Information and Communication Technologies (ICT) Programme

Project N°: H2020-ICT-2016-1-732105

D2.4: CERBERO Scenarios Description

Lead Beneficiary:	TASE
Workpackage:	WP2
Date:	06-03-2018
Distribution - Confidentiality:	Public

Abstract:

This report is an update of deliverable D2.3, and it is meant to provide user needs and use case requirements update of the originally defined CERBERO use-cases: Smart Travelling, Self-Healing for Planetary Exploration and Ocean Monitoring.

Disclaimer

The CERBERO Consortium is the following:

Num.	Beneficiary name	Acronym	Country
1 (Coord.)	IBM Israel – Science and Technology LTD	IBM	IL
2	Università degli Studi di Sassari	UniSS	IT
3	Thales Alenia Space Espana, SA	TASE	ES
4	Università degli Studi di Cagliari	UniCA	IT
5	Institut National des Sciences Appliquees de Rennes	INSA	FR
6	Universidad Politécnica de Madrid	UPM	ES
7	Università della Svizzera italiana	USI	CH
8	Abinsula SRL	AI	IT
9	Ambiesense LTD	AS	UK
10	Nederlandse Organisatie Voor Toegepast Natuurwetenschappelijk Ondeerzoek TNO	TNO	NL
11	Science and Technology	S&T	NL
12	Centro Ricerche FIAT	CRF	IT

For the CERBERO Consortium, please see the http://cerbero-h2020.eu web-site.

Document Authors

The following list of authors reflects the major contribution to the writing of the document.

Name(s)	Organization Acronym
Joost Adriaanse	TNO
Andreea Balau	TNO
Antonio Lopez Varona	TASE
Manuel Sanchez	TASE
Sergio Tardon	TASE
Leszek Kaliciak	AS
Hans Myrhaug	AS
Ayse Goker	AS
Francesca Palumbo	UniSS
Michael Masin	IBM

Document Revision History

Date	Ver.	Contributor (Beneficiary)	Summary of main changes
05-01-2018	0.1	TNO/AS/TASE/CRF/S&T	First draft
15/01/2018	0.2	TNO	Adding sections (4.1 - 4.3)
17/01/2018	0.3	AS	Adding sections (4.1 - 4.3)
22/01/2018	0.4	AS	Refining sections (4.1 - 4.3), adding section 4.4 and 4.5
27-01-2018	0.5	TASE	Adding sections (3.1 – 3.5)
29-01-2018	0.6	UNISS	UniSS review
23-02-2018	0.7	TNO	Refining sections (4.1 - 4.5)
26-02-2018	0.8	TASE	Refining sections (4.1 - 4.5)
02-03-2018	0.9	AS	Refining sections (4.1 - 4.5)
06-03-2018	1.0	UNISS	UniSS second review

Table of contents

1. Executive Summary 5
1.1. Structure of Document 5
1.2. Related Documents 5
2. Use Case description Smart Travelling for Electric Vehicles 6
2.1. Introduction 6
2.2. Challenge and goals 6
2.3. System architecture and components 9
2.4. Use-Case vs. Technology mapping 12
2.5. Update on requirements 13
3. Use Case Self-Healing System for Planetary Exploration 14
3.1. Introduction 14
3.2. Challenge and goals 14
3.3. System architecture and components 15
3.3.1. Robotic Arm 17
3.3.2. BLDC Motor 18
3.4. Use-Case vs. Technology mapping 19
3.5. Update on requirements 21
4. Use Case Ocean Monitoring 22
4.1. Introduction 22
4.2. Challenge and goals 23
4.3. System architecture and components 25
4.3.1. Sensors and Information Fusion 28
4.3.2. Adaptive Camera System – design-time and physical prototype 29
4.3.3. Battery system and prediction modeling 33
4.4. Use-Case vs. Technology mapping 33
4.5. Update on requirements 35
5. Conclusions 37
6. References 38

1. Executive Summary

This document is meant to provide an overview of all the application scenarios that are going to be used to assess the CERBERO framework and methodologies.

For each scenario:

- the main challenges and goals are going to be presented, along with the solutions featured by CERBERO to address them.
- the overview of the different parts composing the cyber-physical environment are presented, to clarify where CERBERO technologies/tools intervene to facilitate and improve design, integration, deployment and verification phases.

Please note that more detailed descriptions on how technologies and tools will be used in each scenario is going to be provided in D6.7 - Demonstration Skeleton (Ver 1).

1.1. Structure of Document

This document is organized per scenario: Section 2 reports on the Smart Travelling for Electric Vehicle, Section 3 on the Self Healing for Planetary Exploration, and Section 4 on the Ocean Monitoring.

For each scenario, this document presents challenges, general system set-up and involved CERBERO technologies.

1.2. Related Documents

This document is related to:

- D2.3 (CERBERO Scenarios Description - Ver 1) and D2.6 (CERBERO Technical Requirements - Ver 1) that have been already approved by commission. These two deliverables described the generalities and features of the three CERBERO use cases, the technical specifications we derived from them and the project scientic challenged to drive all the project activities. In particular, the present document is an evolution of D2.3 and represent the base for the evolution of D2.6.
- Task 6.1 of the CERBERO project is meant to define the link between CERBERO technologies and advances and their assessment in the demonstrators. D6.7 (Demonstration Skeleton - Ver 1) and its updates (the main outcomes of T6.1) will include the details of the use case to technology mapping for each demonstrator, which are partially introduced in the present document, and will describe the testing environment and assessment plans for each use case.
- D6.8 (Space Demonstrator - Ver 1), D6.9 (Ocean Monitoring Demonstrator - Ver 1) and D6.10 (Smart Travelling Demonstrator - Ver 1) will receive the present deliverable and D6.7 as input to discuss the results of the demonstration and to derive indications and guidelines for the next 18 months of the project based on achieved results.

2. Use Case description Smart Travelling for Electric Vehicles

2.1. Introduction

In this section the latest insights on the Smart Travelling for Electric Vehicles use case are described. The generalities of this scenario have already been covered in D2.3. In this document we intend to include new insights from the last reporting period on challenges, goals and prototyping architecture/components. Mapping among use case and CERBERO technologies and tools is also provided, which will be refined in D6.7 (demonstration skeletons) and in the demonstration-related deliverable (D6.10 due at M18).

2.2. Challenge and goals

Real time system-in-the-loop simulations are generally extremely challenging, since they require the co-simulation of a running system with physical and/or cyber models, requiring timely data exchange among them. Things get even worse when adaptive scenarios have to be supported. The Smart Travelling use case presents mainly functional adaptation needs, and it is hybrid since physical environment and components are controlled by the cyber parts.

In this scenario CERBERO technologies and tools will coexist with the SCANeR electric vehicle simulator, made available by CRF. Therefore, in addition to the issues described above, to meet CRF desiderata on the SCANeR side, CERBERO framework has to enable the possibility of:

1. Testing different driving experience in electric vehicles, being able to simulate different and very specific electric motors and batteries (not vendor dependent, like Oktal). Modularity, re-usability and composability features are, therefore, mandatory for the co-simulation environment that CERBERO should offer.
2. Supporting future vehicles, with extensive intelligent support functionalities that will greatly influence the driving experience. Dedicated modules have to be added to SCANeR for evaluation and testing of human interaction.
3. Providing situation-aware driver support functionalities, which can pro-actively make accurate predictions of the impact of detected behaviours, and consequently support the driver on future choises. The co-simulation environment has to allow functional adaptivity based on predictions and drivers' preferences. (Self-)adaptation has to be effectively provided by CERBERO technologies: runtime monitored data should feed proper models running on (self-)adaptation managers, to trigger the approriate reconfiguration behaviours executed by (distributed) reconfiguration agent(s).
4. Providing strong real-time guarantees. Prediction calculations are time critical, as the driver should be provided with advice well before choice (e.g. for route or charging) needs to be known.

Table 2-1 –Smart Travelling General Needs

ID	Need	High Level Requirement
ST1	Reduction of costs, increase of reuse in different simulation scenarios.	Development of parametric, modular and extendable cyber-physical co-simulation environment.

ID	Need	High Level Requirement
ST2	Reduce time of development, verification, integration, along with the related costs, exploiting a library of reusable components/metrics integrated by common framework in different levels of abstraction. Incremental prototyping.	Development of an integrated open-source or commercially available toolchain for design space exploration and co-simulation, with system-in-the-loop capabilities.
ST3	Efficient support of functional adaptivity, according to system, human and environment triggers.	Development of a (self-)adaptation methodology with supporting tools.

Table 2-1 (as well as Table 3-1 and Table 4-1 in the following sections) is meant to generalize the fundamental needs, derived from the considered use case general challenges. The high-level part of the CERBERO stack (as it becomes evident in Table 2-3 of Section 2.5) will be primarily used to address those needs. This use case is meant to demonstrate how CERBERO will allow easier system-in-the-loop co-simulation and execution, and how it will allow easier accessessment of the impact and performance of new additional components.

In order to address these needs, the following technological challenges have been identified:

- CERBERO methodologies and tools will be used to model and implement situation-aware adaptive CPS.
- The DynAA simulation tool will serve as a bridge among SCANeR and CERBERO components extending SCANeR functionality (MECA, additional copies of DynAA, and, later, AOW), requiring DynAA to be extended to operate as system-in-the-loop during real-time simulations.
- MECA tool, which is a decision making system, will be adapted to monitor the environment by collecting data from external systems, and to act as (self-)adaptation manager to trigger functional reconfiguration of vehicle routing at the system level.
- The combination of DynAA and MECA will enable time-constrained look ahead predictions calculations of vehicle status an potential routes (by executing in real-time multiple simulation scenarios), providing also cross-layer KPI optimisation (where optimization is done using different models, such as physical and functional models).

For the first demonstrator implementation (M18) we expect to:

Verify the condition awareness capabilities of the CERBERO framework for real-time system-in-the-loop simulation:

1. Integration of external electric motor and battery models into the SCANeR driving simulator using DynAA as a system-in-the-loop (to emulate the dynamics of a real electric vehicle). The integration should support extensive monitoring of the vehicle

behaviour and its environment, to simulate behaviour of motor and battery in a realistic manner and adapt in case of relevant changes of the environment.

2. Synchronise logging data from different distributed simulation modules, to obtain consistent and accurate overall logging data (to be used for detailed scenario analysis).

Continuous system monitoring for adaptiveness management:

1. Integrate and provide basic mechanism in DynAA for vehicle simulations, which can be used to make predictions of the impact of specific routes. Integration should provide sufficient monitoring data in order to adapt based on changes in the environment.
2. Integrate MECA with DynAA and SCANeR and provide the basic mechanism in MECA needed to support driver support. Figure 2-1 shows the intended interfaces and describes the main steps of the mechanism. The interfaces are shown as arrows and steps are indicated with number:
   (1) Data will continuously flow into MECA: vehicle data (SCANeR), sensor data (DynAA), and user data. User data will be input via a TBD development purpose User Iinterface[1] (e.g. dislay, tablet, etc.), UI.
   (2) Data is processed by MECA to trigger an action (e.g., battery level low; re-optimize route).
   (3) A route (or multiple routes) is (are) sent to DynAA for impact analysis.
   (4) Impact analysis result and a first selection of feasible route(s) is returned to MECA.
   (5) Action is triggered in MECA to process the analysis result(s)
   (6) Results (based on specific KPIs) are presented to user via TBD UI[2].

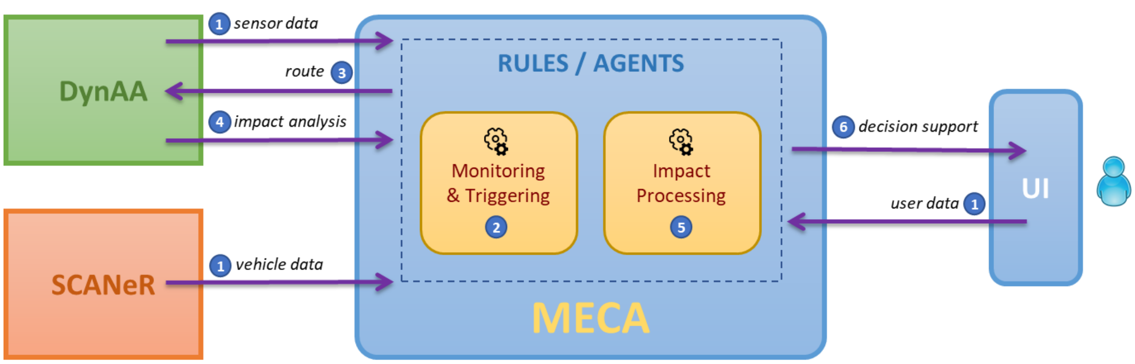

Figure 2-1: Schematic overview of planned activities for MECA for M18.

[1] More details on the interfaces will be provided in D6.10, where the details of the Smart Travelling demonstrator are going to be described. Please note that the user of CERBERO results are not the final drivers, but driving simulator developers. Therefore, the definition of the UI is mainly for demonstration purposes.
[2] More details on the interfaces will be provided in D6.10 where the details of the Smart Travelling demonstrator are going to be described.

The goal related to shortening the 3D database development are no longer in scope of the CERBERO project, as the related technologies are very use case specific and it is expected that these developments will not contribute to the development of a generic CERBERO tool chain for CPS.

Although most essential elements will be integrated in SCANeR driving simulator for the first demonstrator, it is expected that only parts of the use case scenarios can be demonstrated in the first demonstrator release at M18. Details will be provided in D6.7. The demonstrator of M18 will include the skeleton in which the different tools (SCANeR, DynAA and MECA) are integrated and which can be used to implement individual use case scenarios.

Further integration, during M18-M36 timeframe, of the DynAA, MECA and also AOW (an optimization tool by IBM, which is capable of exploring and pruning a large design space exploiting Mathematical Linear Programming) tools will be needed to implement complete use case scenarios, which will include an integrated set of functionalities like triggering of the driving assistant, route calculation, charging optimisation and route predictions during an actual driving simulation run. This complete integration will be performed during the implementation of the second demonstrator (M36).

Goals for the second demonstrator will include:

1. Simulation of complete use case scenarios where the electric vehicle and driver support functionalities (using MECA, DynAA and AOW) will respond to user actions and triggers from the environment to support user during the travel with appropriate advice;
2. Use of simulation modules in DynAA for calculation predictions for driver support, which will be initiated by MECA;
3. Use AOW for optimal routing calculation and finding optimal charging solution for use in driver advice by MECA;
4. Evaluate “what-if” scenarios using MECA for preferences based driver support advice.

2.3. System architecture and components

In Figure 2-2 an overview of the Smart Travelling demonstrator components is depicted: the CRF driving simulator is connected to different tools of the CERBERO framework (DynAA, AOW and MECA). This picture shows also some models and blocks that will be developed during the project timeframe. TNO models (Electric Model, Battery Model, Vehicle Model and Charging Infrastructure Model) will adehere to the methodologies defined in WP3, while the Route Calculation and the Knowledge Base will be developed by S&T in order to provid input to MECA.

The components added by CERBERO are:

- **Model simulator** (DynAA) in which different simulation modules can be executed, either concurrently with the simulation in SCANeR (and thus executed in real-time) or as part of a predicted route provided by MECA (and thus executed much faster than real-time). The model simulator will be based on the DynAA tool, which will run in a system-in-the-loop mode and execute and synchronise with different simulation modules required for the Smart Travelling use case. DynAA, on the one hand, will be connected to SCANeR to provide parts of the vehicle simulations and, on the other, to MECA to provide simulated predictions (using optional parallel execution of several possible routes). In future, there could also be connections to AOW, to provide simulations of, for example, the charging infrastructure.

- **Driver Support** (MECA) which provides requested or pro-active advice (i.e. on optimal route) to the driver of the vehicle on optimal routes to travel (based on battery charge and predicted impact). The driver support functionality will be based on the generic MECA tool, which will be extended with generic monitoring and situation-aware supervisor functionalities.
- **AOW optimizer** which can calculate optimal routes or charging schemes (where optionally complete fleet charging could be taken into account). The AOW optimizer will use the generic AOW tool to calculate optimal solutions (i.e. for routes and charging options) as requested by the driver support functionality.

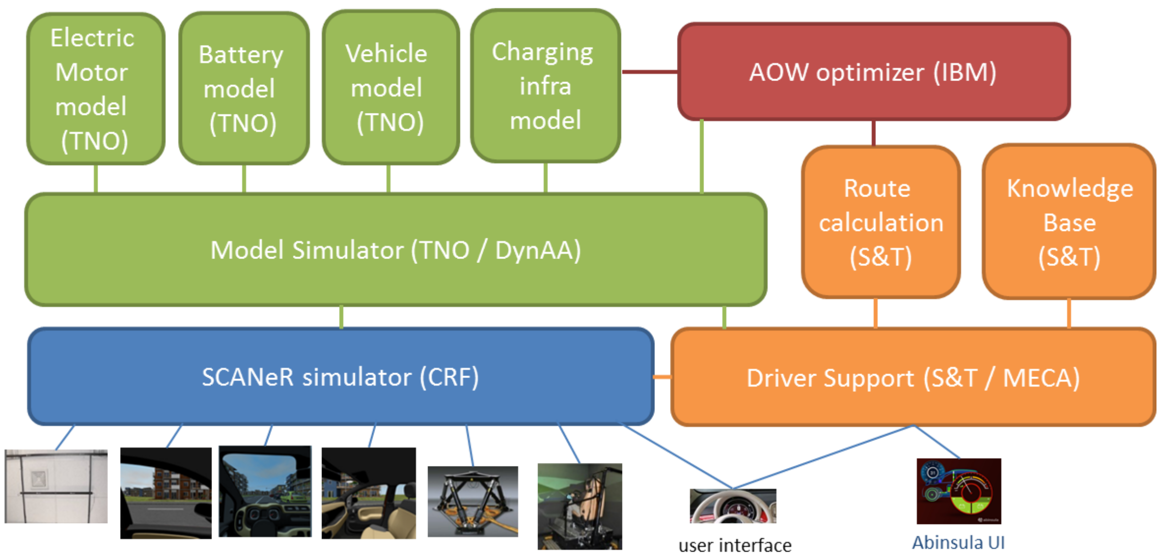

Figure 2-2: Smart Travelling demonstration components

Modes of operation

In the Smart Travelling use case there are actually two modes of operation in which the different components are used: the vehicle simulation mode and the driver support mode.

In the vehicle simulation mode the simulations are focussed on real time simulation of the behaviour of the different components of the electric vehicle (like battery and electric motor). In the driver support mode, simulations are used to predict the impact of possible future routes on energy usage and battery consumption. These simulations are executed in a much higher frequency than real time and the requirements on accuracy are probably less restrictive (given many other uncertainties not included in the calculations will affect the energy consumption).

The different simulations will have to focus on different KPIs. The vehicle simulation should focus on accurate and real-time behaviour of the (sub-)systems, while the driver support related predictions should focus on cost and energy reduction, response time (when advice should be available) and driver preferences.

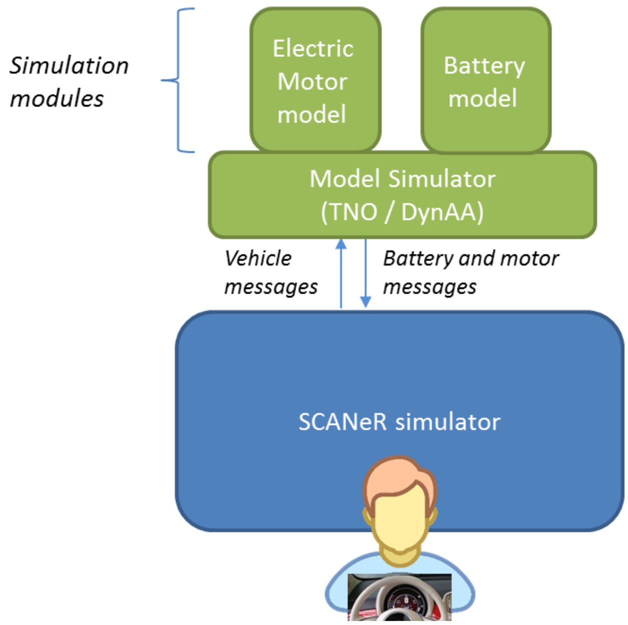

Figure 2-3: Smart Travelling - vehicle simulation mode

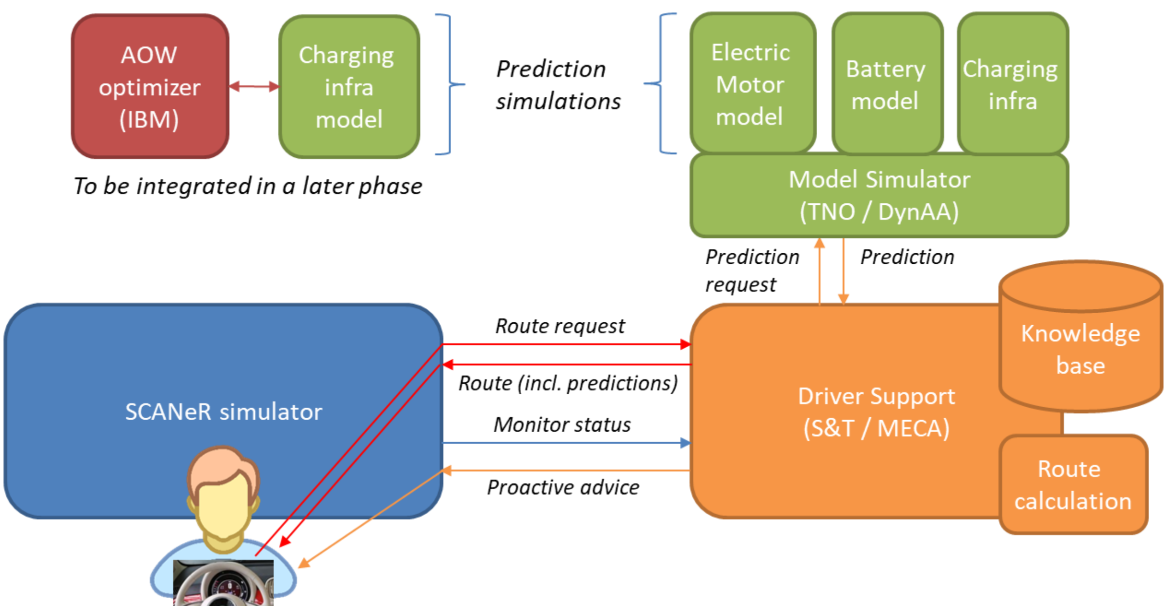

Figure 2-4: Smart Travelling - driver support and predictions mode

Both simulations modes should be able to handle and adapt based on different type of triggers, such as:

▪ Environment (environment-awareness)

▪ System (self-awareness)

▪ Human (user-commanded).

The battery model for example will need to react to changes in temperature (which will change the reacheability of capacity and affect the efficiency) while the driver advice module should react when driver deviates from earlier advised route (and probably recalculate for new situation) or planned charging station becomes out of order (requiring a replanning). The two modes will thus demonstrate that the system can handle the goals and challenges set for the use case (condition awareness and adaptiveness).

The adaptation strategies include:

- *Battery model* - will adapt its charging capability based on temperature and its age (number of charging cycles performed);
- *Charging infra model* – will adapt based on events like outage of electricity in specific charging poles areas, defects (poles out of order), occupation of specific charging poles, etc.
- *Driver support* – will adapt its advice based on possible deviation from the route (by driver), detected congestions on the route, issues with planned charging poles, etc.
- *Route calculation* – will adapt based on traffic jams, accidents, road maintenance, etc

2.4. *Use-Case vs. Technology mapping*

In the Smart Travelling use case a number of different models and tools are used to accomplish the added functionalities and implement the requested adaptivity support. In Table 2-3 a summary is given, we report all the CERBERO technologies (tools and/or models) that will be used to meet the Smart Travelling needs (see Table 2-1), their purpose in the use case and the specific KPI they address and/or optimize. Not all the following CERBERO technologies are expected to be demonstrated at M18 in the context of smart travelling, the details on M18 demonstrators will be provided afterwards in D6.7 and D6.10.

Table 2-3 Smart travelling Use case vs Technology mapping

Component (model / tool)	**Functionality**	**Purpose**	**(Generic) KPIs addressed**
DynAA system-in-the-loop adapter	System-in-the-loop simulation	Synchronised simulation to run multiple simulation modules which interact with the SCANeR simulation.	• Response time to trigger • Latency
Battery model	Simulation of battery behaviour (to be executed in DynAA)	Equip vehicle with parametric simulation model of battery and use it in the loop for predicting impact of possible future routes.	• Energy • Response time to trigger
Electric motor model	Simulation of electric motor behaviour (to be	Equip vehicle with parametric simulation model of battery and use it in the loop for predicting impact of	• Energy • Response time to trigger

Component (model / tool)	**Functionality**	**Purpose**	**(Generic) KPIs addressed**
	executed in DynAA)	possible future routes.	
Charging infra model	Model of charging infrastructure, including current availability (to be executed in DynAA)	Used to optimize charging of the vehicle and to optimize the charging infrastructure itself	• Energy • Cost • Response time to trigger
MECA Driver support	(Self-)adaptation manager, handling functional adaptivity.	Provide user with advice on route and charging. Generate on request and pro-active advice on optimal route (based on predicted impact calculation)	• Cost • Energy • Response time to trigger
AOW optimizer	Find optimal solution in solution space	Determine optimal soltion for charing charging car of driver and possibly create functionality that could provide overall optimal solution for overall charing pole network (given that multiple drivers would issue requests)	• Cost • Response time to trigger
Preesm / Spider	Support for simulator signal processing	Used to implement simulator adaptive agent in a parallel, energy aware fashion.	• Response time to trigger • Energy
Verification tool	Requirements and properties verification	Formal methods for (i) requirement consistency verification; and (ii) property verification on formal models.	• Cost • Energy • Response time to trigger

2.5. Update on requirements

The demonstrator is still in line with the detailed requirements defined in D2.3. The most important deviation based on the recent work and analysis are:

- *TNO-010 (Automatically generate 3D map data):* this requirement is no longer seen as a CERBERO requirement. Although still required by CRF, realisation of this functionality does not part of CERBERO effort and is therefore removed.
- *New requirement = Optimisation of charging infrastructure*: apart from optimizing the charging of a specific car (e.g. which charging pole would be best to use), the charging infrastructure simulation could also take into account the optimisation of all electric vehicles. It is not clear yet to which extend this functionality can be included (in M36 demonstrator) but by adding optimisation of charging, the functionality could extended with overall optimisation using AOW.

3. Use Case Self-Healing System for Planetary Exploration

3.1. Introduction

In this section the latests insights and updates on the Self-Healing System for Planetary Exploration use case are described. The generalities of this scenario have already been covered in D2.3. In this document we intend to include new insights from the previous months on challenges, goals, architecture and incremental prototyping architecture/components.

Mapping among use case and CERBERO technologies and tools is also provided, which will be refined in D6.7 (demonstration skeletons) and in the demonstration-related deliverable (D6.8 due at M18).

3.2. Challenge and goals

The main challeges to be addressed in the Planetary Exploration use case are summarized hereafter:

- Failure detection and correction: Integrated Circuit malfunctions due to radiation effects are very common in space applications. The effects of subatomic radiation particles are frequently referred as “single event effects”. Particle radiation can cause flip-flops and memory cells to change state. These are errors that can change the functionality of the FPGA. Strategies to autonomously to detect and correct these errors, in order to avoid system malfunction may significantly improve currently adopted technologies.

- Environment adaptation and self-learning: Planetary exploration (e.g. Mars exploration) has harsh environmental conditions. Adaptability to the dynamic environment by adaptation, learning and evolution in robotic applications is mandatory to cope with such extreme situations.

- Power measurement and optimization: Rovers for Planetary Exploration are solar-powered. The energy is absorbed through solar arrays on panels that sit atop. Future missions for planetary investigation require high efficiency and low power consumption to be able to ensure the autonomy of the system. All the energy saved during computation can be used either for transmission or operation, but in any case advanced power optimization strategies will be of paramount importance.

Table 3-1 – Planetary Exploration General Needs

ID	Need	High Level Requirements
PE1	Minimization of energy consumption and costs, while keeping/improving resiliency.	Enable Dependable Hardware / Software (HW/SW) co-design.
PE2	Provide multi-objective design space exploration and multi-view analysis.	Develop integrated open-source or commercially available toolchain

ID	Need	High Level Requirements
	Reduce development time of complex heterogenous systems by increasing the level of abstraction. Increase quality and verification level to ensure proper operation of the system.	environment for the design and assessment of heterougenous cyber-physical systems.
PE3	Efficient support of architectural adaptivity, according to radiation effects and harsh environmental conditions.	Development of a HW/SW (self-)adaptation methodology and supporting tools.

Table 3-1 summarizes, for the Planetary Exploration use case, the general needs that have to be addressed using CERBERO technologies, in particular, with the bottom part of the CERBERO stack (as it becomes evident in Table 3-2 of Section 3.4). This use case is meant to demonstrate how CERBERO will facilitate the design of heterogeneous system, along with their runtime management. Runtime strategies for hardware and software adaptivity will be assessed, to answer to system and environmental changes in a self-adaptive manner.

Technically speaking, Planetary Exploration key challenges and goals will be addressed in CERBERO by supporting the following features:

1. **Failure detection and correction**: Due to single event effect errors, dynamic hardware-sofware reconfiguration will be provided, measuring fault tolerance improvements and redundant execution for critical task execution. Software and hardware runtime monitors information will be defined and implemented to automously provide this kind of support in combination with proper healing techniques.
2. **Environment adaptation and self learning**: The goal is to provide adaptation of the system to the harsh physical environment and self-awareness. The system will be able to combine functional, reconfigurable and real-time description of the application to ensure the proper operation and adaptation. Countinuous KPIs monitoring coupled with system adaptation strategies will allow to guarantee reactiveness and resiliency.
3. **Power measurement and optimization**: The system will be able to provide energy measurements, estimation from performance monitors and autonomous optimization in order to optimize (and minimize wherever possible) consumption. Software and hardware adaptation monitors information will be used to feed the adaptation manager, that will be used to make decisions on what to do to minimize the system power consumption, while guarantee the requested performance level. The adaptation engine then will implement reconfiguration.

3.3. System architecture and components

An overall view of the Self-Healing System for Planetary Exploration system components is presented in Figure 3-1. It consists of a robotic arm composed of two main parts: the Robot Control Unit (RCU) and the Servo Control Unit (SCU). These parts must perform Adaptive Motion Planning and Self-Healing functionalities. For this purpose and to guarantee the proper runtime operation of the system, under radiation and harsh environmental conditions, CERBERO consortium will make available technologies and tools, spanning from design-time formal verification to runtime monitoring and self-adaptation support for advanced model-based trade-off management (i.e. performance/resiliency vs. energy consumption).

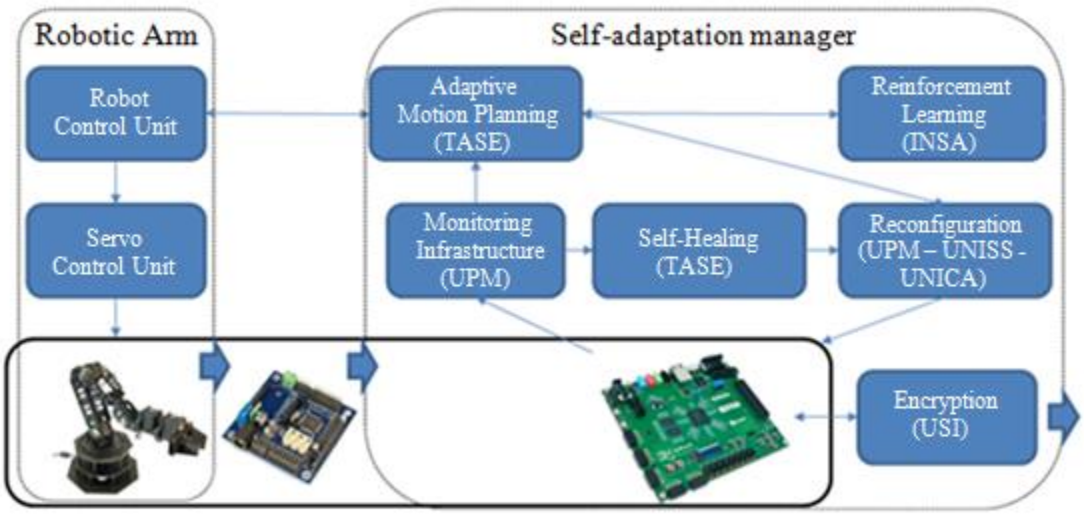

Figure 3-1 Self-Healing System for Planetary Exploration demonstration components

Physically, two units will be deployed:

- **The Robot Control Unit (RCU)**: This unit performs the most computation intensive high level tasks, like motion planning and interpolation. It sends commands to the physical system, which in this case is the robotic arm, and receives status telemetrys from it.
- **The Servo Control Unit (SCU)**: It drives a robot joint, in the WidowX Robot Arm we selected for demonstration porposes [TROSSENROBOTICS]. It is equipped with Dynamixel actuators. The SCU implements functions of position, velocity, current and torque control.

Functionally speaking, SCU and RCU will feature:

- **Adaptive motion planning:** The purpose is to provide arm motion planning in complex environments. The electronic on board of the robotic arm must be able to make decisions and adapt to complex environments and situations, in a completly autonoums manner.
- **Self-Healing:** This consist of ensuring the proper operation of the system, which must be able of detect and correct errors due to harsh environments. The heterogenous computing support will put in place monitoring and reconfiguration capabilities to meet this demand.

CERBERO technologies/components will be used to design and operate the described system. At design time, we will use:

- **Formal verification:** The VT tool by UNISS provides a way to prove the correct by contruction correctness of the system. It is based on formal methods of mathematics.
- **Monitoring Support:** PAPI will be used to properly intrument the software with the APIs to read software and hardware performance monitors, which will then be used at runtime to feed the models on the (self-)adaptation manager that will decide upon optimal (trade-off and situation aware) heterogenous system configuration.
- **Heterogenous system design:** PREESM, MDC and ARTICo3 will be used to customize the heterogenous computing infrastructure. PREESM will decide upon the prelimary core mapping, while the MDC tool and ARTICo3 will be used to deploy accelerator units with different flavours of hardware reconfiguration, which may be useful to address the different functional/non-functional trade-off categories.

At runtime, we will use:

- **Reinforcement Learning**: In the self-adaptation manager, reinforcement learning implemented using PREESM and Spider will be used to handle adaptation to physical environments and self-awareness.
- **Self-adaptation manager**: Different functional/non-functional (represented mainly by the CERBERO KPIs) trade-off categories will have to be served at runtime, i.e. to provide resiliency to errors (by implementing redundant execution) or to minimize consumption (at the expense of parallelism exploitation). The self-adaptation manager, according to information derived by proper monitors/sensors, will have to master system execution, putting in place adaptivity management (chosing what to do to meet the given KPI and minimize the costs of reconfiguration), by means of distributed engines (which can be software or hardware ones). For the planetary exploration case, we will face both functional and architectural adaptivity, since the heterogenous computing infrastructure is meant to be adaptable.

To provide and access Adaptive Motion Planning and Self Healing for a robotic planetary exploration, addressing the goals described in Section 3.2, two execution scenarios have been defined:

- **Robotic arm:** This scenario is focused on the RCU that will provide Adaptive motion planning of the robotic arm.
- **Brushless DC (BLDC) motor:** This scenario is focused on SCU that will drive a brushless motor, which is representative of space applications.

3.3.1. Robotic Arm

The Robotic Arm scenario is shown in Figure 3-2 . It will be in charge of generation of arm movement trajectories, validation of collision-free motion paths and self-healing.

- **PC** - The purpose of the PC is to provide serial communication with the RCU in order to send and provide debug options. This communication will be encrypted using CCSDS cryptographic algorithms, which are standards for Space Data System, in order to guarantee the confidentiality and integrity of the data in critical applications.
- **Development Board** - This board performs the RCU functionalities. The RCU receives commands from the PC and also send status information to it. This module

interprets and executes the commands, performs robot motion planning and interpolation and sends servo level commands to the SCU.

- **Robotic Arm**: WidowX robotic arm is equipped with some smart Dynamixel actuator joints. These actuators provide SCU that implements functions of Torque Limit, PID Gain, Moving Speed, Status, etc. Dynamixel actuators are not representative of space use case applications. For this reason, a BLDC motor scenario with its own SCU must be provided.

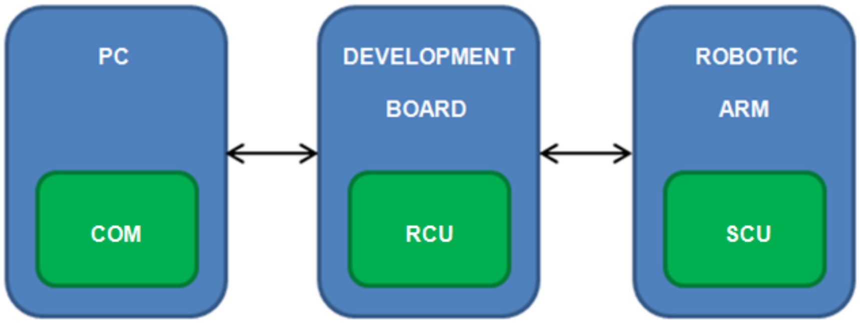

Figure 3-2 Robotic Arm Scenario

3.3.2. BLDC Motor

The BLDC Motor scenario is shown in Figure 3-3. Its goal is to control a space representative motor with self-healing characteristics.

- **CONTROL**: This module is in charge of generating the control signals that are fed to the driving stage. It performs, depending on the control strategy, the calculation of the duty cycle of these Pulse Width Modulation (PWM) signals in order to achieve the desired conditions of operation (speed, torque), according to the information of the current state of the motor. Also, the activation of these signals must commutate between one phase and another in a certain order (called phase commutation sequence) in order to vary the magnetic field induced by the stator windings and keep the rotor spinning.
- **DRIVER STAGE**: The control signals are fed to the driver stage module, which essentially consists of a MOSFET H-bridge capable of generating the driving signal for each one of the motor phases, and also receive the encoded position through the Hall-effect sensor signals.
- **BLDC MOTOR**: A three-phase BLDC motor, with Hall-effect sensors that provide the position of the rotor so the control module can perform the phase commutation in time. For torque regulation, also a measure of the current or the BEMF zero-crossing point for each phase is required.

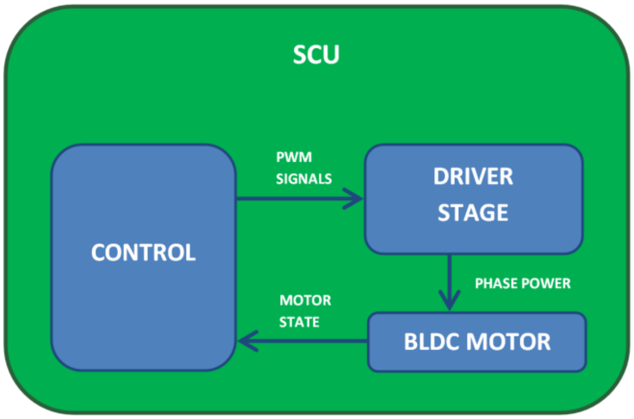

Figure 3-3 BLDC Motor Scenario

3.4. Use-Case vs. Technology mapping

In the Planetary Exploration use case, there are different models and tools that will be used to address needs and challenges identified in this context (see Table 3-1). Models, methodologies, architecture and tools are summarized in Table 3-2, describing their functionality, purpose in use case and KPIs addressed and/or optimized. Not all the following CERBERO technologies listed in the table are expected to be demonstrated at M18, the details on M18 demonstrators will be provided afterwards in D6.7 and D6.8.

Table 3-2 Planetary Exploration Use Case vs Technology mapping

Component (model / tool)	**Functionality**	**Purpose in use case**	**(Generic) KPIs addressed**
Preesm / Spider	Reinforcement learning implementation	Proof-of-concept of adapativity in terms of both adaptation to the physical environment and self-awareness.	• Energy • Latency • Safety
PiSDF model	Real time modelling of the application.	Combine a functional, reconfigurable and real-time description of the application with parallelism and energy awareness.	• Latency • Energy • Throughput
VT Tool	Requirements and properties verification	Formal methods for (i) requirement consistency verification; and (ii) property verification on formal models.	• Energy • Latency • Safety
Artico3 architecture	Architecture for dynamic HW adaptation, using DPR, with performance scalability (number of cores) and redundant execution	To show extended HW adaptation with performance/fault tolerance trade-offs	• Throughput • Energy • Fault masking in TMR mode • Fault recovery

Component (model / tool)	**Functionality**	**Purpose in use case**	**(Generic) KPIs addressed**
			time
Coarse-grained MDC-compliant adaptivity	Word-level coarse-grained hardware reconfiguration support	Proof of concept will be provided at M18 on how to implement trade-offs among KPIs.	• Power • Throughput • Latency
Papify	Common instrumentation interface for HW & SW elements	To demonstrate the capability of instrumenting, via PAPI, both HW & SW	• Energy • Throughput • Faults detected in Artico3 fabric
Custom PAPI-compatible components	Structured method for PAPI-compatible access to HW monitors	To provide unified access, via PAPI, to both HW and SW components	• Energy • Throughput • Faults detected/masked in Artico3 associated fabrics
Performance monitors	HW modules for performance monitoring of HW accelerators. Compatible with both MDC accelerators and ARTICo3 architecture.	Instrumentation of HW accelerators	• Throughput • Latency
ARTICo3 fault monitors	Fault diagnosis monitors for redundant HW execution modes	Measuring fault tolerance improvements when in redundant execution mode for critical HW task execution	• Fault detection time • Fault recovery time
Dedicated library of components to support data redundancy	Hardware modules for on-the-fly error detection and correcting of memory data.	Guarantee the integrity of data while maintaining the performance and energy consumption limited	• Reliability (integrity of the data)
Energy consumption monitors	Various modes of power/energy measurement in SW (PS) and HW (PL) subcircuits in Zynq devices.	Provide energy measurements to optimise energy consumption by autonomous adaptation	• Power
Energy consumption models	To estimate the energy consumption from performance monitors	Model validation by correlation between models and actual measurements	• Energy estimation
Papify-viewer	Visualization tool for papify values	User-centric tool for performance analysis	• Energy • Throughput • Faults detected in Artico3 fabric
Hardware	Compose hardware	Capability of runtime	• Latency

Component (model / tool)	**Functionality**	**Purpose in use case**	**(Generic) KPIs addressed**
Composition Tool	accelerators by mapping different predefined processing elements.	generating new HW functions by tiny blocks composition. It includes both static mapping from an intermediate format and dynamic mapping by goal functions.	• Energy • Throughput
Runtime Adaptive Hardware/Software Task Manager	Runtime Mapping and Scheduling of Tasks (Actors) on the Heterogeneous Platform	To move dynamically actors on the available CPUs or on the available portions of Reconfigurable Logic in order to achieve different goals (combinations of KPI)	• Throughput • Energy • Self-Healing • Latency
Self-adaptation manager	Manager for various reconfiguration types (ARTICo3/MDC/SW-like)	To show how dynamic adaptation in heterogeneous systems is achieved under the provided reconfiguration types	• Throughput • Energy • Self-Healing • Latency

3.5. Update on requirements

The Planetary Exploration demonstrator is still in line with the requirements defined in D2.3. Only the following changes must be taken into account:

- TASE-002 (Robotic Control Unit architecture): this requirement is divided into two requirements. TASE-002 (1) will be based on ZedBoard Zynq-7000 in order to implement HW & SW solutions by using the integrated MPSoC as a first approarch. TASE-002 (2) will be based on ZCU 102 Zynq UltraScale+, which Xilinx plans to develop a rad-tolerant version.
- TASE-012 (Servo Control Unit architecture): this requirement is divided into two requirements. TASE-012 (1) will be based on Zybo Zynq-7000 7000 in order to implement HW & SW solutions by using the integrated MPSoC as a first approarch. TASE-012 (2) will be based on ZCU 102 Zynq UltraScale+, which Xilinx plans to develop a rad-tolerant version.

4. Use Case Ocean Monitoring

4.1. Introduction

In this chapter the latests insights and updates on the Ocean Monitoring use case are described. The use case itself is already described in D2.3. This chapter includes new insights from the previous months on challenges, goals and architecture. The testing environment will be described in D6.7, while implementation details and assessment results are going to be detailed in D6.9. This use case differs slightly from others in that, the focus is more on incremental prototyping [Goker 2008] with higher exposure to real environment.

Ocean monitoring as a field in general can encompass a range of purposes from observing and tracking marine ecology, and climate to subsea maintenance of equipment. These can be enabled via satellites, and onsea and subsea vehicles that may be manned, unmanned or hybrid through local or remote control. The Ocean Monitoring use case, in this project, comprises of smart video-sensing unmanned vehicles with immersive environmental monitoring capabilities. As also referred to in the first deliverable, they serve as *marine eyeballs* that can capture live videos and images of the local on-sea and subsea surroundings. The marine robots will be remote controlled within wireless reach and visible sight, but also designed for self-operation and navigation. The vehicles will perform smart sensing and processing capabilities for several purposes including navigation and visual sensing.

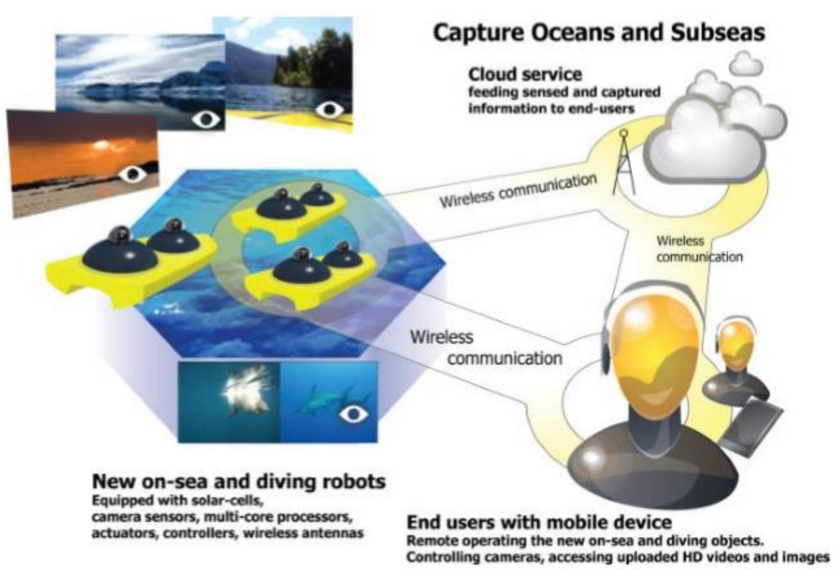

Figure 4-1 Ocean Monitoring – Overall system architecture and components

The overall system architecture, introduced originally in D2.3, is still valid and provides a summary of the key cornerstones for our Ocean Monitoring use cases. This is also shown in Figure 4-1. In short, there are surface and subsea vehicles that contain or carry sensors to help them *sense* the environment. These communicate with local or remote users. The marine robots and sensors as well as the users have the ability to download/upload with cloud services. Sensed data can be stored locally or streamed to a cloud service from where relevant information can be retrieved. The marine robot can perform on-the-fly data analysis and data fusion in order to make decisions and adapt to changing environment.

4.2. Challenge and goals

The goal is to research and develop CPS-prototypes for the Ocean Monitoring scenarios, towards commercial products and for demonstration purposes. The challenge, in this multidisciplinary field, is in considering the requirements and implementation aspects when balancing the order and development of the functionalities. We consider the Ocean Monitoring scenarios to consist essentially of three areas of work in the CERBERO project. These areas, and their challenges, are described hereafter.

- *Marine robot* (physical / mechanical parts) – the cost of deploying subsea marine robots is currently very high. This is generally due to the cost of running the supply ships to deploy the marine robot and the marine robot size and weight itself. Most of the current approaches to designing and deploying marine robots rely on legacy approaches of using supply ships to go offshore or to submerge robots and parts from the coast. We consider it timely to investigate alternative methods for moving and deploying marine robots for small to medium size robots to improve their affordability. However, small to medium size robots arise additional requirements specified below.
- *Enhanced vision and sensing capabilities* – current vision and sensing technologies for ocean monitoring are overwhelmingly based on default software on existing off-the-shelf cameras (e.g. GoPro©), and other sensors. These have generally been developed for use in other fields such as security, and social internet applications. They are short on meeting specific vision and sensing challenges that occur on sea surface, but particularly subsea. Their use in oceans seems to generally be perceived as a niche area and as yet is not fully developed. Current approaches to monitoring, assessing and decision-making around visual ocean data tend to revolve around human operators manually inspecting hours of videos in what is also less clear visibility. Real-time monitoring is generally focused on video data availability on a supply ship. Both of these are expensive processes in terms of time, effort, and cost and hence would benefit from semi-automated approaches and use of COTS components.
- *Autonomous driving and human control* – there are three main issues here: the environment, wireless communication, availability of technologies. The ocean environment has both extreme weather and onsea/subsea water movements. Wireless communication is an issue. Autonomous shipping benefits from satellite communication, but wireless communication subsea is a challenge, because wireless signals are greatly attenuated by water [Zhang et. al, 2015]. The state-of-art in this area is low frequency acoustic communication that can be used for remote controlling, but not for video transmission. The technologies available for ocean monitoring are few, in comparison to road and air, and are less standardised. This is mainly a reflection of the time and effort spent on road and air vehicles, as

they have been more visible and usable transport by humans. Current approaches are focused around military applications for underwater gliders and detection of submarines, submerged vehicles, and ships on the sea horizon. Autonomous driving for civilian use has had less focus in comparison. Hence, low-cost and flexible solutions are less readily available for maritime sectors and ocean monitoring. We will address this through using standardised wireless communication for the auto-pilot and remote control components and consider civil use cases.

Table 4-1 Ocean Monitoring General Needs

ID	Need	High Level Requirements
OM1	Faster development cycles and cost reductions due to early-stage system characterization.	Provide complete design cycle from system level design to mapping over COTS SW and HW components.
OM2	Provide multi-objective design space exploration and multi-view analysis at the system level, facilitating development cycles and reducing time to market. Increase reuse among cycles, along with quality and verification level by fast prototyping from high level of abstraction directly to working real time applications.	Develop integrated open-source or commercially available toolchain environment for cyber-physical systems, with focus on fast prototyping thanks to high-level system characterization. The open toolchain environment contains open source or commercially available tools and integration framework.
OM3	Efficient support of functional adaptivity, according to system, human and environment triggers.	Development of a SW (Self-) Adaptivity methodology and supporting tools.

Table 4-1 summarizes the needs posed by the Ocean Monitoring use case to the CERBERO project. The upper part of the CERBERO stack (as it becomes evident in Table 4-3 of Section 4.4) will be primarily used. The CERBERO project offers several technological possibilities and ways to progress beyond the current state of the art in ocean monitoring using marine robots, addressing the abovementioned challenges and goals. A marine robot generally comprises several sub-systems/components, i.e. a multitude of sensors, actuators, controllers, and power units, that need to function independently, but also must interoperate and communicate with each other using several wireless and wired network communication protocols for the human operator to be able to operate it. CERBERO will enable these robots to be able to effectively support functional adaptivity to react to system, human and environmental triggers. More details on how CERBERO will help in addressing the identified goals are provided hereafter.

1. **Marine robot: Interoperability of CPS components through model-based development** - The overall technological challenge is in the integration of the sub-systems (components) to achieve interoperability. Although a marine robot can be seen as a CPS system in itself, each sub-component of the marine robot can also be a potential CPS-system in itself too with different adaptive behaviours. This integration and interoperability can only happen at a software level, and therefore, the CERBERO software framework with its new model-based development approach for adaptive CPS development are important

considerations when it comes to interfacing components in this use case. The goal is to have efficient and light-weight yet robust marine robots for on sea and subsea monitoring. Information structures will be represented and modelled to: 1) interface between components, 2) to represent time and context, and 3) to enable adaptive behaviour of CPS components.

2. **Enhanced vision and sensing capabilities: better situational awareness through adaptive sensing** - The second challenge is in providing adaptive vision and sensing capabilities to the human operator for remote sensing and monitoring of the seas. This challenge requires new adaptive image processing methods for enhancing the captured imagery, along with object/motion detection. Datafeed from multiple sensors need to be adapted and fused for better vision, then the data must be indexed, and stored for search/ retrieval, and quickly communicated via wireless communication to the human operator or cloud service. This will be addressed by using adaptive image enhanement methods in combination with the CERBERO model-based approach for representing, indexing, storing and searching of sensed data. The adaptive image enhancement and information fusion methods are used to enhance the situational awareness. State-of-the-art video compression methods are also analysed and tested. The goal is to have adaptive and immersive visual sensing capabilities that have been specifically developed for ocean monitoring use cases. Adaptive sensing capabilities will be researched in the form of: 1) a physical adaptive camera prototype, and 2) simulation of system configuratons with multiple cameras with a combination of DynAA and/or AOW, which in the Ocean Monitoring case are meant to be used at design time before the system prototyping, depending on the simulated/ optimised aspects, and 3) hybrid image and video retrieval models for smart / adaptive cameras.
3. **Autonomous driving and human control -** The goal is to have flexible and affordable hybrid solutions for autonomous driving and human-controlled vehicles in ocean monitoring. Several components will be integrated to support autonomous driving and human control. The marine robot’s CPS components for driving, include: engine, battery system, navigational sensors, and propulsion and steering software. In terms of the battery system, it is important to be able to predict how long the battery will last given the currenty energy usage and chosen navigation. The marine operator will be able to control the robot from remote or engage the autopilot. State of the art, standardised encryption will be used for secure wireless communication, as per the use case requirements. The autopilot will use GPS and navigational sensors to detect objects, for example.

4.3. System architecture and components

An overall view of the Ocean Monitoring architecture is presented in Figure 4-2. It consists of the marine robot (with its mechanical and physical parts), its movement and steering capability (with auto-pilot and remote control) the interfaces (both physical/mechanical and user interaction interfaces) and, illustrated at the top of the diagram, a series of components to navigate, move, sense, and store information. These components are: navigational sensors; battery system; engine; camera system; and information storage. The components related to visual sensing, such as the camera system, and operation of the robot is developed by AS in the CERBERO project as part of demonstration activities in WP6. In order to move, sense and observe the seas more effectively, various optimisations and simulation techniques need to be used. We differentiate between design-time and runtime. Both DynAA and AOW components are being used for design-time support. Additionally, DynAA will also be used at runtime. Figure 4-2 shows runtime aspects. DynAA has been developed by TNO and AOW by IBM, both are being adapted within the CERBERO project to fit the use cases. In Ocean Monitoring they will be trialed together with AS, for a representative range of tasks.

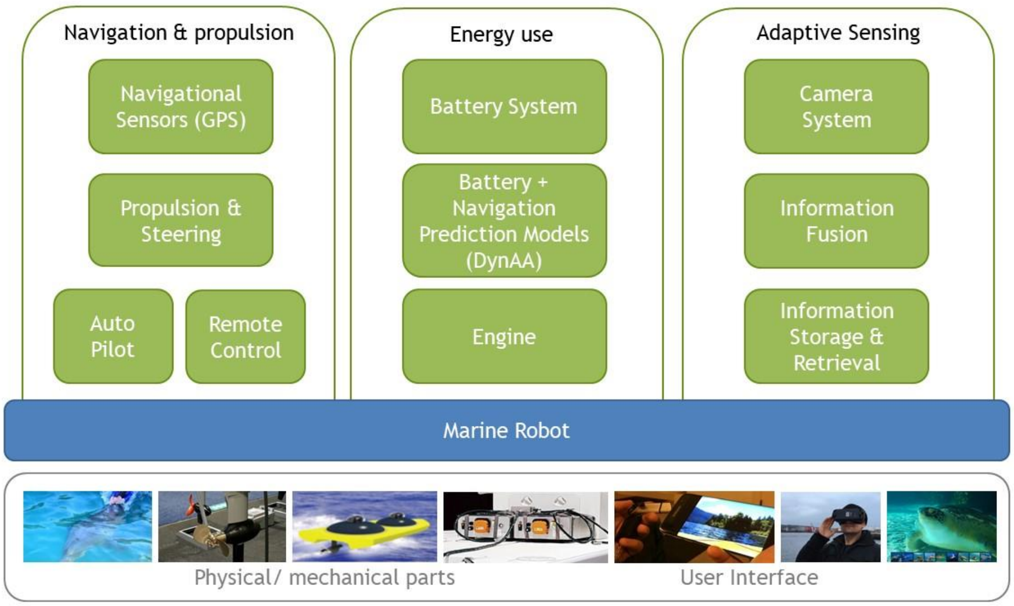

Figure 4-2 Ocean Monitoring System Components

Below are the descriptions of the Ocean Monitoring components illustrated in the figure (Figure 4-2). All components are being integrated specifically for this use case.

<u>Navigation & Propulsion</u>:

- **Navigational sensors** - These are sensors that help the robot to navigate in the seas in order to make observations on surface (on sea) and subsea.
- **Propulsion and steering -** This is software that integrates and communicates with micro-controllers that control the propulsion and steering actuators.
- **Auto-pilot -** This is GPS based and is needed in conditions to minimise risk, pre-plan trips and missions, or increase safety.
- **Remote control -** A human operator / user can control the marine robot and/or its sensors and equipment from remote.

<u>Energy Use:</u>

- **Battery system** - This includes both the physical batteries, their alternative connectivities and the management of their energy and use.

- **Modeling and Prediction (DynAA) – Battery and Navigation Prediction Models -** This component can be used to model, simulate, and predict, the performance of other components and algorithms. The DynAA simulation is developed by TNO. DynAA will be used for assessing how much power is left and to predict how much power is needed for navigation, propulsion, and camera use (i.e. the power demands).
- **Engine -** This consist of off-the-shelf components integrated to suit Cerbero project Ocean Monitoring needs.

Adaptive sensing:

- **Camera system -** This comprises of two or more camera sensors to monitor the local environment around the vehicle or within its reach.
- **Information Fusion –** This is to fuse images from the camera system and any additional sensor information to enable adaptive sensing of the surroundings.
- **Information storage and retrieval**- This is the information storage units along with the indexing and retrieval algorithms to access the information. It consists of information storage units (Terabytes) and high speed data buses for storing image and video streams in parallel when necessary. These can then be further processed and analysed for categorisations and decision making.
- **Marine robot** (physical / mechanical parts): A prototype with some key physical and mechanical components will be developed. Some components will be trialed and tested off-the-shelf components. Others will be developed to help showcase aspects of CERBERO.

The overall system architecture will be implemented using a set of system components that instantiate the reference architecture, and which include enabling data flow components to be black-boxed within the overall architecture. This provides a practical basis for the iterative development approach within this use case.

Needed software stack and hardware platform with firmware

There are several components that need to be integrated and communicate both internally in a marine robot and externally with human operator and the cloud. Wireless mobile communication will be used to extend the range and possible area of the vehicle. Also internal network in the robot is needed. Due to the number of components to integrate with in terms of sensors, actuators, power, and computing units, the chosen hardware platform is inevitably multicore based, and hybrid, where three state of the art hardware platform architecture families are recommended. These are the: Snapdragon reference architecture (i.e. Snapdragon 835/ Snapdragon 845), the Intel i7 reference architecture with hyper threading, and the Nvidia Jetson TX1/TX2. All are 64-bit processors and thus allow for much for more RAM, and sensor and information processing. The Snapdragon processor family is energy efficient with very good support for wireless communication. The needed firmware must be based on Linux operating system with a Java Virtual machine running on top. Thus, for the Snapdragon 835 and 845 reference architectures the firmware needs of the use case translate into Android Nougat 7.0 and beyond as software stack, whereas for the Intel i7 reference architecture to the Ubuntu 16.04 LTS version, again with Java virtual machine on top. As the table below illustrates, the

Snapdragon architectures are more suitable for sensor integration and wireless communication, where the Intel i7 architecture is more specialised for towards data processing and server applications, including cloud services. The Snapdragon architectures are efficient low energy processors, whereas Intel i7 architectures consume significantly more. The Nvidia Jetson modules are also designed for embedded use, with a relatively more powerful GPU and extensive software support for efficient and performant video handling.

Table 4-2 Needed Software stack and Hardware Platform with Firmware

Platform	**Snapdragon 835 / 845**	**Intel i7 Kaby Lake**	**Nvidia Jetson TX1 / TX2**
OS	Android 7.0	Ubuntu 16.04LTS/Android 7.0	Ubuntu 16.04LTS
CPU	ARM (8 cores, 64-bit)	Intel (8 cores/2 threads, 64-bit)	ARM (4 cores, 64-bit)
GPU	Adreno	Intel	Nvidia Pascal/Maxwell (256 cores)
Vision API	FastCV	OpenCV	VisionWorks (CUDA)
Video	H.264, HEVC, V9	H.264, HEVC, V9	H.264, HEVC, V9
Camera	Dual image sensor processors	Dual image sensor processors	6 image sensor processors
Wireless	Gigabit LTE, Wifi, Bluetooth5, NFC	None	Wifi, Bluetooth4
Location	GPS, etc. (6 satellite systems)	None	None
Battery	Quick charge chipset	None	None
Security	Accelerated AES/SHA	Accelerated AES/SHA	Accelerated AES/SHA
Wire	USB-3, USB-C, CSI-2, HDMI via USB	Gigabit Ethernet, USB-3, PCIe 3.0, HDMI	Gigabit Ethernet, USB-3, PCIe 3.0, CSI-2, HDMI, CAN (on TX2 only)

4.3.1. Sensors and Information Fusion

Sensors and information fusion are key aspects within Ocean Monitoring. -The main components for sensors and information fusion in the Ocean Monitoring case are:

- **Sensors** - such as camera, GPS, ultrasound, laser, sense the local environment and provide the relevant data.
- **Information Fusion Models** – are needed to combine the information coming from the sensors in the most optimal way in order to make decisions, improve image quality, retrieve relevant data, etc.

- **Compression** – needed to reduce the size of data which have to be sent to the user's receiving device like laptop, smartphone, and tablet. Different compression algorithms and different compression ratios can be considered.
- **Object Detection** and **Image Enhancement Models** – are the techniques to enhance the underwater image to alleviate the poor visibility conditions and detect and track moving objects for the enhanced situational awareness of the user. The image enhancement can be based on edge detection and noise removal for example, and thus improve and enable object detection.

The key performance indicators (KPIs) that will access system performance in this context are Response-Time, Throughput, Power, and Image Quality (Ranked Feature) . First, different type and number of active sensors require different power levels. Second, different fusion and compression algorithms also require different power levels and need different times to complete. However, the image quality is also important during any adjustments.

With respect to CERBERO technologies**, DynAA Simulation Model** and **AOW Optimizer** will be used to model and simulate configurations in terms of different sensors activated, compression algorithms, and fusion models. They can be used for assisting the selection architectures for data throughput at design-time.

4.3.2. Adaptive Camera System – design-time and physical prototype

Sensing, observing, gathering, fusing, and analyzing information is highly important in ocean monitoring systems. We consider visual sensing in particular to be key in this. The updated use case prioritization reflects this aspect. In that regard, below in this section, we illustrate (Figure 4-3, Figure 4-4) a relevant use case relating to adaptive cameras, its system components and technologies developed and used within the project. The figures illustrate both a simulation (Figure 4-3) and a physical prototype (Figure 4-4).

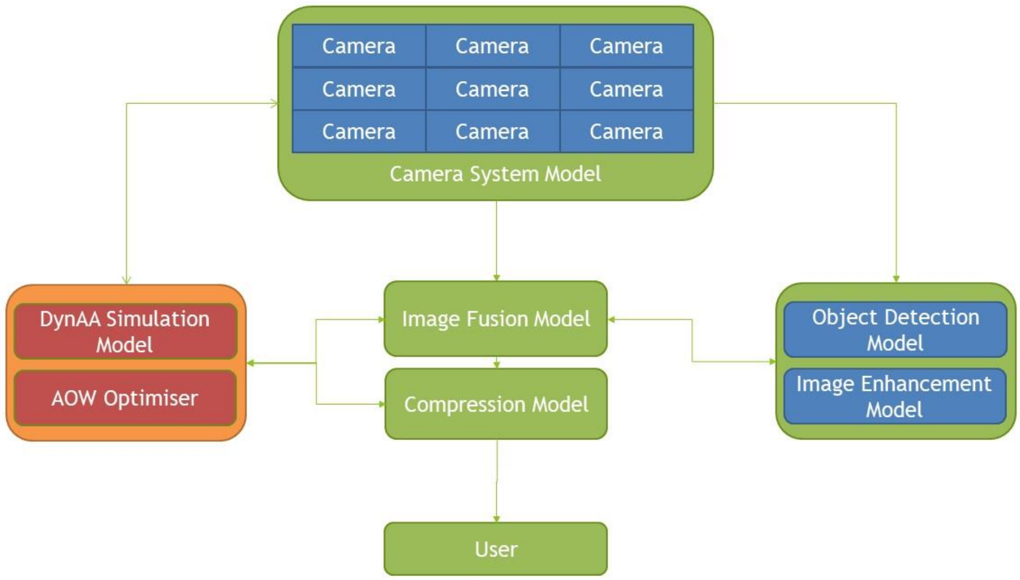

Figure 4-3 Adaptive Camera System – Design-time model in Ocean Monitoring

The main models in the Adaptive Camera System for simulation/optimization purposes are:

- *Camera System Model* – a system of cameras/lenses arranged into a grid, providing images of the same environmental area from different perspectives. All cameras or just a subset of them can be active at a time. The more cameras active, the better the image quality but higher the energy cost.
- *Image Fusion Model* – used to fuse images after their registration, in order to remove the noise or create higher resolution images.
- *Image Compression Model* – needed to reduce the size of images which have to be sent to the user's receiving device like laptop, smartphone, and tablet. Different compression algorithms and different compression ratios can be considered which are typically H.264 (AVC), H.265 (HEVC), VP9, JPEG, and PNG.
- *Object Detection* and *Image Enhancement Models* – are the techniques to enhance the underwater image to alleviate the poor visibility conditions and detect and track moving objects for the enhanced situational awareness of the user. The image enhancement can be based on edge detection and noise removal for example, and thus improve and enable object detection. The number of active cameras and thus the image quality can affect both the object detection and the image enhancement methods.

The key performance indicators (KPIs) that will access system performance in this context Image Quality (Ranked Feature), Response Time, Throughput, Energy, and

Power. First, different number of active cameras requires different energy levels and will produce images of different quality. There is a trade-off between the energy consumption, time, and image quality. The more cameras active, the better the image quality but the higher the energy cost and time constraint. Second, different compression algorithms and compression ratios will also result in different energy consumption, image quality (if a lossy compression is used), and different times required to compress.

With respect to CERBERO technologies, DynAA and AOW will be used to simulate different camera configurations in terms of different number of cameras activated and different compression algorithms and compression ratios. The results from different DynAA simulations will be fed into the AOW Optimizer in order to find the optimal configurations with respect to the KPI. The Ocean Monitoring use of the aforementioned CERBERO tools involves design-time adaptation of the camera system to the demands of the marine robot and the robot's operator, based on DynAA and AOW, and runtime adaptation for energy usage and navigation using DynAA with case-specific adaptation components. DynAA is developed by TNO, AOW is developed by IBM, together with AS they will work these aspects of the case.

The components of the physical prototype of the adaptive camera system in Figure 4-4 are:

- *Camera sensor* – an HD camera for capturing the imagery.
- *Laser* (optional) – for the distance measurement which may be needed in visibility estimation.
- *Illumination sensor* (optional) – sense the level of illumination which may be needed in visibility estimation.
- *Visibility estimation* – estimation of relative or absolute visibility conditions based on: blurriness levels of an image, presence of noise, illumination level, distance from an object and object's characteristics. Distance from an object for the absolute visibility can be estimated from the two cameras or the laser measurements, for instance. The relative visibility is a deviation of the current conditions and the ideal subjective visibility - no clouds, noise, good illumination. Relative visibility would be based on characteristics such as luminance, contrast, object clarity (quality of edges, blurriness, etc.).
- *Image fusion* – fusion of registered images from two cameras for better quality image. There is also an adaptive fusion of an edge image with the original one where the influence of each corresponds to different visibility levels, thus de-blurring and de-noising the image in an adaptive manner.
- *Adaptive image enhancement* – where the techniques such as illumination correction are applied to the image in an adaptive manner, affecting the original image to a different degree depending on the visibility conditions. More specifically, adaptive image enhancement includes techniques for deblurring, histogram equalization, contrast enhancement, illumination correction, edge detection and adaptive fusion.

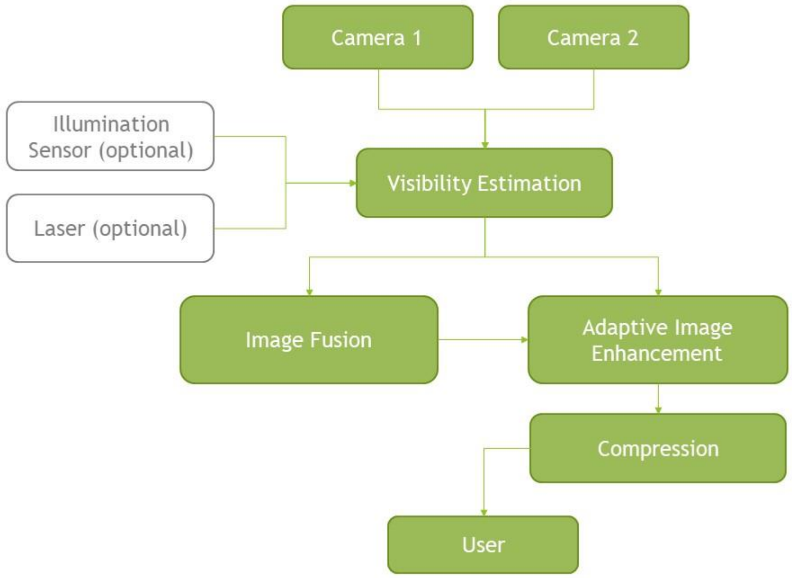

Figure 4-4 Adaptive Camera System – Physical Prototype in Ocean Monitoring

Below is a list of the existing and new algorithms that will be used mainly in the context of computer vision and the adaptive camera system:

- Canny edge detector (Canny, 1986): For image enhancement - overlay of edges on the original image. It can be also used in the movement detection and tracking algorithms.

  http://www.intelligence.tuc.gr/~petrakis/courses/computervision/canny.pdf
- Frame difference movement detection (background subtraction) (Godbehere et. al., 2012), (KaewTraKulPong, and Bowden, 2002): For movement detection and object tracking.

  https://docs.opencv.org/3.3.0/db/d5c/tutorial_py_bg_subtraction.html
- Colour based movement detection (Gevers, and Smeulders, 1997), (Khan et. Al, 2012): For movement detection and object tracking.

  http://ieeexplore.ieee.org/document/1341216/
- Superresolution techniques (e.g. averaging) (Boyce, 1992): For enhancement of images taken from multiple cameras, mainly the average image algorithm for noise reduction

  http://aishack.in/tutorials/noise-reduction-averaging-theory/

  http://www.infognition.com/articles/what_is_super_resolution.html
- Bilateral filter (Tomasi, and Manduchi, 1998), (Rompelman, and Ros, 1986): For smoothing an image, de-noising while preserving the edges.

  https://people.csail.mit.edu/sparis/bf_course/course_notes.pdf

- New adaptive model for automatic compensation of the loss of light at different depths: For automatic compensation of the loss of light at different depths starting with the red colour. Part of adaptive image enhancement.
- New information fusion model for combination of frame difference and colour based movement detection: For the fusion of the background subtraction movement detection algorithm and the colour-based movement detection in order to utilise the strengths of both.
- New edge detector: Based on the application of bilateral filtering for edge preservation and image derivatives computed at eight different orientations.
- New image enhancement algorithm: Based on the fusion of original image and the image processed by the new edge detector. Part of adaptive image enhancement.
- New hybrid image retrieval algorithms: For retrieval of images from the database based on the combination of visual features and text.

  http://www.iiis.org/CDs2017/CD2017Summer/papers/SA542OS.pdf

4.3.3. Battery system and prediction modeling

Battery lifetime prediction and modeling are very important in ensuring the marine robot can continue its sensing mission of the seas as efficiently, effectively and as long as possible – whilst not missing any key objects, wildlife, and environments. The modeling is for predicting how much battery power is left and adapting other parts of the system to this information – for example, when to return from trip / mission. In other words, there is an interplay of the battery power left and the navigation:

- *Battery System* – This consists of the lithium battery cells arranged in series or parallel. The different battery topologies provide different voltages and different electric storage capacities.
- *Battery and Navigation Prediction Models (DynAA)* – This is about predicting and assessing how long the battery will last during mission so that marine robot components can adapt accordingly. The relevant key performance indicators for this part are Power and Energy.

4.4. Use-Case vs. Technology mapping

In the Ocean Monitoring use case a number of different components and technologies are used to accomplish the functionalities listed below. In the table below, a summary is provided of the CERBERO components used, their purpose in the use case, and the specific KPI (Key Performance Indicator) they address and/or optimize.

Table 4-3 Ocean monitoring components, tools, their purpose in the use case and the specific KPI they address and/or optimize

Component (model / tool)	**Functionality**	**Purpose in use case**	**(Generic) KPIs addressed.**

Component (model / tool)	**Functionality**	**Purpose in use case**	**(Generic) KPIs addressed.**
Camera System Model	System simulation	A model of a system of cameras/lenses arranged into a grid, providing images of the same environmental area from different perspectives. This Model is for design time. The Camera Model uses DynAA and AOW.	• Response time • Image quality (Ranked feature) • Throughput
Battery & Navigation Prediction Model	Prediction of remaining battery time.	Create model for predicting and assessing how long the battery will last during mission so that marine robot components can adapt accordingly. This prediction model uses DynAA.	• Energy • Power
Object detection & Image Enhancement Model	Automatic object detection techniques	Enhance the underwater image to alleviate the poor visibility conditions and detect and track moving objects for the enhanced situational awareness of the user.	• Image quality (Ranked feature)
Adaptive camera system	Adaptive sensing	Prototype of adaptive camera to adapt to underwater visibility conditions. It will use models of object detection and image enhancement to refine and adapt the image analysis process.	• Power • Response time • Image quality (Ranked feature)
Information fusion	Combining different types of information	Needed to enhance images and videos, make decisions based on different information types.	• Image quality (Ranked feature)
Compression	Compress the data	Needed to reduce the size of images which will be sent to the user's device (laptop, smartphone, tablet, etc.). There will be switch between different compression algorithms offering different features.	• Image quality • Response time
Reference architecture implementation	Implement / instantiate the reference architecture	To provide a well defined framework for integrating data-flow black-box components (networked black-box) within the overall architecture.	• Response time • Cost

4.5. *Update on requirements*

Overall, the Ocean Monitoring demonstrator is still in line with the detailed requirements defined in first deliverable (D2.3). In the first deliverable, a description of the scenarios, use cases and detailed list of the requirements was provided. Here, in the two tables below, we refer to the scenarios and their derived use cases in summary along with the category of requirements they map on to.

Table 4-4 Ocean monitoring scenario, use cases and requirement group

Overall Scenario	Derived Use Cases	Requirements Group Code
Observing wildlife	Enabling adaptive camera system for parallel delivery of videos and images	R2.2
	Ocean monitoring – surface	R1.1
Subsea monitoring	Eye in the water	R1.2
	Searching for missing vehicles	R3.1
	Storing data streams from multiple sensors	R3.2
Marine robot propulsion and transport	Reconfiguration of battery module in runtime	R2.1
	Control of thrusters and steering through software	R1.3
	Set the course and forget – autopilot	R2.3
	Remote control of marine robot	R1.4

Table 4-5 Ocean monitoring requiremens groups and their priorities

Requirements Group Code	Name of Requirements Group	Old Priority	New Priority
R1.1	Requirements for ocean monitoring - surface	Priority 1	Priority 2
R1.2	Requirements for ocean monitoring - subsea	Priority 1	Priority 1
R1.3	Requirements for control of thrusters and steering through software	Priority 1	Priority 2
R1.4	Requirements for remote control of marine robot	Priority 1	Priority 2
R2.1	Requirements for reconfiguration of battery module in runtime	Priority 2	Priority 2

Requirements Group Code	**Name of Requirements Group**	**Old Priority**	**New Priority**
R2.2	Requirements for adaptive camera system	Priority 2	Priority 1
R2.3	Requirements for autopilot – navigation	Priority 2	Priority 2
R3.1	Requirements for searching for missing vehicles	Priority 3	Priority 3
R3.2	Requirements for storing data streams from multiple sensors	Priority 3	Priority 3

The main update with respect to requirements and their priorities is in relation to surface/subsea requirements and visual sensing or adaptive camera system.

In the course of working on the plan of prototypes it became evident that there were more challenges for subsea aspects of the use cases, and arguably less technologies available, for these than the sea surface use cases. Also, some techniques and tools used for subsea could also be deployed on the sea surface. To minimise risk and give more flexibility in work, we prioritised the subsea over sea surface requirements. Similarily, to minimise risk and allow for more time to meet the challenges, we upgraded the priority of visual sensing or adaptive camera sytems related requirements. Another, secondary factor, relates to the availability and accessibility of other CERBERO tools and documentation within the current time-frame for prototyping. These were also considered in reassessing the requirements with respect to possible timescales for implementation and testing.

5. Conclusions

In this document, all the CERBERO assessment scenarios, their challenges and goals, architecture and components, testing environments planned within them, technology mapping for each use case and update on requirements with previous deliverables.

This document provides a high level description, which will be elaborated in more detail in future deliverables. Updates will take place in M19 and M25. In this way we will guarantee an effective industry-driven deployment of the CERBERO framework, which will be updated or adjust to serve different or changed scenario requirements.

This document serve as a basis for the Technical Requirements Elicitation, which is part of D2.7, and for the definition of the Demonstrators skeletons, which will be part of D6.7.

6. References

[CERBERO 2017]	http://www.cerbero-h2020.eu
[SCANeR Studio]	http://www.oktal.fr/en/automotive/range-of-simulators/software
[DynAA]	http://youtu.be/ZP6q9J5wX4k (short introduction movie)
[CIRCUS]	http://www.esa.int/Our_Activities/Space_Engineering_Technology/Automation_and_Robotics/CIRCUS
[TROSSENROBOTICS]	http://www.trossenrobotics.com/widowxrobotarm
[Canny 1986]	J. Canny, A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679-714, 1986.
[Tomasi 1998]	C. Tomasi and R. Manduchi, Bilateral filtering for gray and color images. *Sixth International Conference on Computer Vision*, Bombay, pp. 839-846, 1998.
[Godbehere 2012]	A. B. Godbehere, A. Matsukawa and K. Goldberg, Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. *American Control Conference (ACC)*, Montreal, QC, pp. 4305-4312, 2012.
[KaewTraKulPong 2002]	P. KaewTraKulPong, R. Bowden, An improved adaptive background mixture model for real-time tracking with shadow detection. In: Remagnino P., Jones G.A., Paragios N., Regazzoni C.S. (eds) *Video-Based Surveillance Systems*, Springer, Boston, MA, 2002.
[Gevers 1997]	T. Gevers, A.W.M. Smeulders, Color-based object recognition. In: Del Bimbo A. (eds) Image Analysis and Processing, *Lecture Notes in Computer Science*, vol 1310, Springer, Berlin, Heidelberg, 1997.
[Khan 2012]	F. S. Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, M. Vanrell and A. M. Lopez, Color attributes for object detection. *IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, pp. 3306-3313, 2012.
[Boyce 1992]	J. M. Boyce, Noise reduction of image sequences using adaptive motion compensated frame averaging. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, San Francisco, CA, pp. 461-464 vol.3, 1992.
[Rompelman 1986]	O. Rompelman, H., H. Ros, Coherent averaging technique: A tutorial review Part 1: Noise reduction and the equivalent filter. *Journal of Biomedical Engineering*, vol 8, issue 1, pp. 24-29, 1986.
[Goker 2008]	Goker and H. Myrhaug Evaluation of a mobile information system in context. *Journal of Information Processing & Management*, ISSN: 0306-4573, Vol: 44, Issue: 1, Page: 39-65, 2008
[Zhang 2015]	X. Zhang, J. H. Cui, S. Das, M. Gerla and M. Chitre, "Underwater wireless communications and networks: theory and application:

Part 1 [Guest Editorial]," in *IEEE Communications Magazine*, vol. 53, no. 11, pp. 40-41, November 2015.