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	D6.2: RAWFIE Platform Validation (a)		
Confidentiality	PU	Deliverable type[1]	R
Deliverable File	D6.2	Date	31.07.2016
Approval Status[2]	WP Leader	Version	1.0
Contact Person	Marcel Heckel	Organization	Fraunhofer
Phone	+49 351 / 4640-645	E-Mail	marcel.heckel@ivi.fraunhofer.de

[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
Marcel Heckel	Fraunhofer	marcel.heckel@ivi.fraunhofer.de
Philippe Dallemagne	CSEM	Philippe.dallemagne@csem.ch
Kakia Panagidi	UoA	kakiap@di.uoa.gr
Kostas Kolomvatsos	UoA	kostasks@di.uoa.gr
Giovanni Tusa	IES	g.tusa@iessolutions.eu
Vasil Kumanov	Epsilon Bulgaria	Vasil.kumanov@epsilon-bulgaria.com
Nikolaos Pringouris	HAI	Priggouris.nikolaos@haicorp.com
Jason Ramapuram	HES-SO	Jason-emmanuel.ramapuram@hesge.ch

REVIEWERS TABLE

Name	Company	E-Mail
Giovanni Tusa	IES	g.tusa@iessolutions.eu
Kakia Panagidi	UoA	kakiap@di.uoa.gr
Kostas Kolomvatsos	UoA	kostasks@di.uoa.gr
Michail Chatzidakis	UoA	mhatzi@di.uoa.gr

DISTRIBUTION

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

CHANGE HISTORY

Version	Date	Reason for Change	Pages/Sections Affected
0.1	2016-06-13	TOC / Initial version	all

[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).

0.2	2016-06-30	Refined structure	all
0.3	2016-07-12	Validation by requirements: table added	Section 3
0.4	2016-07-15	Methodology and questionnaire updated	Section 2, 4, A
0.5	2016-07-18	Roadmap defined	Section 7
0.6	2016-07-20	Performance and technical evaluation updated	Section 2, 6
0.7	2016-07-25	Showcase scenarios updated	Section 5
0.8	2016-07-27	Review all contributions	all
0.9	2016-07-29	1st Review	all
0.10	2016-07-30	2nd Review	all
0.11	2016-07-31	Handle review comments	all
1.0	2016-07-31	Final version	all

Abstract:

The objective of this deliverable is a report on the validation of the RAWFIE platform. It describes the validation and evaluation procedures and their outcomes.
The document will be released as a live document in three phases/cycles according to the roadmap.
This deliverable is based on the validation plan setup in D4.3 and on the results of tasks T6.1 and T6.2.

Keywords: tests, validation, evaluation, methodology, requirements, showcase, questionnaires, interviews

Part II: Table of Contents

List of Figures

List of Tables

Part III: Executive Summary

The objective of this deliverable is a report on the validation and evaluation of the RAWFIE platform.

The first chapter introduces the methodology that was used and will be used for the next iteration of this document. Not all mentioned methods are applied in this iteration of the deliverable, due to the early state of the system. However, they are still mentioned for completeness and will be used in the following iterations.

The validation starts with a list stating which of the requirements from D3.2 are currently met. This gives a high-level overview of the state of the system.

Then, a questionnaire to be submitted to the end users is described in short. The current version of the questionnaire is mainly aimed at getting feedback from the interviewed end users, therefore to understand whether the building of the RAWFIE platform is on the right track, as well as to identify potential areas of improvements.

A dedicated chapter describes the showcase that was presented to the users. An execution of the validation scenarios of D4.6 was not feasible, as not all necessary components were ready. Based on the showcase however, the already mentioned questionnaire was completed by the users and the results and conclusions also presented in this chapter. The results of the questionnaire showed that RAWFIE is on the right track, but a lot of work is yet to be done.

Finally, an initial performance and technical evaluation of the RAWFIE system is done, mainly focusing on the latency and round trip time when using the message bus communication, in 2 different environments: a simulated one, with generic, no actual RAWFIE components used, and a more realistic one, where actual RAWFIE publish / consuming software, running on actual UxVs, was used. Such preliminary tests will help the consortium in order to have a general idea of the performance when using this specific communication pattern. More tests have to be carried out in the coming months during the second validation iteration, when the quantitative validation metrics will be more exhaustively evaluated against specific success criteria, that are going to be defined with Testbeds and UxVs owners.

The last chapter gives a short roadmap of the validation steps along with the conclusion and outlook.

Part IV: Main Section

1 Introduction

1.1 Scope of D6.2

This deliverable presents the approach and the results of the first evaluation and validation of the RAWFIE system. In addition to verification ("Are we building the product right?"), the validation ("Are we building the right product?") also benefits from end-user feedback. However, until now no direct validation by end-user has been done because essential components required will be completed in the current implementation period. Also, most of the validation scenarios outlined in D4.3 cannot be executed completely because several implementations are not ready and will be realised in the ongoing implementation phase. However, a showcase was prepared, where the current state of the system was presented to end-users, and a questionnaire followed the presentation. The evaluation of the system performance is conducted in this deliverable for a limited number of metrics and specific workflows, as described in Section 6. Future versions of the deliverable (starting with D6.4), will provide more comprehensive technical evaluation results, based on the quantitative metrics defined in D4.3 and related success criteria, that will be defined in D4.6.

Therefore, this deliverable focuses on the following aspects:

- Define the exact methodology to realise the validation by taking into account the validation scenarios and metrics defined in D4.3.
- Validate which requirements from D3.2 are currently met.
- Prepare the steps needed for end-user validation e.g., questionnaires.
- Describe a first showcase for the end-users based on the current system prototype.
- Evaluate the questionnaires that were filled out after the showcase.
- Preform an initial performance and technical evaluation of the system.
- Define a roadmap on how the validation will be realised in the next versions of this deliverable.

1.2 Relation to other deliverables

The methodology of D6.2 is based on the outcome of D4.3 and it checks if the validation-related requirements defined in D3.1/D3.2 are met.

D6.4 will be the second version of the "RAWFIE Platform Validation" deliverable. It will contain the end-user feedback, especially from the users of the first Open Call. The validation scenarios and validation template of D4.3 will be used to perform the validation test. It will also contain the evaluation, based on the metrics defined in D4.3 and related success criteria that are going to be defined in D4.6.

2 Methodology

The evaluation in this and the following deliverables will be based on the following methods

- Formally check which requirements are met (D3.2).
- Let end-user operate the system – under guidance and supervision – using the validation scenarios from D4.3 and following.
- Perform questionnaires fill-out and interviews with the end-user after using the system.
- Evaluate the data collected from observations and questionnaires/interviews.
- Execute performance and technical evaluation, using metrics and stress tests described in D4.3 and following.

In this deliverable the actual operation of the system is presented by a showcase that shows the current possibilities of the system to the users (the users will not interact with the system at this stage). This is unavoidable as several implementations are not ready and many validation scenarios cannot be executed. Based on the showcase, the end-users will fill out the questionnaire.

2.1 Check which requirements are met

The check of requirements is reported in a table where each requirement corresponds to a row. For each requirement it is stated if it is currently met or not. If a requirement is not met, a short comment field must be filled out explaining the reason.

The table provides a quick overview about the readiness of the system.

2.2 Observing the end-user while operating the system

The end users will operate the system by execution of the validation scenarios defined in D4.3 (and following documents). For each step it will be recorded whether the step could successfully be executed or not and a comment will be added in case of an error. Also, the metrics defined in D4.3 (and following documents) will be checked whether they are met during the execution of the scenario or not and then used for the evaluation against the corresponding success criteria.

While the end users operate the system they will be observed [3]. The observation may be done in two different ways:

- Direct watching the users
    - For this the observer will directly watch the end-user while operating the system.
    - This may also be recoded via a video camera.
    - The observer constantly takes notes of the actions and failures of the user.
    - The observer may also answer questions of the user, if the user in not able to perform the scenario steps on his/her own (this will also be noted).
    - For small number of users.

- Recording remote actions on the system
  - If the users interact with RAWFIE via the web portal, their actions on the web page will be recorded for later evaluation. This can be done with an website analysis tool like Piwik4, Etracker5, Reinvigorate6, Mint7 or Open Web Analytics8 (the decision which one to use will be made in the near future).
  - These systems collect different statistics like how long a user stayed on a page, which pages were visited more frequently, or section on page which was clicked frequently.
  - For large amount of users.

The main advantage of observing users is that it "does not rely on people's willingness or ability to provide information" [3]. On the downside, you may not understand "why people behave as they do". For direct watching this can be compensated by querying the users afterwards for the reason of the unusual behaviour. Another disadvantage of direct watching is that it is expensive and time-consuming.

2.3 Questionnaires and interviews

Questionnaires [2] and interviews [4] will be performed after the users have interacted with or have seen the RAWFIE system. Questionnaires can be filled out on paper or online (online questionnaires will reduce the amount of work to evaluate them). Interviews will be based on the questionnaires but may also collect additional information should they occur during the conversation.

The questionnaire uses closed-ended (include a list of predetermined answers from which participants can choose) and open-ended questions (free text answers). Open-ended questions are used when possible answers are not known. But they are time-consuming to fill in and to analyse. Therefore, closed-ended questions are preferred as they are easier to fill in and to analyse. Additionally, most closed-ended questions will have an "other" option with a free text answer, to cover non-foreseen information.

2.4 Evaluation of data from observations and questionnaires/interviews

The evaluation of the collected data from the observations and questionnaires/interviews analyses two main categories of data: qualitative data [5] (data in information in non-numeric form) and quantitative data [6] (data in information in numeric form). Another category is

[4] https://piwik.org/
[5] https://www.etracker.com
[6] https://www.reinvigorate.net
[7] http://haveamint.com/
[8] http://www.openwebanalytics.com/ T

formed by the occurred errors and unexpected behaviour of the system during the validation scenarios.

2.4.1 Qualitative data

Qualitative data mainly results from the direct watching the users and open-ended questions. The analysis of qualitative data relies heavily on interpretation and is time-consuming and labour-intensive. The process will adopt the schema described in [5]

- *Review the data*: review the data several times until one has a general understanding on the ideas.
- *Organize the data*: Group data based on topics, stakeholder or date.
- *Code the data*: Identify and label common trends or ideas that appear repeatedly.
- *Interpret the data*: start by key themes/ideas and factor these themes/ideas by revisiting the review.

2.4.2 Quantitative data

Quantitative data is mainly a result of action recording on the system or of closed-ended questions. Quantitative data can easily be visualized by using graphs and charts [1]. For example pie charts could be used at some preferred options of the system or bar chars to show the percentage of acceptance of different parts of the system.

2.4.3 Errors and unexpected behaviour

Errors and unexpected behaviour during the execution of the validation scenarios will be recorded and documented for each validation scenario step. These will be handled and fixed in later implementation phases.

2.5 Performance and technical evaluation

The technical evaluation will be based on the measurements of quantitative metrics, as the ones already defined and classified in deliverable D4.3, and further improvements / modifications that will be elaborated in subsequent iterations (e.g. in the D4.6). Such metrics will be for example, all the ones belonging to the “Interconnectivity/Data communication” category (section 4.10 of D4.3).

The methodology for performance and technical evaluation will include the following steps:

- Definition of specific test cases, according to the metric/s that should be measured and evaluated
    - test cases may be the validation scenarios already defined in D4.3, or just a subset of the steps and workflows included on each validation scenario, meaning that the metrics under investigation can be measured, in such cases, while running the validation scenarios themselves.

  - or they may consist of dedicated tests and workflows (e.g. stress tests with real or simulated sending/receiving components)
- Preparation of the software components for monitoring and collecting the investigated metrics, also using external monitoring tools if needed and where applicable, to record (persist) the results.
- Execution of the tests while recording the results.
- Analysis of the results and final evaluation of the metrics against the defined "success criteria" for each of them.

Success criteria are the criteria to evaluate if the recorded results about a specific metric, meet the expectations and are in line with the performance requirements. For example, in the case of the "End-to-End Latency" metric, a possible success criterion would be that if the recorded values are below the 20ms threshold, then the evaluation is positive. In the opposite case the evaluation is negative, normally meaning that some adjustments/interventions at the software or hardware level, is needed.

3 Validation by requirements

The following Table 1 lists all requirements from D3.2 and states if they are currently met or not. The "OK" column contains a Y(yes) in the requirement is met and a N (no) if not.

Regarding the development plan the most planed features are fulfilled, except the booking functionalities (which are missing completely until now):

- Exploring testbeds & UxVs
- Define and execute a simple mission (e.g. drive a cycle and send some measurements) via EDL
- Send UxV position and measurements to middleware
- Visualization of experiments and measurements in GUI
- Execute a simple data analysis scenario

However, there is still a lot of work to be done, to enable the full workflow from exploring testbeds & UxVs, booking UxVs, write the EDL script, executed the experiment automatically, visualize experiment, store measurement, analyse and visualize measurements and charge the experimenter for resource usage. The upcoming development iteration will go a big step forward in this direction.

No	ID	Component	Title	OK	Comment
1	PT-GEN-R-001	General	RAWFIE Platform should adopt Sliced Federated Architecture (SFA)	N	Planned for 2nd dev. iteration
2	PT-GEN-R-002	General	RAWFIE platform shall support various roles with different privileges at every level of access.	N	Roles not evaluated
3	PT-GEN-R-003	General	The RAWFIE Data model should include all basic entities that are used or/and exchanged by the various components of the RAWFIE Platform	Y	
4	PT-GEN-R-004	General	RAWFIE platform shall provide appropriate data storage for information that needs to be persisted, exchanged, or analysed by the various tools and services.	Y	POSTGRES Database used for storage
5	PT-WEB-P-001	Web Portal Tool	A web portal interface shall be provided to the users of the platform to access almost all main functionalities.	Y	Main access to implemented services and tools is achieved via a

					web portal
6	PT-WEB-P-002	Web Portal Tool	Web portal usage shall be allowed only to authenticated users	Y	
7	PT-WEB-P-003	Web Portal Tool	A tutorial or similar type of documentation shall be provided to the users of the platform	N	A Wiki Tool will be integrated in the next iteration.
8	PT-BOO-T-001	Booking Tool	Booking Tool should allow booking of resources at the experimenter level for a specified period and for selected resources	N	Planned for 2nd dev. iteration
9	PT-BOO-T-002	Booking Tool	Booking Tool functionality shall be compatible with the SFA myslice architecture and the notion of slices reservations	N	Planned for 2nd dev. iteration
10	PT-BOO-T-003	Booking Tool	Booking Tool should delegate all its actions related to Booking of a resource to the Booking Service	N	Planned for 2nd dev. iteration
11	PT-BOO-T-004	Booking Tool	Booking Tool may also interact with the Testbeds Directory Service in order to retrieve information on unallocated testbed resources	N	Planned for 2nd dev. iteration
12	PT-BOO-T-005	Booking Tool	Booking Tool should communicate with the underline services using JSON formatted messages (through an RPC or REST API)	N	Planned for 2nd dev. iteration
13	PT-BOO-T-006	Booking Tool	Booking Tool should provide appropriate functionality for viewing the reservations of a user/experimenter	N	Planned for 2nd dev. iteration
14	PT-BOO-T-007	Booking Tool	Booking Tool should allow editing of existing Reservations	N	Planned for 2nd dev. iteration
15	PT-BOO-T-008	Booking Tool	Booking Tool should allow cancellation of existing Reservations	N	Planned for 2nd dev. iteration
16	PT-BOO-T-009	Booking Tool	Booking Tool should allow creation of bookings through an intuitive UI interface	N	Planned for 2nd dev. iteration
17	PT-BOO-T-010	Booking Tool	Appropriate notification mechanism should be provided to the user in case status of reservation request is not directly available.	N	Planned for 2nd dev. iteration
18	PT-BOO-T-011	Booking Tool	Booking Tool may provide assistance of feedback to the potential experimenter during the booking process	N	Planned for 3nd dev. iteration
19	PT-BOO-	Booking Tool	Booking functionality should provide means to ensure	N	Planned for 2nd dev.

	T-012		fairness in resource booking as well as protect for malevolent actions that a user may perform.		iteration
20	PT-BOO-T-013	Booking Tool	RAWFIE platform should allow virtualization of available UxVs resources during reservation process	N	To be checked if feasible during 3rd iteration
21	PT-SYM-T-001	System Monitoring Tool	Listing and/or visualisation of current system health status shall be available	Y	
22	PT-SYM-T-002	System Monitoring Tool	The current system health status should be grouped thematically.	N	Currently only one list for all
23	PT-SYM-T-003	System Monitoring Tool	Filtering of the accessible component health statuses by user roles/rights should be possible.	N	Will be implemented in the next iteration
24	PT-SYM-T-004	System Monitoring Tool	The health statuses webpage should be updated automatically.	Y	
25	PT-SYM-T-005	System Monitoring Tool	The health status information should include a severity indication and possibly textual information with additional details.	Y	
26	PT-REE-T-001	Resource Explorer Tool	The UI interface shall illustrate testbed and UxV information of the RAWFIE federation that the experimenters should take advantage of	Y	
27	PT-REE-T-002	Resource Explorer Tool	Registration of testbeds and UxVs may be possible via the Web Portal	N	Planned for 2nd dev. iteration
28	PT-REE-T-003	Resource Explorer Tool	RAWFIE platform should provide a Resource Discovery tool for fine-grained resource searches	Y	
29	PT-REE-T-004	Resource Explorer Tool	Link to the Booking Tool should be provided	N	Booking Tool not implemented. Planned for 2nd dev. iteration
30	PT-EXA-T-001	Experiment Authoring Tool	Experiment Description Language (EDL) shall be used as a language for the definition of experiment scenarios	Y	A first version of EDL is available
31	PT-EXA-T-002	Experiment Authoring Tool	The EDL should allow the definition of all necessary requirements for an experiment	Y	The EDL already fulfils the requirement

32	PT-EXA-T-003	Experiment Authoring Tool	For each defined experiment specific metadata, i.e. name, version, date and description shall be defined.	Y	The EDL already fulfils the requirement
33	PT-EXA-T-004	Experiment Authoring Tool	An experimenter shall be able to provide initial conditions and/or configuration parameters for an experiment	Y	The EDL already fulfils the requirement
34	PT-EXA-T-005	Experiment Authoring Tool	An experimenter shall be able to manage/guide the available booked resources during experiment authoring	Y	The EDL already fulfils the requirement
35	PT-EXA-T-006	Experiment Authoring Tool	An experimenter shall be able to define the type of information to be gathered and/or stored by UxV resource(s)	N	Planned for 2nd dev. iteration
36	PT-EXA-T-007	Experiment Authoring Tool	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	N	Planned for 2nd dev. iteration
37	PT-EXA-T-008	Experiment Authoring Tool	An experimenter shall be able to provide navigation or movement directives during experiment authoring	Y	The available editors offer this functionality
38	PT-EXA-T-009	Experiment Authoring Tool	An experimenter should be able to provide formation information for a group of UxVs resources	N	The current version of the EDL partially covers this requirement
39	PT-EXA-T-010	Experiment Authoring Tool	A textual editor shall be provided for the authoring of RAWFIE experiments	Y	The RAWFIE already fulfils the requirement
40	PT-EXA-T-011	Experiment Authoring Tool	A visual/graphical editor shall be provided for the authoring of RAWFIE experiments	N	Planned for 2nd dev. iteration
41	PT-EXA-T-012	Experiment Authoring Tool	Platform shall allow saving, editing and/or deletion of an experiment defined via EDL	Y	The RAWFIE already fulfils the requirement
42	PT-EXA-T-013	Experiment Authoring Tool	The visual editor should allow the definition of movement and location waypoints in a map	Y	The RAWFIE already fulfils the requirement
43	PT-EXA-T-014	Experiment Authoring Tool	During authoring of an experiment selection of resources should be limited only to the ones previously reserved from the user at the foreseen time of experiment	N	Planned for 2nd dev. iteration
44	PT-EXA-T-015	Experiment Authoring Tool	Validation of EDL script should be possible prior to or during saving	Y	The RAWFIE already fulfils the requirement
45	PT-EXA-T-016	Experiment Authoring Tool	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	N	Planned for 2nd dev. iteration
46	PT-EXM-	Experiment	Experiment Monitoring Tool shall provide overview of	N	Experiment Monitoring

	T-001	Monitoring Tool	experiments of a user		Tool not implemented
47	PT-EXM-T-002	Experiment Monitoring Tool	Experiment Monitoring and Visualisation should be integrated	N	Experiment Monitoring Tool not implemented
48	PT-EXM-T-003	Experiment Monitoring Tool	Cancellation of running experiments should be possible via Web Portal	N	Experiment Monitoring Tool not implemented
49	PT-NAV-T-001	UxV Navigation Tool	This component will provide to the user the ability to remotely navigate a squad of UxVs through a user friendly interface.	N	Navigation tool not implemented
50	PT-NAV-T-002	UxV Navigation Tool	The tool should provided some validation of user's instructions	N	Navigation tool not implemented
51	PT-NAV-T-003	UxV Navigation Tool	UxV Navigation Tool should be available for the navigation of all moving resources	N	Navigation tool not implemented
52	PT-NAV-T-004	UxV Navigation Tool	UxV Navigation Tool should be available to read from the database a detailed version of the map of the available areas	N	Navigation tool not implemented
53	PT-VIS-T-001	Visualisation Tool	The Visualisation Tool shall allow the visualisation of information about the running experiments, in tabular/graphical form	Y	
54	PT-VIS-T-002	Visualisation Tool	A 3D visualization should be available for the tracking of all moving resources	N	
55	PT-VIS-T-003	Visualisation Tool	The Visualisation Tool may allow visualisation of video streams coming from the experiment, and experiment's camera control	N	Planned for 2nd dev iteration
56	PT-VIS-T-004	Visualisation Tool	The Visualisation Tool shall provide access to information UxV device on the geographic map	Y	
57	PT-VIS-T-005	Visualisation Tool	The Visualisation Tool shall allow organization and manipulation of multiple geographic layers	Y	
58	PT-VIS-T-	Visualisation	Possibility of Adding/Removing/Updating graphical widgets	Y	

	006	Tool	should be provided		
59	PT-VIS-T-007	Visualisation Tool	Possibility to display both actual and expected UxVs' route and position should be provided	Y	
60	PT-DAA-T-001	Data Analysis Tool	Analysis tool will provide interface to data engine.	N	Planned for 2nd dev iteration
61	PT-DAA-T-002	Data Analysis Tool	Analysis tool will provide access to past experiments	Y	Graphite is in place
62	PT-DAA-T-003	Data Analysis Tool	Analysis tool will provide ability to query message bus streams	N	Planned for 2nd dev iteration
63	PT-DAA-T-004	Data Analysis Tool	Analysis tool will provide interface to end running jobs	Y	Access to spark master is in place
64	PT-DAA-T-005	Data Analysis Tool	Analysis tool will provide a simple metric selection interface, a view of the result stream & the job status tab	N	Planned for 2nd dev iteration
65	PT-DIR-S-001	Testbeds Directory Service	The Testbed Directory Service shall provide access to information on all Testbeds registered in RAWFIE	Y	
66	PT-DIR-S-002	Testbeds Directory Service	The Testbed Directory Service should provide access to information on all Testbeds registered in RAWFIE according to predefined filters	N	Planned for 2nd dev iteration
67	PT-DIR-S-003	Testbeds Directory Service	The Testbed Directory Service shall provide access to information about available resources (UxVs) belonging to the testbeds registered in RAWFIE	Y	
68	PT-DIR-S-004	Testbeds Directory Service	The Testbed Directory Service should provide access to information on available resources (UxVs) belonging to the testbeds registered in RAWFIE, and according to predefined filters	N	Planned for 2nd dev iteration
69	PT-DIR-S-005	Testbeds Directory Service	The Testbed Directory Service shoud provide the possibility to register new testbeds in the RAWFIE platform, as well as to unregister (delete) testbeds from the platform	Y	
70	PT-DIR-S-006	Testbeds Directory Service	Some basic query capabilities should be provided	N	Planned for 2nd dev iteration
71	PT-DIR-	Testbeds	The Testbed Directory Service shall provide the possibility to	Y	

	S-007	Directory Service	register new resources belonging to a specific testbed in the RAWFIE platform, as well as to unregister (delete) resources		
72	PT-CPV-001	EDL Compiler and Validator	A tool for translating EDL into user directives shall be provided	Y	A first version is available. To be improved in 2nd dev. iteration
73	PT-CPV-002	EDL Compiler and Validator	An experimenter should have the opportunity to use a code generation engine	Y	The RAWFIE already fulfils the requirement
74	PT-CPV-003	EDL Compiler and Validator	Experiments defined via EDL shall be validated after their authoring	Y	The RAWFIE already fulfils the requirement
75	PT-CPV-004	EDL Compiler and Validator	The compiler and validator should communicate with the authoring tool in order to transfer error indications and hints for solving them	Y	The RAWFIE already fulfils the requirement
76	PT-EXV-S-001	Experiment Validation Service	RAWFIE shall provide a validator to constantly check experiment scenarios during runtime	Y	The RAWFIE already fulfils the requirement
77	PT-EXV-S-002	Experiment Validation Service	The validation service should perform syntactic checking	Y	The RAWFIE already fulfils the requirement
78	PT-EXV-S-003	Experiment Validation Service	The validation service should perform semantic checking	Y	The RAWFIE already fulfils the requirement
79	PT-USR-S-001	Users & Rights Service	User login credentials checking shall be provided	Y	
80	PT-USR-S-002	Users & Rights Service	RAWFIE platform shall support various roles with different privileges at every level of access.	Y	
81	PT-USR-S-003	Users & Rights Service	The Users & Rights Service may provide a proxy service for web application that do not check access rights.	N	To be checked if needed
82	PT-BOO-S-001	Booking Service	Booking Service should support reservations of resources at both user level and experiment level	N	Planned for 2nd dev. iteration
83	PT-BOO-S-002	Booking Service	User level booking should be triggered by the Booking Tool via a REST API.	N	Planned for 2nd dev. iteration
84	PT-BOO-S-003	Booking Service	Experiment level booking should be triggered by the experimenter before issuing a manual or schedule launching	Y	During experiment authoring selection of

			of a validated experiment		resources is available only from a user reservation
85	PT-BOO-S-004	Booking Service	Experiment level booking should support both immediate booking as well as booking at a future time	N	
86	PT-BOO-S-005	Booking Service	Booking Service should provide all the necessary methods to manage the bookings including addition, modification and cancellation/deletion operations	N	Planned for 2nd dev. iteration
87	PT-BOO-S-006	Booking Service	Booking Service should be able to compute and return feedback on conflicting bookings for a provided booking request	N	Planned for 2nd dev. iteration
88	PT-BOO-S-007	Booking Service	Reservation Data should be persisted in order to survive service failures and be available by other services	N	Planned for 2nd dev. iteration
89	PT-BOO-S-008	Booking Service	Historical data retrieval for Bookings/Reservations should be available on demand	N	Planned for 2nd dev. iteration
90	PT-BOO-S-009	Booking Service	Booking functionality shall support reservation of resources involving multiple testbeds	N	Planned for 3rd dev. iteration
91	PT-BOO-S-010	Booking Service	Booking functionality should be able to correctly handle simultaneous Reservations requests by end users	N	Planned for 2nd dev. iteration
92	PT-BOO-S-011	Booking Service	Notification mechanisms may be provided for experiments scheduled for execution in the future.	N	Planned for 3rd dev. iteration
93	PT-LAU-S-001	Launching Service	Launching Service should support short-term or manual launching of an experiment initiated directly by an experimenter	Y	
94	PT-LAU-S-002	Launching Service	Launching Service should support long-term or scheduled launching of an experiment initiated directly by an experimenter	N	Planned for 2nd dev. iteration
95	PT-LAU-S-003	Launching Service	Each executing experiment should be uniquely identified within RAWFIE ecosystem	Y	
96	PT-LAU-S-004	Launching Service	During launching it must be ensured that the experiment to be started has been validated based on spatio-temporal constraints	N	Planned for 2nd dev. iteration (needs Experiment Validation Service to be available)
97	PT-LAU-	Launching	During launching it must be ensured that the experiment to be	Y	

	S-005	Service	started belongs to an authorized user of the RAWFIE platform		
98	PT-LAU-S-006	Launching Service	The Launching Service should be able to address simultaneous requests for starting an experiment	Y	
99	PT-LAU-S-007	Launching Service	The Launching Service should send an appropriate message upon successful starting of an experiment	Y	
100	PT-LAU-S-008	Launching Service	The Launching Service may interact with other components or database services in order to retrieve information needed for deciding on launching an experiment	Y	
101	PT-LAU-S-009	Launching Service	Interactions of the launching service with database services and/or other components should respect the RAWFIE platform boundary	Y	
102	PT-LAU-S-010	Launching Service	Launching service should support requests for experiment cancellation	Y	
103	PT-LAU-S-011	Launching Service	RAWFIE platform shall provide means to ensure fairness in experiments execution	N	
104	PT-LAU-S-012	Launching Service	Launching service should provide appropriate feedback to the requested entity regarding failures on fulfilling a request	N	Planned for 2nd dev. iteration
105	PT-LAU-S-013	Launching Service	Launching service should not alter or modify any information related to the actual execution of an experiment	Y	
106	PT-VIS-E-001	Visualisation Engine	The Visualization Engine shall handle the communication with the Message Bus, for the information that will be coming from the UxVs	Y	
107	PT-VIS-E-002	Visualisation Engine	The Visualization Engine shall provide a GIS server capable of handling geographical layers (overlays)	Y	
108	PT-VIS-E-003	Visualisation Engine	The Visualization Engine may allow cache of data for faster access to the available geographic layers	N	Not planned for now, we do not have in house maps for that
109	PT-VIS-E-004	Visualisation Engine	The Visualization Engine shall provide the possibility to reply experiments using historical data	N	Planned for 2nd dev. iteration
110	PT-EXP-C-001	Experiment Controller	Cancellation of running experiments should be possible	N	Experiment Controller not implemented
111	PT-EXP-C-002	Experiment Controller	RAWFIE platform shall allow experimenters to remotely navigate UxVs.	N	Experiment Controller not implemented

112	PT-EXP-C-003	Experiment Controller	The Experiment Controller shall support the execution of experiments that involve multiple testbeds	N	Experiment Controller not implemented
113	PT-EXP-C-004	Experiment Controller	The Experiment Controller shall be able to support multiple experiments running	N	Experiment Controller not implemented
114	PT-EXP-C-005	Experiment Controller	The Experiment Controller shall be able to analyse the whole experiment script and dispatch the appropriate parts to each responsible testbed facility	N	Experiment Controller not implemented
115	PT-EXP-C-006	Experiment Controller	The Experiment Controller shall support receiving feedback at regular intervals from all testbed facilities about the progress of the experiment in this time interval	N	Experiment Controller not implemented
116	PT-EXP-C-007	Experiment Controller	The Experiment Controller shall be able to override the order of instructions described in the input script while the experiment is running	N	Experiment Controller not implemented
117	PT-EXP-C-008	Experiment Controller	The Experiment Controller shall be able to continuously feed the front-end tier (Experiment Monitoring Tool) giving the experimenter a clear view of the experiment workflow as a whole	N	Experiment Controller not implemented
118	PT-EXP-C-009	Experiment Controller	The Experiment Controller shall send distinct error and warning messages in every case the experiment's state diverges from the aimed target	N	Experiment Controller not implemented
119	PT-DAA-S -001	Data Analysis Engine	Analysis engine will support accepting of analysis jobs	Y	Via distribution from Zeppelin or JAR submit
120	PT-DAA-S -002	Data Analysis Engine	Analysis engine will support compiling analysis jobs	Y	Via Apache Zeppelin
121	PT-SYM-S-001	System Monitoring Service	RAWFIE middle tier shall include a module to monitor the performance of the middle tier components.	Y	
122	PT-SYM-S-002	System Monitoring Service	RAWFIE Testbeds and UxVs statuses should be monitored	N	UxVs statuses currently not sent by the Monitoring Manager of the testbed
123	PT-SYM-S-003	System Monitoring	RAWFIE system administrators should be informed if critical components are down	N	Need to be configured in Icinga

		Service			
124	PT-SYM-S-004	System Monitoring Service	User may register for notifications if special components are down	N	Need to be configured in Icinga
125	PT-SYM-S-005	System Monitoring Service	Notifications about planned downtimes	N	Need to be configured in Icinga
126	PT-ACC-S-001	Accounting Service	The accounting service should be capable to accept different cost models regarding RAWFIE usage on a per service basis	N	Accounting Service not implemented
127	PT-ACC-S-002	Accounting Service	The accounting service should be capable to gather statistics regarding usage of the platform by experimenters.	N	Accounting Service not implemented
128	PT-ACC-S-003	Accounting Service	The RAWFIE platform should record information related to time and type of access for a service by a user.	N	Accounting Service not implemented
129	PT-ACC-S-004	Accounting Service	The cost model used may take into consideration the overall time of experiments executed by a user of the platform.	N	Accounting Service not implemented
130	PT-ACC-S-005	Accounting Service	The accounting service may support different types of charging based on the type of the experimenter (industrial, research, university etc.)	N	Accounting Service not implemented
131	PT-ACC-S-006	Accounting Service	The accounting service may support predefined types of memberships regarding usage of the platform that may depend on various types of parameters	N	Accounting Service not implemented
132	PT-ACC-S-007	Accounting Service	The accounting service should be able to handle the addition of new services that may be incorporated in the RAWFIE platform during time.	N	Accounting Service not implemented
133	TB-GEN-R-001	General	Each UxV Testbed should provide a Slice Interface for federating their capabilities/resources to the experimenter.	N	
134	TB-GEN-R-002	General	Each Testbed should provide the exact boundaries within which its UxVs can operate	Y	
135	TB-GEN-R-003	General	Testbed areas should at least be able to host/operate multiple UxVs of one or more types	Y	
136	TB-GEN-R-004	General	Testbed areas environment should be closely monitored	N	Testbed areas not available during 1st iteration

137	TB-GEN-R-005	General	Indoor spaces of a testbed should provide a shielded indoor environment	Y	
138	TB-GEN-R-006	General	Testbed facility areas should comprise storing spaces and be able to receive inspect and assemble and/or fix UxVs	N	Cannot be validated during 1st iteration
139	TB-GEN-R-007	General	Testbed facilities should provide emergency services in an extraordinary event	N	Cannot be validated during 1st iteration
140	TB-GEN-R-008	General	Testbed areas should provide proper facilities and equipment	N	Cannot be validated during 1st iteration
141	TB-GEN-R-009	General	Testbed must provide dedicated computational resources	N	Cannot be validated during 1st iteration
142	TB-GEN-R-010	General	Testbeds should be supported by on-site personnel	N	Cannot be validated during 1st iteration
143	TB-GEN-R-011	General	Testbeds should conform to all legal restrictions	N	Cannot be validated during 1st iteration
144	TB-MOM-001	Monitoring Manager	The Monitoring Manager component should be able to provide information about the capabilities of each resource node.	N	Monitoring manager not implemented
145	TB-MOM-002	Monitoring Manager	The Monitoring Manager component should collect and report current status of testbed facilities	N	Monitoring manager not implemented
146	TB-MOM-003	Monitoring Manager	The Monitoring Manager component should store periodically all testbed information	N	Monitoring manager not implemented
147	TB-MOM-004	Monitoring Manager	Testbed monitoring manager should be able to transmit the current status to the System Monitoring Service.	N	Monitoring manager not implemented
148	TB-NEC-001	Network Controller	The RAWFIE communication resources shall be managed to offer seamless connectivity in the normal operations of the system.	N	Network Controller not implemented
149	TB-NEC-002	Network Controller	Provision of network communication resource	N	Network Controller not implemented
150	TB-NEC-003	Network Controller	Alternative communication system	N	Network Controller not implemented
151	TB-NEC-004	Network Controller	Management of the communication system	N	Network Controller not implemented
152	TB-NEC-	Network	Time constraint verification and notification	N	Network Controller not

	005	Controller			implemented
153	TB-REC-001	Resource Controller	RAWFIE platform shall support a semi-autonomously way of navigation of the UxVs	Y	
154	TB-REC-002	Resource Controller	RAWFIE platform should be able to activate the “Emergency Scenario”	N	
155	TB-REC-003	Resource Controller	The Resource Controller shall receive location messages from the vehicles at regular intervals	Y	
156	TB-REC-004	Resource Controller	The Resource Controller shall transmit the next location for the current experiment to the vehicles	Y	
157	TB-REC-005	Resource Controller	The Resource Controller shall be able to plan the next location that will be transmitted in the vehicle taking into account the locations of all UxVs that are active in that testbed	Y	
158	TB-REC-006	Resource Controller	For the experiment accomplishment the Resource Controller shall operate in close coordination with the Experiment Controller	N	Experiment Controller not available during 1st dev. iteration
159	TB-PRO-001	Testbed Proxy	Testbed proxy should act as a reverse proxy	N	Removed from architecture
160	TB-PRO-002	Testbed Proxy	Testbed proxy contains Inner and Outer Firewall	N	Removed from architecture
161	TB-MAN-001	Testbed Manager	Testbed Manager shall support permanent storage of all testbed attributes and resources attributes that belong to testbed	N	
162	TB-MAN-002	Testbed Manager	Testbed Manager shall provide information about the capabilities of each resource node	N	
163	TB-MAN-003	Testbed Manager	Testbed Manager shall check periodically the status of all other services running at testbed level	N	Status checked only for Testbed Manager
164	TB-MAN-004	Testbed Manager	Testbed Manager shall contain a registration log for all the experiments executed in the testbed	N	
165	TB-MAN-005	Testbed Manager	Testbed Manager shall be periodically informed about the status of all running experiments in the testbed	Y	
166	TB-MAN-006	Testbed Manager	Testbed Manager shall store configuration parameters for the UxVs in the relevant testbed	N	

167	TB-MAN-007	Testbed Manager	Testbed Manager shall implement a user interface to support the interactions between testbed operators and machines	N	
168	TB-MAN-008	Testbed Manager	Testbed Manager shall be able to store data locally in case of transmission failure	N	
169	TB-MAN-009	Testbed Manager	Testbed Manager may provide statistical data/information about testbed operation	N	
170	TB-UVG-001	General	Compliance of UxV to RAWFIE specification and interfaces	N	Specification still in draft state
171	UXV-NOD-001	UxV Node	Each UxV shall have a unique Identification code.	Y	
172	UXV-NOD-002	UxV Node	Each UxV node should ensure a minimum autonomy of 15-30 minutes.	-	not tested in1st iteration
173	UXV-NOD-003	UxV Node	Each UxV node should ensure payload.	Y	
174	UXV-NET-001	UxV Network and Communication	Capability of taking the control of the UxVs from distance.	-	not tested in1st iteration
175	UXV-NET-002	UxV Network and Communication	UxVs should be able to Synchronize their Time-References between them.	-	not tested in1st iteration
176	UXV-NET-003	UxV Network and Communication	The UxV should provide Access Point functionality.	-	not tested in1st iteration
177	UXV-NET-004	UxV Network and Communication	Each UxV node shall be equipped with primary and secondary communication means.	-	not tested in1st iteration
178	UXV-NET-005	UxV Network and Communication	UxV network interface management	-	not tested in1st iteration
179	UXV-NET-006	UxV Network and Communication	UxV communication interoperability with RAWFIE (incoming)	Y	
180	UXV-	UxV Network	UxV communication interoperability with RAWFIE	Y	

	NET-007	and Communication	(outgoing)		
181	UXV-NET-008	UxV Network and Communication	Neighbouring UxV monitoring	-	not tested in1st iteration
182	UXV-NET-009	UxV Network and Communication	Each UxV node should be able to send navigation state feedback with at least 2 Hz frequency and maximum 1 sec latency when within radio communication reach.	-	not tested in1st iteration
183	UXV-SEN-001	UxV Sensor and Localisation	Each UxV node should tag location and timing capability to each sensor readings	-	not tested in1st iteration
184	UXV-SEN-002	UxV Sensor and Localisation	Each UxV node shall be able to list the available sensors	-	not tested in1st iteration
185	UXV-SEN-003	UxV Sensor and Localisation	UxV location and sensor data should be made available to the experimenter	Y	
186	UXV-SEN-004	UxV Sensor and Localisation	Location sensors should be supported in each UxV unit and can be used remotely during testbed demonstrations.	Y	
187	UXV-SEN-005	UxV Sensor and Localisation	UxVs should sent a notification to the Resource Controller when they reach the desired location	Y	
188	UXV-STO-001	UxV On-board storage	UxVs shall be able to store data on board.	-	not tested in1st iteration
189	UXV-STO-002	UxV On-board storage	UxV's shall provide a management tool of the available storage.	-	not tested in1st iteration
190	UXV-STO-003	UxV On-board storage	UxV's shall provide an authorized access to the data management tool.	-	not tested in1st iteration
191	UXV-STO-004	UxV On-board storage	UxV's shall provide a data log.	-	not tested in1st iteration
192	UXV-STO-005	UxV On-board storage	UxV's may provide an automated syncing of servers.	-	not tested in1st iteration

193	UXV-PRC-001	UxV On-board processing	Each UxV shall be able to operate autonomously.	-	not tested in1st iteration
194	UXV-PRC-002	UxV On-board processing	The UxV should provide collision avoidance mechanism.	-	not tested in1st iteration
195	UXV-PRC-003	UxV On-board processing	Capability of task planning of the UxVs nodes during run-time.	-	not tested in1st iteration
196	UXV-PRC-004	UxV On-board processing	UxVs should be able to cooperate during the execution of an experiment.	-	not tested in1st iteration
197	UXV-PRC-005	UxV On-board processing	Each UxV node shall keep position while waiting for new instructions.	-	not tested in1st iteration
198	UXV-MGT-001	UxV Management	UxVs shall offer on demand resources (Network, Sensor, Processing, and Controller).	-	not tested in1st iteration
199	UXV-MGT-002	UxV Management	UxV shall be capable to revert to a safe mode	-	not tested in1st iteration
200	UXV-MGT-003	UxV Management	UxV shall be capable to restart each component independently	-	not tested in1st iteration
201	UXV-MGT-004	UxV Management	UxV shall be capable to monitor the health of the system	-	not tested in1st iteration
202	UXV-MGT-005	UxV Management	UxV shall be capable to enable/disable each component	-	not tested in1st iteration
203	UXV-MGT-006	UxV Management	UxV shall be capable to offer safe maintenance access for manufacturers	-	not tested in1st iteration

Table 1: Validation by requirements

4 Questionnaire for end-user validation

The main topics of the questionnaire to collect validation data and user opinions is presented here, together with the purpose of the questions and possible evaluation methods. The complete questionnaire can be found in annex A. The first results of the questionnaire are presented in section 5.2.

The current version of the questionnaire focuses on general opinion and validation of the system, to get a feedback whether we are going in the right direction. Further questions related to specific functionalities and metrics will be added in the next version.

The questionnaire currently has six main sections. The purpose of the questions is described in the following:

- About you
  - Simple questions to get an overview of the person that answers the questionnaire.
- UxVs and Testbeds
  - Questions to find out what experience the person has in general with UxVs, testbeds and the experimenting in testbeds.
  - Contains also questions on what he liked or disliked in the testbeds he knows (11, 12, 13).
- Experimenting with RAWFIE
  - Here potential end-users of the RAWFIE system should give their opinion on the possibilities of the RAWFIE system.
  - First, it asks how (14), why (15), what (16), and when (17) he would use RAWFIE. It is important to know the main interests of the users, so we could focus more on the perspectives that are really needed.
  - Then the user is asked what functionalities are of his/her interest, not of his/her interest, or missing (18-22). The purpose is again to focus on needed functionalities.
  - Questions about how experiments would be carried out (23, 24) to optimize the experimentation process.
  - And finally some questions about the value of RAWFIE: paying for RAWFIE services and cost reductions through RAWFIE (for the business model, see WP2) (25-27).
- SFA interface provision
  - Determine the need of a fully functional SFA interface.
- Testbed integration into RAWFIE
  - These question are to get more information on potential new testbeds,
- UxV integration into RAWFIE
  - These question are to get more information on potential new UxV providers.

5 Showcase to inform end-users

A demonstration of the online platform was presented to the end-users. The demonstration included a short presentation of RAWFIE as a project and the components that were implemented in the first development period. Then a platform overview followed.

During the platform demonstration, end users had the opportunity to get an overview of the components included in the portal like Resource Controller, Experiment Authoring Tool, Experiment Monitoring, Data Analytics and System Monitoring. An explanation of their functionality was provided to them.

5.1 Showcase scenario

Afterwards, a demonstration of a simple showcase was given to the end users. EDL parts were demonstrated and explained in order for the end users to write its own experiment, as figured in the following script:

```
Experiment
     Metadata
         Name TestExp
         Version 30.0
         Date 26/02/2016
     ~Metadata
     Requirements
         Nodes 3
         Testbed Porto_Testbed
         Location(+41.18339200, -8.70830300)
         Duration 4
        MaxDistance 100
     ~Requirements
     Declarations
         var x as Integer
     ~Declarations
    Execution
        ExecutionInfo
            LayoutWidth 500
            LayoutHeight 500
        ~ExecutionInfo
        Node
            ID node1
            Route[
                    WP<0, 98 - 60 - 0>
                    WP<1, 57 - 93 - 0>
                    WP<2, 94 - 138 - 0>
                    WP<3, 137 - 106 - 0>
                ]
            DataManagement
                    Time 14 Algorithm average(history = 10)
                    ~DataManagement
            NodeCommunication
                    NIC WiFi
                     ~NodeCommunication
            DataManagement
```

```
                    Time 25 Algorithm average(history = 5)
                    ~DataManagement
        ~Node
        Node
            ID node2
                Route[
                    WP<0, 84 - 229 - 0>
                    WP<1, 32 - 233 - 0>
                    WP<2, 25 - 172 - 0>
                    WP<3, 84 - 169 - 0>
                ]
        ~Node
        Node
            ID node3
                Route[
                    WP<0, 118 - 218 - 0>
                    WP<1, 165 - 182 - 0>
                    WP<2, 193 - 229 - 0>
                    WP<3, 141 - 266 - 0>
                ]
        ~Node
    ~Execution
~Experiment
```

Figure 1: EDL script use during showcase

Due to lack of real devices, a video with the real operation and execution of the aforementioned experiment were shown to the end users.

5.2 Results of questionnaire

A summary of the questionnaire and a table with all answers can be found in Annex B and C.

The following sub-sections summarise the results and derive some requirements out of them.

5.2.1 Conclusions

We have got 7 responses from the following types of stakeholders

- UxV manufactures or UxV service providers (industrial): 2
- Research/university/higher education: 3
    - As experimenter: 3
    - As testbed owner: 1
- Industrial users: 2
    - As experimenter: 2
    - As testbed owner: 1

All of them stated that RAWFIE would be valuable or useful for them (“good” or “great”).

5.2.1.1 Testbed experiences of the users

While the UxV manufactures have never used testbeds, most of the other had some experience with testbeds. One already had even used an UxV testbed. So it can be concluded that these potential experimenters have good understanding about the purpose of RAWFIE.

On the questions what is good or bad in other existing testbeds, we only got answers from two users. The following list summarises the answers:

- Positive aspects
  - Execute experiments on demand.
  - Viewing sensor data in real-time.
  - Repeatability of experiments (via scripts).
  - Ability to experiment with multiple patterns and varying interface/protocol configurations.
  - Ability to record the experiment in detail.
  - Ability to debug experiments.
- Negative aspects
  - Not adequate customisability to the experimental configuration.
  - OTA programming missing.

Regarding the positive aspects of other testbeds facilities: RAWFIE plans to fulfil most of the aspects. Only the debugging of experiments was not planned until now. Concerning the bad aspect about better customisability of experiments, we will have in mind that the EDL must be highly customisable. The OTA programming is already part of our user scenarios (D3.2).

5.2.1.2 Reasons for using RAWFIE in future

The users stated the following potential usage of the RAWIE platform:

- Middleware development for sensing and control of UxVs.
- Early experimentation/feasibility study.
- Can provide a platform where multiple patterns can be tested.

As regards why they would use RAWFIE, most of the participants answered “I need a variety of different UxVs from different vendors in order to test my product sufficiently”. So the variety of different UxVs that RAWFIE will support it also the most valuable aspect that potential users see.

The users see the usage in all phases of the development, while most would use RAWFIE for the early prototype testing. A conclusion for RAWFIE could be that the system should be flexible enough to be helpful in all these phases.

5.2.1.3 Data and measurements

The live visualisation of experiments and sensor values was the most important data feature (5), followed by the data analysis functionality (4) and the raw sensor values (3).

As regards data format for results, the “Homogenised sensor data” was most frequently marked (5). The “Raw sensor data” (3) and “Aggregated and analysed data” (2) followed behind.

The conclusion is that the live visualisation is highly demanded and should be further developed. The data analysis functionality is also of value for the users. But, beside these, there are some requirements for raw sensor data or homogenised sensor data, that should also be provided to the experimenter.

5.2.1.4 EDL and experimenting

Regarding the flexibility of the experiment scripting, the most preferred answer was, that they liked the idea of an EDL, but access to specific UxV commands should be available somehow. Nobody wanted direct access to the UxVs (via a UxV specific scripting/programming language).

This means the EDL is the right way, when it also provides access to UxV specific commands. No effort should be spent to support UxV specific scripting/programming languages.

5.2.1.5 Business model

The cost reductions were estimated between 20% and 85% which is really a broad range. To get better value on this, it would be better to do some real life cost evaluations during the next iteration.

The question about the price per hour and UxV did not result in clear preferences. So some more detailed questions on this topic are needed too.

5.2.1.6 SFA

Only 2 of the 7 users had experiences with SFA. These two would find it good if RAWFIE would have an SFA interface.

RAWFIE will still implement an SFA interface, even if SFA needs better promotion.

5.2.1.7 UxV and Testbed integration

The answers on testbeds and UxV integration showed that there is interest of other organisation/companies to integrate their testbeds and UxV into the RAWFIE system.

Further conclusion could not be drawn from the answers.

5.2.2 New requirements

The following new requirements were identified during evaluations of the answers of the questionnaire.

- Ability to somehow debug experiment scripts.
- EDL must allow high customisability to the experimental configuration.
- EDL must allow access to specific UxV commands somehow.
- Download of raw sensor data and/or homogenised sensor data of experiments.
- The RAWFIE system should be flexible enough to be used during all product development phases.

5.2.3 Improvements for next questionnaire

The questionnaire shows some shortcomings – these need to be addressed for the next iteration.

- Try to formulate more comprehensive questions. E.g., the question "Which functionalities of RAWFIE are most valuable for you?" and "Which functionalities of RAWFIE are of no interest to you?", 3 of 7 marked same items on both questions.
- More detail question to validate the function of the RAWFIE system

6 Performance and technical evaluation

6.1 Benchmarking of the RAWFIE Message Broker

In this section we present the outcomes of a series of performance tests, focused on the measurements of the End-to-end latency for messages exchanged between Kafka producers and consumers. A high level analysis of the results is provided, while the actual evaluation of the observed metrics (End-to-end latency) against the expected success criteria will be provided in subsequent versions of this document (D6.4), where the detailed evaluation of all quantitative metrics defined in D4.3 will be provided. The reason being that the accurate definition of the success criteria, like for example the maximum End-to-end latency admitted for a safe control of the UxVs, is part of an already planned, incoming activity, which will involve the participation of Testbeds and UxVs owners, both partners belonging to the consortium and new consortium partners who are joining the consortium after the conclusion of the 1st Open Call.

Aforementioned tests for the RTT and the End-to-end latency metrics, whose setup and results are presented in the following, are carried out in 2 different environments. A local, controlled one, where simulated RAWFIE software components (messages producers and consumers), running on general purpose laptops, publish and consume messages at high rates. A second setup involves actual RAWFIE software components, the Resource Controller and the UxV RAWFIE Adaptor running on the UxV nodes, acting as producers and consumers of the messages.

6.2 Local environment with simulated components

Hardware

- Forwarder (Consumer+Producer)
  - General purpose laptop
  - CPU:Intel i5 5200U
  - RAM:8gb DDR3L
  - HDD:1TB 5400rpm

- Forwarder (Broker)
  - Server for hostng kafka broker
  - CPU:Intel i5 5200U
  - RAM:8gb DDR3L
  - HDD:1TB 5400rpm

- Server/Zookeeper/Producer+Consumer
  - Virtual PC in rack server
  - CPU:Intel Xeon E5-2640 (assigned 4 cores)
  - RAM:8GB DDR3
  - HDD:60GB RAID 5

Software Components & Configuration

- Producers software
  - Simulated Apache Kafka producer
- Consumer software
  - Simulated Apache Kafka consumer
- Number of Apache Kafka Servers (e.g. cluster or a single server):1

Specific tuning for the tests

- type of messages: Avro
- number of producers:1
- number of consumers:1
- Kafka topics:1

Tests description and results

The latency was measured by synchronising the time of the consumer and the producer with an NTP server (2.gr.pool.ntp.org), and then computing the time from when the packet was published, to when it was received, in milliseconds.

The following figures shows the latency measures in milliseconds, with the varying number of packets sent from the producer to the consumer.

Number of messages: 10.000

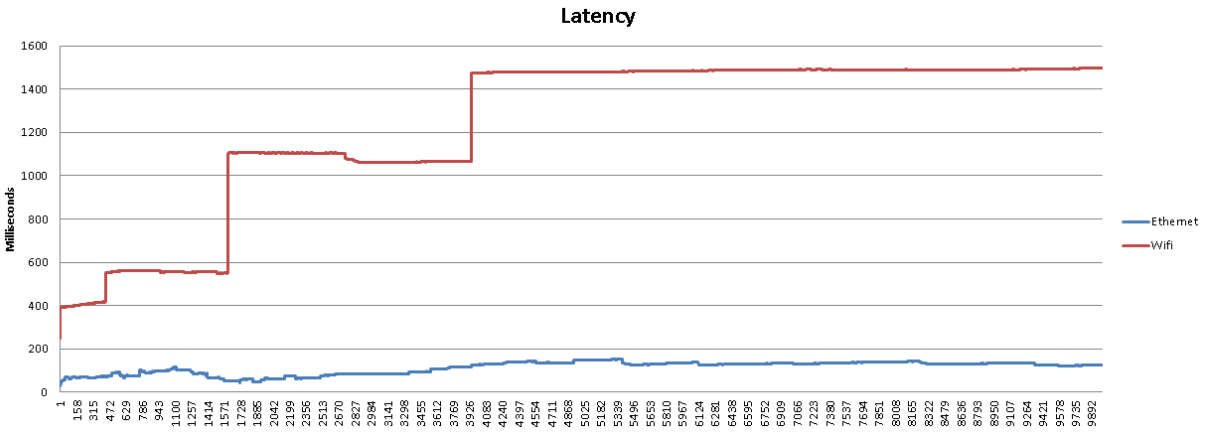


Figure 2: End-to-end latency with the number of messages varying from 1 to 10,000

Number of Messages: 100.000

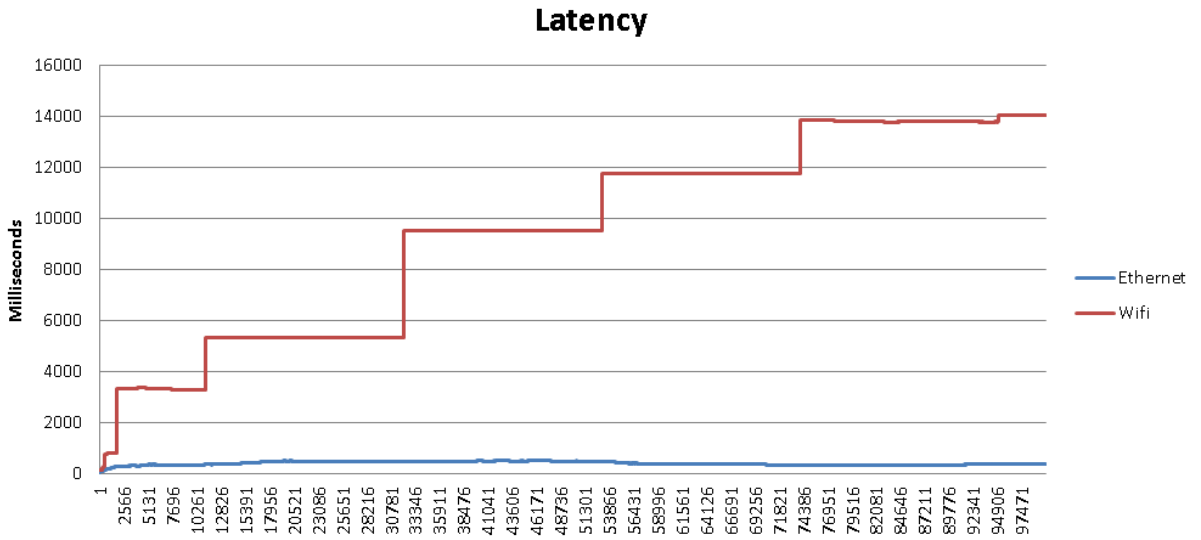


Figure 3: End-to-end latency with the number of messages varying from 1 to 100,000

Number of Messages: 100.0000

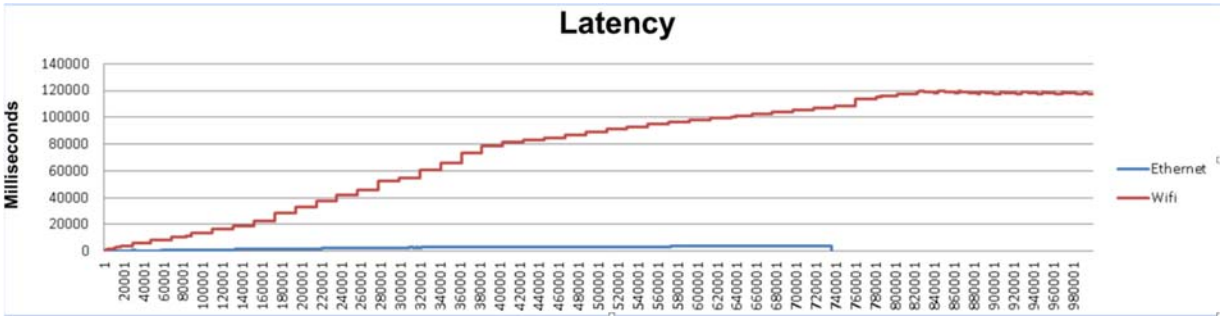


Figure 4: End-to-end latency with the number of messages varying from 1 to 1,000,000

6.3 Test environment with actual RAWFIE components

Hardware

MST UxVs are equipped with one embedded computer with the following specifications:

- CPU: AMD® Geode™ LX 800 @ 500 MHz
- RAM: 1024 MB DDR1 @ 400 MHz
- Disk: Compact Flash 32 GB
- Network: 802.3u, 802.11n @ 2.4 GHz

The MST RAWFIE software runs on server with the following specifications:

- CPU: Intel® Core™ i7-4710HQ @ 2.50GHz
- RAM: 8192 MB DDR3 @ 1.6 GHz
- SSD: Samsung SSD 840 EVO 250 GB
- Network: 802.3ab

Test environment setup

The temporary testing infrastructure of MST, whose network topology is depicted in , comprised the following components:

- Three Light Autonomous Underwater Vehicles (LAUVs) equipped with Conductivity, Temperature, Rhodamine Dye, Chlorophyll, Phycocyanin, Phycoerythrin, and Fluorescein sensors; active dual frequency sonar and high definition camera. Communication with the Manta gateway is performed using a 2.4 GHz 802.11n radio link and 25 kHz acoustic modem. These assets are represented as “AUV 0”, “AUV 1”, and “AUV 2” in the network topology diagram.
- One Manta gateway with WHOI Micromodem Acoustic Modem and one 2.4 GHz 802.11n radio with an omnidirectional antenna. This asset is represented as “GW” in the network topology diagram.

- One 2.4 GHz 802.11n radio with builtin 90º sector antenna, connected to the MST network infrastructure and to the Internet through a firewall. These assets are represented in the network topology diagram as “LAN-GW”, “LAN”, and “Firewall” respectively.

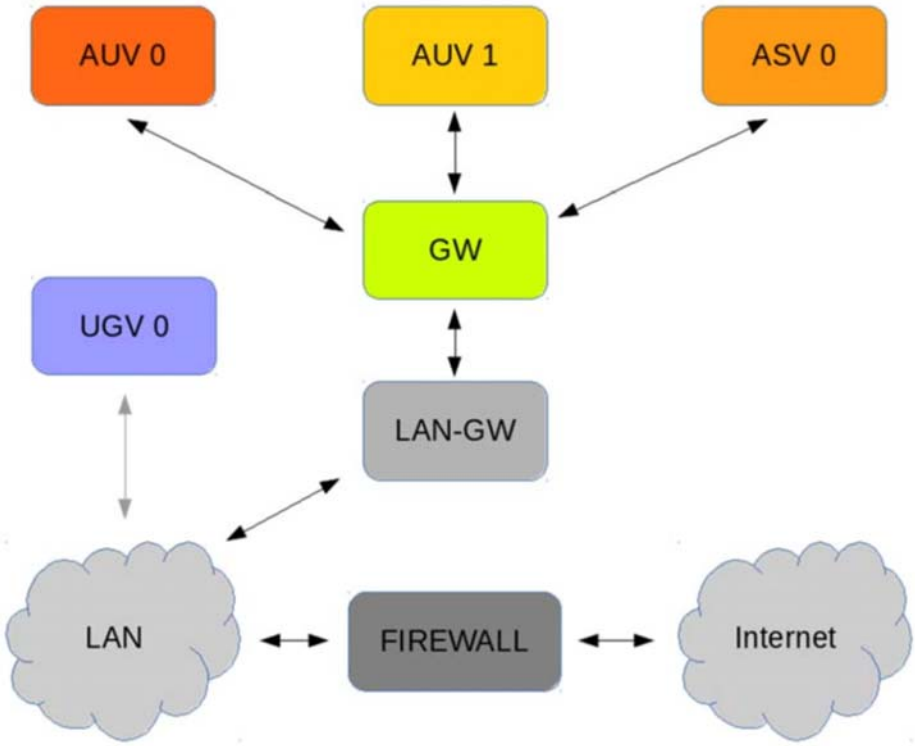


Figure 5: Schematic test setup for RAWFIE component tests

6.3.1 Round Trip Time results

Two different stress tests performed:

1. Sync Test: publishes/receives 1000 records.
2. Burst Test: publishes all records / receives all records.

The first one, was performed by synchronising the publisher with the consumer, i.e. a new message is published right after the “answer” for the first one is received. In the second configuration, messages are sent in burst, i.e. without waiting the feedback. In all cases,

Below is the summary of the results, in textual form, of the 2 different test types.

Sync Test (TX/RX) | 1000 records

- Subscribed Topics : 1
- Elapsed Time : 113226 ms
- Schema Initialization : 8 ms
- Kafka Producer Initialization : 3 ms

- Kafka Consumer Initialization : 5266 ms
- Kafka Consumer Shutdown : 0 ms
- RX – Records : 1000
- **RTT – Minimum : 102.00 ms**
- **RTT – Maximum : 951.00 ms**
- **RTT – Mean : 113.10 ms**
- **RTT – SD : 38.12 ms**
- TX – Records : 1000
- TX – Duration : 0 ms
- TX – Minimum : 0.00 ms
- TX – Maximum : 641.00 ms
- TX – Mean : 0.69 ms
- TX – SD : 20.27 ms

Burst Test (TX/RX) | 1000 records

- Subscribed Topics : 1
- Elapsed Time : 21662 ms
- Schema Initialization : 11 ms
- Kafka Producer Initialization : 3 ms
- Kafka Consumer Initialization : 5075 ms
- Kafka Consumer Shutdown : 611 ms
- RX – Records : 1000
- **RTT – Minimum : 223.00 ms**
- **RTT – Maximum : 20976.00 ms**
- **RTT – Mean : 10634.81 ms**
- **RTT – SD : 6020.21 ms**
- TX – Records : 1000
- TX – Duration : 686 ms
- TX – Minimum : 0.00 ms
- TX – Maximum : 642.00 ms
- TX – Mean : 0.68 ms
- TX – SD : 20.30 ms

6.3.2 End-to-end latency results

The latency was measured by using timestamps between the consumer in the UAVs and the consumer inside the Kafka server. The two consumers were synchronized by using a GPS NTP server by using GPS time definition. The difference between the two consumers were computed and depicted in the following graph. The Schema used for the execution of the tests is described in the Annex D

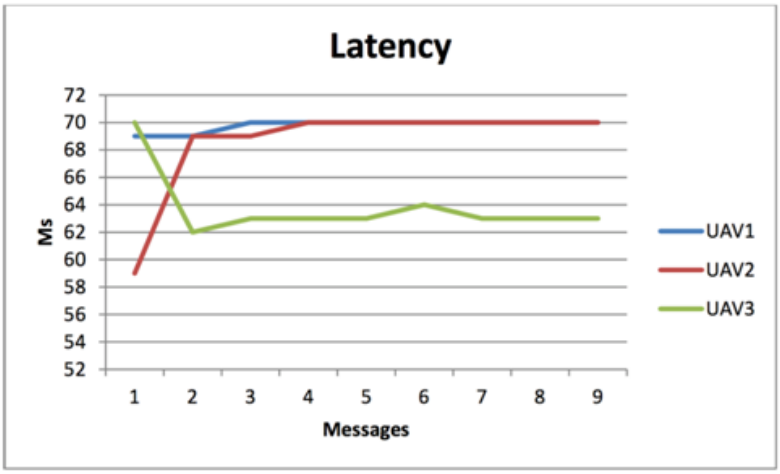


Figure 6: End-to-end latency in the second environment

.

7 Roadmap for the Platform Validation

The following roadmap is planned to perform the validation of the system until M30 (in the first table) and M40 (in the second table)

Year	**2016**						**2017**					
Month	J	A	S	O	N	D	J	F	M	A	M	J
Project Month	19	20	21	22	23	24	25	26	27	28	29	30
Development and implementation of RAWFIE components (2nd iteration)												
Extend questionnaires												
Platform ready for end-user tests												
Perform validation scenarios (observation of participants, recording of validation metrics)												
Do questionnaires or interviews with the users												
Evaluate questionnaires and interviews												
Perform evaluation of quantitative metrics against												

success criteria												
Prepare D6.4												

Year	**2017**					**2018**				
Month	J	A	S	O	N	D	J	F	M	A
Project Month	31	32	33	34	35	36	37	38	39	40
Development and implementation of RAWFIE components (3rd iteration)										
Extend questionnaires										
Platform ready for end-user test										
Perform validation scenarios (observation of participants, recording of validation metrics)										
Do questionnaires or interviews with the users										
Evaluate questionnaires and interviews										
Perform evaluation of quantitative metrics against success criteria										
Prepare D6.6										

8 Conclusion and Outlook

The main drawback of the current state is, that the system is not ready for real end-user validation and comprehensive, quantitative performance tests. A showcase was performed where the system was shown to end-users. Based on this a questionnaire was filled out by the end-users where we get fist responses to improve the RAWFIE system, which showed that RAWFIE is on the right way. Also some new requirements where extracted from the answers of the questionnaire.

Furthermore, some performance evaluation has been made, measuring the end-to-end latency and RTT when using the Kafka message bus for communication. The results show that Kafka provides high performance and therefore may fit to the performance needs of RAWFIE, that will be further evaluated when the success criteria will be defined together with the end users and especially Testbeds/UxVs owners.

For the next version of the “RAWFIE Platform Validation” (D6.4), the platform will be ready for real end-user tests, where also the validation scenarios from D4.3/D4.6 will be executed and metrics will be evaluated.

Annex

A End-user questionnaire

About you

1. How old are you?
   - younger than 20
   - 20 to 29
   - 30 to 39
   - 40 to 49
   - 50 to 59
   - 60 and older
2. Which kind of organisation/company are you from?
   - Free text
3. What is your professional role?
   - Free text
4. What are your activities/responsibilities at your organisation/company?
   - Free text
5. Which roles could be played by your company organisation/company (if any)?
   - Experimenter
   - Tesbed owner
   - UxV manufacturer
   - Regulation body

UxVs and Testbeds

6. In which kinds of UxVs are you interested?
   - UAV (unmanned aerial vehicle - aircraft, commonly known as a drone)
   - UGV (unmanned ground vehicle - vehicle that operates while in contact with the ground)
   - USV (Unmanned surface vehicles - vehicles that operate on the surface of the water)
   - UUV (unmanned underwater vehicle - vehicles that are able to operate underwater)
   - Other: Free text
7. Which kinds of UxVs have you worked with?
   - UAV (unmanned aerial vehicle - aircraft, commonly known as a drone)
   - UGV (unmanned ground vehicle - vehicle that operates while in contact with the ground)
   - USV (Unmanned surface vehicles - vehicles that operate on the surface of the water)
   - UUV (unmanned underwater vehicle - vehicles that are able to operate underwater)

- Other: Free text

8. Have you ever made experiments with UxVs?
   - Yes
   - No
9. Have you ever used testbeds in general (not only UxV testbeds)?
   - Yes
   - No
10. Have you ever used UxV testbeds?
    - Yes
    - No
11. Which testbeds facilities did you use?
    - Free text
12. Which functionalities/procedures of the used facilities according to your opinion are useful?
    - Free text
13. Which functionalities/procedures of the used facilities according to your opinion need improvement?
    - Free text

Experimenting with RAWFIE

For those that would like to execute experiments with RAWFIE

14. How would you incorporate RAWFIE into your projects?
    - Free text
15. Why would you use the RAWFIE platform?
    - I only need to perform some tests and don't want to buy UXVs for these few.
    - It is necessary to have many UXVs involved in one tests, but I only have too little of them.
    - I need a variety of different UXVs from different vendors in order to test my product sufficiently.
    - Other: Free text
16. Which kind of information are most relevant of an experiment?
    - Live visualisation of the experiments including sensor values
    - Performing some standard data analytic algorithms on the gathered sensor data Raw sensor data
    - Other: Free text
17. In which phases during the product life cycle would you use RAWFIE?

- Case study
- Feasibility study
- Early prototype testing
- Late prototype testing
- Final product testing
- Other: Free text

18. Which functionalities of RAWFIE are most valuable for you?

- EDL script editor
- EDL visual/graphical editor
- Resources Explorer
- Booking of resources
- Experiment Monitoring and Visualisation
- UxV remote control (live)
- System Monitoring
- Data Analytics
- Other: Free text

19. Which functionalities of RAWFIE are not of interest for you?

- EDL script editor
- EDL visual/graphical editor
- Resources Explorer
- Booking of resources
- Experiment Monitoring and Visualisation
- UxV remote control (live)
- System Monitoring
- Data Analytics
- Other: Free text

20. Do you like the idea of a specialized EDL (experiment description language)?

- Yes, I don't want to worry about UxV specific command.
- Yes, but access to specific UxV commands should be possible somehow.
- No, I want direct access to the specific UxV (UxV specific scripting/programming language).

21. Which kind of result formats would you like to have?

- Raw sensor data (UxV/sensor specific)
- Homogenized sensor data (homogenized format for RAWFIE)
- Aggregated and analysed data
- Just if experiment was successful or not
- Other: Free text

22. Which additional functionalities or information would you like to have provided by RAWFIE?

- Free text

23. How would you perform most the tests?
    - Large test series (prepare many tests that may run several days and evaluate them afterwards)
    - Short test interactions (prepare only a few tests, run and evaluate them; afterwards the next cycle begins)
    - Both equally
    - Other: Free text
24. Which kinds of experiments would you perform with RAWFIE?
    - Free text
25. How valuable/useful would RAWFIE be for you?
    - Great
    - Good
    - Average
    - Low
    - Valueless
26. How much would you pay per hour and UxV?
    - Free text
27. Please tried to estimate of the cost reduction (person months) by using RAWFIE for your experiments instead of build your experimentation platform yourself (in %)
    - Free text

SFA interface

Slice Federation Architecture interface

28. Have you ever used a testbed via a SFA interface?
    - What is SFA?
    - Yes. It's really useful.
    - Yes. But I like a more specialised interface.
    - No.
29. How valuable is an SFA interface in general for you?
    - Great
    - Good
    - Average
    - Low
    - Valueless
30. How valuable would an SFA interface for RAWFIE be for you
    - Great

- Good
- Average
- Low
- Valueless

Testbed integration into RAWFIE

For those that would like to integrate their testbeds into RAWFIE

31. Which type of testbed can you provide
    - Air
    - Ground, outdoor
    - Ground, indoor
    - Maritime, water outdoor
    - Water indoor
    - Other: Free text
32. Are there any constrains that must be obeyed in your testbed (e.g.: availability, hours of operation, number of UxV simultaneously operated)?
    - Free text
33. What do you expect from an integration into the RAWFIE system?
    - Free text
34. How many UxVs can your testbed host (approximately)?
    - Free text

UxV integration into RAWFIE

For those that would like to integrate their UxVs into RAWFIE

35. Which kinds of UxVs can you provide?
    - UAV (unmanned aerial vehicle - aircraft, commonly known as a drone)
    - UGV (unmanned ground vehicle - vehicle that operates while in contact with the ground)
    - USV (Unmanned surface vehicles - vehicles that operate on the surface of the water)
    - UUV (unmanned underwater vehicle - vehicles that are able to operate underwater)
    - Other: Free text
36. Are there any constrains that must be obeyed with your UxVs?
    - Free text
37. What do you expect from an integration into the RAWFIE system?
    - Free text
38. How many UxVs could you provide (on how many testbeds)?

- Free text

Final comments

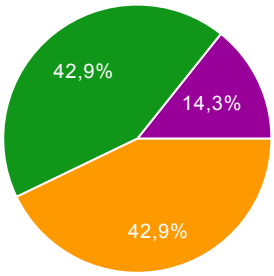
39. Any additional comments that you have about the RAWFIE system?
    - Free text

B Questionnaire summary

The following pages contain the automatic generated summary.

About you

How old are you?



younger than 20	**0**	0 %
20 to 29	**0**	0 %
30 to 39	**3**	42.9 %
40 to 49	**3**	42.9 %
50 to 59	**1**	14.3 %
60 and older	**0**	0 %

Which kind of organisation/company are you from?

UAS service providers

University of Piraeus

Research / Higher Education

Electronics

University of Applied Sciences

IoT, RnD department, AGT International

UxV Addition & Customization

What is your professional role?

Professor

business development

Faculty

Head of RD

Scenior Data Scienst

CEO

What are your activities/responsibilities at your organisation/company ?

Research and Teaching

R&D manager

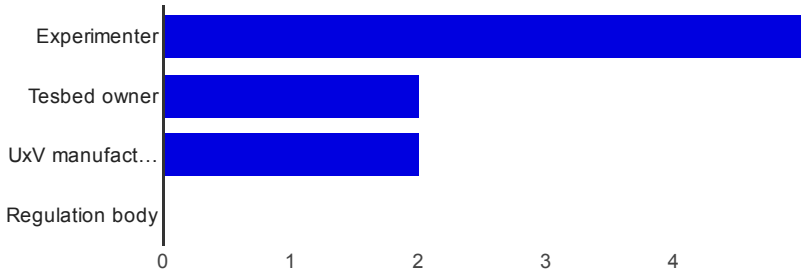
RD Projects/Product Development

Teaching-Research

Researching and supervising application of data analytics solutions, using Internet of Things data. Helping evaluating which sensors and network configurations best fit each application needs.

CEO

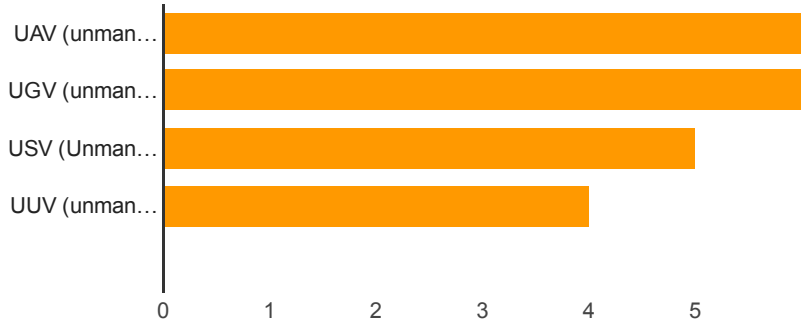
Which roles could be played by your company organisation/company (if any)?



Experimenter	**5**	71.4 %
Tesbed owner	**2**	28.6 %
UxV manufacturer	**2**	28.6 %
Regulation body	**0**	0 %

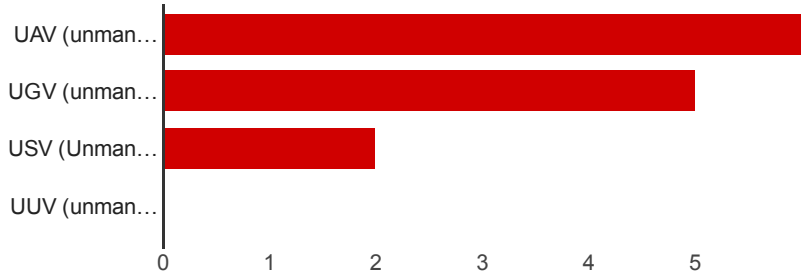
UxVs and Testbeds

In which kinds of UxVs are you interested?



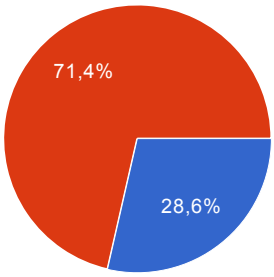
UAV (unmanned aerial vehicle - aircraft, commonly known as a drone)	**6**	100 %
UGV (unmanned ground vehicle - vehicle that operates while in contact with the ground)	**6**	100 %
USV (Unmanned surface vehicles - vehicles that operate on the surface of the water)	**5**	83.3 %
UUV (unmanned underwater vehicle - vehicles that are able to operate underwater)	**4**	66.7 %

Which kinds of UxVs have you worked with?



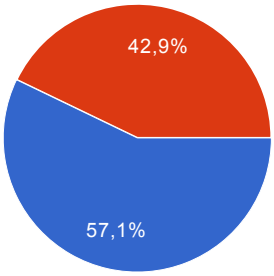
UAV (unmanned aerial vehicle - aircraft, commonly known as a drone)	**6**	85.7 %
UGV (unmanned ground vehicle - vehicle that operates while in contact with the ground)	**5**	71.4 %
USV (Unmanned surface vehicles - vehicles that operate on the surface of the water)	**2**	28.6 %
UUV (unmanned underwater vehicle - vehicles that are able to operate underwater)	**0**	0 %

Have you ever made experiments with UxVs?



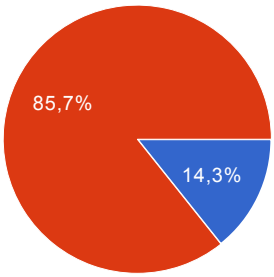
Yes	**2**	28.6 %
No	**5**	71.4 %

Have you ever used testbeds in general (not only UxV testbeds)?



Yes	**4**	57.1 %
No	**3**	42.9 %

Have you ever used UxV testbeds?



Yes	**1**	14.3 %
No	**6**	85.7 %

Which testbeds facilities did you use?

UUV experiment

Testbeds for our products

MSP430 MCU DEBUGGER

Mobile Emulab, PLANETLAB, EMULAB

Which functionalities/procedures of the used facilities according to your opinion are useful?

the bility to start/stop experiments on-demand, and viweing the sensor reading timeseries in real-time

ENERGY TRACER

Repeat-ability of experiments with the use of a script, ability to experiment with multiple mobile patterns to observe interactions, ability to record the experiment in detail, easiness to configure and debug experiments, ability to experiment with varying interface/protocol configurations

Which functionalities/procedures of the used facilities according to your opinion need

improvement?

add more customisability to the experimental configuration (needs to be supported by the platform too though)

OTA programming

Experimenting with RAWFIE

How would you incorporate RAWFIE into your projects?

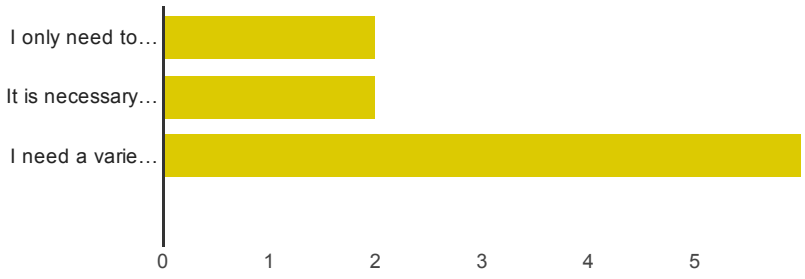
YES

We can discuss it

MIDDLEWARE DEVELOPMENT FOR SENSING AND CONTROL OF UAVs/UGVs

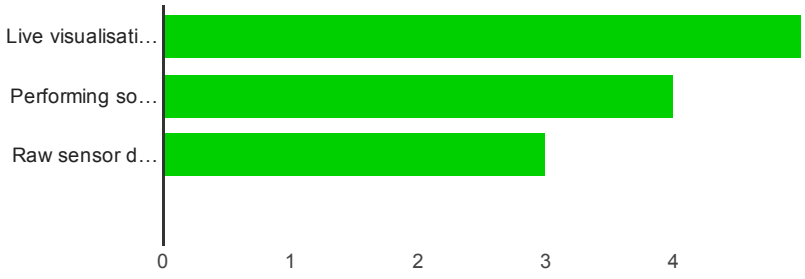
As an easy way for early experimentation / feasibility study. The data acquired could prove valuable to develop and debug planned analytics. It offers a platform where multiple patterns can be tested and the results can help debug and develop the analytics that require data driven approaches such as anomaly detection.

Why would you use the RAWFIE platform?



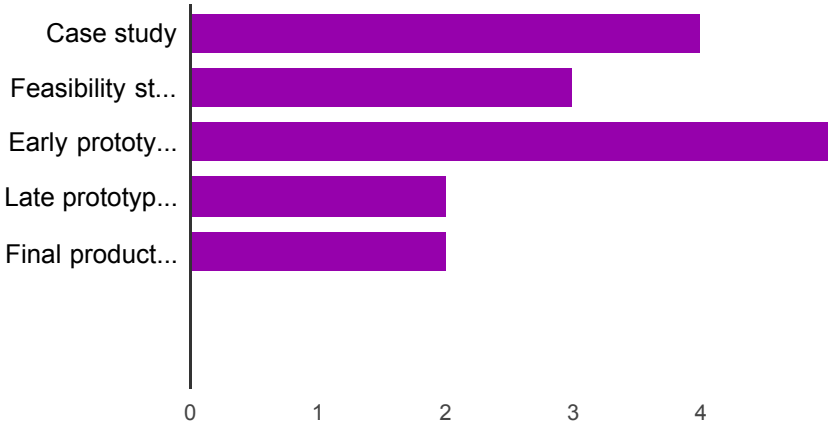
I only need to perform some tests and don't want to buy UXVs for these few.	**2**	33.3 %
It is necessary to have many UXVs involved in one tests, but I only have too little of them.	**2**	33.3 %
I need a variety of different UXVs from different vendors in order to test my product sufficiently.	**6**	100 %

Which kind of information are most relevant of an experiment?



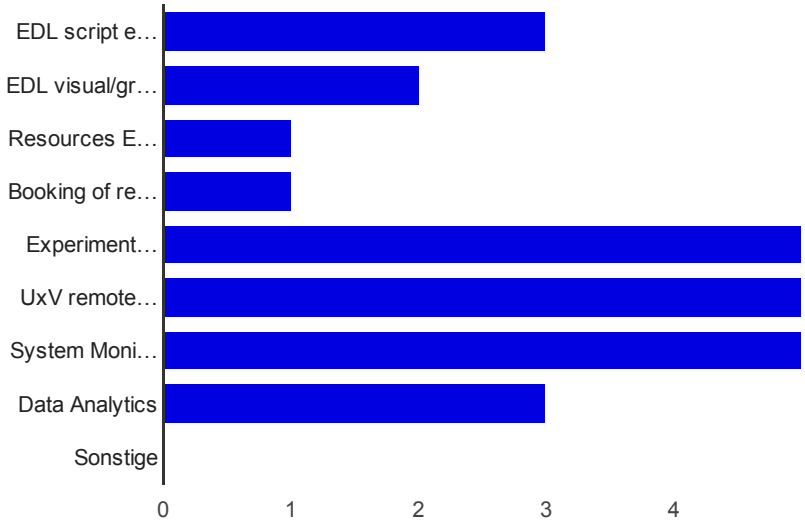
Live visualisation of the experiments including sensor values	**5**	83.3 %
Performing some standard data analytic algorithms on the gathered sensor data	**4**	66.7 %
Raw sensor data	**3**	50 %

In which phases during the product life cycle would you use RAWFIE?



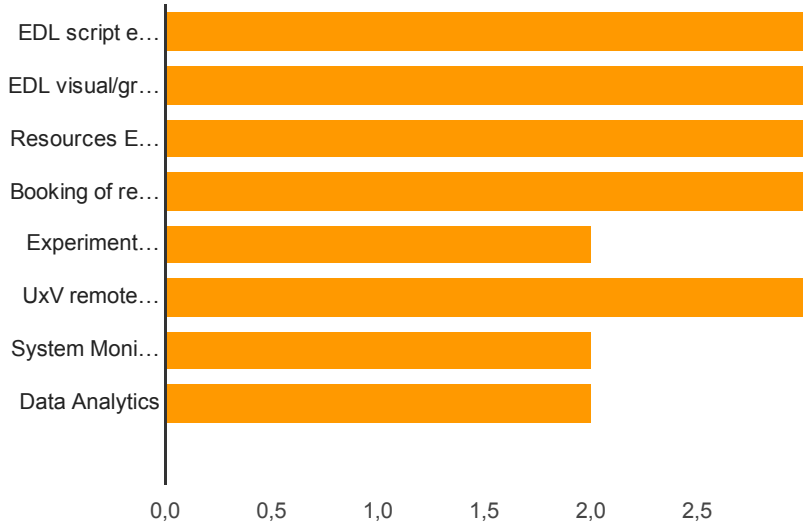
Case study	**4**	66.7 %
Feasibility study	**3**	50 %
Early prototype testing	**5**	83.3 %
Late prototype testing	**2**	33.3 %
Final product testing	**2**	33.3 %

Which functionalities of RAWFIE are most valuable for you?



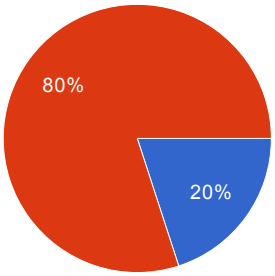
EDL script editor	**3**	50 %
EDL visual/graphical editor	**2**	33.3 %
Resources Explorer	**1**	16.7 %
Booking of resources	**1**	16.7 %
Experiment Monitoring and Visualisation	**5**	83.3 %
UxV remote control (live)	**5**	83.3 %
System Monitoring	**5**	83.3 %
Data Analytics	**3**	50 %

Which functionalities of RAWFIE are not of interest for you?



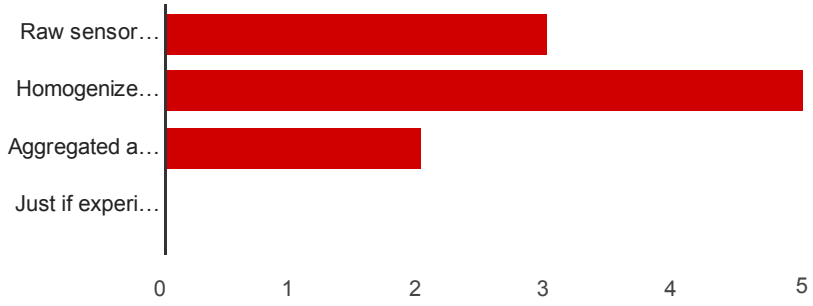
EDL script editor	**3**	75 %
EDL visual/graphical editor	**3**	75 %
Resources Explorer	**3**	75 %
Booking of resources	**3**	75 %
Experiment Monitoring and Visualisation	**2**	50 %
UxV remote control (live)	**3**	75 %
System Monitoring	**2**	50 %
Data Analytics	**2**	50 %

Do you like the idea of a specialized EDL (experiment description language)?



Yes, I don’t want to worry about UxV specific command.	**1**	20 %
Yes, but access to specific UxV commands should be possible somehow.	**4**	80 %
No, I want direct access to the specific UxV (UxV specific scripting/programming language)	**0**	0 %

Which kind of result formats would you like to have?

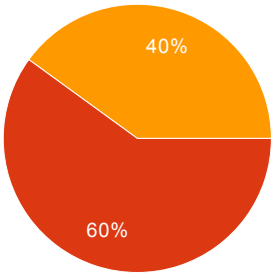


Raw sensor data (UxV/sensor specific)	**3**	50 %
Homogenized sensor data (homogenized format for RAWFIE)	**5**	83.3 %
Aggregated and analysed data	**2**	33.3 %
Just if experiment was successful or not	**0**	0 %

Which additional functionalities or information would you like to have provided by RAWFIE?

Location data

How would you perform most the tests?

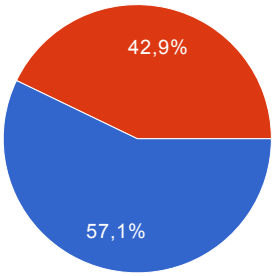


Large test series (prepare many tests that may run several days and evaluate them afterwards)	**0**	0 %
Short test interactions (prepare only a few tests, run and evaluate them; afterwards the next cycle begins)	**3**	60 %
Both equally	**2**	40 %

Which kinds of experiments would you perform with RAWFIE?

real-time sensing and control using Delay Tolerant Networks

How valuable/useful would RAWFIE be for you?



Great	**4**	57.1 %
Good	**3**	42.9 %
Average	**0**	0 %
Low	**0**	0 %
Valueless	**0**	0 %

How much would you pay per hour and UxV?

somethiing equivalent to what Cloud Computing platforms charge for (e.g., $1 per hour, depending on functionality provided)

100

[Disclaimer: I would not be the one to address / decide the cost questions, only advise, and it would fall to the specific product owner to decide based on his budget, his goals, the alternatives, and most importatly and his urgency] Depending on the scale of experiments, for deciding if the testbed should be used it is possible that a trial of the testbed of ~<1K would be approved, then the results internaly presented and benefits evaluated, before proceeding to more investment. I am not sure how this would be epected to be mapped to hours/UxVs

Please tried to estimate of the cost reduction (person months) by using RAWFIE for your experiments instead of build your experimentation platform yourself (in %)

30

70%

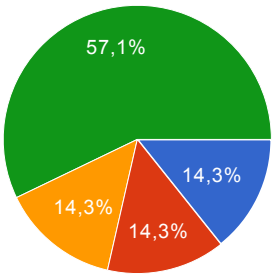
40%

85%

Using RAWFIE would significantly decrease entrance and early phases costs. As often my company would try different approaches before deciding to move to later stages. Reduction in cost would be significant, potentially up to 20% to 50%. However, in my mind the most important aspect is not the reducton in cost, but the reduction in time required to arrive to a first feasibility test. Depending on the urgency of a use case, time to market gains or even the ability to meet strict deadlnes might be more important.might

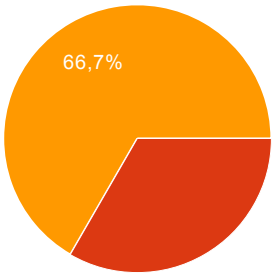
SFA interface

Have you ever used a testbed via a SFA interface?



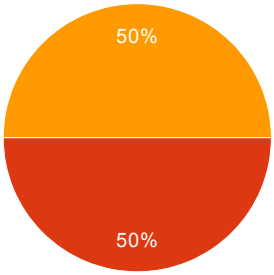
What is SFA?	**1**	14.3 %
Yes. It’s really useful.	**1**	14.3 %
Yes. But I like a more specialised interface.	**1**	14.3 %
No.	**4**	57.1 %

How valuable is an SFA interface in general for you?



Great	**0**	0 %
Good	**2**	33.3 %
Average	**4**	66.7 %
Low	**0**	0 %
Valueless	**0**	0 %

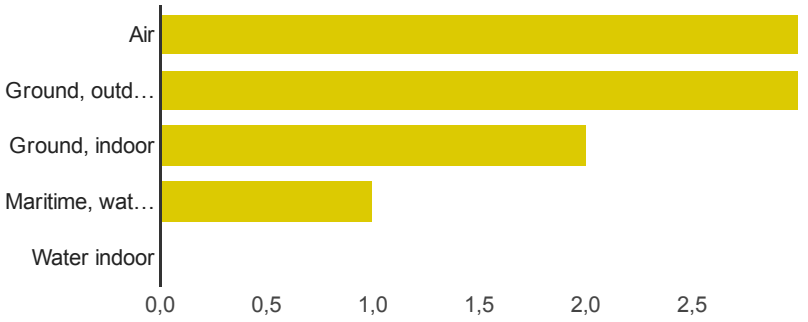
How valuable would an SFA interface for RAWFIE be for you



Great	**0**	0 %
Good	**3**	50 %
Average	**3**	50 %
Low	**0**	0 %
Valueless	**0**	0 %

Test bed integration

Which type of test bed can you provide



Air	**3**	75 %
Ground, outdoor	**3**	75 %
Ground, indoor	**2**	50 %
Maritime, water outdoor	**1**	25 %
Water indoor	**0**	0 %

Are there any constrains that must be obeyed in your testbed (e.g.: availability, hours of operation, number of UxV simultaneously operated)?

number of UxVs operated

no

What do you expect from an integration into the RAWFIE system?

access to a variety of diverse UxV platforms

dissemination of code/experience in developing UAVs/UGVs

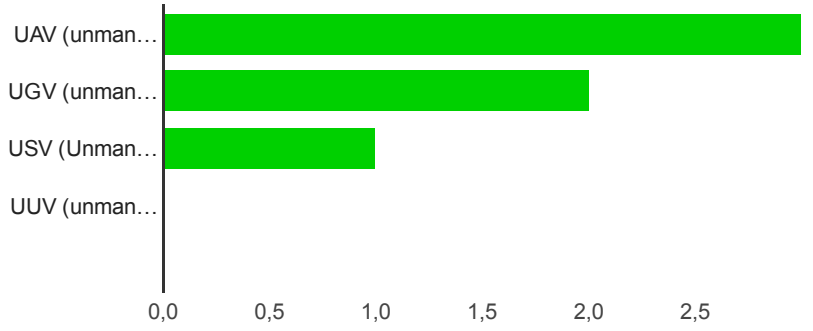
How many UxVs can your testbed host (approximately)?

5

2

UxV integration

Which kinds of UxVs can you provide?



UAV (unmanned aerial vehicle - aircraft, commonly known as a drone)	**3**	100 %
UGV (unmanned ground vehicle - vehicle that operates while in contact with the ground)	**2**	66.7 %
USV (Unmanned surface vehicles - vehicles that operate on the surface of the water)	**1**	33.3 %
UUV (unmanned underwater vehicle - vehicles that are able to operate underwater)	**0**	0 %

Are there any constrains that must be obeyed with your UxVs?

NO

no

MavLink compatible Communication Protocol

What do you expect from an integration into the RAWFIE system?

dissemination of SW/HW design

Feedback for UAV evolution, New test cases,

How many UxVs could you provide (on how many testbeds)?

10

two

12

Final comments

Any additional comments that you have about the RAWFIE system?

no

C Questionnaire single results

In the following pages the raw answers of the questionnaire are listed as table.

Timestamp	How old are you?	Which kind of organisation/company are you from?	What is your professional role?	What are your activities/responsibilities at your organisation/company ?	Which roles could be played by your company organisation/company (if any)?	In which kinds of UxVs are you interested?	Which kinds of UxVs have you worked with?	Have you ever made experiments with UxVs?	Have you ever used testbeds in general (not only UxV testbeds)?	Have you ever used UxV testbeds?	Which testbeds facilities did you use?
25.07.2016 15:02:46	40 to 49	UAS service providers	business development	R&D manager	UxV manufacturer	UAV, UGV,USV, UUV	UAV, UGV,USV	Yes	No	No	
27.07.2016 16:24:33	40 to 49	UxV Addition & Customization	CEO	CEO	UxV manufacturer		UAV, UGV,USV	No	No	No	
25.07.2016 18:04:21	40 to 49	University of Piraeus	Faculty	Research and Teaching	Experimenter	UAV, UGV,USV, UUV	UAV	No	No	No	
26.07.2016 13:59:22	30 to 39	Research / Higher Education	Professor	Research and Teaching	Experimenter	UAV, UGV,USV, UUV	UAV	No	Yes	No	UUV experiment
26.07.2016 22:10:35	50 to 59	University of Applied Sciences	Professor	Teaching-Research	Experimenter, Tesbed owner	UAV, UGV	UAV, UGV	Yes	Yes	No	MSP430 MCU DEBUGGER
26.07.2016 19:14:19	30 to 39	Electronics	Head of RD	RD Projects/Product Development	Experimenter	UAV, UGV,USV, UUV	UGV	No	Yes	Yes	Testbeds for our products
27.07.2016 12:43:23	30 to 39	IoT, RnD department, AGT International	Scenior Data Scienst	Researching and supervising application of data analytics solutions, using Internet of Things data. Helping evaluating which sensors and network configurations best fit each application needs.	Experimenter, Tesbed owner	UAV, UGV,USV	UAV, UGV	No	Yes	No	Mobile Emulab, PLANETLAB, EMULAB

Which functionalities/procedures of the used facilities according to your opinion are useful?	Which functionalities/procedures of the used facilities according to your opinion need improvement?	How would you incorporate RAWFIE into your projects?	Why would you use the RAWFIE platform?	Which kind of information are most relevant of an experiment?
		YES	# I need a variety of different UXVs from different vendors in order to test my product sufficiently.	# Live visualisation of the experiments including sensor values # Performing some standard data analytic algorithms on the gathered sensor data # Raw sensor data
			# I only need to perform some tests and don't want to buy UXVs for these few. # It is necessary to have many UXVs involved in one tests, but I only have too little of them. # I need a variety of different UXVs from different vendors in order to test my product sufficiently.	# Live visualisation of the experiments including sensor values # Performing some standard data analytic algorithms on the gathered sensor data # Raw sensor data
the ability to start/stop experiments on-demand, and viweing the sensor reading timeseries in real-time	add more customisability to the experimental configuration (needs to be supported by the platform too though)		# I only need to perform some tests and don't want to buy UXVs for these few. # I need a variety of different UXVs from different vendors in order to test my product sufficiently.	# Performing some standard data analytic algorithms on the gathered sensor data
ENERGY TRACER	OTA programming	MIDDLEWARE DEVELOPMENT FOR SENSING AND CONTROL OF UAVs/UGVs	# I need a variety of different UXVs from different vendors in order to test my product sufficiently.	# Live visualisation of the experiments including sensor values
		We can discuss it	# I need a variety of different UXVs from different vendors in order to test my product sufficiently.	# Live visualisation of the experiments including sensor values # Performing some standard data analytic algorithms on the gathered sensor data
Repeat-ability of experiments with the use of a script, ability to experiment with multiple mobile patterns to observe interactions, ability to record the experiment in detail, easiness to configure and debug experiments, ability to experiment with varying interface/protocol configurations		As an easy way for early experimentation / feasibility study. The data acquired could prove valuable to develop and debug planned analytics. It offers a platform where multiple patterns can be tested and the results can help debug and develop the analytics that require data driven approaches such as anomaly detection.	# It is necessary to have many UXVs involved in one tests, but I only have too little of them. # I need a variety of different UXVs from different vendors in order to test my product sufficiently.	# Live visualisation of the experiments including sensor values # Raw sensor data

In which phases during the product life cycle would you use RAWFIE?	Which functionalities of RAWFIE are most valuable for you?	Which functionalities of RAWFIE are not of interest for you?	Do you like the idea of a specialized EDL (experiment description language)?	Which kind of result formats would you like to have?	Which additional functionalities or information would you like to have provided by RAWFIE?	How would you perform most the tests?
# Early prototpye testing # Late prototype testing # Final product testing	# Experiment Monitoring and Visualisation # UxV remote control (live) # System Monitoring # Data Analytics	# EDL script editor # EDL visual/graphical editor # Resources Explorer # Booking of resources	Yes, but access to specific UxV commands should be possible somehow.	# Aggregated and analysed data		Short test interactions (prepare only a few tests, run and evaluate them; afterwards the next cycle begins)
# Case study # Feasibility study # Early prototpye testing # Late prototype testing # Final product testing	# EDL script editor # EDL visual/graphical editor # Resources Explorer # Booking of resources # Experiment Monitoring and Visualisation # UxV remote control (live) # System Monitoring # Data Analytics	# EDL script editor # EDL visual/graphical editor # Resources Explorer # Booking of resources # Experiment Monitoring and Visualisation # UxV remote control (live) # System Monitoring # Data Analytics		# Raw sensor data (UxV/sensor specific) # Homogenized sensor data (homogenized format for RAWFIE)		
# Case study # Early prototpye testing	# Experiment Monitoring and Visualisation # UxV remote control (live) # System Monitoring		Yes, but access to specific UxV commands should be possible somehow.	# Homogenized sensor data (homogenized format for RAWFIE)		Both equally
# Feasibility study	# UxV remote control (live)	# UxV remote control (live)	Yes, but access to specific UxV commands should be possible somehow.	# Raw sensor data (UxV/sensor specific) # Homogenized sensor data (homogenized format for RAWFIE)	Location data	Short test interactions (prepare only a few tests, run and evaluate them; afterwards the next cycle begins)
# Case study # Early prototpye testing	# EDL script editor # Experiment Monitoring and Visualisation # System Monitoring # Data Analytics		Yes, but access to specific UxV commands should be possible somehow.	# Homogenized sensor data (homogenized format for RAWFIE), Aggregated and analysed data		Short test interactions (prepare only a few tests, run and evaluate them; afterwards the next cycle begins)
# Case study # Feasibility study # Early prototpye testing	# EDL script editor # EDL visual/graphical editor # Experiment Monitoring and Visualisation # UxV remote control (live) # System Monitoring	# EDL script editor # EDL visual/graphical editor # Resources Explorer # Booking of resources # Experiment Monitoring and Visualisation # UxV remote control (live) # System Monitoring # Data Analytics	Yes, I don’t want to worry about UxV specific command.	# Raw sensor data (UxV/sensor specific) # Homogenized sensor data (homogenized format for RAWFIE)		Both equally

Which kinds of experiments would you perform with RAWFIE?	How valuable/useful would RAWFIE be for you?	How much would you pay per hour and UxV?	Please tried to estimate of the cost reduction (person months) by using RAWFIE for your experiments instead of build your experimentation platform yourself (in %)	Have you ever used a testbed via a SFA interface?	How valuable is an SFA interface in general for you?
	Great		30	No.	Average
	Good			Yes. It's really useful.	Average
	Good			What is SFA?	
	Great	somethiing equivalent to what Cloud Computing platforms charge for (e.g., $1 per hour, depending on functionality provided)	70%	No.	Average
real-time sensing and control using Delay Tolerant Networks	Great	100	85%	No.	Average
	Great		40%	No.	Good
	Good	[Disclaimer: I would not be the one to address / decide the cost questions, only advise, and it would fall to the specific product owner to decide based on his budget, his goals, the alternatives, and most importatly and his urgency] Depending on the scale of experiments, for deciding if the testbed should be used it is possible that a trial of the testbed of ~<1K would be approved, then the results internaly presented and benefits evaluated, before proceeding to more investment. I am not sure how this would be epected to be mapped to hours/UxVs	Using RAWFIE would significantly decrease entrance and early phases costs. As often my company would try different approaches before deciding to move to later stages. Reduction in cost would be significant, potentially up to 20% to 50%. However, in my mind the most important aspect is not the reducton in cost, but the reduction in time required to arrive to a first feasibility test. Depending on the urgency of a use case, time to market gains or even the ability to meet strict deadlnes might be more important.might	Yes. But I like a more specialised interface.	Good

How valuable would an SFA interface for RAWFIE be for you	Which type of test bed can you provide	Are there any constrains that must be obeyed in your testbed (e.g.: availability, hours of operation, number of UxV simultaneously operated)?	What do you expect from an integration into the RAWFIE system?	How many UxVs can your testbed host (approximately)?	Which kinds of UxVs can you provide?	Are there any constrains that must be obeyed with your UxVs?	What do you expect from an integration into the RAWFIE system?2	How many UxVs could you provide (on how many testbeds)?	Any additional comments that you have about the RAWFIE system?
Average	Air, Ground, outdoor, Ground, indoor, Maritime, water outdoor				UAV, UGV,USV	NO		10	
Good					UAV	MavLink compatible Communication Protocol	Feedback for UAV evolution, New test cases,	12	
	Air, Ground, outdoor								
Average	Ground, indoor	number of UxVs operated	access to a variety of diverse UxV platforms	5					
Average	Air, Ground, outdoor	no	dissemination of code/experience in developing UAVs/UGVs	2	UAV, UGV	no	dissemination of SW/HW design	two	no
Good									
Good									

D Avro Shema Messages

Goto.avsc

```
{
  "namespace": "eu.rawfie.uxv.commands",
  "name": "Goto",
  "type": "record",
  "doc": "Command a system to move to a given location at a given speed",
  "fields": [
    {
      "name": "header",
      "type": "eu.rawfie.uxv.Header"
    },
    {
      "name": "location",
      "type": "eu.rawfie.uxv.Location"
    },
    {
      "name": "speed",
      "type": [
        "float",
        "null"
      ],
      "unit": "m/s"
    },
    {
      "name": "timeout",
      "type": "float",
      "unit": "s"
    }
  ]
}
```

Header.avsc

```
{
  "namespace": "eu.rawfie.uxv",
  "name": "Header",
  "type": "record",
  "fields": [
    {
      "name": "sourceSystem",
      "type": "string",
      "doc": "Canonical name of the originating system"
    },
    {
      "name": "sourceModule",
      "type": "string",
      "doc": "Canonical name of the module within a given system that
originated the message"
    },
    {
      "name": "time",
      "type": "long",
      "unit": "ms",
      "doc": "Time elapsed since the Unix epoch"
    }
  ]
}
```

E Abbreviations

Table 2 gives the abbreviations used across the RAWFIE projects in the documents and deliverables.

Abbreviation	Meaning
3D	three-dimensional space
ACL	Access Control List
AGL	Above Ground Level
AHRS	Attitude and Heading Reference System
AJAX	Asynchronous JavaScript and XML
AM	Aggregate Manager (of SFA)
AP	Access Point
API	Application Programming Interface
API	Application programming interface
AT	Aerial Testbed
AUV	Autonomous underwater vehicle
B-VLOS	Beyond Visual Line Of Sight
CA	Certification Authority
CAA	Civil Aviation Authority
CAO	Cognitive Adaptive Optimization
CBNR	Chemical Biological Nuclear Radiological
CEP	Circular Error Probability
CPU	Central Processing Unit
CSR	Certificate Signing Request
DETEC	Department of the Environment, Transport, Energy and Communication
DGCA	Directorate General of Civil Aviation
DoA	Description of Actions
EASA	European Aviation Safety Agency
EC	Experiment Controller
ECC	Error Correction Code
ECV	EDL Compiler & Validator
EDL	Experiment Description Language
EDL	Experiment Description Language
EER	Experiment and EDL Repository
EU	European Union
E-VLOS	Extended Visual Line Of Sight
EVS	Experiment Validation Service
FIRE	Future Internet Research & Experimentation
FOCA	Federal Office of Civil Aviation
FPS	Frames Per Second
FPV	First Person View
GAA	German Aviation Act
GIS	Geographic Information System
GNSS	Global Navigation Satellite System
GPIO	General Purpose Input/Output
GPS	Global Positioning System
GUI	Graphical user interface

HD	High Definition
HTTP	Hypertext Transfer Protocol
HW	Hardware
IAA	Irish Aviation Authority
IaaS	Infrastructure as a Service
IDE	Integrated Development Environment
IDE	integrated development environment
IFR	Instrument Flight Rules
IP	Internet Protocol
ISO	International Standards Organization
JDBC	Java Database Connectivity
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
KPI	Key Performance Indicator
LBL	Long Baseline
LDAP	Lightweight Directory Access Protocol
LS	Launching Service
MEMS	MicroElectroMechanical System
MM	Monitoring Manager
MSO	Multi Swarm Optimization
MT	Maritime Testbed
MOM	Message Oriented Middleware
MVC	Model View Controller
NAT	Network Address Translation
NC	Network Controller
NF	Non Functional
ODBC	Open Database Connectivity
OEDL	OMF EDL
OMF	cOntrol and Management Framework
OMF	Orbit Management Framework
OML	ORBIT Measurement Library
OS	Operating System
OTA	Over The Air
P2P	Point to Point
PSO	Particle Swarm Optimization
PTZ	Pan Tilt Zoom
RC	Resource Controller
RC	Resource Controller
RE	Requirement Engineering
REST	Representational state transfer
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
RSpec	SFA Resource Specification
SaaS	Software as a Service
SAML	Security Assertion Markup Language

SFA	Slice-based Federation Architecture
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Simple Query Language
SSO	Single-Sign-On
SVN	Apache Subversion
TM	Testbed Manager
TMS	Testbed Manager Suite
TP	Testbed Proxy
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UI	User Interface
UML	Unified Modelling Language
USV	Unmanned Surface Vehicle
UUV	Unmanned Underwater Vehicle
UxV	Unmanned aerial/ground/surface/underwater Vehicle
VE	Visualization Engine
VT	Vehicular Testbed
VT	Visualization Tool
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WPS	Web Processing Service
WSDL	Web Services Description Language
XMPP	Extensible Messaging and Presence Protocol

Table 2: Common abbreviations

Table 3 gives the notations used in the RAWFIE documents and deliverables.

Notation	Description
D*X.Y*	Deliverable *X.Y* from the DoW
MS*X*	Milestone *X* from the DoW
WP*X*	Work package *X* from the DoW
OC*X*	Open Call *X*
A*X.Y*	Activity number *Y* in Phase *X*
DL*X.Y*	Deadline number *Y* in Phase *X*
M*X*	Project month number *X*

Table 3: Notation

F Glossary

The RAWFIE glossary consists of generic terms, contributed by all partners.

A

Accounting Service

RAWFIE component. Component that keeps track of resources usage by individual users.

Aggregate Manager

Slice Federation Architecture (SFA) term. The Aggregate Manager API is the interface by which experimenters discover, reserve and control resources at resource providers.

Avro

Apache Avro: a remote procedure call and data serialization framework

B

Booking Service

RAWFIE component. The Booking Service manages bookings of resources by registering data to appropriate database tables.

Booking Tool

RAWFIE component. The Booking tool will provide the appropriate Web UI interface for the experimenter to discover available resources and reserve them for a specified period.

C

Common Testbed Interface

RAWFIE component. The set of software and hardware functionalities each Testbed provider should ensure, for the communication with Middle Tier software components of RAWFIE, therefore for the integration with the RAWFIE platform

Component

A reusable entity that provides a set of functionalities (or data) semantically related. A component may encapsulate one or more modules (see definition) and should provide a well defined API for interaction

D

Data Analysis Engine

RAWFIE component. The Data Analysis Engine enables the execution of data processing jobs by sending requests to a processing engine which will perform the computations specified when the analytical task was defined through the Data Analysis Tool to be transmitted to the processing engine for execution.

Data Analysis Tool

RAWFIE component. The Data Analysis Tool enables the user to browse available data sources for subject to analytical treatment as well as previous analysis tasks' outcomes.

E

EDL Compiler & Validator

RAWFIE component. The EDL validator will be responsible for performing syntactic and semantic analysis on the provided EDL scripts.

Experiment Authoring Tool

RAWFIE component. This component is actually a collection of tools for defining experiments and authoring EDL scripts through RAWFIE web portal. It will provide features to handle resource requirements/configuration, location/topology information, task description etc.

Experiment Controller

RAWFIE component. The Experiment Controller is a service placed in the Middle tier and is responsible to monitor the smooth execution of each experiment. The main task of the experiment controller is the monitoring of the experiment execution while acting as 'broker' between the experimenter and the resources.

Experiment Monitoring Tool

RAWFIE component. Shows the status of experiments and of the resources used by experiments.

Experiment Validation Service

RAWFIE component. The Experiment Validation Service will be responsible to validate every experiment as far as execution issues concern.

M

Master Data Repository

RAWFIE component. Repository that stores all main entities that are needed in the RAWFIE platforms. Is an SQL-database

Measurements Repository

RAWFIE component. Stores the raw measurements from the experiments

Message Bus

Also known as Message Oriented Middleware. A message bus is supports sending and receiving messages between distributed systems. It is used in RAWFIE across all tiers to enable asynchronous, event-based messaging between heterogeneous components. Implements the Publish/Subscribe paradigm.

Module

A set of code packages within one software product that provides a special functionality

Monitoring Manager

RAWFIE component. Monitors the status of the testbed and the UxVs belonging to it, at functional level, e.g. the 'health of the devices' and current activity.

N

Network Controller

Manages the network connections and the switching between different technologies in the testbed in order to offer seamless connectivity in the operations of the system.

L

Launching Service

RAWFIE component. The Launching Service is responsible for handling requests for starting or cancellation of experiments.

R

Resource Controller

RAWFIE component. The Resource Controller can be considered as a cloud robot and automation system and ensures the safe and accurate guidance of the UxVs.

Resource Explorer Tool

RAWFIE component. The experimenter can discover and select available testbeds as well as resources/UxVs inside a testbed with this tool. Administrators can manage the data.

Results Repository

RAWFIE component. Stores the results of data analyses.

Resource Specification (RSpec)

SFA term. This is the means that the SFA uses for describing resources, resource requests, and reservations (declaring which resources a user wants on each Aggregate).

S

Schema Registry

A schema registry is a central service where data schemas are uploaded to. As an added benefit each schema has versions with it can convert allowable formats to other ones (e.g.: float to double) It maintains schemas for the data transferred and keeps revisions to be able to upgrade the definitions as with the simple field conversion. Used in RAWFIE for messages on the message bus.

Service

A component that is running in the system, providing specific functionalities and accessible via a well known interface.

Slice Federation Architecture (SFA)

SFA is the de facto standard for testbed federation and is a secure, distributed and scalable narrow waist of functionality for federating heterogeneous testbeds.

Subsystem

A collection of components providing a subset of the system functionalities.

System

A collection of subsystems and/or individual components representing the provided software solution as a whole.

System Monitoring Service

RAWFIE component. Checks readiness of main components and ensure that all critical software modules will perform at optimum levels. Predefined notification are triggered whenever the corresponding conditions are met, or whenever thresholds are reached

System Monitoring Tool

RAWFIE component. Shows the status and the readiness of the various RAWFIE services and testbed

T

Testbed

A testbed is a platform for conducting rigorous, transparent, and replicable testing of scientific theories, computational tools, and new technologies.

In the context of RAWFIE, a testbed or testbed facility is a physical building or area where UxVs can move around to execute some experiments. In addition, the UxVs are stored in or near the testbed.

Testbeds Directory Service

RAWFIE component. Represents a registry service of the middleware tier where all the integrated testbeds and resources accessible from the federated facilities are listed, belonging to the RAWFIE federation.

Testbed Manager

RAWFIE component. Contains accumulated information about the UxVs resources and the experiments of each one of the federation testbeds.

Tool

A GUI implementation to do a special thing, e.g. the "Resource Explorer tool" to search for a resource

U

Users & Rights Repository

RAWFIE component. Management of users and their roles. Is a directory services (LDAP).

Users & Rights Service

RAWFIE component. Manages all the users, roles and rights in the system.

UxV

The generic term for unmanned vehicle. In RAWFIE, it can be either:

USV - Unmanned Surface vehicle.

UAV - Unmanned Aerial vehicle.

UGV - Unmanned Ground vehicle.

UUV - Unmanned Underwater vehicle.

UxV Navigation Tool

RAWFIE component. This component will provide to the user the ability to (near) real-time remotely navigate a squad of UxVs.

UxV node

RAWFIE component. A single UxV node. The UxV is a complete mobile system that interacts with the other Testbed entities. It can be remotely controlled or able to act and move autonomously.

V

Visualisation Engine

RAWFIE component. Used for providing the necessary information to the Visualisation tool, to communicate with the other components, to handle geospatial data, to retrieve data for experiments from the database, to load and store user settings and to forward them to the visualisation tool.

Visualisation Tool

RAWFIE component. Visualisation of an ongoing experiment as well as visualisation of experiments that are already finished

W

Web Portal

RAWFIE component. The central user interface that provides access to most of the RAWFIE tools/services and available documentation.

Wiki Tool

RAWFIE component. Provides documentation and tutorials to the users of the platform.

References

[1] Evaluation Research Team, *Brief 12: Using Graphs and Charts to Illustrate Quantitative Data*, Centers for Disease Control and Prevention, 2008, https://www.cdc.gov/healthyyouth/evaluation/pdf/brief12.pdf

[2] Evaluation Research Team, *Brief 14: Data Collection Methods for Program Evaluation: Questionnaires*, Centers for Disease Control and Prevention, 2008, https://www.cdc.gov/healthyyouth/evaluation/pdf/brief14.pdf

[3] Evaluation Research Team, *Brief 16: Data Collection Methods for Program Evaluation: Observation*, Centers for Disease Control and Prevention, 2009, https://www.cdc.gov/healthyyouth/evaluation/pdf/brief16.pdf

[4] Evaluation Research Team, *Brief 17: Data Collection Methods for Program Evaluation: Interviews*, Centers for Disease Control and Prevention, 2009, https://www.cdc.gov/healthyyouth/evaluation/pdf/brief17.pdf

[5] Evaluation Research Team, *Brief 19: Analyzing Qualitative Data for Evaluation*, Centers for Disease Control and Prevention, 2009, https://www.cdc.gov/healthyyouth/evaluation/pdf/brief19.pdf

[6] Evaluation Research Team, *Brief 12: Analyzing Quantitative Data for Evaluation*, Centers for Disease Control and Prevention, 2009, https://www.cdc.gov/healthyyouth/evaluation/pdf/brief20.pdf