc basis the infrastructure
<<include>>
Stationary Surveillance Sensors
Check periodically the infrastructure
<<include>>
Check for the accuracy of information
UxVs

Figure 9: Use case scenario for scenario 2

Analytic Description

According to the scenario, security forces and commanders, which are located at an operational center, have an overview of the monitored area through the RAWFIE platform. Picture, indications and generally information, coming from the stationary surveillance sensors (cameras, radars etc) which are deployed along the borders or the perimeter of a crucial infrastructure, are gathered and analyzed by the platform. On a periodic or ad hoc basis, the collected information is checked for accuracy by using UxVs.

In case of an alarm for a potential threat or intrusion detected by the security sensors (fire detector, acoustic sensor, motion sensors, CBNR sensor etc.) or cameras, automatically UxVs are deployed by the platform to collect more specific information. The collected information should be reported to the operational centre in order to be assessed by people in charge and decide the course of action to face the threat (see Figure 8).

In the context of the above described use case, RAWFIE platform can be utilized to execute a series of experiments in order to assess and identify the optimum way to utilize available UxVs resources in order to perform the requested task of extending the monitoring area of the stationary surveillance sensors or monitoring a sensitive area after an alert given from security sensors. Indicatively, the following experiments can be performed:

- Optimization of the used UxVs in relation to the length of the border/perimeter.
- Optimization of the UxVs search pattern.
- Optimization of the UxVs used sensors in relation with the detected violation.

Type of sensors on the UxVs could be:

- Day/night thermal cameras
- Radars
- CBNR sensors
- Acoustic sensors

The collected data could be:

- Live streaming of the monitoring area
- Snapshots of the target or monitoring area, damage on infrastructure
- Pollution due to CBNR attack.

3.3.3 Scenario 3 – On demand deployable Internet facilities

Overview/Rationale

The scenario tackles the rapidly expanding domain of on-demand deployable Internet facilities through UxVs. An illustrative example could be the provision of broadband connectivity to remote locations without such communication capabilities or to areas affected by phenomena/natural disasters like earthquakes, floods etc. where these moving Wi-Fi hotspots will provide Internet in an underdeveloped or semi-urban environment. Imagine a UAV (or swarm of UAVs) overflying the area. Each UAV will offer “an Access Point (AP) like functionality” and will provide Internet connectivity and services to the local population (permanent residents, crisis management groups, etc.). The unmanned systems could form a multi-hop network in order to relay traffic to and from fixed infrastructure that has not been impacted. The same architecture could be based either on other types of UxV (e.g., a USV could provide connectivity to small islands), or UAVs that have landed on suitable locations (thus conserving energy) or even their collaboration in mixed formations.

Potential end user of this scenario can be an Internet Service Provider (ISP) or Search and Rescue (SAR) teams in case of emergency.

In order to accomplish this difficult undertaking task such devices need to be extremely energy efficient (possibly solar-powered) and to operate for long without external intervention. This market is expected to bloom in the coming years as more and more players are attracted to it (e.g., Facebook + Ascenta). Similar architectures are currently being investigated in the US by Google after acquisition of Titan Aerospace[4] or in the literature ([4], [5]).

Picture(s)

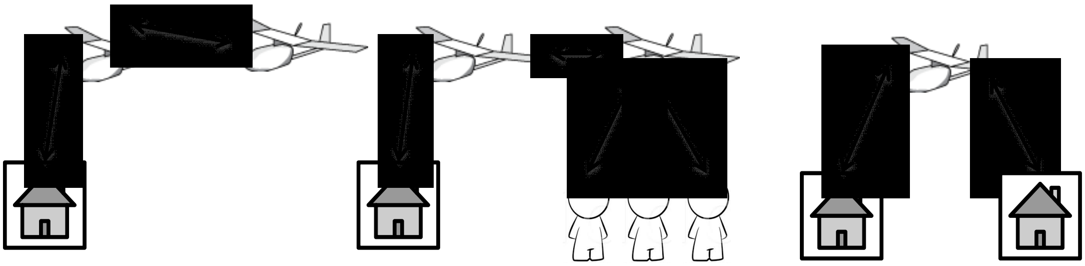


Figure 10: On demand deployable Internet facilities scenario

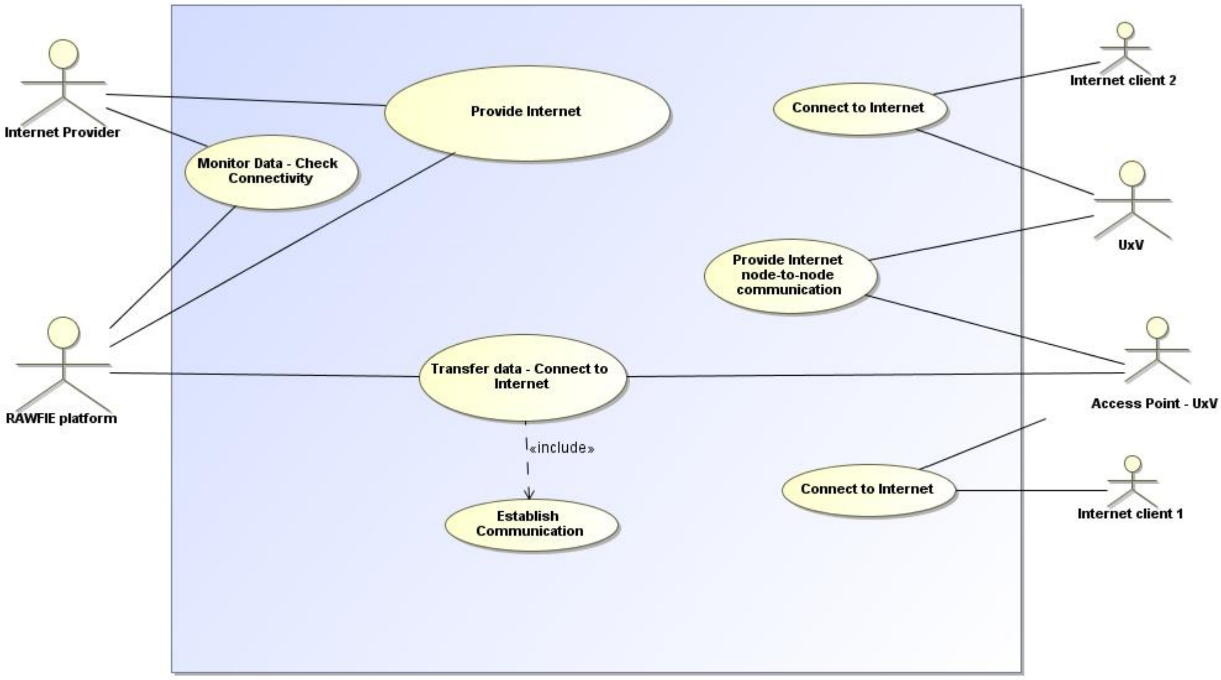


Figure 11: Use Case diagram for scenario 3

[4] http://techcrunch.com/2014/04/14/google-acquires-titan-aerospace-the-drone-company-pursued-by-facebook/

Analytic Description

The technical crew of local SAR team releases a number of UxVs at initial places in the area. The UxV can either "patrol" over/on a specific area or lock-on to the signal of the portable devices carried by each SAR Team. The goal of each UxV will be processed through the RAWFIE platform in order to prioritize needs, maximize data volumes circulated or to conserve its batter lifetime by taking off-line devices that are not needed for the task at hand. Various tasking strategies can be explored and compared accordingly.

The main "backbone" network (in essence, those UxVs that are closer to the source of the Internet feed) can be augmented by implementing Channel Bonding like techniques through the addition of more UxVs in the vicinity. In a similar manner, underexploited UxVs can be either switched-off or put to use by intelligent network routing protocols.

Experiment outcome:

- Scheduling algorithms for the efficient tasking of UxVs
- Percentage of time of network coverage/usage per device
- Network routing protocols than take into account remaining usage

3.3.4 Scenario 4 – Exploration & Assessment of Network Technologies Robustness

Overview/Rationale

This scenario deals with the assessment and benchmarking of network technologies like protocols & low level interfaces in various types of environments. The aim is to assess networking performance and robustness with respect to certain parameters and factors (i.e., communications range, throughput, error distribution) and support a subsequent exploration analysis for identifying the best deployment per case basis.

Potential end users of this scenario can be networking manufacturers.

Picture(s)

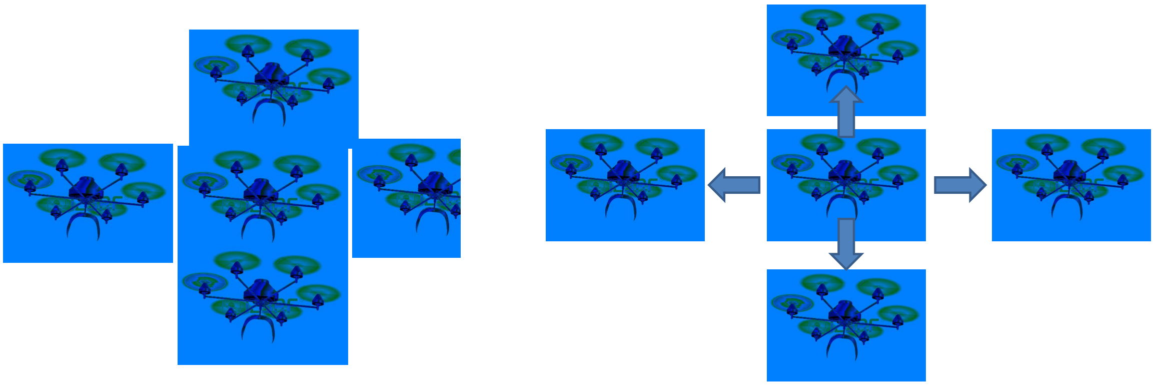


Figure 12: Exploration & Assessment of Network Technologies Robustness scenario

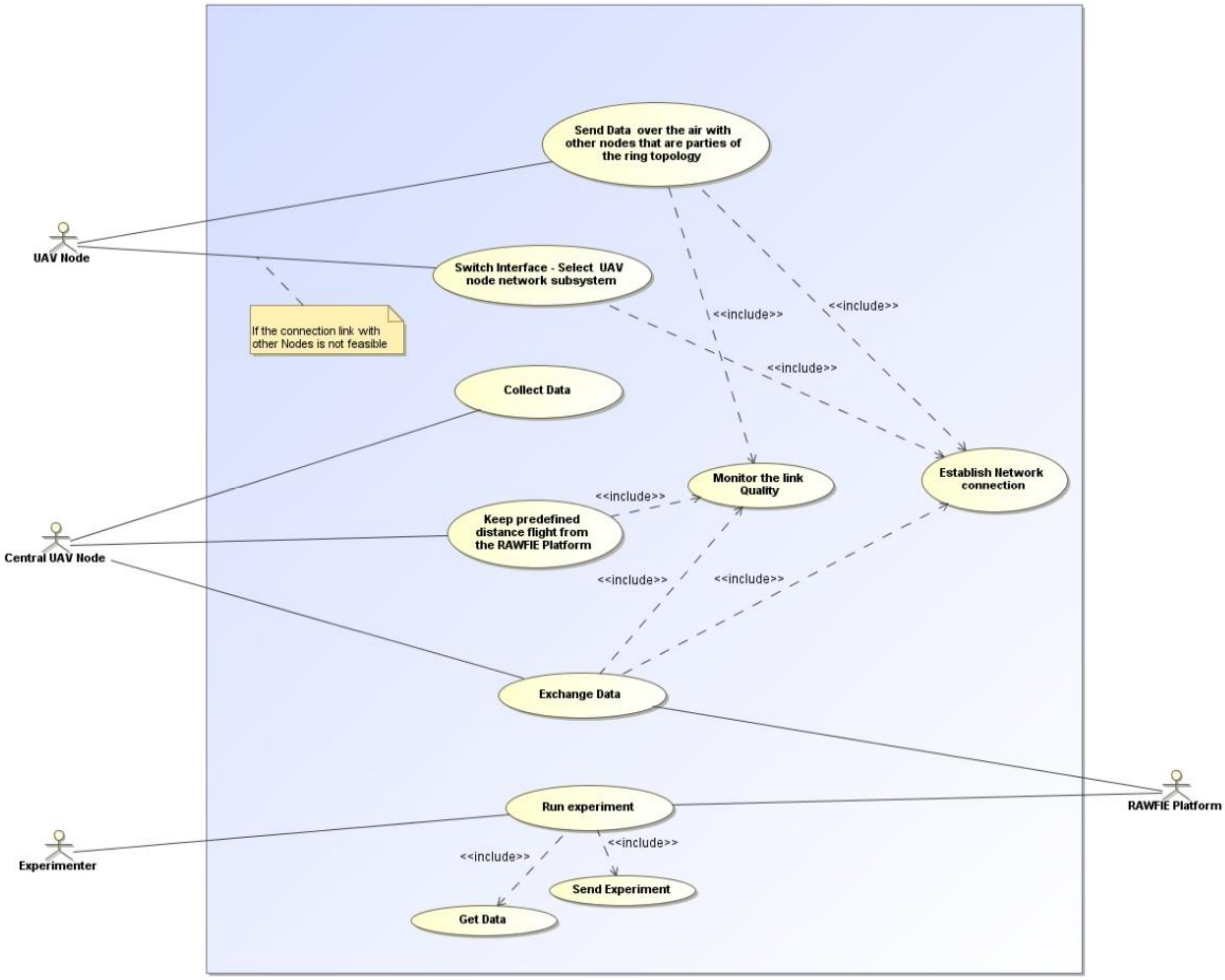

Send Data over the air with other nodes that are parties of the ring topology
UAV Node
Switch Interface - Select UAV node network subsystem
If the connection link with other Nodes is not feasible
<<include>>
<<include>>
<<include>>
Collect Data
Monitor the link Quality
Establish Network connection
<<include>>
Keep predefined distance flight from the RAWFIE Platform
Central UAV Node
<<include>>
<<include>>
Exchange Data
RAWFIE Platform
Run experiment
Experimenter
<<include>>
<<include>>
Send Experiment
Get Data

Figure 13: Use Case diagram for scenario 4

Analytic Description

The prime type of experiment could be related to *network protocol benchmarking*. Imagine a RAWFIE setup where a number of nodes (UxV) diverge from a common location (point). Let us assume a number of UxV in the order of 5 (=N+1). The UxV are interconnected by means of a device-to-device interface. Whenever, the N+1 UxV are close to each other, communication between is trouble-free. The N UxV are instructed to diverge (leave the mustering point at prearranged velocities – expanding ring). The onboard systems communicate with the central node (which remains static) and exchange information. Data collected can be post processed by the RAWFIE platform in order to draw conclusions regarding networking performance indicators such as throughput and error distribution with respect to communications range.). An interesting case is this particular experiment is the assessment of the nodes' capability to switch to alternative network interfaces whenever need arises. Ideally, nodes should be able to seamlessly communicate irrespective of their distance and network i/f involved. Switching between interfaces could cause some time of inactivity at higher layers or even disconnection. In this case, UxVs, when in relative proximity communicate with a short range interface which fails at some point in time. When communication conditions deteriorate, devices switch to medium range connectivity (either autonomously or instructed by ground control) and the impact on higher layers is assessed (e.g., disconnection of large file transfers involving applications – OTA).

Experiment outcome:

- Distributions of errors, SNR, throughput
- Communication disconnections

Another type of experiment may involve the assessment of robustness of low level network components (e.g. radio i/f, ECC implementations etc.) in diverse environments where line of sight communication does not exist. The UxVs could be instructed to communicate behind obstacles (e.g., hills, within forests or through canyons). In this type of experiment, delay tolerant capabilities of the network protocol in use can also be explored. In this experiment, a group of devices may be instructed to remain in static formation (i.e., maintain fixed distances between devices) but the whole group moves at areas where such obstacles are present. The collected information (see the list in the expanding ring case) is recorded and subsequently geo-referenced to assess the exact impact of potential obstacles and the environment as a whole.

Experiment outcome:

- Percentage of network coverage vs geospatial obstacles
- Algorithms for maximization of network coverage based on
  - Obstacles
  - Number of UxVs
  - Computation needs and battery lifetime

3.3.5 Scenario 5 – Efficient Coordination for phenomena or mission coverage

Overview/Rationale

This scenario deals with the efficient coordination of multiple UxVs for the purpose of covering certain phenomena (e.g. fire spreading in an area) or executing a certain sensing mission (e.g. mapping or scanning of an unknown area). The purpose is to explore various issues and strategies that can be adopted for intelligent coordination and control of multiple devices while minimizing resources consumption. In such experiments, sensor placement is changed from the possibly uniform and phenomenon-agnostic strategies to a fully distributed intelligent scheme.

Potential users of this scenario could be National Fire Brigades, Command and Control Centers (C&C) as well UxV manufacturers.

Picture(s)

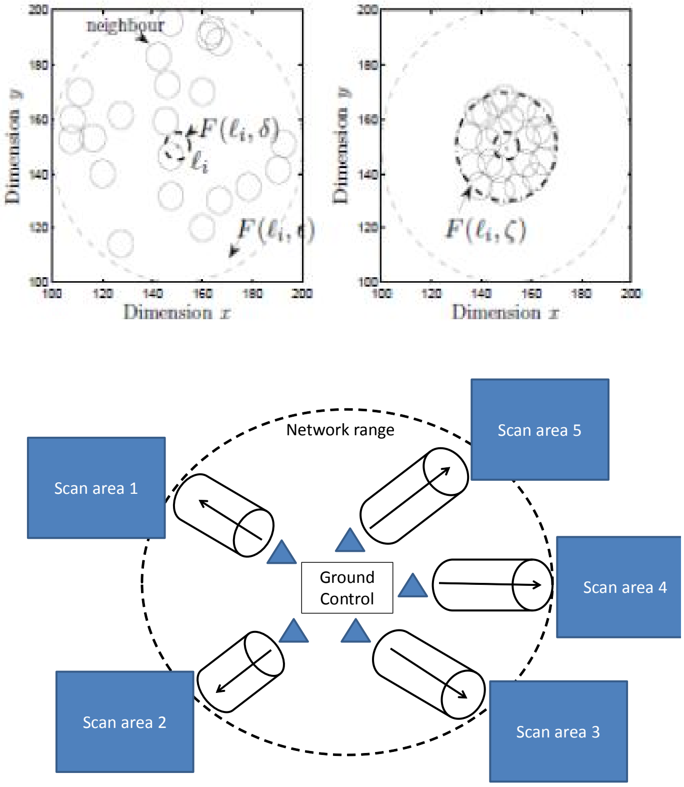


Figure 14: Efficient Coordination for phenomena or mission coverage

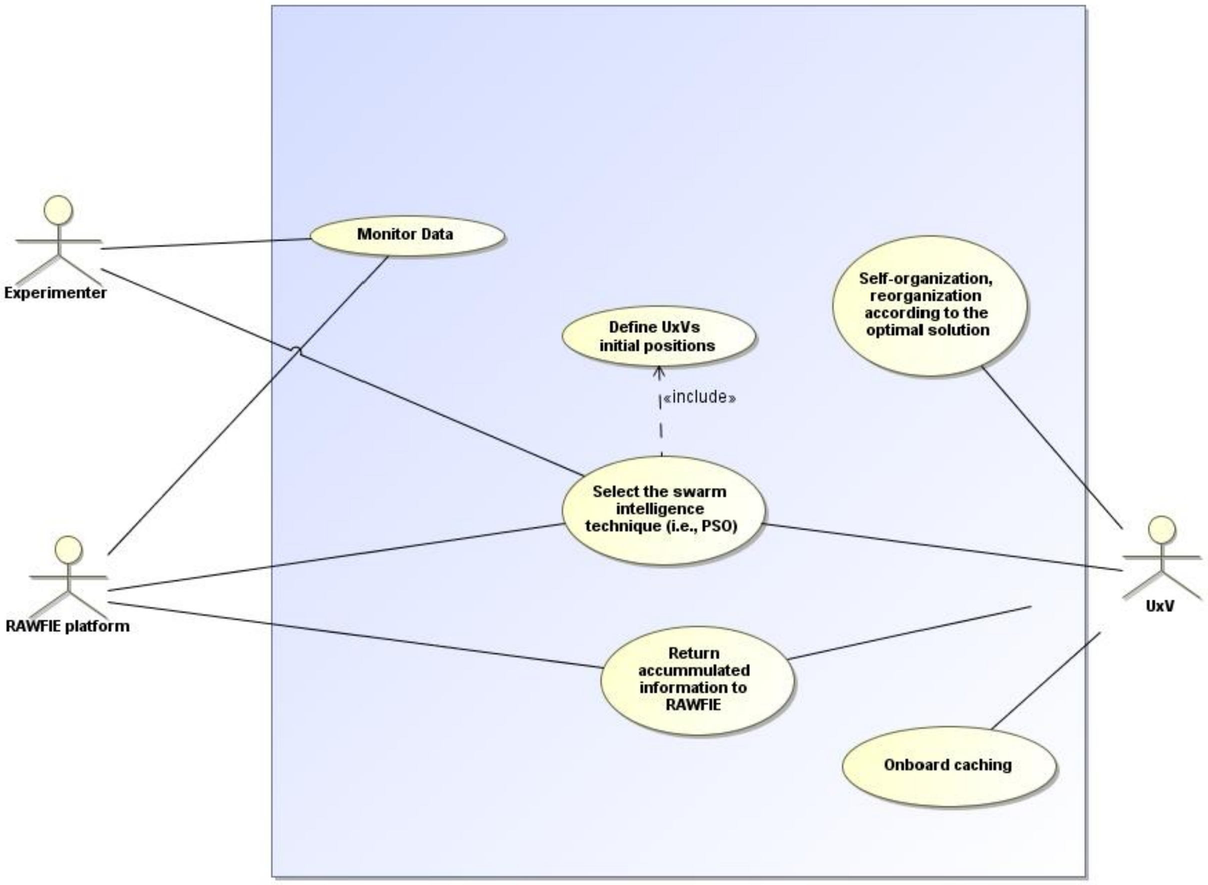

Experimenter
RAWFIE platform
UxV
Monitor Data
Define UxVs initial positions
«include»
Self-organization, reorganization according to the optimal solution
Select the swarm intelligence technique (i.e., PSO)
Return accummulated information to RAWFIE
Onboard caching

Figure 15: Use Case diagram for scenario 5

Analytic Description

A possible experiment may include multiple UxVs, grouped in a swarm and coordinating internally for maximizing coverage of a forest fire event. UxV nodes perform self-organization and reorganization to more accurately capture the spatiotemporal development of the phenomenon. Techniques like i.e. Particle Swarm Optimization (PSO) or Multi Swarm Optimization (MSO) can be adopted by the UxV for their optimal placement. Energy expenditure and node traveling distances are taken into account in these experiments to prolong UxV operational lifetime. The coordination scheme should react promptly to the occurrence of new incidents.

Experiment outcome:

- Spatial coverage
- Algorithms minimizing resources consumption

- Reactivity (i.e., time taken from initial detection to attain max coverage)

In another possible experiment, coordination and onboard data caching capabilities can be used to perform a collaborative mission. Imagine a sensing task which involves multiple UxV at a distant location (without network coverage). The UxVs negotiate a scanning plan within the swarm and set out for the scanning area. Following completion of the sensing task the UxV return to network reachability and communicate cached measurements to ground control. The performance of onboard data caching mechanisms can be investigated through this particular class of experiments. Further studies in the direction of DTN research can be also foreseen. Metrics related to the freshness of information and possible correlation to energy expenditure can be introduced.

Experiment outcome:

- Coverage
- Information quality (freshness, etc.)
- Communication quality to ground control

The above experiments can be extended to cover the case where the accumulated information (sensor readings) must be returned to ground control through a multi-hop network formed by the UxVs. Assuming that the scanning process caused battery depletion in the UxV, the device can remain stationary (e.g., flight termination) within the scanning area and exploit their remaining energy to transmit in low power towards ground control or another UxV acting as rely.

3.3.6 Scenario 6 – Over the Air (OTA) UxV Re-programming

Overview/Rationale

The scenario deals with network-assisted programmability of devices. Over the Air programming is a technique that is widely used in the mobile world for performing firmware or software updates mainly of cell phones. Extending this capability to the world to devices with strict real time characteristics such as the UxVs is quite a challenging task. A simple scenario involves the transmission and hot/cold installation of mission or operational related code from the ground control station (over-the-air, OTA, programming).

Picture(s)

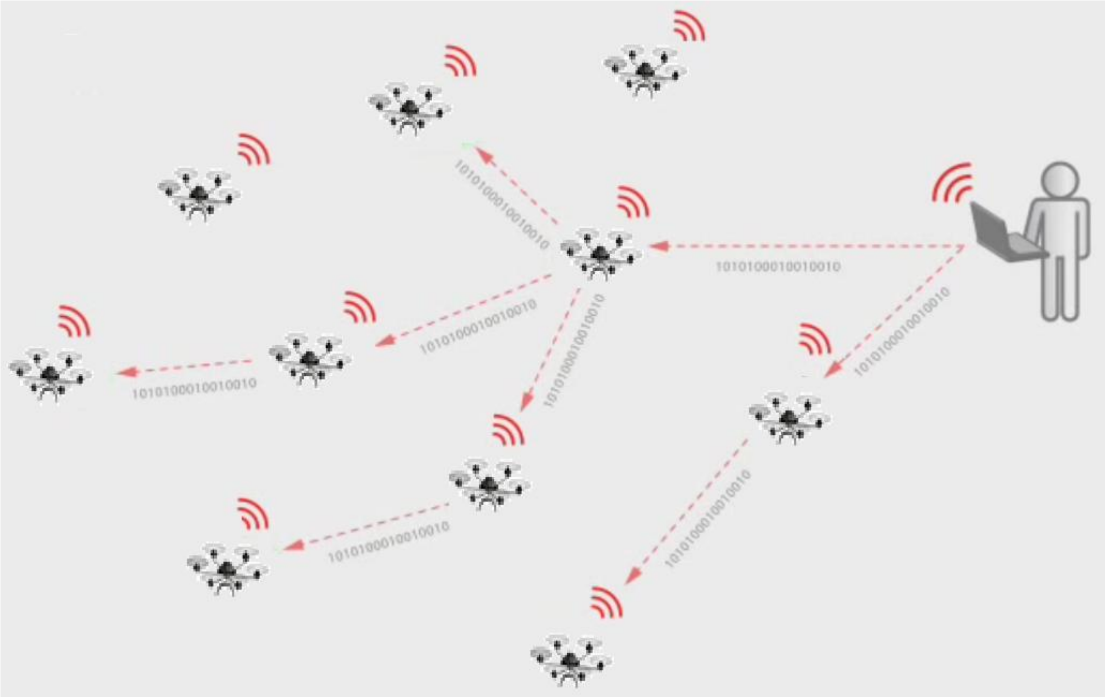


Figure 16: Over the Air (OTA) UxV Re-programming

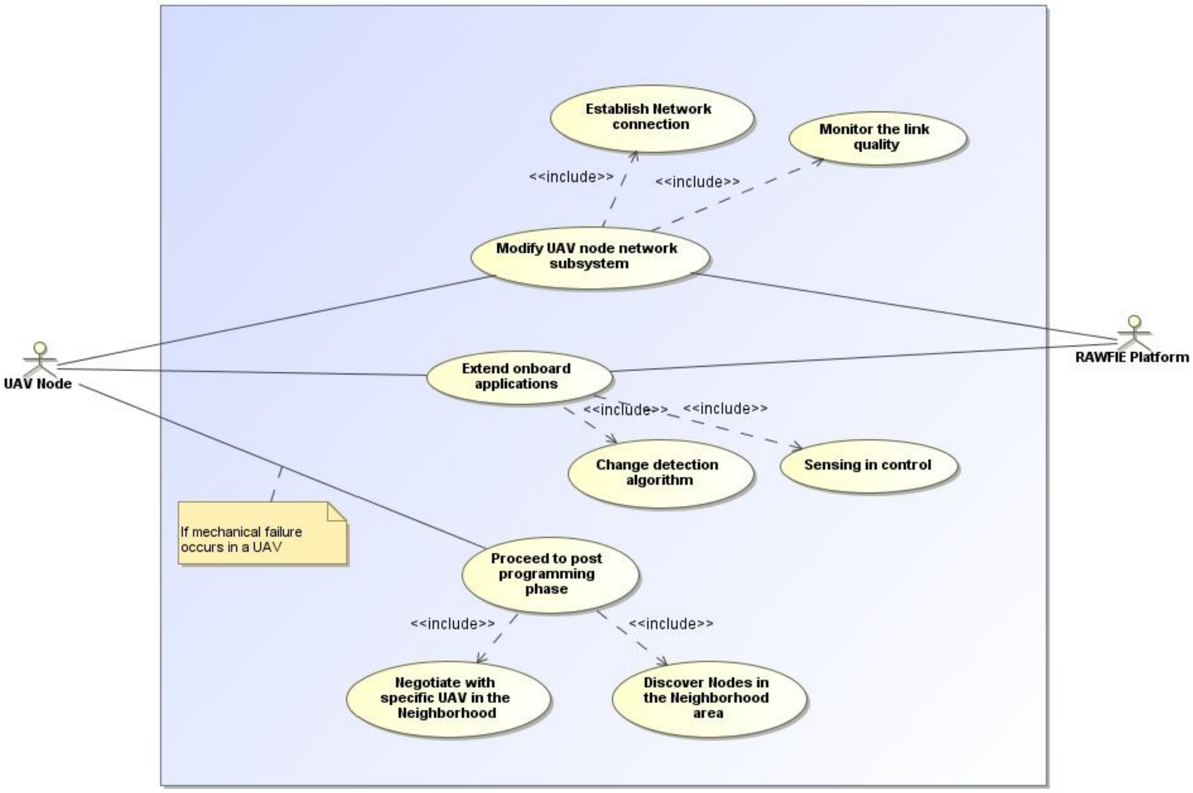

Establish Network connection
Monitor the link quality
<<include>>
<<include>>
Modify UAV node network subsystem
UAV Node
RAWFIE Platform
Extend onboard applications
<<include>>
<<include>>
Change detection algorithm
Sensing in control
If mechanical failure occurs in a UAV
Proceed to post programming phase
<<include>>
<<include>>
Negotiate with specific UAV in the Neighborhood
Discover Nodes in the Neighborhood area

Figure 17: Use Case diagram for scenario 6

Analytic Description

Possible experiments in this direction include the deployment and operation of a new networking protocol (or a complete stack) which may work in parallel to the previous one or perform a full substitution. Evidently, this reprogramming capability needs to be performed error-free and system operation disruption / malfunction should not be experienced. Apart from the networking functionality, the OTA capability may be used in the direction of extending onboard applications that relate to sensing and (in-network) post-processing of sensor readings. A typical scenario is the change of detection algorithms that receive and process the onboard camera feed. Such a feature may require an onboard application supervision module which has complete control over the OTA staging area and can invoke specific methods (callback) in the involved applications (e.g., termination, startup).

Another interesting experiment involved the negotiation between UxV devices while operational and without the supervision of ground control. A member of a swarm that performs a certain task negotiates with another member the offloading of a demanding task (e.g., a fusion process). This offloading process is triggered by a mechanical failure in the former device or energy constraints (device is about to cease operation). Hence the OTA case can be performed either in “hub” mode (i.e., coordinated by ground control) or in P2P mode.

Experiment outcome:

- Algorithms for offloading tasks
- Extending on-board resource capabilities
- Reserve energy

4 Project Constraints

The section provides information related to the testbed facilities and the UxVs resources that are to be used in RAWFIE. Section 4.1 presents brief information about each testbed including possible limitations and constraints that should be considered by potential experimenters. In section 4.2 regulations and standards for each UxV domain are listed wherever available or possible to be obtained. Moreover characteristics, constraints and limitations are given for specific UxV models that are to be provided by the existing UxV operators in RAWFIE (namely OCEANSCAN & ROBOTNIK). This information is expected to be augmented in future versions of the deliverable while additional UxV operators enter the consortium and provide their models.

4.1 Testbed facilities & constraints

3 testbed facilities are initially foreseen to be deployed in RAWFIE. These facilities are:

4.1.1 HAI testbed facility

The testbed facility that HAI will offer for the RAWFIE project is located in the company's industrial complex located in Tanagra around 65 km North of Athens. The testbed facility consists of a runway of around 300m length and 25 m wide which can be used for takeoff of wing or copter UAVs. The facility can support multiple simultaneous flying vehicles but due to the facts that the area is not a reserved test area, the National Regulations regarding UAVs should apply. The regulation poses limitations and the need for special CAA authorization for systems that exceed 25kg in weight. Even for system below that weight constraints are imposed regarding height of flight (<400 feet AGL), area of flight (VLOS is needed; certain distances must be respected from residential areas, military facilities or airports etc.). Insurance is needed in all cases (except for model aircrafts weighting less than 350 gr).

Additional constraints may apply due to the fact that a military airbase is located nearby. This includes:

- the RC frequencies to be used (i.e. no military bands but anyway they must be provided a priori so that a clearance is obtained)
- No ability for direct booking and experiment launching since clearance is needed by the security directorate prior to experiment execution (it may take few days)
- There might be limitations in the way sensors are used and in the sensor data retrieved (applies especially to optical sensors)

Currently HAI has no dedicated protected dock or hangar infrastructure for storage and maintenance of the testbed UAVs. The expected number and especially the dimension size of the UxVs must be clarified in order to see how to address this

4.1.2 HMOD testbed facility

HMOD can provide a number of testbed facilities which are located in the wide area of Attica region. In particular, testing tracks for USVs can be hosted in highly controlled sea areas in front of naval bases which are perfectly described in their characteristics and regulations by the Naval Base of Skaramangas (NW of Athens).

The Naval Base of Skaramngas is able to receive, inspect, launch and store USVs in its facilities. It provides military grade emergency services (i.e. crash, fire or rescue) and has the appropriate radar facilities and systems for tracking and surveillance. The potential test sea area is 2.000 m^2 with depth of 50m, appropriate for surface tests. Accommodation can be provided by hotels in Athens (approximately 15 km) with the provision of each experimenter.

For security reasons, at least one month prior to the executed test-experiment, the potential institution (experimenter) has to provide:

- Completed security clearance forms with data of the personnel, which will like to enter to the Naval Base
- The period of the experiment
- A description of the tests-experiments
- The equipment that will be used (number -dimensions of USVs, sensors etc).
- The infrastructures and consumables which are needed to perform the test (storage facilities, maintenance, fuels etc).
- The RC frequencies that are about to be used

The Naval Base's authorities, considering the submitted data and upon the availability of the testbed facility the requested dates, will issue a license for the use of its infrastructures.

During the activities inside the Naval Base, the hosted personnel will be escorted by a navy officer who will be their liaison and point of contact with the Naval Base's authorities.

Normally, the provided dock inside the Naval Base can facilitate up to 10 USVs (expected WxLxH – 1x2.5x1.5). In case of a different number or dimensions of the USVs, the possibility of facilitate the equipment and perform the test will be consider upon request.

4.1.3 PEGASE testbed facility

A number of testbed facilities are available via PEGASE (see Table 4). The main facility that PEGASE will offer is the CEEMA facility which is part of the Domaine du Planet complex. The facility is located in Pourrieres around 45 km North of Marseille international airport. The testbed facility consists of a runway of around 300m length and 20 m wide which can be used for takeoff of fixed wing or copter UAVs until 150 kg (class F)

The place is the property of a PEGASE member. It offer a private aerial area (1km wide / 2500 feet high), a large field (65 ha) for ROV test and a water area (4500 m2 and 5m deep) for boat and submarine tests. The SUV facility is available 7/7 days during all the aeronautics day. It also could be used during the night (if special demand).

The volume of the aerial area allows simultaneous UAV, ROV and UVS operations. Only the French frequencies band are possible but if some other frequencies are needed the authorization could be obtain (30 days)

Finally, the CEEMA is a resort offering all the facilities (hotel, restaurant, workshops, meeting room, etc.) just 250 meter from the runaway. This place can be rented in exclusivity if necessary.

Platform	Locating	The test-bed facility	Type of UVS	Number of simultaneous flying vehicles	Size max	Height max of flight	RC frequencies uses	Flying authorization	Sensors skill
Technopole Avignon	Avignon airport	runway of around 150m length and 15 m wide which can be used for takeoff of fixed wing or copter UAVs until 150 kg	UAS	Limited, max 3 UAS simultaneous	No limitation	150 m	To be defined	No ability for direct booking and experiment launching since clearance is needed by the security directorate prior to experiment execution (it may take few days)	No
CEEMA	Middle of Provence region near to Aix en Provence, about 40minuts by car around 45 km North of Marseille international airport	runway of around 300m length and 20 m wide which can be used for takeoff of fixed wing or copter UAVs until 150 kg (class F)	All	No limitation	No limitation	150 m	French frequencies	7/7 days Flight at night possible	Optics
Technopole Mer	Near Toulon, with a specific motorway exit	Platform test for equipment 2017							

Table 4: Characteristics of testbed facilities offered by PEGASE

4.2 UxV Constraints & Regulations

The RAWFIE platform seeks to provide an umbrella that will enable virtualization in accessing UxVs resources lying in different testbeds and in disparate geographical locations. The intention is to provide to the potential user/experimenter a common interface for controlling and monitoring these resources that is agnostic of the complexities and particularities pertaining to each UxV domain. . However when implementing the system it is important to be aware of the possible constraints that apply to these systems as well as to regulations and standards that may set limitations to their usage. In this section we will briefly list such constraints and limitations generally for each UxV type. We will also give specific details for each UxV model that partners considered as UxV providers will bring into the project. The latter information is of course subject to update in next versions of the deliverable and while new UxV players enter the consortium (usually as a result of the open call process.

4.2.1 Regulation & Standards

4.2.1.1 Unmanned Aerial Vehicles –UAVs

Currently the regulation is not consistent across Europe, even though a Steering Group is working on the harmonization of the Remotely Piloted Aircraft System (RPAS - European Remotely-Piloted Aircraft Systems Steering Group (ERSG) [13]). In 2014, this group issued a communication on the adopted strategy, entitled "A new era for aviation: Opening the aviation market to the civil use of RPAS in a safe and sustainable manner". This paves the way to a sustainable commercial exploitation of RPAS and the related resources, in accordance with the civil aviation authorities. In the meantime, the national regulations still frame the emergence of UAV activities and services.

UAV belong to five main categories: gliders, balloons, dirigibles, fixed-wing or helicopters. They are available in a great variety of sizes and other characteristics, such as speed, range, carried payload: material, sensors, cameras, etc. In the follow-up of this text, only fixed-wing and helicopters will be discussed, since they are the most popular and useful, from an industry point of view.

Fixed-wing UAVs are more suitable for covering large distances at higher speeds, while helicopters provide the flexibility of landing and taking-off without runway and hovering capabilities. Large UAVs (>150 kg) can deal with payloads up to 50 kg, while small UAVs only up to 2 kg. The size and mass difference also impact their speed and range, with short flights of 30 minutes at 40 km/h for small UAVs to more than 4 hours and up to 200 km/h for large ones. UAVs of a larger dimensions and weight (>25 kg) can better cope with strong weather conditions and be operated at wind speeds of up to 60 km/h as well as under moderate rain and snow.

The weight and payload are the major criteria used in categorizing UAVs. Large UAVs (>150 kg) fall under European Union regulations (EASA), while smaller UAVs (<150 kg) are treated under - currently often very restricted - regulations of the national aviation authorities. Depending on the applicable national regulations, UAVs often have to be operated in the visual line-of-sight (VLOS) of the operator, which represents a total range of 1 km (e.g. in France) or extended to 2 km (e.g. in United Kingdom – UK). We will detail hereafter the regulation below 150 kg in different representative European countries.

UK has two categories under 150 kg: less than 20 kg and between 20 kg and 150 kg. The UAV that are less than 20 kg do not need any airworthiness approval except for aerial work purpose; they do not need any registration, but the pilot needs to be certified (BNUC-S or equivalent) and they must stay away from people, property and congested areas. UAV over 20 kg needs airworthiness, pilot certification and they are prohibited at less than 150m away from people, property and congested areas.

In *Ireland*, the regulation is similar to UK, with the exception that it is forbidden to fly closer than 8km to an airfield, closer than 150 m to any person, vehicle and closer than 8 km to any aircraft. VLOS is also possible for UAV of more than 20kg and less the 150 kg at the altitude of less than 120m in a range of 500m.

In *Switzerland* only VLOS is allowed. Any UAV of less than 30kg are allowed without restriction except at a distance of less than 100m of any person or private property, less than 5km from airfields. Local restrictions may apply. Insurance is required for UAV of more than 0,5kg and less than 30 kg. Between 30 kg and 150 kg, VLOS requires permission.

In *Germany*, UAV over 25 kg are not allowed. For all other UAV, only VLOS in a range of 300m max is allowed, with minimum distance of 1.5 km from airfields. UAV less than 5kg are allowed without restrictions except for commercial purpose (LuftVO flight permit required). Between 5 kg and 25 kg, the LuftVO permit and certification are required.

France regulates under the basis of scenarios (S1 to S4), which combines UAV weights, maximum altitude, population flown over and type of control (VLOS, FPV[5]), and UAV type (A to G), which depends on weight and engine power. They are usable in specific scenarios, with or without airworthiness approval, registration or permit: more details are available at [22] Note that flight out of sight is prohibited. Tethered aerostat is the only allowed operation at night. Any UAV carrying a camera/video needs airworthiness approval. Outside of the predefined scenario, a flight plan must be declared and approved by the flight control authorities.

In Greece there is a distinction between RPAS used for aeromodelling and commercial or experimental usage (characterized as UAVs). The former ones are categorized based on their

[5] This type of control usually implies remotely piloted via an onboard camera, fed wirelessly to video goggles or a video monitor (e.g. smart phone)

weight (from 0-25 kg). VLOS is always required and based on increase of weight there are additional constraints imposed (i.e. height of flight, distance from habited areas, airports, military installations, insurance needed). For UAVs the landscape is not clear but in general the EASA rules apply and an authorization is always needed from CAA as well as insurance for potential damage.

In the *US*, the UAVs fall under the regulation edict produced by the FAA and NASA. It basically bans any commercial use of UAV.

Traditionally in every country, the military and other specially restricted zones, such as populated areas, airports, nuclear power plants are excluded from UAV operations. Insurance also imposes a strict framework on UAV use. For example in UK, VLOS[6], E-VLOS[7], B-VLOS suffer from specific limitations (basically permission is required in all cases, except for VLOS under 20 kg).

The situation of the UAV regulations is representative of those of all sorts of remotely operated maritime or terrestrial observation vehicle. They obey however to different regulations and constraints, since they are interfering differently with the human activities or the government attributions.

It must also be noted that since UAVs comprise a rapidly developing area of business and with their commercial use looming around there is a fast evolution of the involved regulations to address this matter. Therefore although the provided information represent the current status, it is likely to become rapidly outdated.

4.2.1.2 Unmanned Ground Vehicles –UGVs

Regulations in the field of UGVs, provided by official agencies such as ISO standards, are almost non-existent at the moment. Several ISO standards contain regulations, tests and safety criteria regarding robotic manipulators and industrial robots. These standards are listed below:

- ISO 8373:2012: Robots and robotic devices -- Vocabulary
- ISO 9283:1998: Manipulating industrial robots -- Performance criteria and related test methods
- ISO 9409-1:2004: Manipulating industrial robots -- Mechanical interfaces -- Part 1: Plates
- ISO 9409-2:2002: Manipulating industrial robots -- Mechanical interfaces -- Part 2: Shafts
- ISO 9787:2013: Robots and robotic devices -- Coordinate systems and motion nomenclatures

[6] Usually beyond 500m the operations are supported by additional technology

[7] Usually beyond 500m the remote pilot is supported by one or more observers (RPS crews maintains direct unaided visual contact with RPA)

- ISO 9946:1999: Manipulating industrial robots -- Presentation of characteristics
- ISO 10218-1:2011: Robots and robotic devices -- Safety requirements for industrial robots -- Part 1: Robots
- ISO 10218-2:2011: Robots and robotic devices -- Safety requirements for industrial robots -- Part 2: Robot systems and integration
- ISO 11593:1996: Manipulating industrial robots -- Automatic end effector exchange systems -- Vocabulary and presentation of characteristics
- ISO/TR 13309:1995: Manipulating industrial robots -- Informative guide on test equipment and metrology methods of operation for robot performance evaluation in accordance with ISO 9283
- ISO/FDIS 13482: Robots and robotic devices -- Safety requirements for personal care robots
- ISO 14539:2000: Manipulating industrial robots -- Object handling with grasp-type grippers -- Vocabulary and presentation of characteristics
- ISO/DTS 15066: Robots and robotic devices -- Safety requirements for industrial robots -- Collaborative operation
- ISO/CD 18646-1: Robots and robotic devices -- Performance criteria and related test methods for service robot -- Part 1: Wheeled mobile servant robot

4.2.1.3 Unmanned Surface Vehicles (including Autonomous Underwater ones) –USVs & AUVs

There’s no special legislation considering the sailing of USVs, the Unmanned Vehicles will follow the national applicable rules of safe navigation and maritime traffic, provided by the legislation of the typical sea vessels.

4.2.2 Specific Model Characteristics & Constraints

This section describes the characteristics of specific UxV model’s that are provided by UxV providers and will be available for use in the initial RAWFIE testbeds. For the first iteration only models provided by MST and ROBOTNIK will be available. Below details are given per UxV type and model.

4.2.2.1 Unmanned Aerial Vehicles –UAVs

Currently there are no UAV suppliers or manufacturers in the consortium. The table below summarizes desired characteristics (not strictly imposed though) for UAVs to be used in the RAWFIE project.

Launch type	using catapult,vertical, wheeled
Recovery type	vertical (autonomous, use of parachute), wheeled
Endurance	> = 30 min
Mission Radius	>= 500 meter (depends on UAV type and scenario also)
Ceiling	50 m to 2 km
Weight	<= 100 kg

Payload	>= 0.5 kg
Side Wind Endurance	until 14 m/s
Contrary Wind Endurance	until 4 m/s
Operational Conditions	Day/Night
Operational Temp	-15°C until 45 °C
Guidance	Manual and pre-defined flight plan by autopilot
GPS localization	GNSS (GPS, GLONASS,GALILEO,EGNOS etc) Possibility to use alternative system (INS, hybrid INS+GNSS)
Guidance capabilities	• Manual and Pre-defined flight plan by autopilot • Possibility to change the flight plan when the UAV is at the ground or during the flight • In case of losing contact with the base, ability to return to a predefined area.
Transmitting data Specifications	• Range of live streaming transmission (video, picture, sensor's indicators) > 2 km • Automatic wireless connection between UAV –Control Center
Other	On board data cashing

Table 5: Expected characteristics of RAWFIE project UAVs

4.2.2.2 Unmanned Ground Vehicles –UGVs

Info below is based on example models provided by partner ROBOTNIK (UGV supplier).

	SUMMIT XL ROBOT PLATFORM	**TURTLEBOT 2 NUC**	**TURTLEBOT**
Main Computer	MITAC Desktop board J1900 (Intel Celeron, 2Ghz processor)	Intel NUC i5	Standard netbook Ubuntu compatible
Main use	outdoors/indoors	indoors	indoors
Main applications	surveillance, remote monitoring, teleoperation	high performance indoor applications	Navigation, localization and mapping based on Kinect (Slam 2d and 3d)
Gear	4 motorwheels with included encoders	2 wheels +castor	2 wheels +castor
Steering	skid-steering		
Sensor 1	HOKUYO URG04	Microsoft Kinect	Microsoft Kinect
Sensor 2	Microsoft Kinect	Odometry	Odometry
Sensor 3	PTZ Camera: 360º pan with Auto-flip, 12x optical zoom, Day/night, H.264	Gyro (100 deg/s)	Gyro (100 deg/s)
Sensor 4	3DR GPS uBlox LEA-6	Bumpers, cliff sensors, wheel drop sensors: left, center, right	Bumpers, cliff sensors, wheel drop sensors: left, center, right
Dimensions	722x610x392		
Weight	45 kg	5 Kg	5 Kg
Speed	3 m/s	65 cm/s	65 cm/s
Control Software	open architecture ROS	open architecture ROS	open architecture ROS
connectivity	Internal: USB, RS232, GPIO y RJ45	Power connectors: 5V/1A, 12V/1.5A,	Power connectors: 5V/1A, 12V/1.5A, 12V/5A

		12V/5A	
connectivity	External: USB and power supply 12 VDC	3 x touch buttons, 2 programmable 2 colored leds	3 x touch buttons, 2 programmable 2 colored leds
Communication	Long Range Connectivity with Wi-Fi 802.11n Short Range Connectivity by Bluetooth (10 meters range)		
Autonomy	180 minutes	up to 7h	up to 7h

Table 6: Configuration of UGV Systems supplied by ROBOTNIK

Below some additional constraints and limitations related to vehicles movement, their control system and the equipment capabilities are listed. The figures are based on provider's experience.

SUMMIT XL ROBOT PLATFORM

- Max speed 3 m/s
- Max Slope Angle 45º
- Min Round Radius: 50 cm
- Suspension allows robot to manage hard roads (Outdoor/Indoor purposes)
- Carrying capacity 20 kg
- Protection Class IP54
- LiFePo4 Batteries provide 180 minutes of Autonomy
- Easy Wheel Change, Omni-directional Wheels Available
- Embedded Linux OS
    - ROS Control Architecture. Robot Control and Data can be brought to other systems since The ROS middleware provides these facilities:
        - publish/subscribe anonymous message passing
        - recording and playback of messages
        - request/response remote procedure calls
        - distributed parameter system
- Localization by GPS with 3drUblox LEA6
    - update rate 5Hz
    - Accuracy 2.0 m CEP
    - Acquisition: Cold Starts 26s. Hot Starts 1s.
    - Sensitivity -148 to -162 dBm
- Hokuyo Laser Sensor for indoor area scanning:
    - detection area 240º,
    - max distance 4000 mm

TURTLEBOT PLATFORM

- Indoor purpose.

- Max speed: 700 mm/sec, 180 deg/sec.
- High resolution wheel encoder (11.7 ticks/mm) and 3-axis gyrometer for precise localization (100deg/s).
- 3 h of Autonomy or 7 h with extended battery.
- Automatic charge with charging station.
- Up to 5 kg carrying capacity.
- Overcomes obstacles 12 mm high.
- Dimensions: 31,5x43x34,7 cm, weights 5 kg
- Security bump sensors and wheel drop sensors prevent collision and falls.

It should be noticed that these capabilities are related to the actual equipment of the vehicles as referred to in the Specific Model Characteristic collected in Table 6. There's a wide range of additional sensors and equipment improvements that could be implemented upon request, i.e. more powerful processors up to i7 latest releases or longer range laser sensors.

ROBOTNIK has developed a great amount of ROS architecture Drivers for controlling these sensors within the ROS system.

As a conclusion, the main limitations of these platforms are related to movement regarding typical indoor scenarios for the TURTLEBOT platform and the need of communicating via the ROS middleware in order to control robot functionalities.

Further limitations and constraints may apply based on the final configuration of the platforms.

4.2.2.3 Unmanned Surface Vehicles (including Autonomous Underwater Vehicles) –USVs & AUV

Info below is based on example models provided by partner MST-OCEANSCAN (USV-AUV supplier).

	USV #1	USV #2	LAUV
Main Computer	CPU: Single Core ARM Cortex-A8 @ 1GHz RAM: 1 GB Storage: 64 GB (Solid State)	CPU: Single Core ARM Cortex-A8 @ 1GHz RAM: 1 GB Storage: 64 GB (Solid State)	CPU: Single Core Geode LX 800 @ 500 MHz RAM: 1 GB Storage: 16 GB (Solid State)
Auxiliary Computer	CPU: Dual Core Atom @ 1.33 GHz RAM: 1 GB Storage: 128 GB (Solid State)	CPU: Dual Core Atom @ 1.33 GHz RAM: 1 GB Storage: 128 GB (Solid State)	CPU: Single Core ARM Cortex-A8 @ 1GHz RAM: 1 GB Storage: 64 GB (Solid State)
Dimensions	1x2.5x1.5 meters (WxLxH)	1x2.5x1.5 meters (WxLxH)	

Weight	<120 kg	<120 kg	
Speed	2.2/m/s (4knots)	2.2/m/s (4knots)	
Camera	IP Camera, H.264, 720p HD, 30 FPS, Pan Control	IP Camera, H.264, 720p HD, 30 FPS, Pan Control	
GPS	56-channel GNSS	56-channel GNSS	56-channel GNSS
Thrusters	Two	Two	One
Steering	Differential Thrust	Differential Thrust	Four Independent Fins
Single Beam Echosounder	Frequency: 675 kHz Maximum Range: 50 m	-	-
Multi Beam Echosounder	-	Frequency: 260 kHz Beams: 480 Maximum Range: 100 kHz	-
AHRS	MEMS	MEMS	MEMS
Weather Station	Temperature, Pressure, Humidity, Anemometer	Temperature, Pressure, Humidity, Anemometer	-
On-Board Software	DUNE	DUNE	DUNE
Command & Control Software	NEPTUS	NEPTUS	NEPTUS
Communication Protocol	IMC	IMC	IMC
Radio	802.11bgn 2.4 GHz / 802.11ac 5 GHz	802.11bgn 2.4 GHz / 802.11ac 5 GHz	802.11bgn 2.4 GHz
Navigation Lights	Four (Green/Red/White/White)	Four (Green/Red/White/White)	Four (Green/Red/White/White)
Autonomy	2-4 hours	2-4 hours	4-8 hours

Table 7: Configuration of USV Systems supplied by MST OCEANSCAN

Below some additional constraints and limitations related to vehicles movement and their communication capabilities are listed. The figures are based on provider's experience.

- In the case of the standalone AUV the maximum radio range measured in the field was around 1 km.
- In the case of the standalone USVs the maximum radio range measured in the field was around 2.5 km.
- Bandwidth near the maximum range is around 200-400 KiB/s.

When the AUV is coupled with a USV the maximum radio range will be that of the USV. These results assume an access point radio at a height of approximately 3 meters. Note than when the AUV is submerged, radio communication is not feasible due to the high attenuation of radio waves in water.

5 Functional Requirements

5.1 The Scope of the Work

RAWFIE is expected to leverage a multi-tier design pattern in order to facilitate the implementation of a highly and easily extensible experimental platform accessible via the internet. The functionalities for the presentation of the information to the experimenters, the implementation of the core parts of the business logic and the software interfaces for the integration of the different modules along with the data persistence should be separated in different tiers. A bird eye view of RAWFIE will reveal a system that consists of a central entity, the *RAWFIE platform*, possibly residing in a distributed cloud infrastructure and peripheral islands of UxV resources that interact with the Platform via a gateway server. These islands of resources together with their respective gateway constitute the *Testbed* entities. These two entities: *Platform* and *Testbed* drove the top level classification of identified requirements. Further sub-categories were introduced in order to facilitate reading as well as aiming in assisting the work of system designers in allocating requirements to architecture elements.

5.2 Platform Requirements

The term Platform refers to the middleware solution responsible for managing and monitoring the lifecycle of an experiment in the context of the RAWFIE system. An experiment's lifecycle can be analyzed in a number of major phases that include: authoring, booking, launching and evaluation of an experiment. The Platform provides also an appropriate UI and infrastructure services for user management, data storage and testbed management.

Based on the above considerations, an attempt was made to classify the requirements identified at platform level, according to the usage lifecycle phase they belong to. The proposed categories are depicted in Figure 18. Besides the experiment's related phases an additional preparation phase was introduced. This category is not directly related to an experiment. It includes requirements needed before even writing an experiment related i.e. to management of users or data. Finally, a few generic requirements that do not fit into any of the introduced lifecycle phases were classified as common aspects.

A summary table of all platform functional requirements is provided in the Annexes sections.

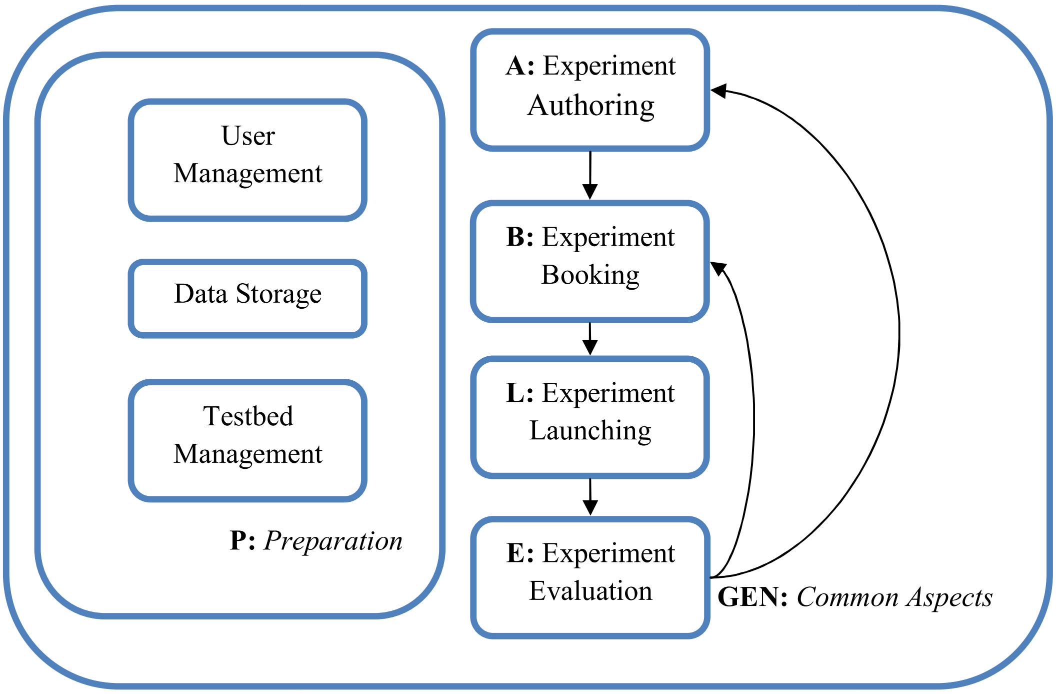

User Management
Data Storage
Testbed Management
P: *Preparation*
A: Experiment Authoring
B: Experiment Booking
L: Experiment Launching
E: Experiment Evaluation
GEN: *Common Aspects*

Figure 18: RAWFIE usage lifecycle phases

5.2.1 Generic Requirements

Id:	PT-GEN-001	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	A web portal interface shall be provided to the users of the platform								
Description:	The RAWFIE web portal shall provide a user-friendly Graphical User Interface (GUI), acting as a central point of access to all the necessary resources and services used by the experimenters.								
Additional Info (comments):	The GUI should be based on W3C standards.								
Related Scenario(s)									

Id:	PT-GEN-002	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	RAWFIE platform shall support various roles with different privileges at every level of access.								
Description:	The platform shall provide a set of different roles with predefined privileges. Every platform user should be assign to a role. At least the following roles shall exist: • Experimenter • Admin • Testbed Operator each of them providing different access rights to the various platform services. Definition of additional roles should be possible.								
Additional Info (comments):	The user's role must be validated in every service access request This is important as some of the experiment data could be highly sensible (this depends on the types of experiments) and the testbeds with their UxVs can be quite expensive								
Related Scenario(s)									

Id:	PT-GEN-003	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	Web portal usage shall be allowed only to authenticated users								
Description:	An experimenter should firstly be registered by creating an account through the portal (initial sign up). Access to the portal functionality shall be allowed only after the information is reviewed and approved by a RAWFIE administrator. Single sign-in authentication (login) process should be provided.								
Additional Info (comments):									
Related Scenario(s)									

Id:	PT-GEN-004	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:	RAWFIE middle tier shall include a module to monitor the performance of the middle tier components.								
Description:	This module will check the performance of the middle tier components by utilizing Key Performance Indicators (KPI) and this way ensure that all critical software modules will perform at optimum levels.								
Additional Info (comments):	Indicators could be: utilization of the different components, response times of the data storage, availability of the web services etc.								
Related Scenario(s)									

5.2.2 Preparation Phase

Id:	PT-P-001	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	The UI interface shall illustrate ALL the essential information of the RAWFIE federation that the experimenters should take advantage of								
Description:	Essential information provided shall include at least: • Testbed facilities information • Available experiments • List of simulation tools								
Additional Info (comments):	The information pertaining to each testbed must be presented in a uniform way. It should include at least: • UxV resources info (available and total) • Testbed particular constraints								
Related Scenario(s)									

Id:	PT-P-002	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	A tutorial or similar type of documentation shall be provided to the users of the platform								
Description:	A well-organized and self-contained tutorial and any other kind of documentation needed shall be provided to the experimenters for considering the design, the use and the variety of resources, the testbed facilities, etc. This can be in the form of a wiki. These functionalities shall be available to all possible future experimenters that may be interested in RAWFIE federation and want to explore its capabilities								
Additional Info (comments):									
Related Scenario(s)									

Id:	PT-P-003	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	RAWFIE platform shall provide a Testbed Directory where all the integrated testbeds are listed								
Description:	The Testbed Directory will include information relevant to the testbeds and possibly their resources (location, facilities) as well information on the capabilities of a particular resource and its requirements for executing experiments e.g. in terms of interconnectivity or dependencies.								
Additional Info (comments):	For each testbed at least the following information must be available: • its name • its location • a short description (possibly mentioning guidelines applying to the testbed usage) • type of resource(s) available • total number of resources available • total number of resources in use • list of resources with an indication as “available” or “booked” • EDL control capabilities supported • Connectivity status Additional (more detailed) information for the various UxV resources may be provided on demand.								

Related Scenario(s)	

Id:	PT-P-004	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	A testbed registration service shall be available								
Description:	Each participating testbed shall be registered in order to participate in RAWFIE Platform. During initial registration important details needed to access the testbed shall be provided and stored in an appropriate testbed directory service.								
Additional Info (comments):	The registration service should allow for periodic or testbed initiated updates of the registered data								
Related Scenario(s)									

Id:	PT-P-005	**Type:**	DATA	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	**1**
Title:	A spatial model for the data storage shall be provided								
Description:	Most of the data inside the data storage, especially sensor readings, will have a spatial aspect, i.e. some form of coordinate associated with it. The spatial model will define the coordinate reference system used as well as enable spatial queries against the storage.								
Additional Info (comments):	Potentially some form of spatial index technology will be provided to accomplish good performance for the spatial queries.								
Related Scenario(s)									

5.2.3 Authoring Phase

	PT-A-001	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	Experiment Description Language (EDL) shall be used as a language for the definition of experiment scenarios								
Description:	A Domain Specific Language combining some common characteristics of well-known XML based or similar languages shall be developed for the effective creation and handling of simple or complex experiment scenarios.								
Additional Info (comments):	The EDL shall provide: • common constructs like loops, conditional statements, synchronization blocks, task definitions etc. • location/topology specific elements • domain specific elements specific to each UxV testbed • elements for describing the UxV behavior								
Related Scenario(s)									

Id:	PT-A-002	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	For each defined experiment a short description of the experiment and its purpose shall be defined.								
Description:	In RAWFIE experimenters that create an experiment will need to provide a short high-level description of the experiment and its purpose. This allows infrastructure providers to keep track of the usage of the infrastructure, and enables them to report about this to their funding sources.								
Additional Info (comments):									
Related Scenario(s)									

<table>
<tr><td><b>Id:</b></td><td>PT-A-003</td><td><b>Type:</b></td><td>FUNC</td><td><b>Importance (priority):</b></td><td>HIGH</td><td><b>Source:</b></td><td>DoW</td><td><b>Ver:</b></td><td>1</td></tr>
<tr><td colspan="2"><b>Title:</b></td><td colspan="8">A tool for translating EDL into user directives shall be provided</td></tr>
<tr><td colspan="2"><b>Description:</b></td><td colspan="8">A tool will be able to translate the user directives defined in the experimentation scenario. The user directives will be translated into specific framework commands that will be understandable by the Testbed.</td></tr>
<tr><td colspan="2">Additional Info (comments):</td><td colspan="8">This can be seen as a compiler for the EDL definitions.
In comparison the EDL definitions, the framework commands are associated to real points in time. This is done by scheduling a certain experiment and then calculating the times from there on.
The real transformation of this commands into control commands for the real UxVs can only be done on testbed level as they depend on the concrete UxVs to be used</td></tr>
<tr><td colspan="2">Related Scenario(s)</td><td colspan="8"></td></tr>
</table>

<table>
<tr><td><b>Id:</b></td><td>PT-A-004</td><td><b>Type:</b></td><td>FUNC</td><td><b>Importance (priority):</b></td><td>HIGH</td><td><b>Source:</b></td><td>Scenario</td><td><b>Ver:</b></td><td>1</td></tr>
<tr><td colspan="2"><b>Title:</b></td><td colspan="8">An experimenter shall be able to describe sensor activation or deactivation directives during experiment authoring</td></tr>
<tr><td colspan="2"><b>Description:</b></td><td colspan="8">The experimenter shall be able to define for an available UxV sensor or sensor type, activation or deactivation policies based on:
• temporal predicates
• spatial predicates
• triggers
• events</td></tr>
<tr><td colspan="2">Additional Info (comments):</td><td colspan="8">Triggered based activation can be initiated based on the fulfillment of certain constraints (i.e. battery below a certain level). The constraints supported for triggered based activation/deactivation are still to be defined.
Type of events may relate to a failure or malfunction (or other criteria). List of supported events is still to be defined
This requirement should be regarded as a refinement of PT-006.4 for onboard sensors</td></tr>
<tr><td colspan="2">Related Scenario(s)</td><td colspan="8">1,2,5,6</td></tr>
</table>

Id:	PT-A-005	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Scenario	**Ver:**	1
Title:		An experimenter shall be able to address a specific UxV resource, group of UxVs or type of UxV resource in an experiment							
Description:		During an experiment authoring the experimenter should be able define actions and behavior for a single UxV or a group of UxVs. This stands true also during the experiment execution phase in case the experimenter is able to remotely controlled the UxVs							
Additional Info (comments):									
Related Scenario(s)		1,2,3,4,5,6							

Id:	PT-A-006	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	Scenario	**Ver:**	1
Title:		An experimenter shall be able to define the type of information to be gathered and/or stored by UxV resource(s)							
Description:		During experiment authoring the experimenter should be able to prescribe for an UxV resource the type and characteristics of the (sensor's) information that should be gathered. Examples type of info could be video, audio, image, raw data etc. Characteristics will depend on information type. I.e. if video is assumed possible characteristics may include video format, resolution etc.							
Additional Info (comments):		The EDL should support some basic information types and be extendable if additional ones are identified in the future.							
Related Scenario(s)		1,2,5,6							

Id:	PT-A-007	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	Scenario	**Ver:**	1

Title:	An experimenter shall be able to define the type of metrics to be gathered and/or stored during an experiment and/or per UxV resource
Description:	During experiment authoring the experimenter should be able to define specific metrics or performance indicators that need to be collected and stored for later analysis. These metrics may include: • network related metrics (i.e. distributions of errors, SNR, throughput, etc) (check scenario 4) • energy/consumption related metrics (i.e. coverage vs energy expenditure) (check scenario 5) • information quality metrics (i.e. information freshness) (check scenario 5)
Additional Info (comments):	The EDL should support some basic type of metrics and be extendable if additional ones are identified in the future.
Related Scenario(s)	4,5,6

<table>
<tr><td><b>Id:</b></td><td>PT-A-008</td><td><b>Type:</b></td><td>FUNC</td><td><b>Importance (priority):</b></td><td>HIGH</td><td><b>Source:</b></td><td>Scenario</td><td><b>Ver:</b></td><td>1</td></tr>
<tr><td colspan="2"><b>Title:</b></td><td colspan="8">An experimenter shall be able to provide navigation or movement directives during experiment authoring</td></tr>
<tr><td colspan="2"><b>Description:</b></td><td colspan="8">The EDL should provide the capability to define navigation or movement directives. This can be done in the form of:
• single geo-referenced waypoint (go to specific point)
• multiple geo-referenced waypoints
• via loading of a proper movement plan (not mandatory but good to have)
• predefined movement patterns</td></tr>
<tr><td colspan="2">Additional Info (comments):</td><td colspan="8">Additional details related to the UxV movement between waypoints should be supported like:
• Speed of movement
• Height at each waypoint (for UAVs only)
• Height interval in which the UAV can operate (for UAVs only)
More advanced options may be supported like:
• Obstacle avoidance rules (minimal needed distance to obstacles)
• Definition of optimization goals (shortest vs. fastest way)
• UxV characteristics (especially weight and height) preventing an UxV from using certain roads, bridges, tunnels etc.</td></tr>
<tr><td colspan="2">Related Scenario(s)</td><td colspan="8">1,2,3,4,5,6</td></tr>
</table>

Id:	PT-A-009	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	Scenario	**Ver:**	1
Title:	An experimenter shall be able to provide initial conditions and/or configuration parameters for an experiment								
Description:	The EDL should support experimenter in defining initial conditions and/or configuration parameters for an experiment. Such conditions may include (not an exhaustive list): • initial position of UxV resources • specific communication interface to be used • etc.								
Additional Info (comments):	It should also be possible to define what happens if the initial conditions are not met (abort the experiment, run it with additional sensors needed to gather the initial situation etc.).								
Related Scenario(s)	5								

Id:	PT-A-010	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Scenario	**Ver:**	1
Title:	An experimenter should be able to provide formation information for a group of UxVs resources								
Description:	The EDL should support the definition of formation information and/or coordination directives that a group of UxV resources should follow during an experiment execution.								
Additional Info (comments):	Formation info may be provided in the form of certain algorithms (i.e. PSO, MSO) that should be adopted by the UxVs for their optimal placement								
Related Scenario(s)	5								

Id:	PT-A-011	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1

<table>
<tr><td><b>Title:</b></td><td>A textual editor shall be provided for the authoring of RAWFIE experiments</td></tr>
<tr><td><b>Description:</b></td><td>A textual editor tool providing access to all EDL elements and all the functionality needed to edit experiment scenarios shall be provided.</td></tr>
<tr><td>Additional Info (comments):</td><td>Ideally the editor will be an IDE with a code completion, syntax highlighting, syntax checking, debugging capabilities, code folding functionality as well as other features making the authoring process easier and more productive.</td></tr>
<tr><td>Related Scenario(s)</td><td></td></tr>
</table>

<table>
<tr><td><b>Id:</b></td><td>PT-A-012</td><td><b>Type:</b></td><td>FUNC</td><td><b>Importance (priority):</b></td><td>HIGH</td><td><b>Source:</b></td><td>DoW</td><td><b>Ver:</b></td><td>1</td></tr>
<tr><td colspan="2"><b>Title:</b></td><td colspan="8">A visual/graphical editor shall be provided for the authoring of RAWFIE experiments</td></tr>
<tr><td colspan="2"><b>Description:</b></td><td colspan="8">The visual editor tool shall provide a graphical interface for handling EDL elements. The experimenter shall be able to choose and connect graphical elements from a customized palette and create graphically full functional experiments.
For each EDL element an appropriate property editor window may be available that will allow customization of EDL element’s properties/attributes.</td></tr>
<tr><td colspan="2">Additional Info (comments):</td><td colspan="8"></td></tr>
<tr><td colspan="2">Related Scenario(s)</td><td colspan="8"></td></tr>
</table>

<table>
<tr><td><b>Id:</b></td><td>PT-A-013</td><td><b>Type:</b></td><td>FUNC</td><td><b>Importance (priority):</b></td><td>HIGH</td><td><b>Source:</b></td><td>DoW</td><td><b>Ver:</b></td><td>1</td></tr>
<tr><td colspan="2"><b>Title:</b></td><td colspan="8">Spatial information shall be provided for the currently available resources for the authoring of new experiments</td></tr>
<tr><td colspan="2"><b>Description:</b></td><td colspan="8">The experimenter shall be equipped with tools representing the spatial coverage needs of the experiment and its fulfillment using a combination of currently available testbed resources prior to the reservation of these resources.</td></tr>
</table>

Additional Info (comments):	The information is given in terms of geospatial information (3D points, lines and polygons), associated with the defined spatial reference system.
Related Scenario(s)	

Id:	PT-A-014	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	Experiments defined via EDL shall be validated after their authoring								
Description:	Both textually or visually defined experiments shall be validated based on a predefined set of rules (i.e. syntactically, regarding spatial and/or spatiotemporal availability of selected resources) providing feedback to the author of the experiment about syntactic or semantic errors and possible restrictions/contradictions.								
Additional Info (comments):									
Related Scenario(s)									

Id:	PT-A-015	**Type:**	DATA	**Importance (priority):**	HIGH	**Source:**	Other	**Ver:**	1
Title:	Platform shall allow saving, editing and/or deletion of an experiment defined via EDL								
Description:	The experimenters shall have the option to save an experiment and retrieve it later on demand. They shall also be allowed to delete or modify existing scenarios owned by them. Experiments still in progress shall not be deleted Every version of the scenario will be saved and can be retrieved later on, i.e. an integrated version control system will be available. This is done by assigning a unique ID to every saved scenario version.								
Additional Info (comments):	Need to clarify whether deletion of an experiment would imply complete deletion from the RAWFIE platform or just not be accessible by the user (information including results can be kept and be available only for administrators).								
Related Scenario(s)									

Id:	PT-A-016	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	RAWFIE platform shall provide a Resource Discovery tool for fine-grained resource searches								
Description:	A Resource Discovery tool shall be available that will provide some basic query capabilities to facilitate the experimenter in identification of certain capabilities (testbed or/and UxV resource specific) that may need for an experiment. An experimenter shall be able to fill in some specific technical details about the hardware he/she is looking for, and it should be possible for the resource discovery tool to construct a suitable response based on the resource information provided for a testbed. When the query in the resource discovery phase returns a certain list of resources, it should be possible for the experimenter to select the resources they would like to include in the experiment. This should be supported in relation with a specific resource ID								
Additional Info (comments):	Need to define what exactly these capabilities could be for the testbed node and its various resources (i.e. CPU, RAM, Op. system, battery state, communication interfaces, sensor types, capabilities regarding resource controller, etc.) Need also to agree whether query capabilities would be available via an SQL query like language or via appropriate drop down menus or catalogues (the latter might be preferable for novice users but may limit the complexity of queries and consequently the granularity of searches).								
Related Scenario(s)									

5.2.4 Booking Phase

Id:	PT-B-001	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	Booking functionality shall be provided to experimenters								
Description:	The experimenters shall be allowed to access the testbeds list and for a given testbed reserve the UxV resources for the time interval required for the execution of the defined experiment. The Booking tool shall be responsible for the coordination in use of the testbed resources among experimenters.								

Additional Info (comments):	Experimenters should be able to create new bookings as well as to view and edit the bookings they have already done.
Related Scenario(s)	

Id:	PT-B-002	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:		Booking tool shall provide an intuitive UI interface to facilitate the reservation process							
Description:		Experimenters shall be allowed to access the testbed list first and, for a given testbed, reserve the UxV resources required for the execution of the defined experiment beforehand through the experiments' editor tool. The booking tool interface should allow to: • Visualize, in a calendar view, the available dates and timeslots for each testbed's resources • Select the preferred date, timeslot and/or space fragment in a testbed (based on the availability of the required nodes) to execute the desired experimentation scenario							
Additional Info (comments):		Experimenters should be able to create new bookings as well as to view and edit the bookings they have already done.							
Related Scenario(s)									

Id:	PT-B-003	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:		Booking functionality shall allow reservation of resources involving multiple testbeds							
Description:		The booking module must allow for the purpose of a single experiment the possible reservation of resources from different physical testbeds if this is explicitly requested from an experimenter.							

Additional Info (comments):	
Related Scenario(s)	

Id:	PT-B-004	**Type:**	FUNC	**Importance (priority):**	LOW	**Source:**	Consortium	**Ver:**	1
Title:	Notification mechanisms may be provided for experiments scheduled for execution in the future.								
Description:	A notification mechanism to remind an experimenter the date and the timeslot allocated for running his/her experiment on the RAWFIE infrastructure may also be envisaged to improve the user experience. The time of notification prior to the experiment launch should be configurable.								
Additional Info (comments):									
Related Scenario(s)									

Id:	PT-B-005	**Type:**	FUNC	**Importance (priority):**	LOW	**Source:**	DoW	**Ver:**	1
Title:	Booking assistance mechanisms may be provided for ensuring fairness in resource bookings								
Description:	Experimenters may be provided with some assistance mechanisms, either automated or involving manual intervention (i.e. by an administrator or testbed operator) for booking experiments. These could try to resolve conflicts (e.g. if an experiment should be booked but not enough UxVs are available at this timeslot, a slightly different timeslot could be proposed) and generally ensure fairness (the system could ask another experimenter - via an asynchronous mechanism - if he is willing to change his experiment if someone else needs an experiment with the resources already blocked by this booking).								
Additional Info (comments):	The advantages for the testbeds are a more efficient utilization and for the experiments it is quite beneficial because it simplifies the booking process.								
Related Scenario(s)									

<table>
<tr><td><b>Id:</b></td><td>PT-B-006</td><td><b>Type:</b></td><td>FUNC</td><td><b>Importance (priority):</b></td><td>MEDIUM</td><td><b>Source:</b></td><td>Consortium</td><td><b>Ver:</b></td><td>1</td></tr>
<tr><td colspan="2"><b>Title:</b></td><td colspan="8">RAWFIE platform should allow virtualization of available UxVs resources during reservation process</td></tr>
<tr><td colspan="2"><b>Description:</b></td><td colspan="8">Unless an experimenter explicitly requests reservation of specific testbeds/resources for an experiment, the RAWFIE platform should offer to an experimenter the ability to reserve resources in a topology agnostic manner thus offering virtualization of available resources.
Internally the service should attempt to reserve resources in the same physical testbed and if this is not possible then consider resources from multiple testbeds.</td></tr>
<tr><td colspan="2">Additional Info (comments):</td><td colspan="8">Exact level of virtualization that will be available in RAWFIE will be defined in next iteration of the deliverable</td></tr>
<tr><td colspan="2">Related Scenario(s)</td><td colspan="8"></td></tr>
</table>

<table>
<tr><td><b>Id:</b></td><td>PT-B-007</td><td><b>Type:</b></td><td>FUNC</td><td><b>Importance (priority):</b></td><td>LOW</td><td><b>Source:</b></td><td>DoW</td><td><b>Ver:</b></td><td>1</td></tr>
<tr><td colspan="2"><b>Title:</b></td><td colspan="8">An accounting and billing service shall be provided that will track resource usage by individual users.</td></tr>
<tr><td colspan="2"><b>Description:</b></td><td colspan="8">The common accounting and billing subsystem of RAWFIE platform will receive events from the other subsystems and will keep track of resources usage by individual users. This billing, in credit units, can be used in the early days of the federation to enable a policy of fair resource sharing among users, by assigning every user a specific number of credits periodically and charging them for their use of distinct platform resources (i.e. UxV launching, post-experiment processing etc.). After the EU funding period, the billing system will play a major role in the viability of RAWFIE as the cost (the model of which is built by identifying and quantifying all costs involved in setting up, maintaining, developing and managing the different facilities that are part of the federation) will be entirely covered by the beneficiaries of the experiments (either be it private or public funding).</td></tr>
<tr><td colspan="2">Additional Info (comments):</td><td colspan="8"></td></tr>
<tr><td colspan="2">Related Scenario(s)</td><td colspan="8"></td></tr>
</table>

5.2.5 Launching / Execution Phase

Id:	PT-L-001	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	RAWFIE shall provide a validator to constantly check experiment scenarios during runtime								
Description:	The validator will check for semantic errors including spatiotemporal constrains. These errors will occur during the execution of the code, after it has been parsed as grammatically correct by the EDL component. For instance booking out testbed and nonexistent coordinates can be checked by this validation level.								
Additional Info (comments):	An error message will be shown to the experimenter as a feedback after the experimentation validation. The message will be provided through the front end tier.								
Related Scenario(s)									

Id:	PT-L-002	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	Launching tool shall be provided to experimenters allowing execution only of experiments that have been successfully validated by the platform								
Description:	The launching tool shall initiate the execution of an experiment or set of experiments. It shall communicate with all the necessary resources the booked timeslots in order to initiate every booked action for the experiment.								
Additional Info (comments):	A successfully launched experimentation scenario is considered as the one that all the involved procedures comply with RAWFIE architecture. That means that the experimentation scenario has been validated in the Experiment Validator of the Middle Tier and been fed in the UxV Testbed Tier								
Related Scenario(s)									

Id:	PT-L-003	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	Launching tool shall be kept informed upon an experiment's state								
Description:	The launching tool should be informed about changes of an experiment's state, especially the completion or the abort of an experiment.								
Additional Info (comments):	In the case of the entire use of a testbed, the Launching tool could initiate the next booked scenario (applies to scenarios without strict timeslot constraints) when the current one is finished								
Related Scenario(s)									

Id:	PT-L-004	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	Monitoring tool shall be provided to experimenters								
Description:	Monitoring tool shall manage the presentation of the information needed for monitoring the status of the nodes and the data collected during the experiments. The values of the sensing modules and the status of the different networking modules are some essential elements of the monitoring tool.								
Additional Info (comments):	The monitoring tool will display the output of the visualization engine (part of the experiment manager).								
Related Scenario(s)									

Id:	PT-L-005	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	Visualization engine shall be part of the middleware system								

Description:	A rather powerful (i.e., in terms of parallel processing and 3D presentation capabilities) Visualization Engine, that will provide its results to the experimenter through the Visualization tool of the Experimentation Suite in the Front End tier, should be developed.
Additional Info (comments):	Also the spatial UxV / sensor data will be part of the engine, i.e. some form of GIS will be provided.
Related Scenario(s)	

Id:	PT-L-006	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	A 3D visualization tool shall be available for the tracking of all moving resources								
Description:	The experimenter shall be able to track UxV resources during the execution of the experimentation scenario with a 2D and/or 3D GIS technology dependent on geo-referenced available 2D or 3D GIS or image data of the experiment area for visualization.								
Additional Info (comments):	Real time tracking may be restricted by the communication technology of the UxV data transmission. Cloud Technology may be not fast enough for real time tracking.								
Related Scenario(s)									

Id:	PT-L-007	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	RAWFIE platform shall provide means to ensure fairness in experiments execution								
Description:	RAWFIE platform shall provide mechanisms, either automated or involving manual intervention (i.e. by an administrator) that will ensure fairness in experiments execution thus avoiding a resource being perpetually used by a certain experiment/experiment.								
Additional Info (comments):									
Related Scenario(s)									

Id:	PT-L-008	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	RAWFIE platform shall allow experimenters to remotely navigate UxVs.								
Description:	RAWFIE experimenters shall have ability to guide the unmanned vehicles through a virtual remote controller provided by the application's interface. Either the experimenter directly controls the UxV or the provided instructions are translated into a "global form" of waypoints (a reference scheme compatible with the build-in navigation system of the UxVs) and transmitted to the controlled units.								
Additional Info (comments):	The virtual remote controller will act as proxy control unit communicating with the real control unit that lies on each testbed The exchanged messages should be designed in respect to open standards possibly using well know formats (i.e. JSON or XML).								
Related Scenario(s)									

Id:	PT-L-009	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	RAWFIE platform shall support a semi-autonomously way of navigation of the UxVs								
Description:	Experimenters provide details about the mission that UxVs will execute as well as comprehensive information about the algorithms to be used to process this task. RAWFIE undertakes the evaluation of all the employed elements and in each time step the system assesses the validity of the decisions of the involved algorithms. The internal control mechanism alters the trajectory of the units so as to ensure both, the vehicle's safety and the success of the mission. At each time step next optimum/appropriate waypoint for each UxV is transmitted to it. The vehicles communicate their sensor measurements together with their exact positions (so as to correct possible localization issues) back to the RAWFIE framework.								
Additional Info (comments):	Real time tracking may be restricted by the communication technology of the UxV data transmission. Cloud Technology may be not fast enough for real time tracking.								
Related Scenario(s)									

Id:	PT-L-010	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Scenario	**Ver:**	1
Title:	An experimenter shall have the means to define actions or tasks that should run on a periodic or ad hoc basis during execution of an experiment								
Description:	The EDL should support the definition of actions or sequence of actions (tasks) that may run periodically or triggered based on predefined criteria or events. Such actions may related to: • enablement/disablement of certain functionality (or modules) • data storage (or caching) • data transmission • error reporting Additional type of actions may exist based on scenario specific needs								
Additional Info (comments):									
Related Scenario(s)	1,2,3,4,5,6								

5.2.6 Evaluation Phase

Id:	PT-E-001	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	Each run experiment should be uniquely identified within RAWFIE platform								
Description:	Each successfully validated experiment should be assigned, upon execution, a unique experiment run ID								
Additional Info (comments):	The experiment ID can be used for correlating the experiment with certain data later on.								
Related Scenario(s)									

Id:	PT-E-002	**Type:**	DATA	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:		RAWFIE platform shall include a service enabling the data collection, analysis and processing.							
Description:		This service is responsible for storing the measurement streams gathered from experiments. RAWFIE will provide a large, secure, cloud-based central repository on which collected data can be made available to all platform users. Moreover processing algorithms will be applied to the collected data transforming them to usable information and giving more detailed insights on the nature of the experiments.							
Additional Info (comments):		The experiment data will include detailed information about the UxVs' state: position and other properties. The size of the generated data might be much bigger from what can be stored in the available storage capacity, in which case we might need to only store the most recent data (using some time window) and/or keep meaningful data summaries over the same or longer temporal windows. All stored experiment data will use an absolute time reference (e.g. the GPS time).							
Related Scenario(s)									

Id:	PT-E-003	**Type:**	DATA	**Importance (priority):**	HIGH	**Source:**	Other	**Ver:**	1
Title:		RAWFIE platform shall allow the retrieval of data and post-analysis information for a particular experiment.							
Description:		This data gathered for a particular experiment should be accessible during and after the experiment, and should be clearly correlated to the experiment run ID Experimenters shall be allowed to choose if collected data can be shared with third parties							
Additional Info (comments):									
Related Scenario(s)									

Id:	PT-E-004	**Type:**	DATA	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	A data description language should be provided								
Description:	A data description language describing the collected data should be developed. It could support different levels of complexity depending on the project constraints but it should contain at least time-stamps and device identifiers for the devices that generated the data.								
Additional Info (comments):	The description language should also support to describe the data types generated – support for numerical, textual and discrete types is mandatory.								
Related Scenario(s)									

Id:	PT-E-005	**Type:**	DATA	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	A data analytical software should be provided								
Description:	Data analytical software that will deliver a set of analytical functionalities such as: • outlier detection, • distribution shift detection, • classification, More details on available capabilities are to be provided in subsequent versions of the requirement deliverable - see also section 7.1								
Additional Info (comments):									
Related Scenario(s)									

5.3 Testbed Requirements

Testbed requirements include all the requirements pertaining the testbed facility, the UxV resources and interconnectivity needs between them and with the RAWFIE platform. The identified high level requirements were classified in 4 main categories:

- General Testbed Requirements

- Interconnectivity Requirements
- Resource (UxV & Sensors) Requirements
- Data Storage Requirements

A summary table of all Testbed functional requirements is provided in the Annexes sections.

5.3.1 General Testbed Requirements

Id:	TB-G-001	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	The Testbed shall implement a monitoring service to observe and report the operation of the testbed's resources								
Description:	The testbed's monitoring service acts as an observer for the seamless operation of the testbed's resources. The service periodically checks the current status of available resources in the testbed facility by interacting with every UxV node. At the end of every monitoring "round" an observation report is provided.								
Additional Info (comments):									
Related Scenario(s)	All scenarios								

Id:	TB-G-002	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:	The testbed shall be able to support static resources (sensors)								
Description:	Beyond UxVs the testbed shall be able to support different kinds of static sensors (Day/night cameras, radars, flame detectors). The UxVs belonging to the testbed shall be able to communicate with static sensors for data collection, activation/deactivation and configurable parameters programming.								
Additional Info (comments):									
Related Scenario(s)	1,2								

Id:	TB-G-003	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	The Testbed should provide (near) real time communication between the sensors,								

	the UxVs and the rest of RAWFIE infrastructure.
Description:	(near) Real time transmission of the collected data to the RAWFIE infrastructure (e.g. live streaming of the monitoring area, coordinates of the potential threat, snapshots of the target, monitoring area, damage on infrastructure, pollution due to CBNR attack).
Additional Info (comments):	
Related Scenario(s)	2, 3, 4, 5, 6

Id:	TB-G-004	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:		Each Testbed shall provide information about the capabilities of each resource node.							
Description:		The testbed has to provide a complete set of each resource node capabilities This will be updated periodically or on request by the RAWFIE platform. Such information for UxV nodes may include: • HW characteristics (CPU architecture and speed, RAM). • Communication capabilities (i.e. supported 802.11 standards, optical networking interfaces, software defined radio) • Sensing capabilities • measurement resource type							
Additional Info (comments):									
Related Scenario(s)									

Id:	TB-G-005	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:		The Testbed should be able to provide accurate location information for each UxV and/or static node.							
Description:		The Testbed should be able to provide accurate location identification for each UxV and/or static node using the UxV's location identification system. Coordinates and information about UxV's track should also be stored and should be provided to the Visualization Engine.							

Additional Info (comments):	
Related Scenario(s)	All scenarios

Id:	TB-G-006	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:		The Testbed shall provide error notification.							
Description:		The Testbed tier should devote specific functionalities for UxVs error notification to the other tiers of RAWFIE platform and, eventually, to the experimenters.							
Additional Info (comments):									
Related Scenario(s)									

Id:	TB-G-007	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:		The Testbed should allow execution of multiple experiments in parallel.							
Description:		The Testbed shall allow for the simultaneous execution of several experiments at the same time provided that there are available resources							
Additional Info (comments):									
Related Scenario(s)									

Id:	TB-G-008	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:		The Testbed shall provide a navigation service							

Description:	Both, in case of remote control as well as in case of the semi-autonomously way of navigation of the UxVs, the testbed will validate the next candidate position for each vehicle.
Additional Info (comments):	This functionality will be developed keeping in mind previous EU funded projects. The algorithm to be used is based on the so called Cognitive-based Adaptive Optimization (CAO) approach. CAO transforms the navigation problem into an optimization one, which in every time step the goal is to optimize the location of the UxVs so to meet the objectives of the mission with respect to a set of constraints. CAO is a lightweight algorithm which for the purpose of the RAWFIE project, will be significantly enhanced so as to operate as a web-service.
Related Scenario(s)	All Scenarios

Id:	TB-G-009	**Type:**	FUNC	**Importance (priority):**	LOW	**Source:**	Consortium	**Ver:**	1
Title:	The Testbed shall be able to support simulated UxVs resources.								
Description:	The Testbed shall be equipped with appropriate adapters/interfaces that will enable communication with UxVs simulators or even with network simulators of UxVs resources. The objective is to give the end-user the ability to conduct experiments and evaluate its aspects of interest (e.g. communication protocols) when the amount of actual resources is limited and the cost of experiment execution prohibitive taken into account that the loss of realism in the simulation models is permissible.								
Additional Info (comments):									
Related Scenario(s)									

5.3.2 Interconnectivity Requirements

Id:	TB-I-001	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:	The UxV Testbed shall be able to communicate with the RAWFIE platform infrastructure using standard based communication means								

Description:	For each Testbed an appropriate mechanism should be developed, responsible for handling the communication between the facility and the rest tiers of RAWFIE architecture.
Additional Info (comments):	Communication should be possible via the internet using either IPv4 or IPv6. Support of VPN tunnelling should also be possible.
Related Scenario(s)	

Id:	TB-I-002	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:	The communication system shall be able to use UxVs to relay information to and from other UxVs								
Description:	In absence of any other traditional communication link, or due to mission-related constraints, the communication system shall use UxVs to relay information to and from other UxVs								
Additional Info (comments):									
Related Scenario(s)	2,3,5,6								

Id:	TB-I-003	**Type:**	FUNC	**Importance (priority):**	LOW	**Source:**	Consortium	**Ver:**	1
Title:	A Testbed’s communication system may provide at least 3 levels of Service and the communication means will adapt to these Levels of Service.								
Description:	The communication means shall adapt the Level of service (e.g. offer several Qualities of Service and characteristics) depending on demand and on the available resources. This can be used to optimise the communication system with respect to several metrics, such as coexistence of several swarms of UxV, degraded modes in case of damage or scrambled channels, etc.								
Additional Info (comments):									
Related Scenario(s)									

Id:	TB-I-004	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	The Testbed shall be able to dispatch UxV information on demand								
Description:	Testbed shall ensure that each resource on the RAWFIE facilities will be discoverable from the rest infrastructure.								
Additional Info (comments):									
Related Scenario(s)	All scenarios								

5.3.3 Resource (UxV & Sensors) Requirements

Id:	TB-R-001	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:	The UxV shall be able to operate autonomously								
Description:	The UxV shall be able to operate autonomously (without any external control). The objective is to give it the capability to make the flight as planned even if there are some disturbances, deviations, unexpected events, etc.								
Additional Info (comments):	See also PT-L-009								
Related Scenario(s)									

Id:	TB-R-002	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	The UxV should provide collision avoidance mechanism								
Description:	The UxV shall be able to autonomously avoid collision, for example by defining an "intimacy zone" in which no other object or UxV is allowed to enter without any specific reaction.								

Additional Info (comments):	See also PT-L-009
Related Scenario(s)	

Id:	TB-R-003	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:	Each UxV shall have a unique Identification code								
Description:	Each UxV shall have a unique Identification code across the testbed								
Additional Info (comments):									
Related Scenario(s)									

Id:	TB-R-004	**Type:**	DATA	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:	UxV’s shall be able to store data on board								
Description:	Capability of data storage in case of transmission failure (Link loss between UxV’s and the platform) and retransmission of data as soon as the link is established again.								
Additional Info (comments):	The UxVs usually communicate their sensor measurements together with their exact positions back to the RAWFIE framework. In certain cases there might be a need to store data on board. Example cases include: • Transmission failure (Link loss between UxV’s and the platform) and retransmission of data as soon as the link is established again • Sensors data content too large to be transmitted in real time, it will be collected and stored and downloaded on the RAWFIE platform for post analysis after the mission or experiment.								
Related Scenario(s)									

Id:	TB-R-005	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	Capability of task planning of the UxVs nodes during run-time.								
Description:	The user must have the capability to plan the course of a UxV and the tasks that it would have to execute during this course								
Additional Info (comments):	This requirement would need an appropriate and easy-to-use User Interface								
Related Scenario(s)	2								

Id:	TB-R-006	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	Capability of taking the control of the UxVs from distance.								
Description:	Capability of taking the control of a UxV node in real-time and on-demand, after receiving appropriate alarm inputs								
Additional Info (comments):									
Related Scenario(s)	2								

Id:	TB-R-007	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	Each UxV node should ensure a minimum autonomy of 15-30 minutes.								
Description:	Multiple UxVs will provide to the experimenters a minimum duration of 45 to 90 minutes.								
Additional Info (comments):	Provided figures are to be checked and updated in next version of requirements document								
Related Scenario(s)									

Id:	TB-R-008	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	Each UxV node should ensure payload.								
Description:	Multiple UxVs will provide to the experimenters a minimum payload of 0.5-1kg per unit.								
Additional Info (comments):	Provided figures are to be checked and updated in next version of requirements document								
Related Scenario(s)									

Id:	TB-R-009	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	Each UxV node should be equipped with a location identification system.								
Description:	UxVs will be equipped with a powerful location identification system such as on board GPS system, Long baseline (LBL) for underwater Positioning, etc. and geo-locate the UxVs in the area of operation.								
Additional Info (comments):									
Related Scenario(s)									

Id:	TB-R-010	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	UxVs should be able to cooperate during the execution of an experiment.								
Description:	The UxV should be able to exchange some data in real-time, at least between the nearest neighbour. This information may be used for the local and fine coordination inside or between UxV swarms or for cooperative monitoring of an area.								
Additional Info (comments):									
Related Scenario(s)	3,5,6								

Id:	TB-R-011	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	UxVs should be able to Synchronize their Time-References between them								
Description:	The UxV shall have and maintain an on-board independent time reference, which is synchronized with the other UxV running the same mission (or in a pre-defined group of cooperative UxVs). This can be used for triggering global actions or for group flying, tighter safety distances, etc.								
Additional Info (comments):									
Related Scenario(s)	3,4,5,6								

Id:	TB-R-012	**Type:**	FUNC	**Importance (priority):**	MEDIUM	**Source:**	Scenario	**Ver:**	1
Title:	The UxV should provide Access Point functionality								
Description:	For the purpose of certain experiments a UxV should be able to act as Access Point/Repeater and provide internet connectivity								
Additional Info (comments):									
Related Scenario(s)	3								

Id:	TB-R-013	**Type:**	FUNC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	Each UxV node shall be equipped with primary and secondary communication means.								
Description:	At least two different networking modules will be utilised in order to support a network problem recovery scheme.								
Additional Info (comments):									

Related Scenario(s)	

5.3.4 Data Storage Requirements

Id:	TB-D-001	**Type:**	DATA	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:	The Testbed shall be able to store data in case of transmission failure.								
Description:	Capability of data storage in case of transmission failure (Link loss between Testbed and the rest of RAWFIE infrastructure) and retransmission of data as soon as the link is established again.								
Additional Info (comments):									
Related Scenario(s)									

Id:	TB-D-002	**Type:**	DATA	**Importance (priority):**	LOW	**Source:**	Consortium	**Ver:**	1
Title:	The Testbed may provide statistical data/information about its operation.								
Description:	Statistical data such as: number of experiments; experiments duration; number of UxV nodes used; Testbed time alive; etc.								
Additional Info (comments):									
Related Scenario(s)									

6 Non-functional Requirements

Non functional categories include requirements that do not directly prescribe functionality of the system but rather focus on aspects related to usability, performance, safety and security etc. For a complete list of the various types of non-functional requirements refer to Table 3 of the deliverable.

Summary tables of all non-functional requirements are provided in the Annexes sections.

6.1 Platform Requirements

Id:	PT-NF-001	**Type:**	SEC	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	RAWFIE platform shall support secure data exchange								
Description:	In cases where data privacy of experiments is considered critical, communication between the middleware and other components (Front-end, Testbed, Data storage) shall be encrypted.								
Additional Info (comments):									
Related Scenario(s)									

Id:	PT-NF-002	**Type:**	SEC	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:	RAWFIE platform shall provide a reservation/booking system with adequate security and privacy								
Description:	RAWFIE platform must provide a reservation/booking system with adequate security to provide assurance to industrial users. More specifically, a potential user/experimenter (with the exception of a RAWFIE platform administrator) should not be allowed to view detailed information on experiments currently executing or scheduled for execution by other experiments.								
Additional Info (comments):									
Related Scenario(s)									

Id:	PT-NF-003	**Type:**	SUPP	**Importance (priority):**	MEDIUM	**Source:**	DoW	**Ver:**	1
Title:	RAWFIE platform should be able to support backups of all critical data								
Description:	Critical data of RAWFIE experiments should be able to be recovered in case of temporal failure.								
Additional Info (comments):	This also includes data which would normally not be stored permanently.								
Related Scenario(s)									

Id:	PT-NF-004	**Type:**	PERF	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	RAWFIE platform shall exhibit high degree of network availability								
Description:	Sufficient network bandwidth shall be available for undisrupted experiments execution and data communication especially in cases where streaming services like video or 3D visualization are requested by end users.								
Additional Info (comments):	Indicative required bandwidth (exact values to be determined in 2nd iteration): • 50-100Mbps symmetrical communication for wired Wide Area Networking, • 1Gbps for wired Local Area Networking, • 50-100 Mbps for wireless Local Area Networking and • 4G for wireless Wide Area Networking								
Related Scenario(s)									

Id:	PT-NF-005	**Type:**	PERF	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	RAWFIE platform shall be able to support (near) real-time information gathering from the UxV sensors								
Description:	RAWFIE platform shall be able to execute experiments with certain response time constraints when real-time information gathered from the UxV sensors is a strict constraint for the successful execution of the experiment.								
Additional Info (comments):	This relates to PT-NF-004 as it requires guaranteed low network latency.								
Related Scenario(s)									

Id:	PT-NF-006	**Type:**	PERF	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	RAWFIE platform shall exhibit high degree of scalability								
Description:	RAWFIE platform shall be able to serve the unobstructed execution of a large number of experiments provided the required UxV resources are available.								
Additional Info (comments):									
Related Scenario(s)									

Id:	PT-NF-007	**Type:**	PERF	**Importance (priority):**	MEDIUM	**Source:**	DoW	**Ver:**	1
Title:	RAWFIE platform should exhibit high degree of availability								

Description:	RAWFIE platform should be available for service at almost any time. This means that the system runs without regular downtimes and updates to the system should be done without interrupting the system for a long period of time.
Additional Info (comments):	Users of the system should be informed about planned downtime so they can be prepared for such times. Non-interrupting updates can be achieved by having at least two instances of all services so that one instance can be updated while the other one still serves all incoming requests.
Related Scenario(s)	

Id:	PT-NF-008	**Type:**	SUPP	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	RAWFIE architecture should be compatible to previous FIRE projects.								
Description:	The architecture of previous FIRE projects should be taken into account in during the process of architecture definition, to check what is reusable. Reuse of existing interfaces and workflows should done as much as possible, so that RAWFIE testbeds could be linked into other FIRE projects.								
Additional Info (comments):									
Related Scenario(s)									

Id:	PT-NF-009	**Type:**	SUPP	**Importance (priority):**	MEDIUM	**Source:**	DoW	**Ver:**	1
Title:	RAWFIE architecture should adopt a modular design approach.								
Description:	The platform architecture should be modular providing functionalities grouped/separated in different software elements in order to allow for easy adaptation and extension.								
Additional Info (comments):									

Related Scenario(s)	

Id:	PT-NF-010	**Type:**	ENV	**Importance (priority):**	HIGH	**Source:**	DoW	**Ver:**	1
Title:	RAWFIE platform shall be deployed as a cloud based service (or list of services).								
Description:	RAWFIE’s platform software will be deployed on the Cloud, and suitable solutions will be employed for ensuring high availability in case of increase in the processing requests.								
Additional Info (comments):	The IaaS platform of GRnet, offering virtualized computing resources, ~okeanos [23] may be used for this purpose.								
Related Scenario(s)									

Id:	PT-NF-011	**Type:**	SUPP	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:	RAWFIE software modules should be implemented as Web Service or as REST								
Description:	To allow an easy cloud setup, all software modules should provide Web Service or REST interfaces								
Additional Info (comments):									
Related Scenario(s)									

Id:	PT-NF-012	**Type:**	SUPP	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1

Title:	RAWFIE modules should use Open Standards and Open Software as far as possible
Description:	The interfaces amongst components should be designed keeping in mind Open Standards and they should use these standards as much as possible. Open Software should be preferred to proprietary software.
Additional Info (comments):	
Related Scenario(s)	

6.2 Testbed Requirements

6.2.1 General Testbed Requirements

Id:	TB-NF-G-001	**Type:**	PERF	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	The Testbed shall provide concurrent requests capacity								
Description:	The Testbed infrastructure should be able to handle requests coming from multiple sources								
Additional Info (comments):									
Related Scenario(s)									

Id:	TB-NF-G-002	**Type:**	SUPP	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	The Testbed infrastructure should provide reliability and robustness of all components/modules.								
Description:	If one of components/modules of the system stops functioning, this should not affect the state of the other components/modules and the system should continue providing services.								

Additional Info (comments):	
Related Scenario(s)	

Id:	TB-NF-G-003	**Type:**	SUPP	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:		The communication system shall offer a high availability							
Description:		The communication system shall offer a high availability (ratio(s) to be defined, depending on the criticality of the communication link) during the use of the testbed.							
Additional Info (comments):		See also PT-NF-004							
Related Scenario(s)									

Id:	TB-NF-G-004	**Type:**	SEC	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:		The communication interfaces shall offer security mechanisms							
Description:		The communication interface of the testbed as well as the links between the testbed and the UxVs shall offer security mechanisms to support authentication, integrity, confidentiality, privacy, and possibly also non-repudiation. A highly encrypted security protocol, that protects mobile units against hacking, being reprogrammed, and potentially used them for malevolent/criminal/terrorist abuses should be used. This will be particularly useful when several users will be using the same testbed at the same time.							
Additional Info (comments):		Needed to address ethical issue mandate.							
Related Scenario(s)									

Id:	TB-NF-G-005	**Type:**	OTH	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	Testbed shall abide with legal restrictions								
Description:	All testbeds have to abide with any legal (mostly safety-related) restrictions imposed by the use of remote controlled vehicles (UxVs) and provide accurate experimental results.								
Additional Info (comments):									
Related Scenario(s)									

Id:	TB-NF-G-006	**Type:**	PERF	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:	The communication system shall be robust to a number of issues typical of wireless communication, such as interference, scrambling, coexistence, etc.								
Description:									
Additional Info (comments):	See also TB-NF-004								
Related Scenario(s)									

6.2.2 Resource (UxV & Sensors) requirements

Id:	TB-NF-R-001	**Type:**	USE	**Importance (priority):**	MEDIUM	**Source:**	Consortium	**Ver:**	1
Title:	UxVs shall offer on demand resources (Network, Sensor, Processing, and Controller).								

Description:	The UxVs shall be able to provide a scalable platform with expandable (on demand) networking and smart sensing facilities along with a rather powerful processing board to tackle the need of hosting an Operating System (OS).
Additional Info (comments):	
Related Scenario(s)	

Id:	TB-NF-R-002	**Type:**	USE	**Importance (priority):**	MEDIUM	**Source:**	Standards	**Ver:**	1
Title:	UxVs sensor system shall be compliant to connection standards and communication interfaces.								
Description:	RAWFIE's UxV sensor system should provide widely accepted connection standards and communication interfaces. The supported sensors can be simple, i.e., microsensors that monitor measure such as temperature, humidity, gas, etc., or more advanced like optical sensors								
Additional Info (comments):									
Related Scenario(s)									

Id:	TB-NF-R-003	**Type:**	SEC	**Importance (priority):**	HIGH	**Source:**	Consortium	**Ver:**	1
Title:	UxV shall be capable to revert to a safe mode								
Description:	UxV shall be capable to revert to a safe mode (e.g. self landing) when a certain failure occurs (e.g. communication failure, out of range, low battery). This should be possible while still under remote control as well as on their own.								
Additional Info (comments):									
Related Scenario(s)									

7 Other Issues

The section provides information regarding potential solutions that exist in important areas that RAWFIE project seeks to provide viable solutions or enhancements. The information should serve as state of art/state of practice reference.

7.1 Datasets & Data Analysis

Data analysis is an important part of the RAWFIE project. High volume of data structures and streams are expected to be provided during an experiment execution and therefore the need for efficient solutions in this area is mandatory. The three solutions are sorted in increasing order of relevance.

ParStream

ParStream [15] is a commercial solution. It is capable to perform parallel streaming of data structures. It uses a client/server model, where server is offered as SaaS. Basic features include:

- Advanced Analytics feature: Integration of R for Machine Learning.
- Provides user defined extensions at the scale of IoT, enabling customers and third parties to define, program and run user-defined algorithms as part of any SQL query.
- Native Integration of Knime Analytics platform (Open source).
- No need for any use of pre-aggregation, data can be continuously imported, indexed and queried with real time analytics feature and therefore low latency.
- Query-type searching adapted to real time analytics rather than batch mode processing.
- Combines usual CPU processing with GPU processing,
- Indexing specifically adapted to CSV files import, although it performs imports on column stores (standard structured datasets) as well.
- Several interfaces, at high level (queries submitted via a Java Applet though the use of a provided JDBC driver), mid level (queries submitted as SQL code) and low level (provided C++ API enabling users to write their queries).

RapidMiner

RapidMiner [16] provides an advanced analytical solution though template-based frameworks. Basic version of the tool is open source, whereas stream processing extension is not. It uses a client/server model, where server is offered as SaaS. Basic features include:

- High-level interface, nearly eliminating the need to write code.
- Provides data ETL operations (Extract Transform Load), data processing and visualization, predictive analytics and statistical modeling.
- Provides a GUI to design and execute analytical workflows.

- The engine can either be used as an API or called from other programs.
- Provides learning schemes and models and algorithms from Weka (popular suite of machine software) and R scripts that can be used through extensions.
- Provides a platform for developers to create data analysis algorithms.

Samoa (Scalable Advanced Massive Online Analysis)

Samoa [17] should be regarded as a platform for mining big data streams. Samoa is released as open source software under the Apache Software License (version 2.0). It provides a collection of distributed streaming algorithms for the most common data mining and machine learning tasks such as classification, clustering and regression, as well as programming abstractions to develop new algorithms. Basic features include:

- Designed in a pluggable architecture allowing it to run on several distributed processing engines (Storm, S4 and Samza).
- Provides an API for algorithm developers that simplify implementing distributed streaming algorithms.
- Can be regarded as both a framework and a library. As a framework, enables developers to reuse their code on different engines. As a library, contains implementation of state-of-the-art algorithms for distributed machine learning on streams.

In order to be able to use Samoa, data has to be accessed via a distributed stream processing engine such as Storm [18], S4 [20] or Samza [19]. This might prove overkill for the current analytical needs. A better, more scalable solution, would be data to be originally continuously stored on a distributed architecture (cluster typically) by the devices themselves however

ParStream and RapidMiner however are both based on a client/server model, server offered as a Saas. Both provide an extensive amount of data analysis routines. Among the two, RapidMiner seems to suit our current purposes best since it'll enable us to focus on algorithm development almost exclusively. ParStream is more performance oriented, putting the emphasis on parallel programming, GPU computing, and the most convenient interface is based on SQL programming, which makes it not fitted for the type of computations we want to run on the datasets.

All in all, Samoa could be the best solution if the data are already managed via streams in a distributed architecture providing a stream processing engine. However this is far from necessary, insofar as a simple column-organized dataset is enough for our analytical plans, this dataset being stored in a given repository. Again, as long as all the data (distributed throughout the architecture or simply aggregated and stored in a single dataset) can be accessed through a single well-defined interface (Stream engine interface for example), an analysis work can be performed. At the end of the day, what is needed is a way to access and freely process this data, which constitute the primary requirement.

Dataset structure

When it comes down to the dataset structure, rows and columns content could simply follow this pattern: a row is associated to a record, a column is associated to an attribute/measurement/result. For example, the row template could be: RecordID, DeviceID, locationX, locationY, locationZ, date, Speed, Temperature, Pressure, etc. In this case, column two contains all the identifiers, column three contains all the latitudes of the drones, column four contains the longitudes of the drones, etc. When a new record is made, then a row is added to the table with the associated attributes. The repository therefore behaves like a queue. If there is any storage capacity issue, several options could be considered (time window, memory window, data summaries, etc.).

This conceptual table structure is usually implanted with the combination of two files in Machine Learning: a ".names" file and a ".data" file. The first one contains all the attribute types and all the elements necessary to understand and process the content of the second file. Given these two files, data analysis can be performed.

The software implementing the algorithms to which the data will be submitted could structurally be partitioned into several modules. Each of them may be associated with one of the associated use cases and therefore have specific roles and functionalities. To enable some kind of extensibility, it should be possible to easily add new modules associated with a different algorithm if any new analysis type is needed in the future.

This software must be easily pluggable on any other dataset provided that the structure of the latter is compatible with the data structure formulated in the software specification. Any dataset following the reference organizational structure can be processed on.

It should also be fault tolerant when it comes down to data loss. The absence of one field in one record of the dataset should not compromise the entire computation. No backtracking signal to the device should be involved, but all the algorithms implemented should be able to overcome this kind of issue.

7.2 Experiment Description Languages

A subject under investigation during the subsequent phases of the RAWFIE project is the specification of a Domain Specific Language capable and expressive enough to describe complex experiments in federated UxVs testbeds. Examining previous work in this area can provide valuable input to find unambiguous ways to describe experiments enabling other users to repeat them. The possibility of augmenting previously defined experiment description languages with specific characteristics from the RAWFIE focus must also be examined.

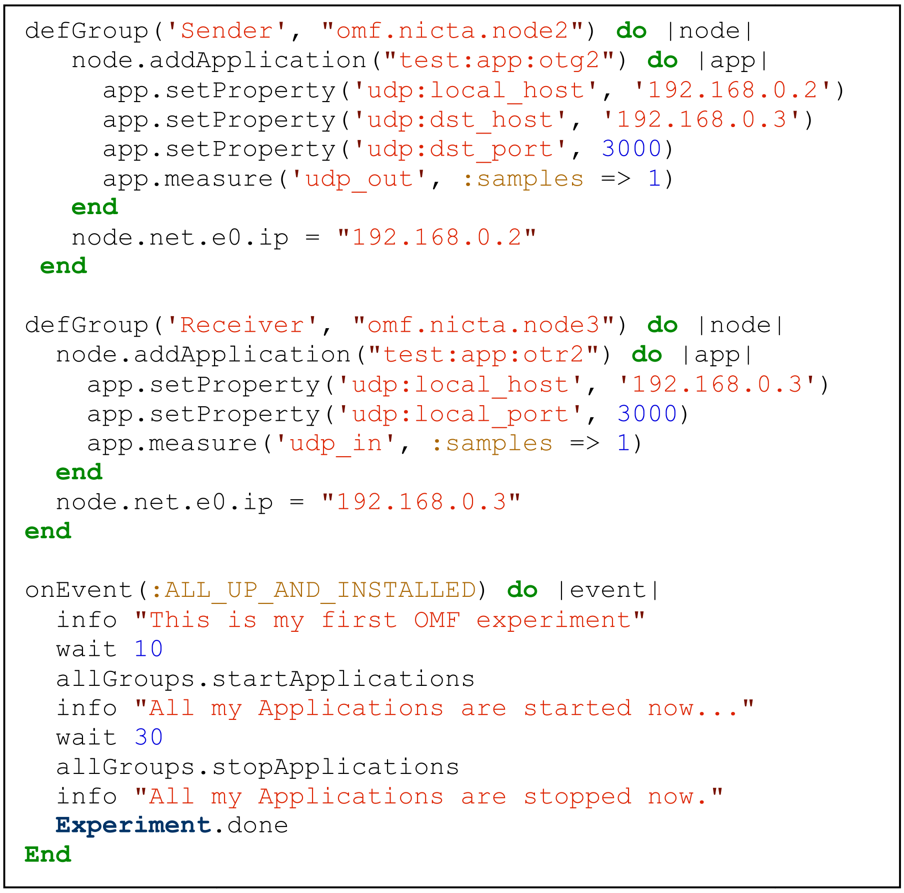
cOntrol and Management Framework (OMF)

OMF [6] is a suite of software components which provides management, control and measurement tools and services to users and operators of networking testbeds. Within OMF a comprehensive experiment description language that allows an experimenter to describe resource requirements and their configurations, as well as the experiment orchestration was developed. OMF Domain-specific Language (OEDL) is based on the Ruby scripting language [7] and uses

Ruby's meta-programming capabilities to provide experiment-specific commands and statements. An experiment description written in OEDL is composed of two parts:

- Resource Requirements and Configurations which enumerate the resources that are required by the experiment and describe the different configurations that need to be applied on them,
- Task Descriptions which are contained in a state-machine that enumerates the different events, states, and associated tasks to perform with the resources in order to realize the experiment.

The entire list of APIs can be found at the OMF page [8]. OMF is currently supported by most of the networking testbeds participating in Fed4FIRE (Virtual Wall, w-iLab.t, Norbit, NITOS). A simple example of OEDL usage is presented in Figure 6. In this experiment two networking groups each one consisting of one node have been defined along with a wired Ethernet connection between them. The 'Sender' node acts as UDP traffic generator while the 'Receiver' acts as traffic sink. OMF uses an event-based approach to describe the different actions required to perform during an experiment. In this simple example the actions performed during the experiment is the start of applications loaded on all groups ('Sender' and 'Receiver' nodes) and the stop of applications 30 sec later. The results of the experiment which are the collection of measurements from two Measurement Points (MPs, `udp_out` provided by the OTG2 application and `udp_in` provided by the OTR2 application) are stored in a SQLite database and can be accessed or further processed depending on user needs.



```
defGroup('Sender', "omf.nicta.node2") do |node|
   node.addApplication("test:app:otg2") do |app|
     app.setProperty('udp:local_host', '192.168.0.2')
     app.setProperty('udp:dst_host', '192.168.0.3')
     app.setProperty('udp:dst_port', 3000)
     app.measure('udp_out', :samples => 1)
   end
   node.net.e0.ip = "192.168.0.2"
 end

defGroup('Receiver', "omf.nicta.node3") do |node|
  node.addApplication("test:app:otr2") do |app|
    app.setProperty('udp:local_host', '192.168.0.3')
    app.setProperty('udp:local_port', 3000)
    app.measure('udp_in', :samples => 1)
  end
  node.net.e0.ip = "192.168.0.3"
end

onEvent(:ALL_UP_AND_INSTALLED) do |event|
  info "This is my first OMF experiment"
  wait 10
  allGroups.startApplications
  info "All my Applications are started now..."
  wait 30
  allGroups.stopApplications
  info "All my Applications are stopped now."
  Experiment.done
End
```

Figure 19: Experiment example in OMF Experiment Description Language

Network Experiment Programming Interface (NEPI)

Another attempt of development of a tool that enables control over different experimentation environments including simulators, emulators and testbeds is NEPI (Network Experiment Programming Interface) [9][10]. NEPI is a python library whose goal is to provide all the facilities needed to accomplish every task of a typical experimentation workflow across various testbeds:

- describe the network and application level aspects of an experiment
- enable trace collection at various key locations within an experiment
- start, monitor and stop a running experiment
- collect the trace results of an experiment once it is completed

Detailed documentation about writing scripts can be found in NEPI web page [11]. NEPI has been used for networking experimentation using Planetlab nodes and an interface for connecting OMF testbeds has also been implemented.

DES-Cript

DES_Cript is a domain specific language designed to define and describe networking experiments which can be executed on a testbed with primary focus Wireless Sensor Network testbeds [12]**Error! Reference source not found.**. DES-Cript is based on XML allowing the experiment description in a clear and hierarchical structure and introducing an abstraction layer by separating the experiment description from its actual execution. Basically a DES-Cript file contains commands that will be executed on nodes at specific time or under specific conditions. The repeatability of experiments has been taken into account in DES-Cript allowing the specification of the number iterations of experiment execution in order to enable the comparison of the results of multiple test runs of the same experiment. Reuse and rescheduling of experiments is also possible enabling the execution of experiment with different parameters. An experiment description is divided into three major sections. First a *general* section contains all meta-data of the experiment. Next follows the *init* section in which the testbed configuration prior to the experiment execution is specified. Finally the *actions* section lists all commands that will be executed in the experiment run. The structure of XML experiment description in DES-Cript is presented in Figure 20

```
<experiment>
    <general>
        <name>...</name>
        <start_time>...</start_time>
        <owner>...</owner>
        <description>...</description>
        <iterations>...</iterations>
        ...
        <groups>
            <group name="groupname" role="Server">
                <members>...</members>
            </group>
        </groups>
        ...
    </general>
    <init>
        <action id="0">
            ...
        </action>
        ...
    </init>
    <actions>
        <action_block id="1">
            <action id="1">
                <group>...</group>
                <command>...</command>
                <start_time>...</start_time>
                <duration>...</duration>
                <evaluation_script>...</evaluation_script>
            </action>
            ...
        </action_block>
        ...
    </actions>
</experiment>
```

Figure 20: Basic structure of DES-Cript

8 Conclusion

The present deliverable reflects the effort to yield a first preliminary list of system and user requirements that will support the definition of the RAWFIE architecture performed in subsequent WPs. Initially an attempt was made to define some basic terms, rules to be followed during requirement authoring and an efficient requirement elicitation methodology. The stakeholders of RAWFIE were identified and a list of representative scenarios were identified and elaborated with the purpose to support requirement analysis process. Defined requirements were classified in two (2) broad categories following a bird-eye view separation of the system to Platform and the various Testbeds (including testbed server node and UxV resources). In both cases functional and non functional requirements were listed. The requirement definition was based on a appropriate custom template 'card' inspired by the Volere template including among other fields an priority field which defines how important is the requirement for the RAWFIE system eco-system. The requirements defined in this first version of the requirement analysis were mainly based on what was prescribed in DoW, in the defined scenarios. A number of requirements were also originated from a review of work performed in other FIRE projects dealing with experimental platforms development and from the consortium expertise.

Subsequent versions of the deliverable, in future iterations, are expected to perform a more fine grained specification and analysis of requirements focusing on specific components of the identified architecture or aspects like the EDL.

The deliverable includes also brief information regarding the testbed areas, their facilities and specific types of UxV models that are initially available in RAWFIE.

9 Annex I

9.1 Platform Requirements Overview

Table 8: Platform functional Requirements table (Version 1)

ID	Category	Title	Type	Priority	Source	Version
PT-GEN-001	GEN	A web portal interface shall be provided to the users of the platform	FUNC	HIGH	DoW	1
PT-GEN-002	GEN	RAWFIE platform shall support various roles with different privileges at every level of access.	FUNC	HIGH	DoW	1
PT-GEN-003	GEN	Web portal usage shall be allowed only to authenticated users	FUNC	HIGH	DoW	1
PT-GEN-004	GEN	RAWFIE middle tier shall include a module to monitor the performance of the middle tier components.	FUNC	HIGH	Consortium	1
PT-P-001	Preparation Phase	The UI interface shall illustrate ALL the essential information of the RAWFIE federation that the experimenters should take advantage of	FUNC	HIGH	DoW	1
PT-P-002	Preparation Phase	A tutorial or similar type of documentation shall be provided to the users of the platform	FUNC	HIGH	DoW	1
PT-P-003	Preparation Phase	RAWFIE platform shall provide a Testbed Directory where all the integrated testbeds are listed	FUNC	HIGH	DoW	1
PT-P-004	Preparation Phase	A testbed registration service shall be available	FUNC	HIGH	DoW	1
PT-P-005	Preparation Phase	A spatial model for the data storage shall be provided	DATA	HIGH	DoW	1
PT-A-001	Authoring Phase	Experiment Description Language (EDL) shall be used as a language for the definition of experiment scenarios	FUNC	HIGH	DoW	1
PT-A-002	Authoring Phase	For each defined experiment a short description of the experiment and its purpose shall be defined.	FUNC	MEDIUM	Consortium	1

PT-A-003	Authoring Phase	A tool for translating EDL into user directives shall be provided	SEC	HIGH	DoW	1
PT-A-004	Authoring Phase	An experimenter shall be able to describe sensor activation or deactivation directives during experiment authoring	FUNC	HIGH	Scenario	1
PT-A-005	Authoring Phase	An experimenter shall be able to address a specific UxV resource, group of UxVs or type of UxV resource in an experiment	FUNC	MEDIUM	Scenario	1
PT-A-006	Authoring Phase	An experimenter shall be able to define the type of information to be gathered and/or stored by UxV resource(s)	FUNC	HIGH	Scenario	1
PT-A-007	Authoring Phase	An experimenter shall be able to define the type of metrics to be gathered and/or stored during an experiment and/or per UxV resource	FUNC	HIGH	Scenario	1
PT-A-008	Authoring Phase	An experimenter shall be able to provide navigation or movement directives during experiment authoring	FUNC	HIGH	Scenario	1
PT-A-009	Authoring Phase	An experimenter shall be able to provide initial conditions and/or configuration parameters for an experiment	FUNC	HIGH	Scenario	1
PT-A-010	Authoring Phase	An experimenter should be able to provide formation information for a group of UxVs resources	FUNC	MEDIUM	Scenario	1
PT-A-011	Authoring Phase	A textual editor shall be provided for the authoring of RAWFIE experiments	FUNC	HIGH	DoW	1
PT-A-012	Authoring Phase	A visual/graphical editor shall be provided for the authoring of RAWFIE experiments	FUNC	HIGH	DoW	1
PT-A-013	Authoring Phase	Spatial information shall be provided for the currently available resources for the authoring of new experiments	FUNC	HIGH	DoW	1

PT-A-014	Authoring Phase	Scenarios defined via EDL shall be validated after their authoring	FUNC	**HIGH**	DoW	1
PT-A-015	Authoring Phase	Platform shall allow saving, editing and/or deletion of a scenario/experiment defined via EDL	DATA	**HIGH**	Other	1
PT-A-016	Authoring Phase	RAWFIE platform shall provide a Resource Discovery tool for fine-grained resource searches	FUNC	**MEDIUM**	Consortium	1
PT-B-001	Booking Phase	Booking functionality shall be provided to experimenters	FUNC	**HIGH**	DoW	1
PT-B-002	Booking Phase	Booking tool shall provide an intuitive UI interface to facilitate the reservation process	FUNC	**HIGH**	DoW	1
PT-B-003	Booking Phase	Booking functionality shall allow reservation of resources involving multiple testbeds	FUNC	**HIGH**	DoW	1
PT-B-004	Booking Phase Launching Phase	Notification mechanisms may be provided for experiments scheduled for execution in the future.	FUNC	**LOW**	Consortium	1
PT-B-005	Booking Phase	Booking assistance mechanisms may be provided for ensuring fairness in resource bookings	FUNC	**LOW**	DoW	1
PT-B-006	Booking Phase	RAWFIE platform should allow virtualization of available UxVs resources during reservation process	FUNC	**MEDIUM**	Consortium	1
PT-B-007	Booking Phase Launching Phase	An accounting and billing service shall be provided that will track resource usage by individual users.	FUNC	**MEDIUM**	DoW	1
PT-L-001	Launching Phase	RAWFIE shall provide a validator to constantly check experiment scenarios during runtime	FUNC	**HIGH**	DoW	1
PT-L-002	Launching Phase	Launching tool shall be provided to experimenters allowing execution only of scenarios that have been successfully validated by the platform	FUNC	**HIGH**	DoW	1
PT-L-003	Launching Phase	Launching tool shall be kept informed upon an experiment's state	FUNC	**HIGH**	DoW	1
PT-L-004	Launching Phase	Monitoring tool shall be provided to experimenters	FUNC	**HIGH**	DoW	1

PT-L-005	Launching Phase	Visualization engine shall be part of the middleware system	FUNC	HIGH	DoW	1
PT-L-006	Launching Phase	A 3D visualization tool shall be available for the tracking of all moving resources	FUNC	HIGH	DoW	1
PT-L-007	Launching Phase	RAWFIE platform shall provide means to ensure fairness in experiments execution	FUNC	MEDIUM	Consortium	1
PT-L-008	Launching Phase	RAWFIE platform shall allow experimenters to remotely navigate UxVs.	FUNC	HIGH	DoW	1
PT-L-009	Launching Phase	RAWFIE platform shall support a semi-autonomously way of navigation of the UxVs	FUNC	MEDIUM	Consortium	1
PT-L-010	Launching Phase	An experimenter shall have the means to define actions or tasks that should run on a periodic or ad hoc basis during execution of an experiment	FUNC	MEDIUM	Scenario	1
PT-E-001	Evaluation Phase	Each run experiment should be uniquely identified within RAWFIE platform	FUNC	HIGH	DoW	1
PT-E-002	Evaluation Phase	RAWFIE platform shall include a service enabling the data collection, analysis and processing.	DATA	HIGH	DoW	1
PT-E-003	Evaluation Phase	RAWFIE platform shall allow the retrieval of data and post-analysis information for a particular experiment.	DATA	HIGH	Other	1
PT-E-004	Evaluation Phase	A data description language should be provided	DATA	MEDIUM	Consortium	1
PT-E-005	Evaluation Phase	A data analytical software should be provided	DATA	MEDIUM	Consortium	1

Table 9: Platform non-functional Requirements table (Version 1)

ID	Category	Title	Type	Priority	Source	Version
PT-NF-001		RAWFIE platform shall support secure data exchange	SEC	HIGH	DoW	1
PT-NF-002		RAWFIE platform shall provide a reservation/booking system with adequate security and privacy	SEC	HIGH	Consortium	1

PT-NF-003		RAWFIE platform should be able to support backups of all critical data	SUPP	MEDIUM	DoW	1
PT-NF-004		RAWFIE platform shall exhibit high degree of network availability	PERF	HIGH	DoW	1
PT-NF-005		RAWFIE platform shall be able to support (near) real-time information gathering from the UxV sensors	PERF	HIGH	DoW	1
PT-NF-006		RAWFIE platform shall exhibit high degree of scalability	PERF	HIGH	DoW	1
PT-NF-007		RAWFIE platform should exhibit high degree of availability	PERF	MEDIUM	DoW	1
PT-NF-008		RAWFIE architecture should be compatible to previous FIRE projects.	SUPP	MEDIUM	Consortium	1
PT-NF-009		RAWFIE architecture should adopt a modular design approach.	SUPP	MEDIUM	DoW	1
PT-NF-010		RAWFIE platform shall be deployed as a cloud based service (or list of services).	ENV	HIGH	DoW	1
PT-NF-011		RAWFIE software modules should be implemented as Web Service or as REST	SUPP	HIGH	Consortium	1
PT-NF-012		RAWFIE modules should use Open Standards and Open Software as far as possible	SUPP	MEDIUM	Consortium	1

9.2 Testbed Requirements Overview

Table 10: Testbed functional Requirements table (Version 1)

ID	Category	Title	Type	Priority	Source	Version
TB-G-001	General	The Testbed shall implement a monitoring service to observe and report the operation of the testbed’s resources	FUNC	HIGH	DoW	1
TB-G-002	General	The testbed shall be able to support static resources (sensors)	FUNC	HIGH	Consortium	1

TB-G-003	General	The Testbed should provide (near) real time communication between the sensors, the UxVs and the rest of RAWFIE infrastructure.	FUNC	MEDIUM	Consortium	1
TB-G-004	General	Each Testbed shall provide information about the capabilities of each resource node.	FUNC	HIGH	Consortium	1
TB-G-005	General	The Testbed should be able to provide accurate location information for each UxV and/or static node.	FUNC	MEDIUM	Consortium	1
TB-G-006	General	The Testbed shall provide error notification.	FUNC	HIGH	Consortium	1
TB-G-007	General	The Testbed should allow execution of multiple experiments in parallel.	FUNC	MEDIUM	Consortium	1
TB-G-008	General	The Testbed shall provide a navigation service	FUNC	MEDIUM	Consortium	1
TB-G-009	General	The Testbed shall be able to support simulated UxVs resources	FUNC	LOW	Consortium	1
TB-I-001	Interconnectivity	The UxV Testbed shall be able to communicate with the RAWFIE platform infrastructure using standard based communication means	FUNC	HIGH	Consortium	1
TB-I-002	Interconnectivity	The communication system shall be able to use UxVs to relay information to and from other UxVs	FUNC	HIGH	Consortium	1
TB-I-003	Interconnectivity	A Testbed's communication system may provide at least 3 levels of Service and the communication means will adapt to these Levels of Service	FUNC	LOW	Consortium	1
TB-I-004	Interconnectivity	The Testbed shall be able to dispatch UxV information on demand	FUNC	HIGH	DoW	1
TB-R-001	Resource	The UxV shall be able to operate autonomously	FUNC	HIGH	Consortium	1
TB-R-002	Resource	The UxV should provide collision avoidance mechanism	FUNC	MEDIUM	Consortium	1
TB-R-003	Resource	Each UxV shall have a unique Identification code	FUNC	HIGH	Consortium	1
TB-R-004	Resource	UxV's shall be able to store data on board	DATA	HIGH	Consortium	1
TB-R-005	Resource	Capability of task planning of the UxVs nodes during run-time.	FUNC	MEDIUM	Consortium	1
TB-R-006	Resource	Capability of taking the control of the UxVs from distance.	FUNC	MEDIUM	Consortium	1

TB-R-007	Resource	Each UxV node should ensure a minimum autonomy of 15-30 minutes.	FUNC	HIGH	Consortium	1
TB-R-008	Resource	Each UxV node should ensure payload.	FUNC	HIGH	Consortium	1
TB-R-009	Resource	Each UxV node should be equipped with a location identification system.	FUNC	HIGH	Consortium	1
TB-R-010	Resource	UxVs should be able to cooperate during an experiment execution.	FUNC	MEDIUM	Consortium	1
TB-R-011	Resource	UxVs should be able to Synchronize their Time-References between them	FUNC	MEDIUM	Consortium	1
TB-R-012	Resource	The UxV should provide Access Point functionality	FUNC	MEDIUM	Consortium	1
TB-R-013	Resource	Each UxV node shall be equipped with primary and secondary communication means	FUNC	HIGH	Consortium	1
TB-D-001	Data Storage	The Testbed shall be able to store data in case of transmission failure.	DATA	HIGH	Consortium	1
TB-D-002	Data Storage	The Testbed may provide statistical data/information about its operation.	DATA	LOW	Consortium	1

Table 11: Testbed non – functional Requirements table (Version 1)

Id	Category	Title	Type	Priority	Source	Version
TB-NF-G-001	General	The Testbed shall provide concurrent requests capacity	PERF	MEDIUM	Consortium	1
TB-NF-G-002	General	The Testbed infrastructure should provide reliability and robustness of all components/modules.	SUPP	MEDIUM	Consortium	1
TB-NF-G-003	General	The communication system shall offer a high availability	SUPP	HIGH	Consortium	1
TB-NF-G-004	General	The communication interfaces shall offer security mechanisms	SEC	HIGH	Consortium	1
TB-NF-G-005	General	Testbed shall abide with legal restrictions	OTH	MEDIUM	Consortium	1
TB-NF-G-006	General	The communication system shall be robust to a number of issues typical of wireless communication	PERF	HIGH	Consortium	1

TB-NF-R-001	Resource	UxVs shall offer on demand resources (Network, Sensor, Processing, and Controller).	USE	MEDIUM	Consortium	1
TB-NF-R-002	Resource	UxVs sensor system shall be compliant to connection standards and communication interfaces.	USE	MEDIUM	Standards	1
TB-NF-R-003	Resource	UxV shall be capable to revert to a safe mode	SEC	HIGH	Consortium	1

10 References

[1] I. Sommerville; Software Engineering, 7th Ed.; John Wiley and Sons, 2004.

[2] International Organization for Standardization; ISO/IEC WD 29148.3; Software and Systems Engineering – Life Cycle Processes – Requirements Engineering, 2010.

[3] Volere Template Edition 13, 2007, http://www.volere.co.uk/template.htm

[4] Patibandla, S.T.; Bakker, T.; Klenke, R.H. "Initial evaluation of an IEEE 802.11s

[5] S. Morgenthaler, T. Braun, Zhongliang Zhao, T. Staub and M. Anwander, "'UAV Net: A mobile wireless mesh network using Unmanned Aerial Vehicles," Globecom Workshops (GC Wkshps), 2012 IEEE, pp. 1603

[6] Thierry Rakotoarivelo, Max Ott, Guillaume Jourjon, Ivan Seskar, "OMF: a control and management framework for networking testbeds", in ACM SIGOPS Operating Systems Review 43 (4), 54-59, Jan. 2010.

[7] http://www.ruby-lang.org

[8] http://omf.mytestbed.net

[9] Mathieu Lacage, Martin Ferrari, Mads Hansen, Thierry Turletti. NEPI: Using Independent Simulators, Emulators, and Testbeds for Easy Experimentation, ROADS 2009

[10] Alina Quereilhac, Mathieu Lacage, Claudio Freire, Thierry Turletti and Walid Dabbous. NEPI: An integration framework for Network Experimentation, in proceedings of *19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2011*

[11] http://nepi.inria.fr

[12] http://www.des-testbed.net/node/231

[13] European Remotely-Piloted Aircraft Systems (RPAS) Steering Group (ERSG) Roadmap for the integration of civil Remotely-Piloted Aircraft Systems into the European Aviation System, Final report from the European RPAS Steering Group (June 2013) http://ec.europa.eu/enterprise/sectors/aerospace/files/rpas-roadmap_en.pdf

[14] European Commission, "A new era for aviation: Opening the aviation market to the civil use of remotely piloted aircraft systems in a safe and sustainable manner", Communication from the Commission to the European Parliament and the Council, http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52014DC0207&from=EN, COM(2014) 207 final

[15] ParStream – https://www.parstream.com/

[16] RapidMiner – https://rapidminer.com/

[17] Apache Samoa – http://samoa.incubator.apache.org/

[18] Apache Storm – https://storm.apache.org/

[19] Apache Samza – http://samza.apache.org/

[20] Apache S4 – http://incubator.apache.org/s4/

[21] http://www.riot.ch/legal-information-about-flying-multicopter-drones-commercial/

[22] http://www.developpement-durable.gouv.fr/Quelle-place-pour-les-drones-dans.html (in French)

[23] https://okeanos.grnet.gr/home/