Ref. Ares(2017)4928436 - 09/10/2017

Information and Communication Technologies (ICT) Programme

Project N^{o}: H2020-ICT-2016-1-732105

D2.3: CERBERO Scenarios Description

Lead Beneficiary:	TNO/AMBIESENSE/TASE
Workpackage:	WP2
Date:	14-09-2017
Distribution - Confidentiality:	Public

Abstract:

This report is meant to provide a description of the CERBERO scenarios: Smart Travelling, Self-Healing for Planetary Exploration and Ocean Monitoring. This is the first document of its kind. We have foreseen different iterations to refine use case

description and requirements, to maintain the technical specifications of the project (elicited from them) constantly updated during the entire project lifetime.

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 España, SA	TASE	ES
4	Università degli Studi di Cagliari	UniCA	IT
5	Institute National des Sciences Appliqués 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
Antonella Toffetti	CRF
Giovanni Turi	CRF
Andreea Balau	TNO
Yunyun Ni	S&T
Hans Myrhaug	AMBIESENSE
Antonio Lopez Verona	TASE
Manuel Sanchez	TASE

Document Revision History

Date	Ver.	Contributor (Beneficiary)	Summary of main changes
04-05-2017	0.1	TNO/AMBIESENSE/TASE/ CRF/S&T	First draft
16-05-2017	0.2	UNISS and IBM	comments and refinements
18-05-2017	0.3	TNO	Updated structure
30-05-2017	0.4	TASE/UPM	Updated use case
01-09-2017	0.5	IBM	Update on the structure to include use case requirements
13-09-2017	0.5	TASE	Alignment and refinements
13-09-2017	0.5	UNISS	Check and refinements

Table of contents

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. This document, for each scenario, is meant to highlight challenges and goals to be addressed and will provide a description of the basic characteristic of each scenario. Moreover, a preliminary overview of the use-cases and demonstrators to be implemented within each scenario is going to be tackled. Finally, a first set of requirements are presented for each use case.

This is the first document of its kind. We have foreseen different iterations (M13 and M19) to refine use case description and requirements, to maintain the technical specifications of the project (elicited from them) constantly updated during the entire project lifetime.

In this first document, each use case has been defined by the different use case leaders and contributions, according to common domain specific methodologies, Requirements have been identified per use-case, without any final cross-domain definition phase.

In the next version of this document, more homogeneous descriptions of use case will be provided and project common requirements will be elaborated.

1.1. Structure of Document

This document is organized as follow:

- Section 2, Section 3 and Section 4 explore the different Use Cases description.
    - Section 2 -Travelling for Electric Vehicle – describes the Smart Travelling scenario, along with a detailed description of its goals.
    - Section 3 - Healing for Planetary Exploration – describes the Planetary Exploration scenario, along with a detailed description of its goals.
    - Section 4 - Ocean Monitoring – describes the Ocean Monitoring scenario, along with a detailed description of its goals.

  For each scenario, and overview of the use case and its goals will be provided; followed by the system-level description of the demonstration environment and its main constituting software and hardware components, and the preliminary list of evaluation scenarios with the expected outcomes for each of them.
- Section 5 describes the CERBERO Use Case Requirements gathering methodology. Convention and template used by all the use case providers are reported.
- Section 6, Section 7 and Section 8 describe the gathered detailed requirements per use case.
    - Section 6- Smart Travelling for Electric Vehicle
    - Section 7- Self Healing for Planetary Exploration
    - Section 8- Ocean Monitoring

2. Use Case description Smart Travelling for Electric Vehicles

2.1. Introduction

In this chapter the use case Smart Travelling for Electric Vehicles is described.

Beside the generic description of the Smart Travelling scenario (which covers the overall functionality of the use case), a number of individual Smart Travelling use cases are described as examples of those to be developed and validated in WP6, using the CERBERO toolchain developed in WP3, WP4 and WP5.

For the use case Smart Travelling we need to make a clear distinction between users of the CERBERO tool chain that are the CRF engineers responsible for development of the Virtual Driving Simulator, users or test drivers which will execute test drives on the component of the simulator to be developed as Smart Travelling demonstrator and end users being all the people who will drive electric vehicles now and in the future. During the CERBERO project, end users will only be involved during WP2 during Focus Group sessions. For testing the developed demonstrator the CERBERO team itself will act as test drivers.

In order to determine the most relevant and important individual use cases and scenarios, Smart Travelling end user requirements (where end users are test drivers) will be collected using a number of focus groups in Italy and in The Netherlands. Based on the outcome of these interviews a number of Smart Travelling individual use cases will be selected and/or added (in case they were not specified yet in the examples described in this deliverable version). In Chapter 11 the used interview scripts are included, which are executed to determine Smart Travelling use cases through the real drivers' interview

In the current version of the document a number of sample simulation (software) modules are included. Once the scenarios are defined, the required simulation modules will be updated and finalized. Also high level user requirements are defined for the Virtual Driving Simulator.

The selected scenarios and related simulation modules will be used as input for the definition of the CERBERO user requirements (D2.3), which will be the input for the CERBERO technical requirements (D2.6). Here the "users" are the CRF experts who need to develop the Smart Travelling functionality for the Virtual Driving Simulator of CRF. The technical requirements will be used to develop the CERBERO toolset (in WP3, WP4 and WP5) and driving demonstrator using the toolchain (in WP6).

In Figure 1 the development approach to generate relevant CERBERO deliverables for the Smart Travelling use case is sketched.

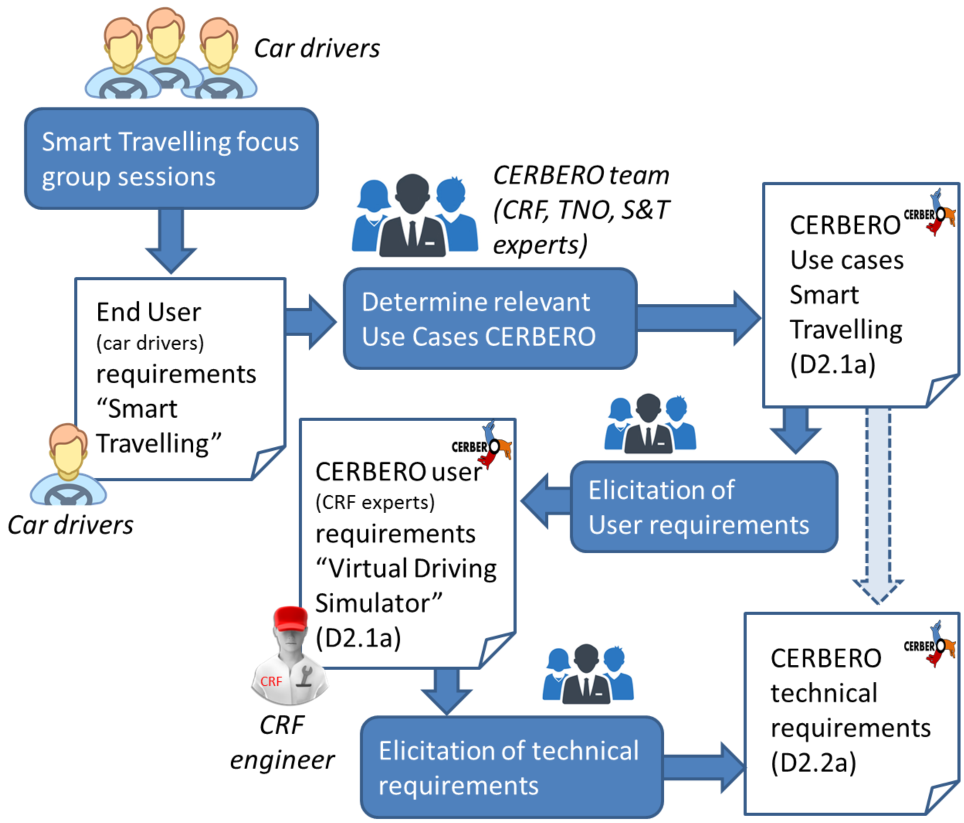


Figure 1: Approach to develop CERBERO deliverables for the Smart Travelling use case

2.2. Goals

The goal of the use case Smart Travelling for Electric Vehicles is testing and validating the Driver Support Interface for Electric Vehicles developed in CERBERO, and interacting with the CRF Virtual Driving Simulator.

The added values in adopting the software modules provided and/or developed by CERBERO ("the CERBERO toolchain") in the Virtual Driving Simulator architecture could be, for example:

- Shortening the 3D database development process for implementing individual use cases, scenarios and simulated environment;
- Optimizing the integration of an external electric vehicle dynamic model;
- Optimizing the data synchronization coming from different distributed modules.
- Build and/or accelerate modules using CERBERO tools and technologies

In Task 2.2 and its deliverable we will map Smart Travelling use case on CERBERO framework components, but we can already said that the top-most part of the CERBERO framework, i.e. the one more system-level oriented, will be assessed by means of this scenario.

2.3. *Use case description*

The use case Smart Travelling focuses on the interaction between human and the Electric/ Vehicle, on the assistance an electric vehicle can give to the driver, when confronted with the task of driving the car from a given origin place A to a given destination B. This assistance must also assure that sufficient battery capacity is available to complete the route.
CRF plans to develop a virtual driving demonstrator during the CERBERO project in which human driver can be monitored during the task and where environment and most of the functionality of the car itself will be simulated (apart from the seat, direct controls like steering wheel and pedals and the user interface). Figure 2 shows a picture of the Virtual Driving Simulator available at CRF, which will be used for developing the demonstrator.

Figure 2: CRF Virtual Driving Simulator which will be used for the Demonstrator

The current virtual driving simulator is already used for other projects by CRF and the intention now is to use the CERBERO toolchain in the development of a virtual demonstrator for testing and validating the **Driver Support Interface**, using a newly developed user interface.

CERBERO Driver Support Interface
The CERBERO driver support interface integrates/adds some new functionalities to the current driving simulator, such as monitoring the driver behaviour. New aspects have to be monitored and researched and new simulation software needs to be developed in order to run new scenarios as emerged from focus group results. The aim of the adoption of the CERBERO toolchain is to contribute in making more efficient the development process of CRF for the creation of new 3D database and data synchronization module.

Important steps in the development of the demonstrator are the integration of different modules used in the simulation, like simulation of the road, simulation of the battery behaviour (related to performed route and driver behaviour), the simulated charging infrastructure, adaptive route planning, and possibly some functionalities from other domains (e.g. agenda) which can interfere with the planned path.

The actual functionality to be included in the Smart Travelling use case will depend on the individual use cases, which will need to be implemented. Individual use cases reflect specific needs or circumstances expected when driving an electric vehicle and involve specific scenarios of interaction. The individual use cases will be selected based on focus group interviews (with end users of the electric/hybrid and conventional powertrain vehicles too), which will indicate what issues they foresee or currently struggle with when driving an electric or hybrid (some problems can be similar with electric vehicle) vehicle. See Section 11 for script used during the user group interviews. To get a better understanding of future users, user group sessions will be organised in both the Netherlands and Italy. Based on the interviews, a number of individual use cases will be developed for the virtual driving demonstrator (including an essential user interface to support the user during the drive) in order to test and verify the CERBERO Driver Support Interface.

In Section 2.5 a number of sample scenarios (called individual use cases) are sketched. As interviews are not completed yet, a final decision on which individual use cases to implement will still need to be made. The complete description of the chosen individual scenarios that will access the CERBERO Driver Support Interface is going to be presented in the subsequent updates of this document (see Section 5).

By integrating all selected functionalities, it will be possible to use the virtual driving demonstrator to perform the routes of the selected individual use case and experience the simulated environment with requested interactions as driver of the demonstrator. By simulating and integrating the different models it will also be possible to respond to unexpected changes (e.g. caused by driver decisions and actions), where demonstrator is able to automatically adapt to new circumstances. It is expected that tools from the CERBERO toolchain can help to integrate the different functionalities and models and provide the required adaptability.

Integration will also involve synchronisation of time (clocks) between all the simulated parts. In order to allow all selected individual use cases to be experienced, the individual use cases and scenarios will be designed with a proper length, so the parts of the routes can be executed in real time.

CRF currently uses SCANeR software (see [SCANeR Studio]) to implement and run the different individual use cases and scenarios.

Currently the ongoing activity is on understanding integration opportunity offered by SCANeR towards available tools in the CERBERO integrated design environment, which have been already surveyed at the time being and may be extended according to the scenario needs.

In particular, investigation are going on to understand if some of the CERBERO tools (like DynAA from TNO (see [DynAA])) could be used to execute some of the models and to interface with the SCANeR software during the actual simulations.

Moreover, some “open doors” in virtual driving simulator architecture were identified, in order to facilitate the decision making on tools to be integrated, chosen accordingly to needs that will emerge from focus group results. The three opportunities (taking into account the virtual driving simulator input requirements) are the following:

1) Generation of a simulated environment starting from commercial existing maps (with path representative for electric vehicle missions);
2) Integration of an external Electric Vehicle and its relative consumption model into the SCANeR development environment;
3) A tool to synchronize, in a post processing phase, the data logged (with different rates) coming from different systems (e.g. vehicle performance, video of the driver, HMI input, etc…).

2.4. System architecture and components

Based on the Smart Travelling individual use cases selected for CERBERO (see Section 2.5) a number of simulation and prediction (software) modules need to be developed and/or integrated in the CERBERO toolchain in order to execute the selected scenarios. The simulation models are needed to provide a (real time) simulated environment to the driver of the demonstrator. The prediction modules are required to provide advice to the driver (e.g. on routes or options for loading of batteries) and calculate predicted results of specific option (e.g. feasibility, cost, duration, impact on batteries, etc.).

Depending on the selected individual use cases (determined based on the results from the focus group interviews) a subset or superset of the following sample modules will need to be provided by the demonstrator.

Simulation modules:

- **3D graphic databases** (representing roads, car and other objects in the environment);
- **Route maps** (street maps like Tele Atlas or Navteq);
- **Car simulator**, which will interface with the controls and actuators of the virtual demonstrator and interface to the battery and engine simulators;
- **User Interface** via which user can be informed and can control the vehicle and route;
- **Driver Support** function, which will generate user advice towards the UI and help users to accomplish successful trip (like route advice or charging predictions and advice based on user preference, such as using different categories of drivers). Input for the device could be collected from a number of remote applications (like weather forecast, charging poles information, traffic information, parking reservation). The module could also be extended with more advanced functionalities to support the driver;
- **Battery simulation** (which, based on the charging strategy, engine usage, heating or cooling of car and other environmental input, will simulate the behaviour of the battery);
- **Engine simulator** (which will simulate behaviour of engine based on driving situation like slope, acceleration, deceleration);
- **Traffic simulator** (which will simulate the other cars on the road around the vehicle driven using the driving simulator);

- **Smart grid (loading pole) infrastructure** simulator of (possible multiple) energy providers, which provide loading poles for simulation of charging the car;
- **Smart home / smart office** (as one of the charging alternatives);
- **Personal agenda** (which can be used to determine route to travel);
- **Weather simulation** (were weather could have impact on battery consumption);
- **Scenario control** functionality, which will control the complete execution and of the simulated scenarios (including raise of triggers during use case execution or setting of specific use case and/or module configurations);
- **Time synchronisation** function which needs to provide mechanisms that ensures all modules are run in a synchronised time and resulting logging of the different modules can easily be matched.
-

Prediction modules:

- **Route planner** (to calculate different options like possible routes to reach destination);
- **Traffic predictor** (to predict future traffic, which could be used to determine best route to take);
- **Battery simulation** where simulation will be used to make predictions and simulations will run much faster than real time (used to calculate impact on battery for a possible route);
- **Engine simulator** where simulation will be used to make predictions and simulations will run much faster than real time (used to calculate impact on engine / need to power for a possible route);
- **Smart grid charging infrastructure** predictor which could tell something about future usage (and for example occupation level of specific poles along the route);
- **Smart home / smart office** predictor, which could indicate availability and price of energy at home location (e.g. based on time of day and expected energy use);
- **Expected behaviour of driver predictor** for the driver (which could be based on known preferences and/or detected driver category and used to propose routes and alternatives for charging options, which could include related activities like rest stops);
- **Personal agenda**, which could be source of predicted travelling needs of the driver (and thus need for specific future battery loads).

Eye movement will be registered during the simulation to register human behaviour during the specific moments of the trip. No specific simulation is needed for this, only synchronised registration of simulation events and the eye movement tracking.
In a sample demonstrator architecture is sketched, with different simulation and prediction models that need to be interfaced and integrated to create a complete driving experience running one or more of the selected scenarios.

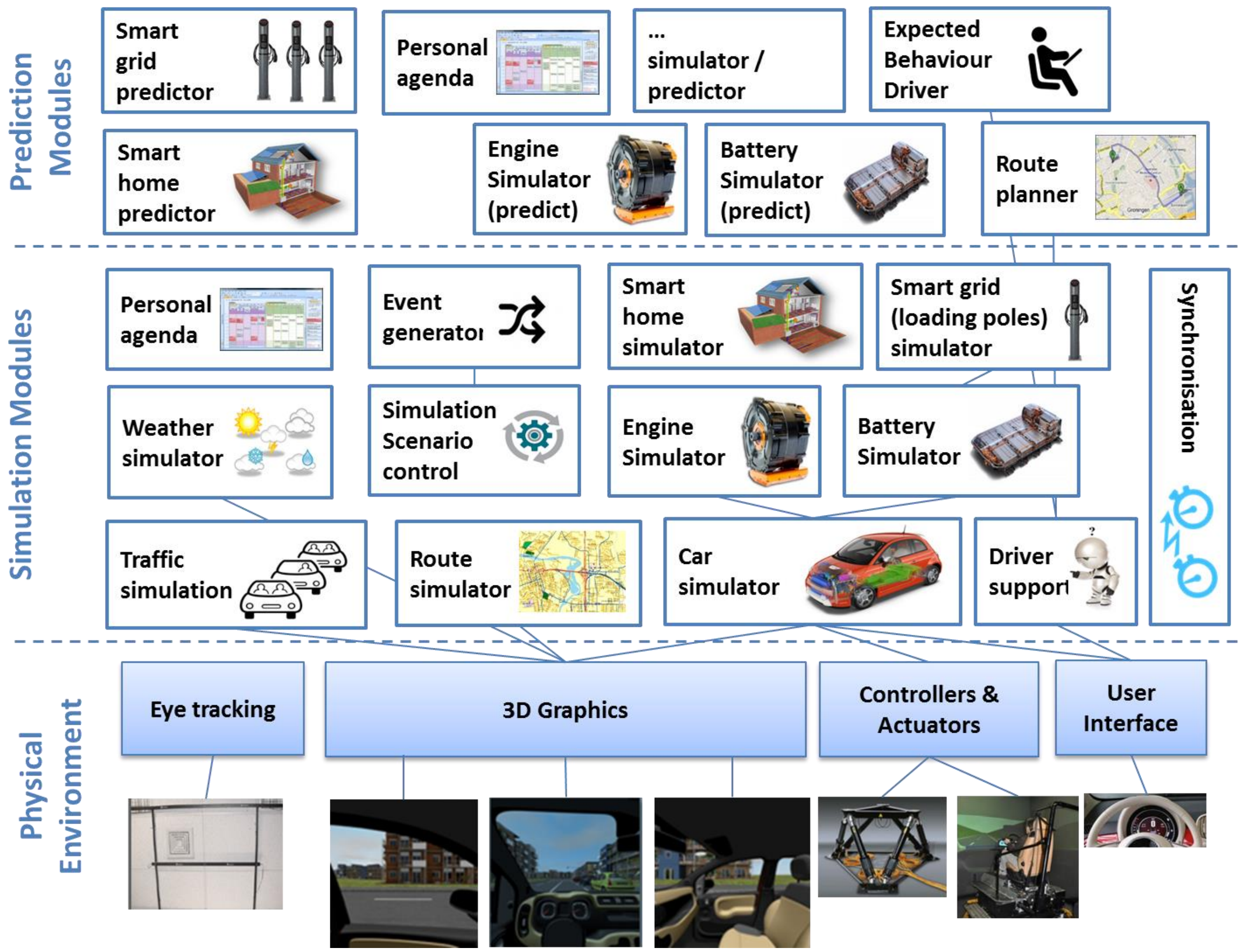


Figure 3: Sample virtual driving demonstrator architecture

Integration of the modules will be a very important part of the activities to be performed within the CERBERO project and for which it is expected that the CERBERO toolchain can help to optimize, improve and accelerate the integration process.

CRF already has the physical environment set up in the Virtual Driving Simulator, with simulation modules for conventional cars. To create the CERBERO demonstrator some modules need to be replaced (like engine simulator) and some need to be added (like the driver support module and other EV related simulation and prediction modules).

It is expected that some of the modules, such as the (electrical) engine simulator and battery simulator from TNO, can be integrated in the CERBERO demonstrator. S&T will provide logic for the (to be developed) driver support module. The TNO DynAA tool will be used to execute some of the simulation modules and run prediction modules to calculate different alternative (e.g. in the case driver support wants to compare number of alternative routes and needs to calculate impact on engine and battery). CRF also wants to create a new synchronisation mechanism between the modules, in which DynAA could play a role. Especially because DyNAA will be involved in both real time simulation and simulation in accelerated time (as used for prediction modules). DynAA could also be used to model and create new simulation modules for the driving simulator. Usage of other tools from the CERBERO toolchain needs to be investigated further.

2.5. Scenarios

In this chapter some possible scenarios (called individual use cases) are sketched (using different scenarios of interaction) which could be executed using the virtual driving demonstrator of CRF.

In each of the individual use cases some of the following high level functional and non-functional requirements are included:

Functional:

- **Battery capacity management**: Distance which can be travelled with certain amount of battery capacity will depend on: temperature, usage of in car climate control (heating/cooling/fan), age of battery (and number of re-charging), charging strategy, driving and engine behaviour (e.g. speed, acceleration) and the road profiles (e.g. slope of road).
- **Battery charging strategy**: Charging speed will have consequences on costs (faster charging will be more expensive), battery life (faster charging will reduce expected life) and travelling time (charging time will add to total travelling time).
- **Determine availability of charging poles**: The availability of the charging poles will depend on time of day and other traffic along the route. Availability can also be related to cost as some charging poles might be much more expensive but better available than others. Other limitations can depend on availability of electricity in the Smart Grid.
- **Manage cost of charging**: The cost of charging, which can be related to the speed or capacity of the given charging point, will differ for the different charging poles. Cost would in most cases be related to the availability of energy in the grid and the demand for energy at a given location (e.g. home, office or on high-way) and moment in time (day, night, rush hour). Cost of energy for a specific charging pole could potentially thus also vary over the day (which might also depend on local regulations).
- **Plan route**: The user should be supported with planning the route, taking into account costs, duration, other traffic, charging needs, other (successive) trips, etc.
- **Adaptability of route**: When drivers are performing a trip, the user should be able to adapt to occurring circumstances (like traffic jams, change of plans and thus route, availability of planned charging locations), possibly helped or advised by some logic in the car (which for example monitors traffic or availability of charging points on the planned route).
- **User advice and preferences**: When supporting users during the travelling (e.g. on route assistance or charging options), the car should take specific user preferences into account and tailor advice to the user based on the situation (which could involve critically and known/learned user preferences).
- **User interaction**: The user will be able to interact with the driving simulator via a set of controls (like the steering wheel, paddles, HMI). The user will not be able to modify all setting however, as some of the settings will be determined by the specific simulated scenario (e.g. climate temperature settings).

Non-Functional:

- **Real time simulation:** the whole simulation need to run at real time to give driver a realistic experience (and avoid driver becoming dizzy during the drive). The simulation modules therefore also need to be able to run at real time.
- **Adaptability simulator:** It should be possible to modify the simulation by, for example, integrating new or different simulation modules.
- **Reusable simulation modules:** It should be possible to use simulation modules in many different use cases. It is therefore important to make interfaces as generic as possible to optimise reuse.
- **Simulation with software / hardware in the loop:** Simulations will include additional software (e.g. prediction modules via DynAA) and hardware (car control hardware and driver support interface) in the loop.

2.5.1. Commuting trip to and from work - route & charging advice

In this individual use case the driver will perform commuting trip to work. The challenges the driver is confronted with in this individual use case are described by the following scenarios:

- Battery of the car is half charged;
- There is a traffic jam towards work;
- Driver has to consider both the trip towards work as well as return home after work (where charging at work is more expensive than charging at home);
- Driver has to verify if additional trips are required during the day (in scope of work), for which certain amount of battery State Of Charge (SOC) would be required;
- Driver has to decide on costs, the convenience of having a full battery and the time needed to charge during (additional) trips. When possible solutions are presented to the driver, the known preferences should be taken into account (e.g. to select or order options);
- Charging could involve charging at work location, where parking place could be combined with charging pole and reservation of the location is required.

2.5.2. Highway route re-planned - charging pole “out of order”

In this individual use case the driver will perform a trip on the highway, where planned charging location is not available because of detected failure (pole “out of order”) and driver has to re-plan and select an alternative route and charging point. The challenges the driver is confronted with in this individual use case are described by the following scenarios:

- Battery of the car is charged at home but capacity is not sufficient to reach the end destination;
- User has to plan charging points along the trip in order to reach destination;
- Charging pole (which was planned as the next charging point) becomes “out of order” (advice via some remote charging service of energy provider);

- Driver needs to adapt route (as next charging point on the route is too far away) and select other charging point as alternative;
- Selection will involve costs of alternative charging poles and additional distance and time needed to drive via the alternative route.

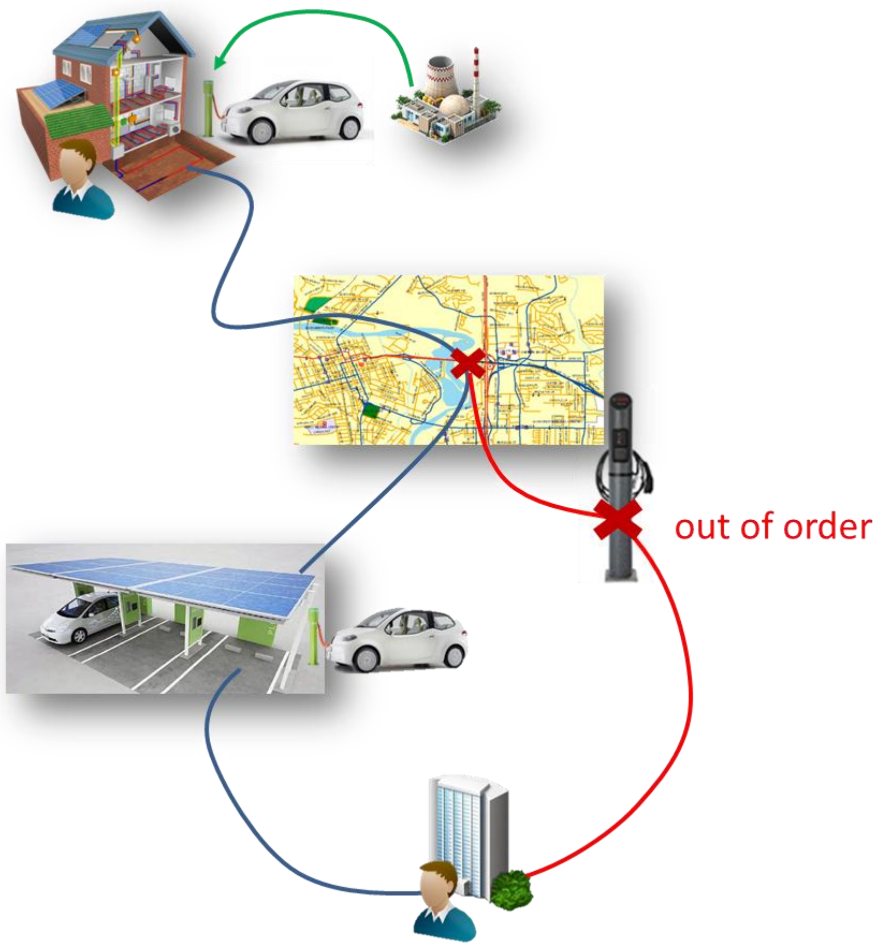


Figure 4: Highway route re-planned

2.5.3. Up-hill trip – how to cope with extra capacity demand

In this individual use case the driver is confronted with additional battery usage because of up-hill trip (which will result in higher usage than estimated earlier). The challenges the driver is confronted with in this individual use case can be described by the following scenarios:

- Battery SOC is estimated to be sufficient to perform trip of given distance;
- Because of terrain (slopes) battery consumption during the trip will be higher than expected for given distance;
- Driver has to consider additional charging during the trip because of the extra consumption or adapt trip to reduce energy consumption (in case driver is able to relate terrain to energy usage).

2.5.4. Emergency call – how to adapt when route is to be changed

In this individual use case the driver will perform a certain trip and because of some emergency call (e.g. from home) he directly needs to alter the destination of the trip. The challenges the driver is confronted with in this individual use case can be described by the following scenarios:

- Route and charging poles to be used are planned for original trip and destination (trip was too long to complete without charging);
- Driver gets an emergency call and needs to adapt the end destination and therefore the whole trip and planned charging scheme;
- Original charging poles possibly need to be released (in case they were reserved for specific time period);
- New route need to be determined, taking into account current and expected traffic and available charging poles;
- Driver will be able to complete route to alternative destination.

2.5.5. Unexpected congestion – how to cope with rerouting

In this individual use case the driver will be confronted with major congestion and needs to adapt the route, which will involve a much longer trip and thus need for additional charging. The challenges the driver is confronted with in this individual use case can be described by the following scenarios:

- Driver is following a route where destination can be reached with available battery SOC and charging is already arranged (reserved) at the destination;
- A major congestion on the planned route (received via some remote route prediction service) forced the driver to consider an alternative route;
- The driver needs to re-plan the route and consider charging locations as alternative route is substantially longer and current battery SOC is not sufficient to reach destination;
- Driver needs to consider if charging at destination is still required (to complete journey back) as additional charging is now performed during the trip;
- Driver could consider to not fully charge battery, and thereby save time on the alternative route, and still perform charging at destination;
- The reservation time slot of charging poles at final destination need to be adapted based on additional travelling time needed (as reservation will be automatically released when car does not arrive within specific time slot).

2.6. User requirements - Virtual Driving Simulator

Although the use cases (and thus modules to develop and or integrate) are not finalised yet, in this paragraph a number of (high level) user requirements are defined. Here the user is the CRF engineers responsible for the development of functionalities on the Virtual Driving Simulator.

User requirements (CRF engineers):

- Easy integration of external simulation (and prediction) modules;

- Real time simulation (to ensure driver can experience real time environment and simulator can record real time behaviour);
- Synchronised simulation (based on 100Hz frequency for most critical real time modules);
- Synchronised logging capability;
- Integration of driver support functionality (including predictions and advice) to drive the Driver Support Interface;
- Logging integration (where logging of different simulation modules is integrated into a single consistent set of logging data which can be analysed by external systems);
- Easy generation and linking of tagging data (during or after simulation runs);
- Easy creation of 3D database content for execution of new use cases for driving simulator (possibly using some external geographic map data like OpenStreetMap (see [OpenStreetMap])).

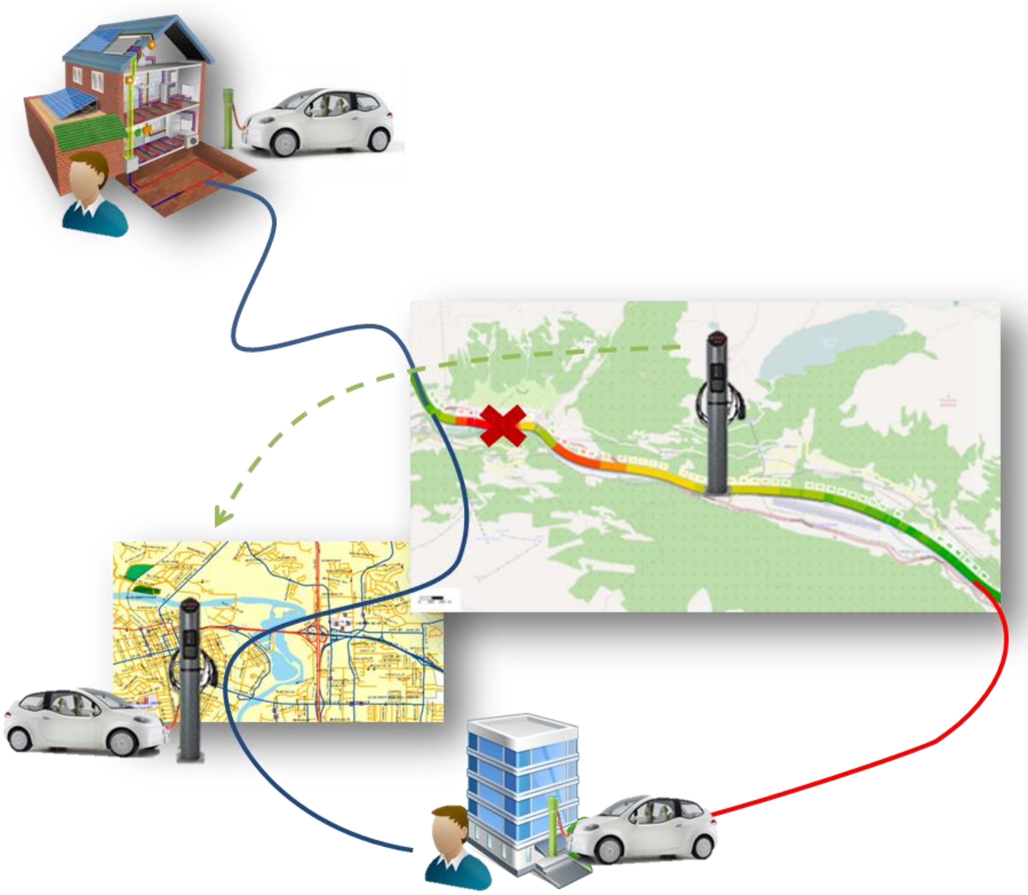


Figure 5: Unexpected congestion

3. Use Case Self-Healing System for Planetary Exploration

3.1. Introduction

In this document the use case Self-Healing System for Planetary Exploration is described.

The space use case will be based on two demonstrators:

- Robotic arm
- Motor control unit

Both demonstrators will be developed and validated in WP6 and using the CERBERO toolchain developed in WP3, WP4 and WP5.

The main controller device of the robotic units (e.g. arm and rover locomotion system) will be designed adopting the CERBERO design environment based on a flexible and heterogeneous MPSoC architecture with self-reconfigurable COTs solutions.

The scenario should cover multiple mission tasks (requires different accelerator types) and should be fault-tolerant and self-healing (the most common approach requires modular redundancy + a voter unit, making it mandatory to increase the number of accelerators used for the same task). In addition, the platform should be able to adapt to changing conditions (e.g. solar storms that require increased hardware redundancy, communication links only available in certain parts of the orbit that impose more stringent real-time deadlines for a short period of time, etc.).

The assessment of both demonstrators will be based on external fault injection using Xilinx *Soft Error Mitigation* IP in order to detect and correct single and multiple bit errors. The behaviour of both demonstrators will be analyzed under fault injection in order to validate the fault-tolerant and self-healing techniques.

3.2. Goals

The main goal of this use case is to provide self-monitoring and self-healing capabilities by means of high performance sensor processing techniques that may trigger dynamic reconfiguration of the embedded computing system to overcome the failures caused by the radiation or the harsh environmental conditions. Such support will ensure long-term maintenance of the system and will guarantee the possibility of dealing with the stringent survival conditions and reliability constraints of a robotic exploration mission. Here is a list of trade-offs that the CERBERO project will be capable of mastering.

1 - Failure level detection and failure detection time vs. Monitoring logic overhead.
2 - Degree of on-board autonomy vs. Downtime and Reconfiguration time.
3 - Degree of ruggedization vs. Increase of logic area and power consumption.

4 - Degree of autonomy / Type and number of correctable errors vs. Predictability of the system.

3.3. System architecture and components

The architecture of the robot arm controller is depicted in next figure. It will be based on a Robot Control Unit (RCU), depicted in Figure 6 and Servo Control Units (SCUs) in a similar way to previous ESA projects [CIRCUS].

The RCU receives commands and sends telemetry from/to the System Controller Device via a RS-232 serial line, it interprets and executes the commands, performs robot motion planning and interpolation, and sends servo level commands to the SCUs.

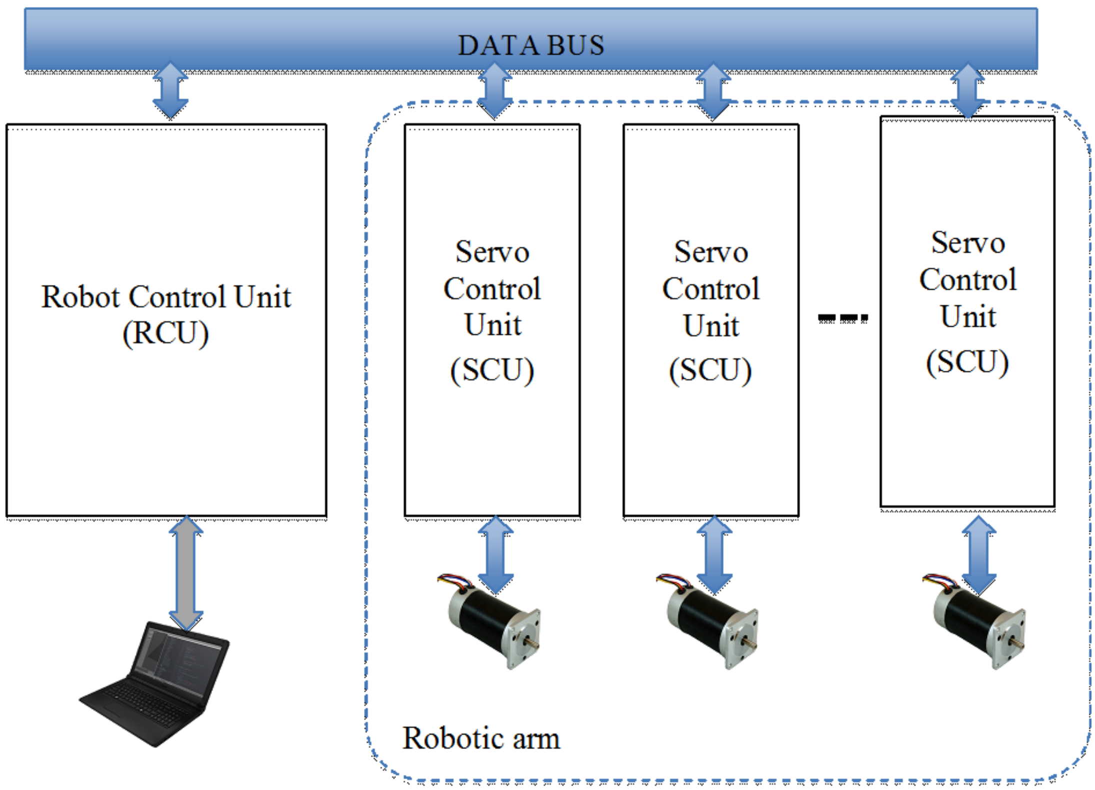


Figure 6: Robot arm controller architecture

The SCU (Servo Control Unit) exchanges data with the RCU, receiving position and velocity set-points and operation commands

As the SCUs are located in robotic arm with PID controller, we plan to focus CERBERO demonstrator in the RCU.

3.3.1. Robot Control Unit

The robot control unit will be in charge of the following arm functionalities:

- Generation of arm movement trajectories
- Validation of collision-free motion paths
- Setting parameters
- Reading sensor data and monitoring the arm status

- Fault detection and recovery
    - Fault - inability to complete a command due to failure of hardware (single event upset, sensor, actuator, electronics, etc.)
    - Event - inability to complete a command due to anything other than a fault (e.g., arm motion impeded by a rock in the digging path)
- Sending arm sensor data to telemetry

The kinematic control will be implemented according to the flow chart depicted in Figure 7.

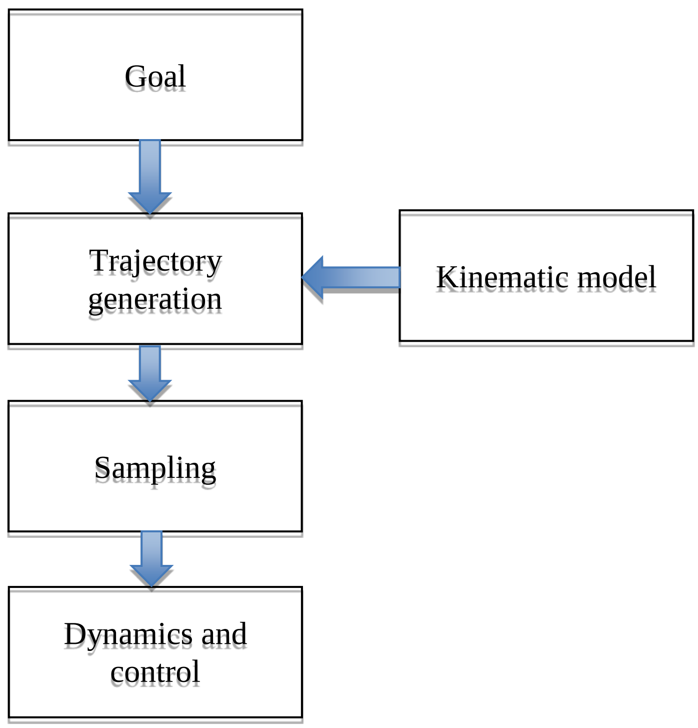


Figure 7: Kinematic control

In order to demonstrate the RCU operation a commercial robotic arm kit has been purchased: WidowX (see Figure 8). This arm is based on Dynamixel servos as SCUs and a microcontroller as a controller. We plan to interface the microcontroller with an MPSOC based on a FPGA Zynq UltraScale+. A ZCU102 evaluation kit has been purchased to use it as a robot control unit.

The next algorithm will be implemented and tested on Zynq UltraScale+:

- Trajectory generation (collision-free motion paths)
- Forward and inverse kinematic model
    - Jacobian matrix (joint velocities to *end-effector* velocities)
    - Optimization algorithms
- Control loops (PID)

3.3.2. Motor Control Unit

In this demonstrator the goal is to control a brushless DC motor (BLDC) with self-healing characteristics. This kind of motor is used in space environment, so the demonstrator will be space representative in terms of motor control.

The motor control architecture will consists of a motor driver circuit to drive each phase of the BLDC motor and a motor controller to implement the loop control of current/speed.

An example of the motor control architecture is depicted in Figure 9.

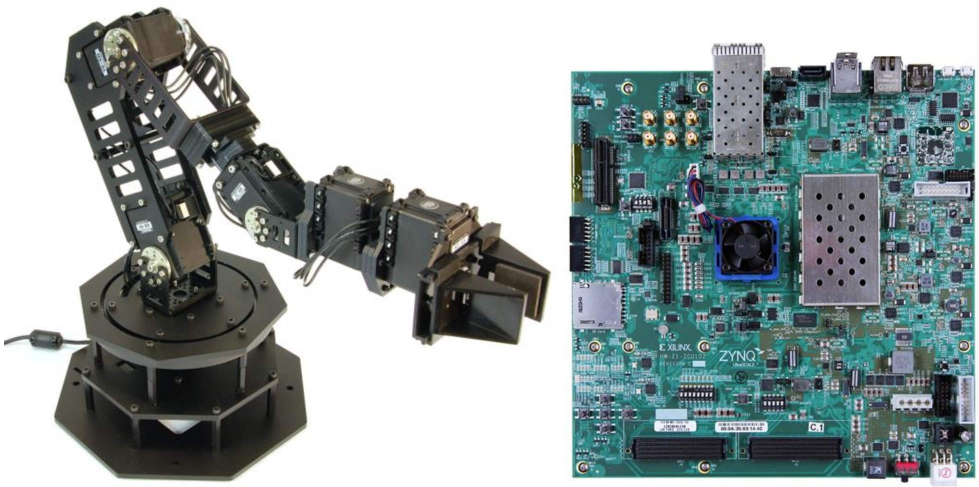


Figure 8: Robotic arm & ZCU102 board

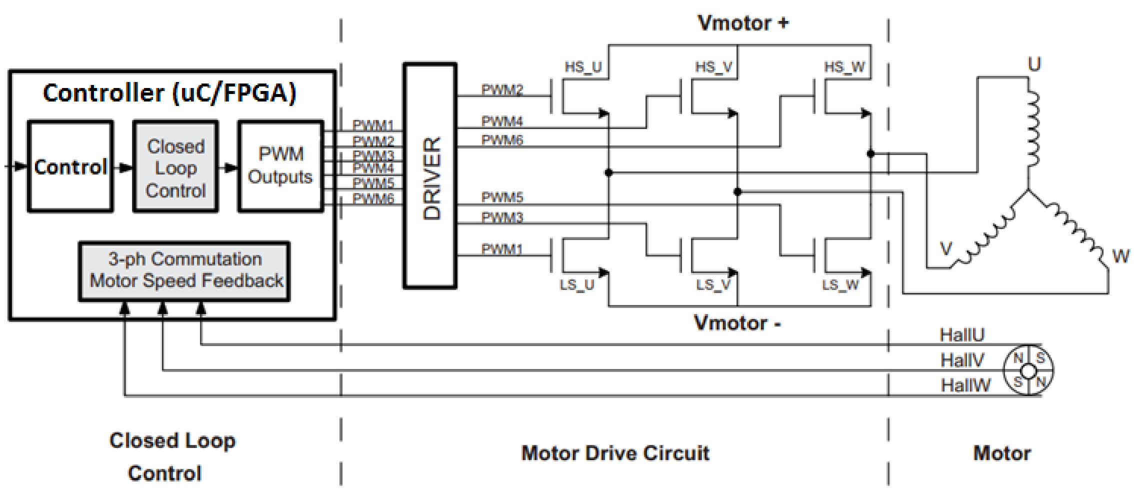


Figure 9: Motor control architecture

The motor controller will be based on a microcontroller or FPGA according to the available commercial boards.

The BLDC motor will be integrated in a dynamometer setup in order to characterize the control loop. An example of the motor setup is depicted in Figure 10.

Figure 10: Motor setup

4. Use Case Ocean Monitoring

4.1. Introduction to Ocean Monitoring

The Ocean Monitoring use case comprises smart video-sensing unmanned vehicles with immersive environmental monitoring capabilities. They serve as “marine eyeballs” that can capture live videos and images of the local on-sea and subsea surroundings. These marine robots can be remote controlled within wireless reach and visible sight, but they will be also designed for self-operation and navigation. The vehicles will be equipped with smart sensing and processing capabilities, used not only to navigate and operate the robot, but also for data analysis and information fusion.

4.2. Goal

The goal is to research and develop CPS-prototypes for the Ocean Monitoring scenarios, towards commercial products and for demo purposes. These marine robots will be developed by following a rapid prototyping strategy, in order to reduce costs and time to market.

4.3. System Architecture and Components

The aforementioned robots can perform on-the-fly data analysis and data fusion in order to make decisions and adapt to changing environment. Sensed data can be stored locally or streamed to a cloud service from where relevant information can be retrieved. To meet emerging environmental needs and standards, and to be as unobtrusive as possible to the surroundings, they should ideally be 100% battery driven - something that also reduces the amount of mechanical components needed.

In terms of information processing, a multi-core processing infrastructure seems to provide the most scalable and flexible computing architecture. This allows for agile integration of sensors and actuators and to manage both navigation and sensors processing. The types of sensors beneficial for implementing the marine robots are: camera, gyroscope, microphone, GPS, light sensor, pressure sensor, and compass. Interoperability and communication can be achieved by combination of WiFi, Bluetooth, Ethernet, USB, mobile network, remote control, and tracking device to assess proximity between sub-systems. A flexible CPS architecture will be chosen, capable of enabling adaptive system behaviour and navigation in rapidly changing marine environments. Actuators that are mainly electrical and digitised are also foreseen. Examples of actuators are: steppers, servos, steering, rudders, winch, anchor, camera position control, and led-lights for dealing with the problem of reduced visibility.

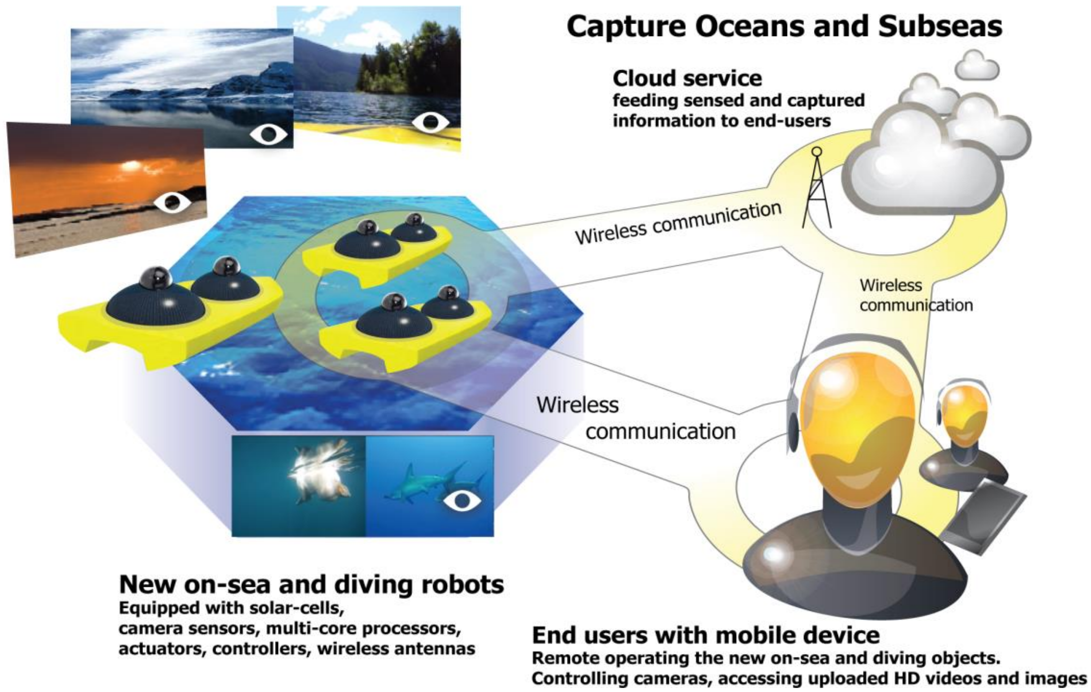


Figure: Ocean Monitoring - System Architecture and Components

Three overall scenarios were identified to help describe complementary aspects and concerns of the marine robots, respectively: i) observing wildlife, ii) subsea monitoring, and iii) marine robot propulsion and transport. A selected set of individual use case descriptions were next described within the contexts of each of these in the following order (see Table 1):

Table 1 - **Ocean Monitoring - Summary of Scenarios and Use Case Descriptions**

Overall scenario	**Use case descriptions**
Observing wildlife	● Enabling adaptive camera system for parallel delivery of videos and images ● Ocean monitoring - surface
Subsea monitoring	● Eye in the water ● Searching for missing vehicles ● Storing data streams from multiple sensors
Marine robot propulsion and transport	● Reconfiguration of battery module in runtime ● Control of thrusters and steering through software ● Set the course and forget - autopilot ● Remote control of marine robot

Together, these scenarios and individual use case descriptions served to elicit functional and non-functional requirements of the marine robots, i.e. the needed electronics, software, along with the physical design and capabilities. The requirements were elicited, grouped with the individual use case description, and enumerated accordingly. The exact list and content of the use cases implemented in CERBERO will be refined in CERBERO D2.2a Technical Specification deliverable and future versions of this deliverable.

The scenario and use case descriptions are presented in the following sections below. The related requirements were then organised as groups associated with each use case description, and included as part of the requirements section/ report. These are:

- Requirements for adaptive camera system
- Requirements for ocean monitoring - surface
- Requirements for ocean monitoring - subsea
- Requirements for searching for missing vehicles
- Requirements for storing data streams from multiple sensors
- Requirements for reconfiguration of battery module in runtime
- Requirements for control of thrusters and steering through software
- Requirements for autopilot - navigation
- Requirements for remote control of marine robot

4.4. Scenario: Observing wildlife

Here follow a list of possible individual test case within this scenario. In the following version of this document they will be prioritized to specify CERBERO test cases.

Title	Observing wildlife
Author	Leszek Kaliciak, Hans Myrhaug
Overview	A marine biologist deploys the marine robot to survey the population of whales
Description	
The user, a marine biologist, wants to study the whale population in the particular area of the ocean. He/she deploys the marine robots from the shore and sets their course for the chosen location. Using their self-navigation capabilities or the remote control, the robots arrive at the specified site. The marine biologist can keep track of the current location of the autonomous boats at all times and can take full control of the robots by using his/her computer system: laptop, tablet, smartphone. When the marine vessels arrive at their destination, the user starts the ocean monitoring process. He/she surveys the surface area around the robot in a natural way - the mounted cameras provide the high resolution view of the surroundings. Next, the marine biologist deploys the underwater device to a chosen depth in order to get a 360 degrees	

view of the subsea. In order to get a more immersive experience and to be able to survey the environment in a more natural way, the user wears the virtual reality (VR) goggles.
Issues
● Loss of direction due to loss of gps, compass, gyroscope, or radio beacon signal ● Offset due to strong wind and waves ● Adrift due to loss of energy ● Wet electronics/seawater ● Loss of communication with the robot
Benefits
● Composability: Auto and manual navigation including obstacle avoidance ● User: Immersive ocean monitoring, data capture and storage ● Environment: Energy efficiency and adaptivity
Needed/ derived use cases
● Enabling adaptive camera system for parallel delivery of videos and images ● Ocean monitoring - surface ● Eye in the water - an augmented reality immersive experience

Name	Enabling adaptive camera system for parallel delivery of videos and images
Author	AS: L Kaliciak, H Myrhaug, A Goker
Description	
Single lens cameras are limited by physics and the future as evidenced by the mobile phone market belongs to camera arrays or multi-lens cameras. Both dual and multi-lens camera systems allows for creating better quality pictures, measuring the depth perception, or generating panoramic or 360 degrees images and videos. Emerging camera solutions are trying to implement smart camera arrays for instance by combining pairs of camera lenses, but for a single specific purpose only e.g. for super-resolution. The marine robots will be equipped with up to several dual and single lens cameras that allow adaptation to the user's current context and needs. The user can make one of the following requests: ● **Immersive** - obtain wide angle, spherical 360, or stereoscopic pictures and videos of the surrounding environment. Multiple cameras will be used to take multiple pictures simultaneously and stitch them together using one of the state-of-the-art SIFT or SURF	

key point detectors and descriptors. Thus generated images and videos can be also viewed, later as well as in the real-time, in the virtual reality VR goggles for enhanced immersive experience. Pairs of lenses use two slightly offset cameras working in tandem. Thus generated images and videos can be viewed, later as well as in real-time, in the virtual reality VR goggles for enhanced immersive experience

- **Quality** - Use the cameras to remove the noise from the picture and improve its quality (super-resolution) which could be especially important in different light environments. Stabilise the image (camera shake reduction).
- **Computer vision** - Enhance the pictures and videos with real-time edge detection and object tracking. Use the cameras working in tandem for depth perception measurement to enhance the foreground objects similarly to the latest mobile phones with twin cameras. Use dual lens cameras for depth perception measurement to detect and track objects based on the background detection and subtraction, and for the obstacle avoidance

After receiving the user request, the "smart" camera array will use the required number of cameras to perform one of the aforementioned tasks e.g. fuse the multiple pictures into one, enhance the edges, or improve the picture's quality.

The obtained data in the form of processed pictures will be then delivered wirelessly to a web application via the http protocol in order to be presented to the user. This process needs to happen in parallel due to the high file sizes of the wide angle/360 degrees images and videos as well as the data being sent from a "fleet" of marine vessels. All the transmitted data will form a visual real-time ocean monitoring system. The images and videos may also require compression before sending and decompression on arrival. The web application/ browser will be running on a smartphone with an android operating system.

Goal	To adapt the number of cameras in use to a specific context: generation of the 360 degrees and panoramic images and videos by stitching them together in real time, generation of stereoscopic images, de-noising and increasing the resolution of pictures and videos, self-monitoring purposes, computer vision purposes - obstacle avoidance, edge enhancement, movement detection and object tracking, depth perception To wirelessly send wide angle/360 degrees images and videos from multiple cameras to a web browser in parallel, using image and video compression to speed up the delivery process thus creating an integrated real-time ocean monitoring system To avoid using motors to rotate camera lenses into position by deploying multiple camera lenses instead
Actor	User, Software

System	Multi-lens cameras, smartphone with android operating system, stitching algorithms for panoramic/360 degrees images and videos, image and video compression techniques, methods for generation of stereoscopic images, thermal camera, computer vision algorithms, depth perception algorithms, super-resolution algorithms
Use Case Figure	
 Copyright AmbieSense Ltd	
Preconditions	
A marine biologist wants to monitor the marine environment using his/her smartphone or VR goggles for more immersive experience. The marine robot equipped with multi-lens cameras needs to deliver the data to the user’s smartphone. The real-time monitoring requires this process to happen wirelessly and in parallel for the seamless user experience. In addition, the marine biologist can adapt the array of cameras in order to (for example) de-noise the picture to a specified level or use multiple cameras to model the depth perception	
Steps	
● The user starts the web application on the smartphone ● The application discovers and connects to the marine robot, a list of camera previews is made available	

• The videos are noisy and the user decides to activate a number of cameras to remove the noise • The application sends the http request to the robot via the internet requesting for the video objects/picture • The “smart” camera solution responds by activating a number of cameras and using one of the super-resolution algorithms. The camera array sends the processed images/videos in parallel • The camera solution compresses video and images beforehand for transfer optimisation purposes • The smartphone receives the data • The received data is aggregated on the smartphone • The application on the smartphone decompresses the video and images with available codecs • The noise-reduced picture/video is presented to the user. The user can activate more cameras to further de-noise the pictures/videos • Next, the user decides to use the cameras in a different way and selects four cameras to obtain the pictures and videos in the spherical view • The application sends the http request to the robot via the internet requesting for the video objects/picture • The camera solution responds by fusing the images and sending them in parallel • The multi-lens camera solution compresses video and images beforehand for transfer optimisation purposes • The smartphone receives the data • The received data is aggregated on the smartphone • The application on the smartphone decompresses the video and images with available codecs • The data is presented to the user in a form of a wide angle/360 degrees picture
Postconditions
The user adapts the camera array to enhance the image quality, reduce the noise to a specified level, enhance the edges of images and videos in real time etc. The user is able to remotely view the environment surrounding the vessel in real time, using his/her smartphone. For more immersive experience, the real-time 360 degrees video can be viewed in a natural way in the VR goggles with additional real-time image processing enhancements
Related Scenarios
Observing wildlife

Subsea monitoring
Related Requirements
Requirements for adaptive camera system

Name	Ocean monitoring – surface
Author	AS: Leszek Kaliciak, Hans Myrhaug
Description	
The user wants to monitor a particular area of the ocean. He/she remotely communicates with the marine robot via his/her laptop, tablet, or smartphone and uses the high resolution cameras mounted on the robot to survey the water surface around the vessel. The camera system takes high resolution images/videos of the surface environment. The marine vessel is also equipped with other types of cameras - infrared and thermal for capturing images and videos in a different electromagnetic spectrum. This additional visual information enhances the surveillance capabilities of the marine robot. The images/videos are sent wirelessly in parallel to the user’s web application via the http protocol. The user is presented with a real-time view of the surroundings and augmented content which highlights and tracks moving objects - potential user interests.	
Goal	Surface surveillance
Actor	Marine biologist
System	User, software
Use Case Figure	

Copyright AmbieSense Ltd

Preconditions
The user wants to survey the surface environment using the high resolution camera system and additional sensors like thermal and infrared cameras. He/she deploys the marine vessel and starts the application on his/her smartphone/laptop/tablet
Steps
● The user opens the ocean monitoring app on the smartphone ● The app discovers and connects to the system via wireless network ● The camera system takes images/videos of the surface environment around the vessel ● The images/videos are sent wirelessly in parallel to the user's web application via http ● The ocean monitoring app displays the real-time view of the water surface around the vessel ● The real-time surface view is also augmented with highlighted moving objects and enhanced edges
Postconditions
The user is presented with a real-time view of the water surface. The video is augmented with enhanced edges and highlighted moving objects.
Related Scenarios
Observing wildlife
Related Requirements

Requirements for ocean monitoring – surface

4.5. Scenario: Subsea monitoring

Here follow a list of possible individual test case within this scenario. In the following version of this document they will be prioritized to specify CERBERO test cases.

Title	Subsea monitoring
Author	AS: Leszek Kaliciak, Hans Myrhaug
Overview	Monitoring of the subsea using smart camera system and additional optional sensors like laser and sonar, for example
Description	
The underwater marine vessels, which can be deployed from the boat, provide the operator with continuous information from multiple sensors when connected with an umbilical cord. The underwater robots can optionally propel themselves and adjust their depth by filling/emptying small cylindrical containers with water. This allows tilting the vehicles, thereby making them float or dive in the sea. There are wireless antennas, reaching above water when the robots are floating on the surface, enabling wireless communication and signalling for the operator. There is also room for optional mounting of additional digital equipment such as temperature sensor, pressure sensor, multibeam echosounder, side scanning sonar, lasers. One type of sensor, a smart multi-lenses underwater camera system, can take wide-angle/360 degrees images and videos of the subsea, which can be stored locally as well as streamed to the user's device. The camera system can operate in poor lightning conditions and can adjust the image quality dynamically by combining information from different lenses. The robots can be launched and picked up from the sea by the user. The ruggedized vehicles are designed to withstand the harsh marine conditions and can be used for equipment and pipe installations surveying, searching for missing vehicles, and monitoring of the marine flora and fauna.	
Issues	
• Data communication • Buoyancy	
Benefits	

• Data: Multiple-sensors, Smart camera system • Composability: Ruggedized, waterproof, airtight • Networkability: Real-time immersive monitoring • Business: Efficient operations and maintenance
Needed/ derived use cases
• Eye in the water • Searching for missing vehicles • Storing data streams from multiple sensors • Control of thrusters and steering through software

Name	Eye in the water - augmented reality immersive experience
Author	AS: Leszek Kaliciak, Hans Myrhaug
Description	
The marine robot's camera system takes wide angle images/videos of the subsea and fuses them together using SURF or SIFT key points detectors and descriptors, or uses the camera geometry itself to stitch the images/videos. In order to get a more immersive experience and to be able to survey the subsea environment in a more natural way, the user wears the virtual reality (VR) goggles. The VR goggles use the underwater 360 degrees camera system of the marine robot and present the user with a real-time view of the surroundings and augment the content by highlighting and tracking moving objects - potential user interests. The user can at any time focus on a particular area of the image and retrieve on-the-fly similar images from the available database. The image retrieval is performed based on the combination of available text and metadata and the visual content. The search results are then displayed in a pop up window and can be further improved by selecting the relevant information objects from the top results. By finding the relevant images in the database of previously known cases, the marine biologist can obtain additional textual information about the surrounding environment as well as the visual data.	
Goal	Augmented reality VR experience
Actor	Marine biologist
System	User, software
Use Case Figure	

 Copyright AmbieSense Ltd
Preconditions
The user wants to enhance the subsea monitoring immersion with the help of virtual reality goggles.
Steps
● The user remotely communicates with the robot ● The user deploys a 360 degrees camera system to a specified depth using the winch ● The 360 degrees camera system presents the user with an underwater real time view of the subsea ● The user wearing VR goggles explores the underwater environment by looking around in a natural way and focuses on a specific area of the 360 degrees image and presses “select” button ● The pop up window presents the user with a set of similar images retrieved from a database, based on the combination of various features

● The user selects the most relevant images and narrows down the search ● The user gets additional visual and textual information about the view that captured his/her interest ● The real-time subsea view is also augmented with highlighted moving objects
Postconditions
The 360 degrees subsea videos can be viewed in the real-time in the virtual reality goggles thus providing enhanced immersive monitoring experience
Related Scenarios
Observing wildlife
Related Requirements
Requirements for ocean monitoring – subsea

Name	Searching for missing vehicles
Author	AS: Leszek Kaliciak, Hans Myrhaug
Description	
The marine robots, which can drift with the ocean current, can be dropped off from the boat to search for missing vehicles. They can be deployed at different locations in the target ocean area, and transmit their location via satellite. The robots can remain adrift for a long period of time, using solar panels to recharge their batteries. They are equipped with cameras and other optional sensors like sonar. One of the cameras monitors the surface surroundings of the robot while the other, the underwater camera system provides an immersive view of the subsea. The marine robot can use a winch to deploy the underwater camera system to a specified depth. The camera system is connected with the marine robot via the umbilical and is specifically designed to operate in poor lighting conditions and to withstand high water pressure. The images and videos are sent wirelessly in parallel to the user's web application via the http protocol. Such design of the marine robot allows for real-time monitoring of the underwater environment including augmenting of the visual content by enhancing the edges and tracking moving objects - potential user interests. The user remotely communicates with the marine robot via his/her laptop, tablet, or smartphone.	
Goal	Localisation of the missing vehicles via real-time ocean monitoring

Actor	User, sensors
System	
Use Case Figure	
 Copyright AmbieSense Ltd	
Preconditions	
The user wants to monitor the surface and subsea ocean environment in order to find the missing vehicle. He is using the surface and immersive underwater camera system specifically designed to operate in poor lighting conditions and to withstand high water pressure. He/she deploys the marine robot and starts the application on his/her smartphone/laptop/tablet.	
Steps	
• The user opens the ocean monitoring app on the smartphone • The app discovers and connects to the system via wireless network • The camera system takes images/videos of the water surface • The images/videos are sent wirelessly in parallel to the user's web application via http • The robot transmits its position and presents the user with the view of the surface environment around the robot • The real-time surface view is augmented with highlighted moving objects and enhanced edges	

● The user deploys the underwater camera system to a specified depth using the winch ● The camera system takes images/videos of the subsea and sends them via umbilical to the vessel ● The images/videos are sent wirelessly in parallel to the user's web application via http ● The camera system presents the user with an underwater real time view of the subsea ● The real-time subsea view is also augmented with highlighted moving objects and enhanced edges
Postconditions
The user is presented with a real-time view of the surface and subsurface environment. The video is augmented with enhanced edges and highlighted moving objects. The robot allows for the constant monitoring of the marine environment thus increasing the chances of finding missing vehicles.
Related Scenarios
Observing wildlife Subsea monitoring
Related Requirements
Requirements for searching for missing vehicles

Name	Storing data streams from multiple sensors
Author	AS: H Myrhaug, L Kaliciak, A Goker
Description	
The marine robot has multiple sensors and cameras used for various purposes, e.g. to capture scenery/ scan surroundings, to navigate, or to detect objects or obstacles. The amount of data generated in parallel from the several sensors therefore implies that multicore processors will be overloaded when writing parallel streams to a hard drive. Additionally, microcontrollers on hard drives will also experience difficulties in writing parallel streams to their file system, simply due to simultaneous concurrent write access. A storage facility capable of writing parallel data streams to multiple hard drives, with diverse data communication topologies, is needed. From his/her smartphone, the user sets the underwater system to capture two streams of 4k spherical videos with additional two photos per second from two other cameras. The video will	

be stitched together by the video streams of four lenses that cover a 360 visual sphere. The user drops the system into the sea in order to later retrieve it as and when it surfaces in 15 minutes. Data streams are recorded and written simultaneously onto separate hard drives through a suitable data communication topology for the current needs. A background program aggregates video streams into spherical videos and stores the results onto another hard drive at 5 Gbits speed, or more. This avoids the data streaming buffers residing in the RAM from becoming oversized, something that will slow down other threads performed by the multi core processor. In parallel to this, two additional lenses capture still images from the surroundings every second. In total, two videos and a set of still images are automatically stored on separate hard drives. An adaptive storage topology can be reconfigured according to the current storage needs of the user by activating separate hard drives for simultaneous use. This can enable energy saving (e.g. energy leakage optimisation). It also shifts computational power away from the multicore processor, and towards communication microcontrollers (e.g. hosts and clients).

When the subsea system returns to the surface, the user is able to play the recorded videos and skim through the captured images from his smartphone app. The data can be streamed in parallel to the user's smartphone from the storage facility, and the user is able to copy/ transfer the video and images of the session onto the storage facility of another robot/ vessel, or to the cloud, whilst performing other tasks in parallel.

Goal	To provide storage facility capable of storing and transfer of sensor data in parallel onto separate hard drives. To utilise high speed data communication topology, at low energy, by activating additional hard drives only when needed.
Actor	User, Sensors
System	Parallel Storage Facility

Use Case Figure

Copyright AmbieSense Ltd

Preconditions

The user needs underwater videos and images from a particular location.

Steps
• The user opens the monitoring app on the smartphone • The app discovers and connects to the underwater system via wireless network • The user opens a new session and adds four camera sensors to use • The system starts the selected camera sensors the user added to the session • The user sets the desired quality of the recordings to be 4k HD, 360 degrees video • The system reconfigures the parallel storage facility to include four parallel SSD drives • The user adds two more cameras, to take pictures every second, to the session • The system reconfigures the storage to include six parallel SSD drives • The user selects the session to last for 15 minutes, and starts the session • The system starts the session and the six cameras start recording • The cameras start streaming video and images onto separate SSD drives in parallel • The user drops the subsea system into the sea, it continues to record multiple video streams and photos as it dives down • The subsea system reappears on the surface after 15 minutes • The smartphone monitoring app reconnects to the subsea system as it surfaces • The user inspects the videos and images of the completed session via the web browser on his smartphone, and copies the session to another robot/ vessel (i.e. target) • The storage facility of the target vessel start receiving the files in parallel • The user is notified by the smartphone app that the session has been transferred, and cross-checks by listing the new content of the target robot/ vessel via the smartphone
Postconditions
The videos and images have been stored in the file system of the separate SSD drives, and transferred in parallel from the local storage facility to a remote storage facility.
Related Scenarios
Subsea monitoring Observing wildlife
Related Requirements
Requirements for storing data streams from multiple sensors

4.6. Scenario: Marine robot propulsion and transport

Title	Marine robot propulsion and transportation
Author	AS: H Myrhaug, L Kaliciak, A Goker
Overview	Transport to remote destination with marine robot

Description

Peter works for a fish farm from the harbour office. The fish farm has 5 clusters of fish tanks, each located between sheltered skerries next to neighbouring islands, between 3-5 km away from the main office. Bags of fish food and other equipment must on a regular basis be transported from the harbour out to the remote fish tanks. Bio-samples also need to be taken for lab tests and staff also need to perform work on the sites. Peter loads cargo into a remote controlled marine robot, closes the cargo doors, unties the robot from the dock, checks the batteries are fully charged, and starts it with his smartphone. He remote-controls it slowly out of the harbour, selects destination, the route, the time to destination, and starts the autopilot. He sends a text message to his colleague at the other destination, that supplies are on their way, and that the robot will arrive in 25 minutes. He then goes on to load the second robot to deliver supplies to another colleague at one of the other sites, and repeats the task. While he is doing this he is able to see video from the first marine robot, on his smartphone: The sea is smooth and he can see that the robot is on its right way. He concludes the first vehicle will be able to return and still have 20% of energy left in the battery module. If the battery level drops below 10%, the battery readjusts its Voltage to guarantee that digital equipment can still operate normally, whereas the autopilot sets the propulsion to standby. It is also possible for Peter to manually set the propulsion to standby and select the desired battery Voltage.

Issues

- The wireless remote control signals will need to reach several kilometres
- Video and data need to be communicated between the operator and marine robot
- If the wireless connection is lost, the vehicle must be able to self operate to some extent

Benefits

- Business: Saving time and money by no longer needing person to onboard
- Work tasks: The work place can focus on other tasks than transportation
- Environment: The marine robots use electric engines for propulsion

Needed/ derived use cases

- Reconfiguration of battery module in runtime from a smartphone

- Control of thrusters and steering through software
- Set the course and forget - autopilot
- Remote control of marine robot from smartphone

Name	Reconfiguration of battery module in runtime
Author	AS: H Myrhaug, L Kaliciak, A Goker
Description	
In terms of energy supply from batteries, there are two main categories of equipment. One is digital equipment, like sensors and processors, which require relatively little but a fairly constant supply of power with a certain voltage. This equipment, that constitutes the mind and senses of the vehicle, ideally needs power supply as long as possible so that the vehicle can communicate information with the user and other remote equipment, even if the vehicle is temporarily unable to navigate or propel itself at sea. The other equipment category, the motoric aspects, are actuators needed to drive the vehicle, which normally need much more power supply than the first category, and as such will empty the batteries faster. A single battery module comprises multiple battery cells, each with a certain voltage and milliAmpere hours when fully charged. A battery cell topology consists of battery cells connected together serially (strings) and in parallel. When the remaining power of the battery module reaches a critically low level, there is a chance that due to the gradually decreased voltage levels, the digital equipment will mal-function or stop completely. Also, if a battery cell string contains a cell that is almost empty, the marine robot can experience a sudden drop in voltage because of this “weak” link. If the battery cell or module becomes 100% discharged, the number of possible recharge cycles can even drop to half, e.g. from 8,000 to 4,000, which translates into a reduced battery lifetime down to half. An adaptive reconfiguration of the topology can therefore obtain more power from the overall battery module and extend the battery capabilities, and also ensure and prolong its lifetime. If a battery cell string continues with the same topology, it could even be rendered useless, despite there being sufficient power and voltage in total. In order to ensure/ achieve an extended operational time of mission critical digital equipment, the possibility to reconfigure the wiring of the battery cells within the battery module through software will make the mind and the senses of the vehicle more resilient. Both human operator (through smartphone app interaction), and the autopilots (through high-level software reconfiguration), should be able to readjust the battery module’s internal connectivity diagram, through a software interface. Thus, the topology of the battery module can be changed dynamically through reconfiguration, connecting the battery cells in series or in parallel, or a mixture of both, in order to obtain a certain voltage level. The software that controls the battery module needs to be able to monitor the power levels and output voltage of each cell, and it would also be beneficial to monitor the temperature of the battery module, or individual cells, in order to detect and isolate faulty thermal runaway cells.	

Goal	To ensure that mission-critical digital equipment still provide certain Voltage level through software reconfiguration as battery module levels drop below certain thresholds. To research battery modules that can adapt its cell topology to better fit the energy needs of the cyber-physical system, and to enable self-repairs and higher safety by eliminating defective cells through reconfiguration.
Actor	User, Software
System	Adaptive, Reconfigurable Battery Module (Lithium Ion based)

Use Case Figure

Copyright AmbieSense Ltd

Preconditions

Battery module comprising 256 battery cells providing 12V when 100% charged has dropped down to 20% of remaining power. The cell series are providing a critically low voltage of 3V and about to steadily drop further. Thus, some sensors and digital equipment will shortly stop working unless a higher voltage is provided

Steps

- The user starts the application on the smartphone
- The application discovers, connects to the marine robot, and requests subscription of battery module notifications
- The autopilot of the marine robot starts forwarding battery module notifications to the smartphone application via Internet as and when they occur

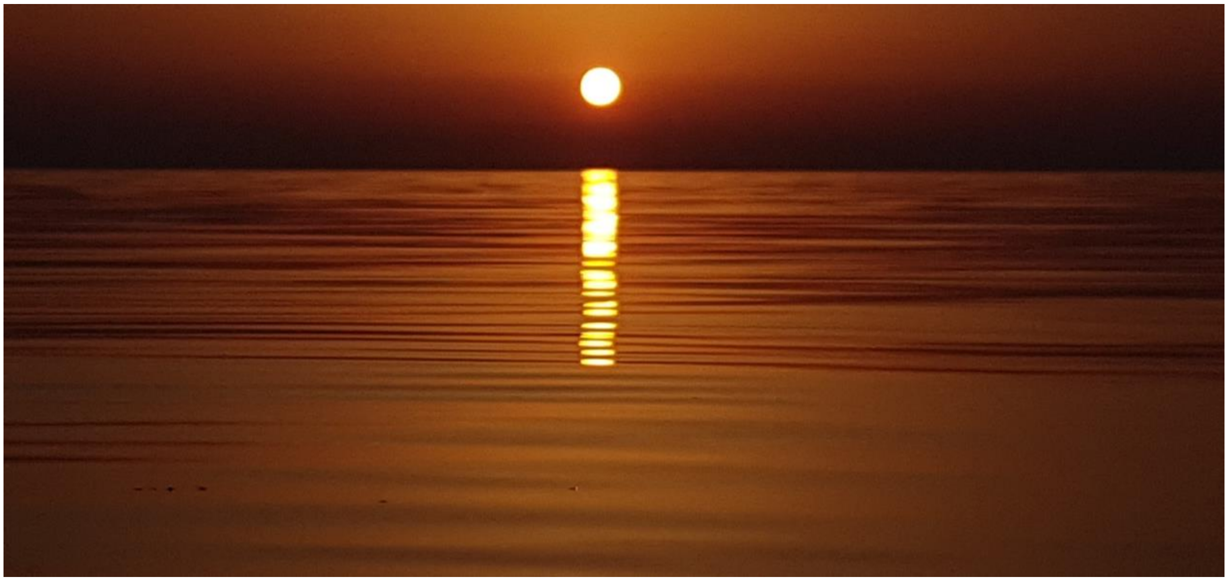
- Both the smartphone application and the autopilot receives a message of 20% energy levels with 3V delivery from the battery module   - The autopilot responds by setting electric propulsion to standby, and notifies the user via the smartphone application   - The autopilot software sends a reconfiguration request to the battery module for 12V   - The battery module software reconfigures its internal cell topology to meet 12V   - The battery module returns a message to the autopilot of successful change   - The autopilot forwards the message to the user's smartphone application about the successful reconfiguration of the battery
Postconditions
The internal connectivity diagram (topology) of the 256 cell battery module has been reconfigured to a new cell topology. The battery module delivers close to 12V: the battery module has switched to a cell topology with fewer but longer cell series. The autopilot has set propulsion to standby, awaiting the battery module to be more fully charged, before further action. Digital equipment has gained first priority in using remaining battery power
Related Scenarios
Observing wildlife Subsea monitoring Marine robots for transportation
Related Requirements
Requirements for reconfiguration of battery module in runtime

Name	Control of thrusters and steering through software
Author	AS: H Myrhaug, L Kaliciak, A Goker
Description	
The marine robots will be operated and navigated by up to several electrical engines that provide safe and clean propulsion, with ultra-low noise levels and vibration. Engines are integrated through electronics and communication. Electrical cables distribute the energy from the batteries to obtain the desired propulsion and steering. Each of the engines are controlled by the propulsion and steering software that in turn is controlled either manually via a user interface by the human operator, or automatically by an auto-pilot. The thrust and steering	

components are operated digitally. The vehicles are considered purely electrical if the battery charging is performed by using electrical charging only, such as when using solar energy, whereas they are considered hybrid if for instance diesel generators are being used to charge the batteries. The advantage of deploying electric or hybrid propulsion systems is that there is no longer need for neither a complex gearbox nor heavy combustion engines. Instead, the propulsion is obtained by setting the speed of the engine by increasing or decreasing its input power. Reverse propulsion is achieved by simply reversing the direction of the electric current. Air intake and exhaust parts also become obsolete in the case of purely electrical vehicles based on solar charging, energy harvesting from surface, or overnight shore hook-up whilst the vehicle is in the port. In the case of hybrid vehicles, where power/ turbine generators are used to recharge the battery modules, e.g. with diesel or gas, the generators can be located away from the engines, which provides more flexibility for the designer of the marine robot. The recharging systems therefore become independent and interchangeable parts by switching to electrical engines. Advanced thrusters can deliver push along a rotating axis, and thus operate as both propeller and steer at the same time. On the other hand, if a thruster has a fixed direction, then additional rudders or fins would be needed to orientate and ensure the direction and stability of the marine robot. Thus, steers, rudders, or foils can be independently controlled by electrical propulsion engines also through software. A vehicle can also be rotated if two propellers are pointing in the same direction by utilising variable thrust. Rudders and foils can function by electromagnetic or hydraulic actuation. Thus, each actuator must be operated through individual microcontrollers that communicate with the thrust and steering software with multi-threading capabilities. Such high-level control allows both human operator and auto-pilot to monitor and set speed, angles, and direction of the interconnected and networked actuators.	
Goal	To enable the user and the autopilot to operate and navigate the marine robot by using electrical engines
Actor	User, autopilot
System	Propulsion and steering software
Use Case Figure	

Copyright AmbieSense Ltd
Preconditions
The marine robot is docked, or in standby mode
Steps
● The user unties the marine robot from its dock ● The user starts the engine by pressing the start button on the screen ● The propulsion and steering software starts and sets the propulsion to neutral ● The user selects throttle operators available and selects forward mode ● The software start the thrusters ● The user increases the forward throttle values with the buttons ● The system increases the amount of power supplied to the thrusters, the marine robot starts to move forward on the surface ● The user increases the speed further and makes a turn ● The thrusters and steering software responds by turning the vehicle in the direction requested by the user ● The user manually changes the course of the marine robot and sets the speed to maximum forward ● The system updates by adjusting the thrusters and steering electronically ● The user slows down and sets the operation in neutral mode ● The thrusters stop operating ● The user sets the throttle in reverse mode and increases the thrust ● The system responds by reversing the marine robot on the surface ● The user navigates the robot to shore, stops the propulsion, and ties, connects the marine robot to the charger
Postconditions
The marine robot has been navigated by the user according to a preferred route and direction through interaction with screen, steering and throttles
Related Scenarios
Observing wildlife Marine robots for transportation
Related Requirements
Requirements for control of thrusters and steering through software

Name	Set the course and forget – autopilot
Author	AS: L Kaliciak, H Myrhaug, A Goker
Description	
The user, a marine biologist, wants to study the whale population in the particular area of the ocean. S/he deploys the marine robots from the shore and sets their course for the chosen location by selecting the destination on an electronic map. Using their self-navigation capabilities, the robots arrive at the specified site. The marine biologist can keep track of the current location of the autonomous vehicle at all times and can take full control it by using his/her computer system: laptop, tablet, smartphone. The marine vessel uses multiple sensors for navigation, maintaining its balance, and adapting its direction and propulsion. The most important ones are gyroscope, compass, and gps, with computer vision for optional short distance object detection and avoidance. The user can also select a detailed path to the destination, and control auto-pilot settings such as: the home location to return to in case of emergency, the time it should take to arrive at the target destination, or the desired speed. The particular navigation algorithm can be interchanged. The algorithm could be to adjust and keep the direction until the selected destination has been obtained, trajectory prediction using a kinematic model, drifting towards location, and so forth. The autopilot capability of the marine robot will be useful especially when no wireless signal is available for remote control. When the wireless communication is back, the user should be able to manually retake control of the marine robot again at any point in time.	
Goal	To enable the marine robot to autonomously reach a chosen target destination/ geocoordinate, optionally by also selecting a detailed path plan/ trajectory.
Actor	User
System	Adaptive autopilot
Use Case Figure	



Copyright AmbieSense Ltd

Preconditions
The marine robot needs to get from point A to point B, avoiding obstacles in its path

Steps

- The user connects and starts the marine robot from his smartphone
- The robot starts up in idle mode
- The user deploys the marine robot from a shore or a boat
- Using tablet or smartphone the user selects a target destination on a map and
- The user can optionally select:
  - A path comprising up to several points
  - The user sets the desired speed
  - home location
- The robot updates its settings
- The user selects the autopilot mode
- The marine robot initiates and starts the autopilot
- The autopilot drives the robot towards the target destination at selected speed
  - The autopilot bases the navigation primarily on incoming sensor data from gps, compass, and the gyroscope
  - Based on the sensed data, the autopilot recurrently adjusts steering and propulsion to fit the direction and speed towards the target destination with a frequent interval
- The user can see the live video feed from the vehicle on his/ her smartphone

• The vehicle reaches the destination and the user is notified by the autopilot • The user selects return home • The robot starts returning home towards the user
Postconditions
The marine robot has reached the target destination though self navigation
Related Scenarios
Observing wildlife Marine robot propulsion and transport
Related Requirements
Requirements for autopilot – navigation

Name	Remote control of marine robot
Author	AS: H Myrhaug, L Kaliciak, A Goker
Description	
The user has just bought a marine robot and transported it to the local marina where he will dock it. The local marina has a charging point where he can recharge its batteries while he is not using it. The marina also offers a wireless network. He will use the robot for transportation and recreational purposes. He likes the possibility to remote control it securely, and to be able to log on to see its status via the Internet, as this makes it convenient to check its status from the office or from home. In order to securely operate the cyber physical system by remote control, he will need to pair his vehicle and his smart phone app with the provided wireless key fobs provided for the individual vehicle. He installs the application on his smartphone and is able to pair the vehicle keys for remote control with his smartphone app securely. Only smartphone apps that have been paired with the remote control keys are able connect to and operate the individual vehicle. In this way, apart from using secure network protocols to establish communication, remote control and operations can happen within areas with mobile or satellite network coverage. This avoids dependencies to traditional RF-remote controllers that are governed by license limitations on maximum allowed signal strength, at the same time as it utilises state-of-the-art secure wireless communication. Every time the user want to check the status of the robot from remote, he starts the smartphone app and taps the key fob onto his phone to enable and wirelessly authenticate remote access and control of the data and equipment onboard. The user can navigate the robot out the harbour area by using propulsion	

throttle and steering via the smartphone.	
Goal	To enable remote control and access to the cyber physical system. To enable delivery of goods to and from docking without the operator being onboard the vehicle during transport. To allow the autopilot to take full control if the wireless signals/ connection becomes unavailable.
Actor	User, Propulsion and steering software, Autopilot
System	Smartphone app, Marine robot, Wireless key fobs
Use Case Figure	
 Copyright AmbieSense Ltd	
Preconditions	
The user has acquired a marine robot and would like to remote control it.	
Steps	
● The user installs the smartphone app for remote control and access ● The user opens the app and registers ● The app creates a user profile and records the user details ● The app prompts the user to tap phone to the marine robots key sign ● The user taps phone to the marine robot to pair the app with the robot ● The app registers and pairs up with the robot ● The app prompts the user to tap the phone onto one of the provided key fobs ● The user taps the phone onto a key fob	

• The app wirelessly connects to the marine robot and presents the remote control pad to the user • The app asks the user if the current location is to be the default location/ home harbour to return to • The user answers yes • The user unties the robot from its dock • The user starts the engines and reverses it out of the dock by the remote control pad • The app asks if the autopilot should be started with a user-defined destination • The user answers no and continues to remote control the robot manually towards a destination across the fjord by looking at the live video feed of the front camera of the robot on his mobile and adjusting the remote controls • The robot arrives near the opposite shore, slows down and stops the vehicle • The user adds the location to the list of locations for the vehicle to remember • The user selects return to home
Postconditions
The user has securely connected and started the vehicle, and has remote controlled it from its home dock to its first destination to remember. It was accomplished by the user watching the live video feed from the front camera of the vehicle and by using remote control pad.
Related Scenarios
Marine robots for transportation Observing wildlife
Related Requirements
Requirements for remote control of marine robot

5. Use Case Requirements Methodology

In this section, we explain the conventions and the template that have been followed by each use case provider to collect and make available their own list of requirements (provided in Sections 6 to 8 of this document).

5.1. Use Case Requirements conventions

When the phrases MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, or OPTIONAL are being used respectively within the short name and description fields of the requirements template, these have the following meaning and interpretations:

- MUST, REQUIRED, SHALL - means an absolute requirement for the specification.
- MUST NOT, SHALL NOT - means an absolute prohibition for the specification.
- SHOULD, RECOMMENDED - valid reasons may exist in certain cases to avoid the requirement. However, its full implication must be have been understood and carefully considered before choosing an alternative direction/ option.
- SHOULD NOT, NOT RECOMMENDED - valid reasons may exist when the requirement could be acceptable or even useful. However, its full implications should have been carefully considered understood and prior to implementing anything in breach of this.
- MAY, OPTIONAL - simply means that the requirement is truly optional, nice to have. An implementation that does not fulfil an optional requirement MUST be prepared to function together with another implementation that fulfils/implements this option, and vice versa.

5.2. Use Case Requirements template

The fields of the requirements template defined for CERBERO (and shown in Table 2) have the following meaning:

- **Req ID:** A unique identifier of the requirements, identifying the partner that made it and a number of the requirement
- **Short name:** A short sentence/ title reflecting the requirement, formulated in line with the provided requirements convention
- **Aspect:** Are considered high-level groupings of cross-cutting concerns (called categories within CERBERO). Also see: [CPS Framework R1.0].
- **Category:** Concerns (or categories) are interests in a system relevant to one or more project stakeholders (see: [CPS Framework R1.0]). For definition of the requirements the most relevant values for Aspect and Category need to be selected from the "Requirements Categories" table in section 5.3, e.g. value:

"Functional, User Task". A concern can be seen as another word for a requirements category.

- **Priority:** The possible values are: 1 (i.e. to be implemented in the first project half); 2 (i.e. to be implemented in the second project half); 3 (i.e. less likely to be implemented during the project, nevertheless important enough to mention)
- **Use case / scenario**: The scenario(s) and/or use case (s) this requirement has been associated with/ elicited from
- **Description:** Clear sentence(s) that concisely communicate this requirement, formulated in line with the provided requirements convention. A rationale, or reason is also beneficial to include to explain a bit more in detail as to why the requirement need to be met
- **Verification:** A fit criterion needed for this requirement to have been met, if applicable
- **Conflicts:** Indicates a possible conflict with other requirements. If a conflict is identified, it might as well be that the two requirements need to be reformulated more precisely
- **Use Cases**: The use case(s) to which this requirement is related to/ elicited from.
- **Additional info:** References to supporting material, if applicable. This could for instance be scientific material, videos, books, other sources, and so forth to help clarify for design and implementation

Table 2 – Requirement Template

Req ID:	*<partner>-<req. number>*
Short name:	*<short (max one sentence) description>*
Aspect:	*<aspect (see CPS framework)>*
Category:	*<category / concern (see CPS framework)>*
Priority:	*<priority of use case>*
Use case / Scenario:	*<use case(s) / scenario(s) for which requirement is applicable>*
Description:	*<description of requirement>* Rationale: *<rationale>*

Verification:	*<fit criterion>*
Conflicts:	*<any known conflicts between defined requirements>*
Additional info:	*<additional info, links to docs and other background material>*
Version:	*<version number of requirement>*

5.3. Use Case Requirements categories

In terms of requirements categories used as means to classify each requirement, CERBERO has adopted and extended the requirements categories from the standard [CPS Framework R1.0], defined in the National Institute of Standards and Technologies (NIST) CPS Public Working Group. A concern (referred to as “category” within the CERBERO project) is specific for a use case, whereas if that concern cuts across several use cases, it becomes an aspect. Aspects are in particularly useful to try to identify as they are good indicators for generic requirements across use cases, and help with adaptation, improvement, and reuse of technologies, components, or tools across use cases.

The categories proposed to be used within CERBERO (based on the Aspect-Concern table used in [CPS Framework R1.0]) are listed in Table 3.

Table 3 – CERBERO requirements categories.

Requirements Categories		
Aspect	**Category (= Concern)**	**Description**
Business	**Enterprise**	Is this a concern related to economic aspects of the CPS though their lifecycle?
Business	**Cost**	Concerns related to direct or indirect monetary flow.
Business	**Environment**	Is this a concern related to the impact of the CPS system on the environment and vice versa?
Business	**Policy**	Is this related to some policy or regulation?
Business	**Time to market**	Is this concern about bringing the CPS products to

		market faster?
Boundaries	**Behavioural**	Does the use case require a system that crosses multiple application areas? If so, which application areas are included?
Boundaries	**Networkability**	Is the concern about the need for interconnectivity through network communication?
Boundaries	**Physical Location**	What are the location requirements of the use case?
Composability	**Reconfigurabilit y**	Is the concern about the need to upgrade or reconfigure a CPS to meet new conditions, needs, goals, objectives?
Composability	**Adaptivity**	Is the concern related to the ability of the CPS to adapt its behaviour to the current situation that it resides/ operates within, to achieve goals, be better adapted to the task at hand, or similar?
Composability	**Control**	Is the concern about how the CPS system components are being controlled/ interfaced/ reconfigured by software programs?
Composability	**Sensors, Actuators**	Is the concern about the need for using specific sensors and/ or actuators to realise the CPS system, make it work?
Composability	**Energy**	Is the concern a about supplying the CPS system with energy?
Composability	**Mechanical/ physical design**	Is the concern about the physical parts, components of the system?
User	**Human factors**	Is this concern about how needed system characteristics seen from the user / operator?
User	**Usability**	Does the system need to meet certain usability criteria, e.g. task completion time, performance, subjective satisfaction measures?

User	**Look and Feel**	Does the user interface need certain look and feel aspects?
User	**Interaction**	The cognitive workload, ease of use, how to interact with the system?
Data	**Volume and Velocity**	Describe the size of the datasets being processed and the speed at which they come into/ out of the system.
Data	**Aggregation**	Describe the requirements to aggregate different data types
Data	**Variability**	Is the size of data being generated/used consistent or is there a growth/ shrinkage trend?
Data	**Fusion**	Is the concern related to the need for data fusion methods?
Functional	**Physical Properties**	What physical properties are being monitored, or acted upon?
Functional	**Computation**	Describe the computation effort and processing required to achieve the use case goals.
Functional	**Error Sensitivity**	Describe the sensitivity of the system to errors in the data
Functional	**Certainty**	What is the level of uncertainty in the data being generated/processed and the assurance of the resulting actions taken by the system?
Functional	**User task**	Is the requirement describing a specific task that the user must be able to perform when using the system?
Functional	**System task**	Is the requirement describing a specific system method/function that must be present for the system to be able perform a task/accomplish a goal?
Timing	**Timeliness**	What are the use case timing constraints?
Timing	**Time Synchronization**	What are the use case time synchronization requirements? Must the CPS system synchronise time

		between themselves to operate together?
Trustworthiness	**Robustness**	What are the robustness requirements? (preventing a fault)
Trustworthiness	**Resilience**	What are the resiliency requirements? (recovering from a fault or sub-fault)
Trustworthiness	**Confidentiality**	What happens if information within the system leaks (or is pulled) out?
Trustworthiness	**Integrity**	What happens if the system acts on incorrect data (including software)?
Trustworthiness	**Availability**	What happens if the system or data it generates is not accessible and prepared to function properly when and where needed?

6. Detailed Requirements for Smart Travelling

In this chapter the technical requirements are included for the use case Smart Travelling for Electric Vehicles.

Req ID:	TNO-001
Short name:	Integration of simulation modules via a system in the loop mechanism.
Aspect:	Functional
Category:	System Task
Priority:	1
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	Easy integration of (new) simulation modules into the overall car simulator. Input and output of integrated module MUST be linked to the existing simulation software. The coupling of the car simulator – considered here as a real system – with a model simulator with system-in-the-loop capabilities SHOULD enable the addition of new client simulation modules without the need to modify or recompile the system (car-simulator) framework software. Rationale: The car simulator is often re-used to test and validate the integration of different driver interfaces and automotive sensors (scenarios). Each scenario involves the simulation of different sets of models and often integration of new ones. This makes it important to be able to include new functionality with minimal effort. The CERBERO methodology and tooling MUST help in creating reference processes for the description, use, and integration of simulation models, such that the phase of preparing a simulation experiment becomes more efficient.
Verification:	Simulation modules MUST execute in parallel with the already existing car simulator software, resulting in successful simulation runs. Adding new modules SHOULD NOT require a modification or recompilation of the whole simulation framework.

Conflicts:	*<any known conflicts between defined requirements>*
Additional info:	<additional info, links to docs and other background material>
Version:	1

Req ID:	TNO-002
Short name:	Updating of simulation modules
Aspect:	Composability
Category:	Reconfigurability
Priority:	1
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	Ability to update existing modules in the simulator with new versions. The simulation framework SHOULD support the update of existing client simulation modules with newer versions without the need to modify or recompile the simulation framework software. Rationale: It SHOULD be possible to update new versions of simulation modules so new or updated designs of modules can be tested and verified with the car simulator.
Verification:	Simulation modules can be replaced with new version and execution of update can be tested by running successful simulations.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-003
Short name:	Synchronization of time between different simulation modules and simulation modules running in different machines.
Aspect:	Timing
Category:	Time Synchronisation
Priority:	1
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	The car simulator concurrently runs different models (power train, kinematics, user interface, etc.) at different machines, which are connected over a network. The local time used by the different simulation model MUST be synchronized continuously for correctness and logging purposes, where the accuracy of synchronization SHOULD be based on a clock frequency of at least 1kHz. The simulation framework MUST provide a synchronization mechanisms that allows client simulation modules to synchronize themselves with the common system time with an accuracy of 1ms. Rationale: For the simulation it is important that all the modules are running in real time and clocks are synchronized so they also register same time.
Verification:	From the logging of the individual modules it SHOULD be possible to verify that clocks of individual modules were synchronized during the simulation.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-004

Short name:	Integral and synchronous logging of module behaviour
Aspect:	Timing / Functional
Category:	Timeliness / System task
Priority:	1
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	The logging of individual simulation modules SHOULD be synchronized and integrated in such way that it is possible to correlate logging messages of different simulation modules. The logging operation is not synchronized, but the data that is logged is time-tagged, such that logging data of different client simulation modules can be correlated offline after the simulation run. Rationale: To analyse results of simulations it is important to be able to correlate actions and behaviour of the individual simulation modules.
Verification:	Logging messages of different modules MUST be correlated using integral logging mechanism.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-005
Short name:	Simulation with system-in-the-loop SHOULD support a model sampling rate accuracy up to up to 1 ms.
Aspect:	Functional / Timing
Category:	System task / Time Synchronisation

Priority:	2
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	Simulators with system-in-the-loop capabilities SHOULD be able to cope with the usual data rate needed in the real system. So for example, the virtual driving platform used in this use case exchanges and produces data that have to be sampled and processed in rates up to 1Khz. This sampling rate accuracy SHOULD be guaranteed by the system-in-the-loop able simulator such that the concurrent simulation of models with a real system can produce timely correct results. <u>Rationale</u>: The real value of a system-in-the-loop simulation is that parts of the system are simulated, but run concurrently with real physical components. The correct time responsiveness of the overall simulation and system can only be reproduced if the system-in-the-loop simulation system can support the sampling rates required by the real part of the system.
Verification:	Measured accuracy of time critical simulation modules and of the system-in-the-loop able simulator SHOULD be in the range of 1 ms.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-006
Short name:	Integration of prediction modules
Aspect:	Composability
Category:	Control
Priority:	2

Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	It SHOULD be possible to include predictions modules able to simulate the future. The predicted outcome of these modules will be used in the real time simulation to propose one of more alternatives. The future simulations MUST run at a much higher speed than real time simulation (time MUST be accelerated by a factor at least 100 or higher). Rationale: During simulation, predictions are required in case driver is to be helped with selection of alternatives (e.g. for routes to taken and/or charging options for loading the battery). Apart from real time running simulation modules, some modules could be run at much higher speed so they can be used to calculate results of specific alternative.
Verification:	During simulation, prediction modules can be used to calculate predicted results of different alternatives.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-009
Short name:	Tag specific events during or after the simulation.
Aspect:	Timing
Category:	Time Synchronisation
Priority:	
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	It MUST be possible to tag specific events during (or optionally after) the simulation. Tagging SHOULD indicate specific event and link to specific

	moment in time (which could be single moment or event with specific duration). The tag MUST also be linked to the logging data produced by the different simulation modules. The simulation framework MUST support a client simulation module to add tagged events to the log data. Rationale: In order to evaluate simulation runs it is important that specific events happened during the simulation can be identified and linked to activities in the different simulation modules. It is therefore important to be able to tag the data generated by the different simulation modules and possible combine logs and activities related to specific tags.
Verification:	During a simulation a specific event MUST be indicated with a tag. It MUST be possible to find the tag in the resulting logging data after completion of the simulation and relate the event to activities which occurred in the different simulation modules.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-010
Short name:	Automatically generate 3D map data.
Aspect:	Composability
Category:	Reconfigurability
Priority:	3
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	It SHOULD be possible to automatically generate 3D map data to support creation of new use cases. The 3D data could be generated based on existing map data (like Google Maps or OpenStreetMaps).

	Rationale: The simulator SHOULD support many different use cases, for which new 3D map data MAY be required. It will be important to reduce the current manual effort and be able to (partly) automate the conversion of existing map data into data which can be used by the simulator.
Verification:	New data is generated based on existing Google Maps or OpenStreetMaps data. The generated map data is used in one or more simulation use cases.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-011
Short name:	Integration of TNO battery model in existing simulator.
Aspect:	Composability
Category:	Control
Priority:	1
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	The TNO battery model SHOULD be integrated into the existing driving simulator. The TNO battery model SHALL be implemented as a separate client simulation model. Rationale: CRF does not have battery models they can integrate into driving simulator. This would be needed to simulate an electric vehicle.
Verification:	Driving simulator will be run and tested to simulate battery of the simulated EV.

Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-012
Short name:	Integration of TNO electric motor model in existing simulator.
Aspect:	Composability
Category:	Control
Priority:	1
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	The TNO battery model SHOULD be integrated into the existing driving simulator. The TNO electrical engine model SHALL be implemented as a separate client simulation model. Rationale: *<rationale>*
Verification:	Driving simulator will be run and tested to simulate electric motor of the simulated EV.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-013
Short name:	Real time simulation

Aspect:	Timing
Category:	Timeliness
Priority:	1
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	The performance of the simulation framework and client simulation modules SHOULD be such that real time vehicle behaviour is simulated. Driving simulator SHOULD thus run driving scenario in real time. The use/driver SHOULD experience scenarios as real time scenarios (thus proving real time behaviour similar to driving a real car). Rationale: The driving simulator is used to collect information of driver and therefore the simulation SHOULD mimic real experience as much as possible (including the speed of interactions).
Verification:	The performance of the simulation framework and client simulation modules SHOULD be such that real time vehicle behaviour is simulated. Driving simulator SHOULD thus run driving scenario in real time. The use/driver SHOULD experience scenarios as real time scenarios (thus proving real time behaviour similar to driving a real car).
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-015
Short name:	It SHOULD be possible to modify use case or create new use cases easily.
Aspect:	Composability
Category:	Adaptivity

Priority:	2
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	In order to test specific use cases it MUST be possible to update, change or create new use cases, e.g. by composing a set of available modules and related data. A use case is comprised of 3D map data, car simulation modules and possibly a set goal for the driver. Rationale: The driving simulator is used to test and validate driver behaviour. To test different use cases it is important that use cases can be adapted and updated in a simple and fast way.
Verification:	
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-016
Short name:	Simulation modules MUST be reusable for many use cases.
Aspect:	Composability
Category:	Reconfigurability
Priority:	2
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	It MUST be possible to use the simulation modules in many different use cases (within model’s functional limits). The interfaces of the modules used should therefore be as generic as possible.

	Rationale: To minimise development cost, it is important that modules can be used in many different use cases (as long as module will fit the specific use case functionalities).
Verification:	Simulation modules can be used in many different use cases and new use cases can be created using existing (or possibly new) simulation modules.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-017
Short name:	Simulation modules MUST be able to use input from other simulation modules during simulation.
Aspect:	Composability
Category:	Control
Priority:	
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	The simulation modules MUST be able to collect input from other modules in order to simulate required interactions. The frequency of interaction MUST be able to go up to 1kHz (depending on the need for the specific modules to interact). Rationale: Simulation modules will have dependencies with other modules and therefore need input from the other modules during the simulation process.
Verification:	During execution simulation modules MUST use input from other modules in order to generate proper simulated behaviour.

Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-018
Short name:	Simulation modules MUST be able to produce output towards other simulation modules during simulation.
Aspect:	Composability
Category:	Control
Priority:	1
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	The simulation modules MUST be able to provide output towards other modules in order to simulate required interactions. The frequency of interaction MUST be able to go up to 1kHz (depending on the need for the specific modules to interact). Rationale: Simulation modules will have dependencies with other modules and therefore need to be able to provide their output towards other modules during the simulation process.
Verification:	During execution simulation modules will use output from other modules in order to generate proper simulated behaviour.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-BATTERY-019
Short name:	The TNO battery model SHOULD model the batteries performance with sufficient accuracy
Aspect:	Functional
Category:	System task
Priority:	
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	The TNO battery model SHOULD model the performance (voltage, temperature and SoC response to current/power request) and incorporate its own limitations. During driving the following accuracies are to be met: • Voltage RMS error <10mV • SoC RMS error <2% • Temperature RMS error <5K Rationale: The function of the battery model is to model the performance of the battery based on some required inputs and providing useful outputs to the vehicle model.
Verification:	The TNO battery model SHOULD be separately verified with realistic test data (automotive drive cycles applied to a battery cell in TNO's testing facilities).
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO- BATTERY-020
Short name:	The TNO battery model SHOULD be provided with the required inputs at

	a sampling rate of at least 1Hz.
Aspect:	Data / Timing
Category:	Volume and Velocity / Timeliness
Priority:	2
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	Requested battery power (by the vehicle (model)), as well as thermal condition (coolant/air temperature and flow) SHOULD be provided to the TNO battery model with a sample rate of at least 1Hz. Rationale: The function of the battery model is to model the performance of the battery based on some required inputs and providing useful outputs to the vehicle model.
Verification:	The TNO battery model will run as part of the vehicle simulator, which handles the models IO.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO- BATTERY-021
Short name:	The TNO battery model SHOULD provide the required outputs at a sampling rate of at least 1Hz.
Aspect:	Data / Timing
Category:	Volume and Velocity / Timeliness
Priority:	2

Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	Actual battery power, voltage, temperature and SoC SHOULD be provided to the vehicle model with a sample rate of at least 1 Hz. Rationale: The function of the battery model is to model the performance of the battery based on some required inputs and providing useful outputs to the vehicle model.
Verification:	The TNO battery model will run as part of the vehicle simulator, which handles the models IO.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO- BATTERY-022
Short name:	Battery cells SHOULD be provided for parameter identification
Aspect:	Composability
Category:	Mechanical/ physical design
Priority:	2
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	Three or more battery cells SHOULD be provided to TNO, to be able to test them and identify cell level model parameters. These cells SHOULD be of the same type as the ones in the vehicle that is being referenced to. Rationale: By testing actual cells, the best battery model accuracy can be realised in the vehicle simulator. This circumvents the need for rescaling other

	battery model parameters to fit the vehicle simulated in this simulator.
Verification:	Cells will be tested in TNO's battery test setup by which model parameters can be identified.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-MOTOR-023
Short name:	The TNO electric motor model SHOULD model the motors performance with sufficient accuracy.
Aspect:	Functional
Category:	System task
Priority:	2
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	The TNO electric motor model SHOULD model the performance (speed, electrical and mechanical power, temperature, etc.) and incorporate its own limitations. The following accuracies are to be met: • Torque RMS error <10% of the nominal Torque • Temperature RMS error <5K
Verification:	The TNO electric motor model will run as part of the vehicle simulator and will be validated with data from the real vehicle (isolating the electric motor as a subcomponent).
Conflicts:	
Additional	

info:	
Version:	1

Req ID:	TNO- MOTOR-024
Short name:	The TNO electric motor model SHOULD be provided with the required inputs at a sampling rate of at least 1Hz.
Aspect:	Functional / Timing
Category:	System task / Timeliness
Priority:	2
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	Requested electric motor power (by the vehicle(model)), as well as thermal condition (coolant/air temperature and flow) SHOULD be provided to the TNO electric motor model with a sample rate of at least 1Hz. Rationale: The function of the electric motor model is to model the performance of the electric motor based on some required inputs and providing useful outputs to the vehicle model.
Verification:	The TNO electric motor model will run as part of the vehicle simulator, which handles the models IO.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-BATTERY-025

Short name:	The TNO battery model SHOULD provide the required outputs at a sampling rate of at least 1Hz.
Aspect:	Functional / Timing
Category:	System task / Timeliness
Priority:	2
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	Actual electric motor power (electrical and mechanical), speed, temperature, etc. SHOULD be provided to the vehicle model with a sample rate of at least 1 Hz. Rationale: The function of the electric motor model is to model the performance of the electric motor based on some required inputs and providing useful outputs to the vehicle model.
Verification:	The TNO electric motor model will run as part of the vehicle simulator, which handles the models IO.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-026
Short name:	The simulation framework SHALL support the configuration of input and output data flows between the client simulation modules.
Aspect:	Composability
Category:	Reconfigurability
Priority:	2

Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	The simulation framework SHALL support the configuration of input and output data flows between the client simulation modules according to required semantics. <u>Rationale</u>: To create or update use cases, new simulation modules will need to be added and interfaces between interfaces need to be created or adapted. It is therefore important to have functionality which supports configuration of the input and output of simulation modules.
Verification:	It will be possible the configuration of input and output of data flows between the client simulation modules.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-027
Short name:	Simulation framework MUST support the cooperation of (instances of) client simulation modules.
Aspect:	Composability
Category:	Control
Priority:	1
Use case / Scenario:	<*u* Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	The simulation framework MUST support the cooperation of (instances of) client simulation modules (as in parallel execution to reduce the module calculation time).

	Rationale: To reduce execution time, it could be required to run multiple (instances of) simulation modules in parallel. To be able to share data during the parallel execution, it will be needed that the modules cooperate (by for example sharing or transferring data with other modules).
Verification:	It will be possible to run multiple (instances of) simulation modules, where modules need to cooperate (e.g. share data).
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-028
Short name:	Use existing driver simulator to test Driver Support interface of Electrical Vehicles.
Aspect:	Business
Category:	Enterprise
Priority:	1
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	The driving simulator of CRF is to be used to test Driver Support interface in an Electrical Vehicle. Rationale: The current driving simulator does not support EVs and SHOULD therefore be adapted and extended to provide required functionalities needed for testing Driver Support interface.
Verification:	The driving simulator SHOULD a simulate EV which can be equipped with Driver Support functionalities.

Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-029
Short name:	Advice for most optimal route (based on predictions).
Aspect:	Functional
Category:	System task
Priority:	1
Use case / Scenario:	Smart Travelling / Highway route re-planned, Emergency call, Unexpected congestion
Description:	The driver MUST be supported with advice on most optimal route to taken, based on available routes, energy consumption (based on simulations / predictions), time and personal preferences (including need to minimize cost for charging). <u>Rationale</u>: The current driving simulator does not support EVs and SHOULD therefore be adapted and extended to provide required functionalities needed for testing Driver Support interface.
Verification:	The driving simulator SHOULD a simulate EV which can be equipped with Driver Support functionalities.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-030
Short name:	Calculations of possible routes.
Aspect:	Functional
Category:	System task
Priority:	1
Use case / Scenario:	Smart Travelling / Highway route re-planned, Emergency call, Unexpected congestion
Description:	Based on map data, current location and requested destination, possible routes will be calculated. Routes can be based on shortest distance, shortest time and minimal height differences (which could impact battery consumption). Rationale: In order to determine the most optimal route, a number of (most logical) routes will need to be calculated. Other criteria could be used to determine the best route(s) to advice to the driver (for which additional processing is required).
Verification:	Based on location and destination and 3D map data a number of alternative routes will need to be calculated.
Conflicts:	
Additional info:	
Version:	1

Req ID:	TNO-031
Short name:	Manage the cost of charging.
Aspect:	Functional

Category:	System task
Priority:	1
Use case / Scenario:	Smart Travelling / all (Highway route re-planned, Up-hill trip, Emergency call, Unexpected congestion)
Description:	The system SHOULD support driver with minimizing the total cost of charging. The cost is related to actual price of the energy at charging station, the speed of charging, status of the battery, the outside and car temperature and the impact on de battery (especially the degradation). <u>Rationale</u>: As cost of charging is depending on many different parameters, the driver SHOULD be supported and advice on how, when and where to change the battery.
Verification:	Based on specific scenario the function SHOULD provide optimal solution (advice) for loading battery. It SHOULD be clear what data, information and knowledge was used to generate the advice.
Conflicts:	
Additional info:	
Version:	1

7. Detailed Requirements for Self-Healing System for Planetary Exploration

In this chapter the technical requirements are included for the use case Self-Healing System for Planetary Exploration.

7.1. Demonstration 1

Req ID:	TASE-001
Short name:	Robotic Arm architecture
Aspect:	Composability
Category:	Mechanical/Physical design
Priority:	1
Use case / Scenario:	Robotic arm control
Description:	The robotic arm will be based on at least 5DOF arm & gripper. Each joint will be commanded by a serial protocol.
Verification:	Inspection
Conflicts:	
Additional info:	Required by use case description
Version:	1

Req ID:	TASE-002
Short name:	Robotic Control Unit architecture
Aspect:	Composability

Category:	Mechanical/Physical design
Priority:	1
Use case / Scenario:	Robotic arm control
Description:	The Robotic Control Unit (RCU) will be based on Zynq UltraScale + in order to implement HW & SW solutions by using the integrated MPSoC.
Verification:	Inspection
Conflicts:	
Additional info:	Xilinx plans to develop a rad-tolerant version of Zynq UltraScale + FPGA.
Version:	1

Req ID:	TASE-003
Short name:	Status monitoring
Aspect:	Functional
Category:	Physical properties
Priority:	1
Use case / Scenario:	Robotic arm control
Description:	It should be possible to monitor the status of the following modules: Self-healing (error counters), runtime adaptation (architecture status), trajectory generation and arm joint status.
Verification:	Inspection
Conflicts:	
Additional	Testing.

info:	
Version:	1

Req ID:	TASE-004
Short name:	Self-healing
Aspect:	Composability
Category:	Reconfigurability
Priority:	1
Use case / Scenario:	Robotic arm control
Description:	The RCU should be protected to tolerate single event upset (SEU) and multiple bit upset (MBUs) by using several techniques like dynamic reconfiguration, TMR, HW/SW adaptation, …
Verification:	Fault injection will be used by using Soft Error Mitigation IP in order to test the robustness.
Conflicts:	
Additional info:	The RCU will be exposed to high radiation conditions.
Version:	1

Req ID:	TASE-005
Short name:	Runtime adaptation
Aspect:	Composability
Category:	Adaptivity
Priority:	1

Use case / Scenario:	Robotic arm control
Description:	RCU shall be able to manage multiple providers of the same services/data to guarantee the system availability or integrity also in case of single/multiple fault. To be able to adjust to current needs (e.g., performance, energy efficiency, substitution of failures), dynamic reconfiguration and adaption of distributed systems (HW/SW) is necessary. It should be possible to command the adaptation strategy in order to validate the functionality.
Verification:	The functionality will be tested by commanding several adaptation strategies and to using fault injection.
Conflicts:	
Additional info:	The RCU will be exposed to high radiation conditions.
Version:	1

Req ID:	TASE-006
Short name:	RCU Communications
Aspect:	User
Category:	Human factors
Priority:	1
Use case / Scenario:	Robotic arm control
Description:	A serial protocol based on RS232 should be used for command and control.
Verification:	The communication will be verified by using a serial terminal.
Conflicts:	
Additional info:	The RS232 protocol is widely spread and it will facilitate to interact with the demonstrator.

Version:	1

Req ID:	TASE-007
Short name:	Arm trajectory generation
Aspect:	Functional
Category:	Computation
Priority:	1
Use case / Scenario:	Robotic arm control
Description:	The Robotic Control Unit (RCU) shall be in charge of the trajectory generation. The trajectory shall be validated by using external constraints to develop collision-free motion paths. Linear and cubic interpolation should be used to generate intermediate points. A configurable security margin should be used in order to avoid collision during testing phase.
Verification:	Several trajectories will be commanded in order to validate the arm movement.
Conflicts:	
Additional info:	Trajectory generation is a building block of robot arms.
Version:	1

Req ID:	TASE-008
Short name:	Inverse Kinematic solver
Aspect:	Functional
Category:	Computation
Priority:	1

Use case / Scenario:	Robotic arm control
Description:	At least one inverse kinematic model will be integrated in RCU in order to execute the trajectory.
Verification:	Several trajectories will be commanded in order to validate the arm movement.
Conflicts:	
Additional info:	Inverse Kinematic models are a building block of robot arms.
Version:	1

Req ID:	TASE-009
Short name:	Error detection and handling
Aspect:	Trustworthiness
Category:	Robustness
Priority:	1
Use case / Scenario:	Robotic arm control
Description:	Implement and verify additional external mechanisms to detect and handle specific errors. In case of a problem occurring on the multicore platform, problem information should be reported and an appropriate action should be taken.
Verification:	
Conflicts:	
Additional info:	
Version:	1

Req ID:	TASE-010
Short name:	Testing
Aspect:	Functional
Category:	Computation
Priority:	1
Use case / Scenario:	Robotic arm control
Description:	The MPSoC/FPGA design must be compatible with Soft Error Mitigation IP in order to test the RCU behaviour under fault injection conditions.
Verification:	
Conflicts:	
Additional info:	
Version:	1

7.2. Demonstration 2

Req ID:	TASE-011
Short name:	Motor hardware
Aspect:	Composability
Category:	Mechanical/Physical design
Priority:	1
Use case / Scenario:	Motor control

Description:	A Brushless DC motor will be used with three hall-effect sensors.
Verification:	Inspection
Conflicts:	
Additional info:	Required by use case description
Version:	1

Req ID:	TASE-012
Short name:	Servo Control Unit architecture
Aspect:	Composability
Category:	Mechanical/Physical design
Priority:	1
Use case / Scenario:	Motor control
Description:	The Servo Control Unit (SCU) will be based on Zynq UltraScale + in order to implement HW & SW solutions by using the integrated MPSoC.
Verification:	Inspection
Conflicts:	
Additional info:	Xilinx plans to develop a rad-tolerant version of Zynq UltraScale + FPGA.
Version:	1

Req ID:	TASE-013
Short name:	Status monitoring

Aspect:	Functional
Category:	Physical properties
Priority:	1
Use case / Scenario:	Motor control
Description:	It should be possible to monitor the status of the following modules: Self-healing (error counters), runtime adaptation (architecture status) and motor status.
Verification:	Inspection
Conflicts:	
Additional info:	Testing.
Version:	1

Req ID:	TASE-014
Short name:	Self-healing
Aspect:	Composability
Category:	Reconfigurability
Priority:	1
Use case / Scenario:	Motor control
Description:	The SCU should be protected to tolerate single event upset (SEU) and multiple bit upset (MBUs) by using several techniques like dynamic reconfiguration, TMR, HW/SW adaptation, …
Verification:	Fault injection will be used by using Soft Error Mitigation IP in order to test the robustness.
Conflicts:	

Additional info:	The SCU will be exposed to high radiation conditions.
Version:	1

Req ID:	TASE-015
Short name:	Runtime adaptation
Aspect:	Composability
Category:	Adaptivity
Priority:	1
Use case / Scenario:	Motor control
Description:	SCU shall be able to manage multiple providers of the same services/data to guarantee the system availability or integrity also in case of single/multiple fault. To be able to adjust to current needs (e.g., performance, energy efficiency, substitution of failures), dynamic reconfiguration and adaption of distributed systems (HW/SW) is necessary. It should be possible to command the adaptation strategy in order to validate the functionality.
Verification:	The functionality will be tested by commanding several adaptation strategies and to using fault injection.
Conflicts:	
Additional info:	The SCU will be exposed to high radiation conditions.
Version:	1

Req ID:	TASE-016
Short name:	SCU Communications
Aspect:	User

Category:	Human factors
Priority:	1
Use case / Scenario:	Motor control
Description:	A serial protocol based on RS232 should be used for command and control.
Verification:	The communication will be verified by using a serial terminal.
Conflicts:	
Additional info:	The RS232 protocol is widely spread and it will facilitate to interact with the demonstrator.
Version:	1

Req ID:	TASE-017
Short name:	Motor control algorithms
Aspect:	Functional
Category:	Computation
Priority:	1
Use case / Scenario:	Motor control
Description:	A PID algorithm will be used for speed control.
Verification:	Test
Conflicts:	
Additional info:	PID controllers are building blocks of robot arms.

Version:	1

Req ID:	TASE-018
Short name:	Error detection and handling
Aspect:	Trustworthiness
Category:	Robustness
Priority:	1
Use case / Scenario:	Motor control
Description:	Implement and verify additional external mechanisms to detect and handle specific errors. In case of a problem occurring on the multicore platform, problem information should be reported and an appropriate action should be taken.
Verification:	
Conflicts:	
Additional info:	
Version:	1

Req ID:	TASE-019
Short name:	Testing
Aspect:	Functional
Category:	Computation
Priority:	1
Use case /	Motor control

Scenario:	
Description:	The MPSoC/FPGA design must be compatible with Soft Error Mitigation IP in order to test the RCU behaviour under fault injection conditions.
Verification:	
Conflicts:	
Additional info:	
Version:	1

8. Detailed Requirements for Ocean Monitoring

The list of requirements has been elicited and grouped according to the scenarios and use cases as means to organize and help prioritize the needed work for Ocean Monitoring.

8.1. Requirements for ocean monitoring - surface

Req ID:	AS-MONITORING-SURFACE-01
Short name:	Camera system - surface
Aspect:	Composability
Category:	Sensors, Actuators
Priority:	1
Use case / scenario:	Ocean monitoring surface
Description:	The marine robot must be equipped with a camera system. The camera system can take pictures and record videos of the marine environment and the user can use the camera system to monitor a particular area of the ocean surface around it. The remote control of the marine robot can also utilize the camera system.
Verification:	The camera system is used to take pictures and record videos, monitor the ocean, and remotely control the robot
Conflicts:	N/A
Additional info:	Ultra-high definition security cameras: https://www.indigovision.com/en-us/products/specialized-cameras/ultra-5k Wide angle lens: https://en.wikipedia.org/wiki/Wide-angle_lens
Version:	1.0

Req ID:	AS-MONITORING-SURFACE-INFRARED-THERMAL-02

Short name:	Infrared and thermal cameras
Aspect:	Composability
Category:	Sensors, Actuators;
Priority:	1
Use case / scenario:	Ocean monitoring - surface
Description:	The camera system may optionally include infrared and thermal cameras to provide additional night-time and heat information and augment the standard visual content in order to alleviate the effects of challenging environmental conditions.
Verification:	The camera system can take thermal and infrared images, and augment the default image with this additional information in order to deal with difficult environmental conditions
Conflicts:	N/A
Additional info:	Pedestrian detection by combining standard and thermal images: www.mdpi.com/1424-8220/15/5/10580/pdf Next-Gen Goggles Combine Night Vision, Thermal and Video Imagery: https://www.youtube.com/watch?v=yRLpDtpgptc Combined use of infrared, thermal and visible imagery: https://www.higp.hawaii.edu/~scott/Workshop_reading/Abrams_etal_1991.pdf
Version:	1.0

Req ID:	AS-MONITORING-SURFACE-WIRELESS-03
Short name:	Wireless communication with the surface camera system
Aspect:	Boundaries

Category:	Networkability
Priority:	1
Use case / scenario:	Ocean monitoring - surface
Description:	Data from cameras must be sent wirelessly to the user's computer system so that the user can monitor the ocean surface in real-time.
Verification:	The user can see the real-time view of the surface environment on his laptop/tablet/smartphone
Conflicts:	N/A
Additional info:	Different types of wireless communication technologies: https://www.efxkits.us/different-types-of-wireless-communication-technologies/ Wireless cameras: www.foscam.co.uk/ www.voltek.co.uk/wireless-cameras.asp
Version:	1.0

Req ID:	AS-MONITORING-SURFACE-AUGMENTED-REALITY-04
Short name:	Augmented reality
Aspect:	Functional
Category:	Computation
Priority:	1
Use case / scenario:	Ocean monitoring - surface
Description:	The camera system may optionally augment the videos by highlighting and tracking moving objects, and enhancing the edges. This can help reduce the information overload and focus the user's attention to enhance the ocean

	monitoring experience.
Verification:	The real-time video is augmented with highlighted moving objects and enhanced edges
Conflicts:	N/A
Additional info:	Image data preparation for intelligent security systems - edge enhancement for improved object tracking: http://ieeexplore.ieee.org/document/4736687/ Real time edge detection based motion tracking: http://www.comm.utoronto.ca/~dkundur/course_info/real-time-DSP/implementation/Demo_Vid_Edge_Motion_Track.pdf Real time object tracking and edge detection for augmented reality: https://ai2-s2-pdfs.s3.amazonaws.com/07d5/c7a69e91c6fad4b8f6d82b8275fd21946fe3.pdf
Version:	1.0

Req ID:	AS-MONITORING-SURFACE-SPHERICAL-05
Short name:	360 degrees surface camera
Aspect:	Composability
Category:	Actuators, Sensors
Priority:	1
Use case / scenario:	Ocean monitoring - surface
Description:	The use of the 360 degrees camera is recommended for the surface monitoring and navigation. The spherical camera provides a natural immersive way of ocean environment monitoring around the marine robot. The 360 degrees camera can be also used for easy remote navigation.
Verification:	The user is presented with 360 degrees view of the surface environment around the marine robot

Conflicts:	N/A
Additional info:	Best 360 degrees cameras: http://www.pocket-lint.com/news/137301-best-360-cameras-the-best-vr-and-360-video-cameras-no-matter-your-budget The navy and 360 degrees cameras: http://www.militaryaerospace.com/articles/2014/06/shipboard-persistent-surveillance.html
Version:	1.0

8.2. Requirements for ocean monitoring - subsea

Req ID:	AS-MONITORING-SUBSEA-RUGGEDIZED-01
Short name:	Ruggedized, waterproof, and airtight
Aspect:	Composability
Category:	Mechanical/ physical design
Priority:	1
Use case / scenario:	Ocean monitoring - subsea
Description:	The underwater robot must be a ruggedized, waterproof, and airtight vehicle capable of withstanding harsh marine conditions including resistance to water pressure to ensure operations, and may optionally be able to propel itself.
Verification:	The robot is able to withstand harsh marine environmental conditions
Conflicts:	N/A
Additional info:	Ruggedized waterproof cameras: https://www.ephotozine.com/article/top-10-best-waterproof-tough-cameras-2017-17302

Version:	1.0

Req ID:	AS-MONITORING-SUBSEA-UMBILICAL-02
Short name:	Umbilical cord
Aspect:	Composability
Category:	Mechanical/ physical design
Priority:	1
Use case / scenario:	Ocean monitoring - subsea
Description:	The underwater robot must have a disconnectable umbilical cord that must be able to communicate live data from multiple sensors between subsea and surface. This will allow for the real-time subsea monitoring on the user's laptop, tablet, or smartphone.
Verification:	The user is presented with the real-time view of the subsea environment
Conflicts:	N/A
Additional info:	Real time underwater streaming with umbilical: http://www.smp-ltd.com/blog/tag/underwater-camera-system/ http://www.scubacenter.com/scubacenter_onlinestore/search_and_salvage/JW_Fishers_Underwater_Video.htm
Version:	1.0

Req ID:	AS-MONITORING-SUBSEA-WIRELESS-ANTENNA-03
Short name:	Wireless antenna for the subsea robot
Aspect:	Composability

Category:	Mechanical/ physical design
Priority:	1
Use case / scenario:	Ocean monitoring - subsea
Description:	The underwater vehicle must have an embedded wireless antenna that reaches above sea level when the underwater vehicle is at surface level. In the case the umbilical cord gets disconnected and the robot resurfaces automatically, the antenna will transmit the underwater vehicle's location for the quick recovery.
Verification:	The robot communicates its position when on surface
Conflicts:	N/A
Additional info:	Wireless antenna for AUV, page 610: https://books.google.co.uk/books?isbn=3319325523 Wireless antennas for long distance communication: http://robotsforroboticists.com/long-distance-wireless-communications-antennas/
Version:	1.0

Req ID:	AS-MONITORING-SUBSEA-CAMERA-SYSTEM-04
Short name:	Underwater camera system
Aspect:	Composability
Category:	Sensors, actuators
Priority:	1
Use case / scenario:	Ocean monitoring - subsea
Description:	The underwater robot must comprise an underwater camera system that optionally may have multiple lenses to cover a wide angle or 360 degrees

	camera view. The camera system will be used for real-time subsea monitoring and will also take pictures and record videos of the underwater environment.
Verification:	The user can visually monitor the subsea environment and the camera system can take underwater pictures and record videos
Conflicts:	N/A
Additional info:	360 degrees underwater cameras: http://www.threesixtycameras.com/waterproof-360-cameras/ Underwater camera robot for farming and fishing: https://www.alibaba.com/product-detail/underwater-rov-robot-camera-for-sea_60599969080.html Deep Trekker with 330 degree field of view: https://www.deeptrekker.com/dtg2/
Version:	1.0

Req ID:	AS-MONITORING-SUBSEA-LIGHTS-05
Short name:	Lights for the subsea robot
Aspect:	Composability
Category:	Mechanical/ physical design
Priority:	1
Use case / scenario:	Ocean monitoring - subsea
Description:	The underwater robot must have underwater lights that optionally can be switched on/off automatically, for the underwater camera system to be able to operate in poor lighting conditions.
Verification:	The underwater camera system can operate in poor lighting conditions

Conflicts:	N/A
Additional info:	New light system for underwater camera: http://www.jwfishers.com/nr/nr152.html Illumination technologies for conventional imaging - page 282: https://books.google.co.uk/books?isbn=0857093525
Version:	1.0

Req ID:	AS-MONITORING-SUBSEA-BUOYANCY-06
Short name:	Buoyancy of underwater robot
Aspect:	Composability
Category:	Mechanical/ physical design
Priority:	1
Use case / scenario:	Ocean monitoring - subsea
Description:	The underwater robot may optionally be able to adjust its buoyancy by filling/emptying small cylindrical containers (syringes) with water. This includes positive and negative buoyancy as well as neutral buoyancy allowing the marine robot to stay at a specific depth
Verification:	The underwater robot can adjust its buoyancy to surface, submerge, or stay at a specific depth
Conflicts:	N/A
Additional info:	Waterbug - buoyancy controlled with syringes: http://cse.unl.edu/~carrick/papers/HigginsC-IROS-2016.pdf Underwater glider - buoyancy controlled with syringes: http://www.instructables.com/id/Underwater-Glider/ Arduino controlled water pump https://www.youtube.com/watch?v=yLEu80HYElE

	Syringe with Arduino https://www.youtube.com/watch?v=-81ECglkDqA https://www.youtube.com/watch?v=-81ECglkDqA
Version:	1.0

Req ID:	AS-MONITORING-SUBSEA-ROBOT-WEIGHT-07
Short name:	Launching underwater robot from a boat - weight consideration
Aspect:	Composability
Category:	Mechanical/ physical design
Priority:	1
Use case / scenario:	Ocean monitoring - subsea
Description:	The user must be able to launch and pick up the underwater robot from a surface boat. Hence, the subsea robot's weight should not exceed 20kg.
Verification:	The user is able to launch and pick up the robot from a boat
Conflicts:	Potential conflict when the components are too heavy
Additional info:	Launching and picking up the robot from a boat: http://www.bluefinrobotics.com/vehicles-batteries-and-services/hauv
Version:	1.0

Req ID:	AS--MONITORING-SUBSEA-ROOM-ADDITIONAL-SENSORS-08
Short name:	Room for additional sensors
Aspect:	Composability

Category:	Mechanical/ physical design
Priority:	1
Use case / scenario:	Ocean monitoring - subsea
Description:	The underwater robot must have place or room for optional mounting of additional digital equipment such as temperature sensor, pressure sensor, multibeam echosounder, side scanning sonar, lasers. The additional sensors can help collect extra data and provide more environmental information for the obstacle avoidance for example
Verification:	The marine vehicle provides room for additional sensors
Conflicts:	Potential conflict when considering the size and weight of the vehicle
Additional info:	Design considerations for engineering autonomous underwater vehicles: http://www.bluefinrobotics.com/vehicles-batteries-and-services/hauv
Version:	1.0

8.3. Requirements for control of thrusters and steering through software

Req ID:	AS-PROPULSION-ENGINES-01
Short name:	Thrusters and steering with software
Aspect:	Composability
Category:	Control
Priority:	1
Use case / scenario:	Use case: control of thrusters and steering through software
Description:	The marine robot may be operated and navigated by more than one engine.

	The rationale for this is that at least one engine must be used to propel the marine robot. If this engine also can rotate along the second axis, i.e. vertically, one engine will be able to thrust and steer the marine robot, although the turning of the marine robot will need some turning radius. If however, the marine robot optionally needs have a catamaran hull, then two parallel engines can be fixed to the hull and steer the vehicle to the left, right, and reverse. If more precision is needed when turning, to avoid e.g. obstacles in a narrow space, then the hull needs to be rotated/ pivoted on a penny, then two or more engines are needed for such precise navigation control.
Verification:	It is possible to thrust and navigate the marine robot on sea surface through software.
Conflicts:	N/A
Additional info:	Propulsion examples with one or more engines: https://www.youtube.com/watch?v=GYAQLwe-Dks https://www.youtube.com/watch?v=WuB8z22UgAo https://www.youtube.com/watch?v=ENVjelLRv5M
Version:	1.0

Req ID:	AS-PROPULSION-SAFE-CLEAN-02
Short name:	Safe and clean propulsion
Aspect:	Composability
Category:	Sensors, Actuators;
Priority:	2
Use case / scenario:	Use case: control of thrusters and steering through software
Description:	The engine should be environmentally friendly with safe and clean propulsion, therefore an engine is recommended to be either an electric stepper, servo, or stepless motor The rationale is to enable safe and clean propulsion, at the same time as

	reducing the complexity of the number of mechanical components needed to enable propulsion. In other words, it is both saving the environment at the same time as reducing costs of developing the marine robot. In terms of which type of electric engine that is recommended will depend on the effect needed in terms of Watt, or kilowatt, vs. the need for precise control over the engine's rotation.
Verification:	No oxygen or hydrocarbon needed to drive the energy. Batteries are present to provide power to the engine.
Conflicts:	N/A
Additional info:	Pros and cons of using stepper engines vs. servo engines: https://www.lifewire.com/stepper-motor-vs-servo-motors-selecting-a-motor-818841 Example of stepper and micro-controller: https://www.youtube.com/watch?v=ay_Yg20k2SU Example of servo and micro-controller: https://www.youtube.com/watch?v=gL7b8E_5aYs
Version:	1.0

Req ID:	AS-PROPULSION-LOW-NOISE-VIBRATION-03
Short name:	Ultra-low noise and vibration levels
Aspect:	Composability
Category:	Sensors, Actuators;
Priority:	2
Use case / scenario:	Use case: control of thrusters and steering through software
Description:	It is recommended that the engines operate at ultra-low noise and vibration levels. The rationale is for the marine robot to operate without unnecessarily disturbing any humans or nature. Silent operations will enable close up filming and photography of the environment. Furthermore, the less vibration

	there is the less maintenance costs it will to ensure daily operations for the owner of the vehicle. Vibrations are known to affect physical material over time, tending to produce microscopic cracks.
Verification:	The vehicle has noise reduction equipment separating electro-mechanical components from the vehicle hull, and/or the hull has embedded material that reduces vibration and by this also actively dampens noise propagation. If all engines for propulsion and steering are electric, this is also satisfies this requirement.
Conflicts:	N/A
Additional info:	Reducing/ isolating sounds and vibrations by using foam, rubber hoses and rubber bearings: https://www.youtube.com/watch?v=fciWTvJPFAA https://www.youtube.com/watch?v=3J45x36a4EQ https://www.youtube.com/watch?v=8_NFUsbXtVw https://www.youtube.com/watch?v=AgRFD_EpQmk https://www.youtube.com/watch?v=8_NFUsbXtVw&t=80s Example of using electric engines to reduce noise and vibration: https://www.youtube.com/watch?v=LTem7YZuDWY
Version:	1.0

Req ID:	AS-PROPULSION-SPEED-DIRECTION-04
Short name:	Set the input power of each engine
Aspect:	Composability
Category:	Control
Priority:	1
Use case / scenario:	Use case: control of thrusters and steering through software
Description:	The software controlling the robot must be able to set the speed and direction of the marine robot by increasing, decreasing, or reversing the input power to the engine

	The reason for this is that the human operator will rely on propulsion of the vehicle to bring it from one location to another. Other forms of 'propulsion' could be performed though sailing, rotor sails, or converting wave movement into motion; however, these forms are dependent on the availability of wind and waves.
Verification:	There marine robot has one or more propels that are controlled by microcontrollers. Microcontrollers communicates with a central computer dynamically can reconfigure the input power to the engine.
Conflicts:	N/A
Additional info:	Controlling a stepper with microcontroller (Arduino): https://www.youtube.com/watch?v=JzR6FV-sdH4 https://www.youtube.com/watch?v=AcLUopVZMco https://www.youtube.com/watch?v=JzR6FV-sdH4 https://www.youtube.com/watch?v=4rK3i2a7WZo Controlling AC motor with microcontroller (Arduino): https://www.youtube.com/watch?v=YH6Di81gGAA https://www.youtube.com/watch?v=gnlcS_xOp9k
Version:	1.0

Req ID:	AS-PROPULSION-BATTERY-05
Short name:	Battery Powered Thrusters and Steering
Aspect:	Composability
Category:	Energy
Priority:	1
Use case / scenario:	Use case: control of thrusters and steering through software
Description:	The engine and the steering must be powered by a battery, and a lithium battery solution is recommended because of its light weight and storage

	capacity.
Verification:	Room has been set aside within the marine robot for mounting and securing batteries. A battery has been installed in this battery compartment, and it is ready to provide power to onboard equipment.
Conflicts:	N/A
Additional info:	An example of using batteries to power boat propulsion: https://www.youtube.com/watch?v=0dAVd6cKDdc
Version:	1.0

Req ID:	AS-PROPULSION-GEARS-06
Short name:	Software control of thrusters and steering
Aspect:	Composability
Category:	Control
Priority:	1
Use case / scenario:	Use case: control of thrusters and steering through software
Description:	The user must be able to control thrust and steering of the marine vehicle through software, and the propulsion control shall include forward, standby, and reverse propulsion.
Verification:	The human operator can control the marine robot's speed and direction through interaction with a user interface on his/ her smartphone, tablet PC, or laptop. This means that there must be software APIs available and implemented in Java for controlling and operating thrusters and steering from application installed on Android.
Conflicts:	N/A
Additional info:	Example of using smartphone for boat navigation purposes: https://www.youtube.com/watch?v=iTR6Wpe8QN8

Version:	1.0

Req ID:	AS-PROPULSION-WITH-SOFTWARE-07
Short name:	Software control of thrusters and steering
Aspect:	Composability
Category:	Control
Priority:	1
Use case / scenario:	Use case: control of thrusters and steering through software
Description:	The user must be able to control thrust and steering of the marine vehicle through software.
Verification:	The human operator can control the marine robot's speed and direction through interaction with a user interface on a his/ her smartphone, tablet PC, or laptop. This means that there must be software APIs available and implemented in Java for controlling and operating thrusters and steering from application installed on Android.
Conflicts:	N/A
Additional info:	Example of using smartphone for boat navigation purposes: https://www.youtube.com/watch?v=iTR6Wpe8QN8
Version:	1.0

Req ID:	AS-PROPULSION-AUTOPILOT-OPTION-08
Short name:	Optional control by autopilot
Aspect:	Composability
Category:	Control

Priority:	2
Use case / scenario:	Use case: control of thrusters and steering through software
Description:	The thrust and steering may be optionally controlled by an autopilot, and in the case an autopilot controls the marine vehicle, it must be able to set and monitor the speed and direction of the marine vehicle.
Verification:	The autopilot can drive the vehicle from one location to another, or maintain its location on the sea.
Conflicts:	N/A
Additional info:	Example of an autopilot for sailing with smartphone: https://www.youtube.com/watch?v=dCafLlMtBiM
Version:	1.0
Req ID:	AS-PROPULSION-RENEWABLE-ENERGY-09
Short name:	Renewable energy recommended for battery charging
Aspect:	Composability
Category:	Energy
Priority:	2
Use case / scenario:	Use case: control of thrusters and steering through software
Description:	It is recommended that the required recharging of batteries for propulsion is based on renewable energy sources, such as solar, waves, wind, hydro, or kinetic.
Verification:	The recharging of the batteries is achieved by using one of solar, waves, wind, hydro, or kinetic energy to generate electric recharging power. The recharging could be happening for instance either from onboard, onshore, or from an on sea charging point.

Conflicts:	N/A
Additional info:	Example of an onboard solar charging solution: https://www.youtube.com/watch?v=eNRELHoD7BQ
Version:	1.0

Req ID:	AS-PROPULSION-SOFTWARE-MICROCONTROLLERS-10
Short name:	Operations and control of actuators, engines
Aspect:	Composability
Category:	Sensors, Actuators; Control
Priority:	2
Use case / scenario:	Use case: control of thrusters and steering through software
Description:	Each actuator, engine must be operated by: Microcontrollers to control and interface with each actuator Software with multi-threading for net-based control Networked communication with standardised protocols A central computer with an embedded emergency battery to ensure remote communication with human operator
Verification:	There are communication cables connecting with each microcontroller that again controls each actuator (i.e. Arduino). The communication cables lead to a central computer that controls each microcontroller with dedicated multithreading software (i.e. Java). The central computer has an embedded emergency battery and remote wireless communication.
Conflicts:	N/A
Additional info:	Example of operations and control of actuators, engines: https://www.youtube.com/watch?v=uwa1M85q8Ao https://www.youtube.com/watch?v=WEaA5QfzExI Example for electric cars, relevant also for marine vehicles:

	https://www.youtube.com/watch?v=_SXwvz9BPZc
Version:	1.0

Req ID:	AS-THRUSTERS-DESIGN-OPTIONS-11
Short name:	Design and operation of thrusters and steering
Aspect:	Composability
Category:	Energy
Priority:	2
Use case / scenario:	Use case: control of thrusters and steering through software
Description:	The thrust and steering is recommended to be designed and operate according to one of the following principles: Use one thruster to deliver push along a rotating axis to control the thrust and steering together Use one thruster to operate with a fixed thrust direction, and where the steering is done by a separate rudder Use two thrusters that both have the same fixed thrust direction, and the steering is achieved by providing variable thrusts on these
Verification:	The thrusters and steering installed in the marine vehicle have been designed following the three principles
Conflicts:	N/A
Additional info:	Examples of rudder control and designs: https://www.youtube.com/watch?v=jEzB-nm-BmA https://www.youtube.com/watch?v=UDBeCqzYO5k Examples of thruster control: https://www.youtube.com/watch?v=vR41QlOoM4s Bow and stern thruster examples: https://www.youtube.com/watch?v=mjdfPhqgiyg https://www.youtube.com/watch?v=tk1PMLY8YfE

Version:	1.0

8.4. Requirements for remote control of marine robot

Req ID:	AS-REMOTE-CONTROL-RECHARGEABLE-BATTERIES-01
Short name:	Rechargeable Batteries Needed
Aspect:	Composability
Category:	Energy
Priority:	1
Use case / scenario:	Use Case: Remote Control of Marine Robot
Description:	The marine robot must incorporate rechargeable batteries.
Verification:	There are rechargeable batteries installed in the marine robot with sufficient capacity to provide power to the propulsion and steering actuators, along with also additional less power-hungry onboard equipment.
Conflicts:	N/A
Additional info:	Selecting a marine storage battery: https://www.westmarine.com/WestAdvisor/Selecting-a-Marine-Storage-Battery What you need to know about boat batteries: http://onthelake.net/boating/batteries.htm
Version:	1.0

Req ID:	AS-REMOTE-CONTROL-ENVIRONMENT-02
Short name:	Charging Based on Renewable Energy

Aspect:	Business
Category:	Environment
Priority:	1
Use case / scenario:	Use Case: Remote Control of Marine Robot
Description:	It is recommended that the required recharging of batteries for propulsion is based on renewable energy sources, such as solar, waves, wind, hydro, or kinetic.
Verification:	It is physically evident that solar, wave, hydro, or a kinetic energy harvesting is used to charge the onboard batteries.
Conflicts:	N/A
Additional info:	Examples of using onboard solar panels: https://www.youtube.com/watch?v=UuZS6fpXii0 https://www.youtube.com/watch?v=MfjZd_zxeFo https://www.youtube.com/watch?v=6ROcWxX5Qrw
Version:	1.0

Req ID:	AS-REMOTE-CONTROL-OPTIONAL-03
Short name:	Recommended Functions related to Remote-Control
Aspect:	Functionality
Category:	User Task
Priority:	1
Use case / scenario:	Use Case: Remote Control of Marine Robot
Description:	For remote-controlling the marine robot, the user should be able to: register his/her profile by signing up

	remotely access and control the data and equipment onboard trigger the autopilot and specify robot's destination add location of the robot to the list of locations select "return home" functionality to make the robot return to the location of origin
Verification:	The user/ human operator can interact with various system parts to sign up, access onboard data and equipment from remote, start/ stop auto-pilot, specify locations for the marine robot, or make the robot return to origin.
Conflicts:	N/A
Additional info:	Illustrative examples of use of remote control and video: https://www.youtube.com/watch?v=5ho3CX4ayiY https://www.youtube.com/watch?v=rAKnu65cRT0&t=7s https://www.youtube.com/watch?v=_b17B6wh6Sc
Version:	1.0

Req ID:	AS-REMOTE-CONTROL-BASICS-04
Short name:	Needed Remote-Control Functions
Aspect:	Functionality
Category:	User Task
Priority:	1
Use case / scenario:	Use Case: Remote Control of Marine Robot
Description:	For remote-controlling the marine robot, the user must be able to perform the following tasks: check the robot's status from office or home pair the vehicle with his/her smartphone app start the engine remotely navigate the robot by adjusting the speed and direction of the marine vehicle via his/her computer system (smartphone, laptop, tablet) and

	also using live camera feed
Verification:	There is an app present on the user's/ human operator's smartphone from where the user can check the status of his/her vehicle, start and stop the vehicle, with interactions to remote control/ drive the vehicle.
Conflicts:	N/A
Additional info:	Example of driving RC boat from smartphone via Bluetooth: https://www.youtube.com/watch?v=txXkEaWvFUc
Version:	1.0

Req ID:	AS-REMOTE-CONTROL-SECURITY-KEYS-05
Short name:	Wireless Key Fobs Recommended
Aspect:	Boundaries
Category:	Networkability
Priority:	1
Use case / scenario:	Use Case: Remote Control of Marine Robot
Description:	It is recommended that the user use a secure wireless Near-Field Communication (NFC) key fob to start the marine robot locally or from remote.
Verification:	The user is able to start the robot by either tapping wireless key fob to a dedicated area on the marine robot, or to the back of his/ her smartphone. The engine is unlocked and the user can start the engine.
Conflicts:	N/A
Additional info:	Example of using NFC for access to data and vehicles: https://www.youtube.com/watch?v=eMZLbl5tFJ4 https://www.youtube.com/watch?v=atJN8w5Raf0 https://www.youtube.com/watch?v=lSxP_OpdVow

Version:	1.0

Req ID:	AS-REMOTE-CONTROL-SECURE-COMMUNICATION-06
Short name:	Secure Network communication recommended
Aspect:	Boundaries
Category:	Networkability
Priority:	1
Use case / scenario:	Use Case: Remote Control of Marine Robot
Description:	It is recommended that marine robot's operations and data are being transmitted over a secure network protocol, to ensure the robot's protection from potential unsolicited use by others.
Verification:	In the case the smartphone app has been implemented as a web application, the use of the https-protocol will most likely be visible from within the web application/ web browser.
Conflicts:	N/A
Additional info:	Lecture on SSL and HTTPS: https://www.youtube.com/watch?v=q1OF_0ICt9A
Version:	1.0

Req ID:	AS-REMOTE-CONTROL-WIRELESS-COMMUNICATION-07
Short name:	Wireless communication and network
Aspect:	Boundaries

Category:	Networkability
Priority:	1
Use case / scenario:	Use Case: Remote Control of Marine Robot
Description:	It must be possible for the user to establish network connection and communicate with the marine robot via local wireless (WiFi or Bluetooth), mobile network, or satellite communication.
Verification:	The marine robot/ vehicle has mobile network (2G, 3G, 4G), emergency network, WiFi, Bluetooth, and optionally Satellite communication installed for two-way communication between the marine robot and the human operator.
Conflicts:	N/A
Additional info:	Microcontroller to Android via WiFi: https://www.youtube.com/watch?v=cxVABKNy7OI Microcontroller to Android via Bluetooth: https://www.youtube.com/watch?v=g9TriZdtlgo Microcontroller to Android via USB: https://www.youtube.com/watch?v=qJ-Knbdmvaw
Version:	1.0

Req ID:	AS-TRANSPORTATION-RECREATION-08
Short name:	Design for transport or recreation
Aspect:	Composability
Category:	Mechanical/ physical design
Priority:	2
Use case / scenario:	Use Case: Remote Control of Marine Robot

Description:	It should be possible to use the marine robot for transport or recreational purposes.
Verification:	In the case the marine robot hull is to be used for transport, there must be sufficient storage space onboard, designed for maximum transportation chosen by the designer. In the case it is to be used for recreational purposes for transporting humans, the marine robot hull has been designed according to national safety regulations. If marine robot is to be only used as ‘eyes on the water’, the hull has still been designed to carry a smaller amount of extra cargo, typically for mounting of extra battery modules, larger/ additional engines, or digital equipment.
Conflicts:	N/A
Additional info:	Example hull designs with assessments of pros and cons: https://www.youtube.com/watch?v=M9N2o4BF2iM https://www.youtube.com/watch?v=s-cK6oCUnvc https://www.youtube.com/watch?v=JhG6R0X28mI https://www.youtube.com/watch?v=G5JOuSuT-lI https://www.youtube.com/watch?v=d-R8B8EBpf0
Version:	1.0

8.5. Requirements for reconfiguration of battery module in runtime

Req ID:	AS-BATTERY-01
Short name:	Battery
Aspect:	Composability
Category:	Energy; Mechanical/ physical design
Priority:	2, 3
Use case / scenario:	Reconfiguration of battery module in runtime
Description:	The marine robot must be equipped with a battery to supply energy to

	propulsion, sensors, actuators, and processors. It is recommended that the battery is lithium-ion comprising multiple cells as lithium-ion batteries are light and offer high energy density, low self-discharge and low maintenance.
Verification:	The marine robot's components get the energy supply from the battery
Conflicts:	N/A
Additional info:	Lithium Ion batteries: http://www.mastervolt.com/products/li-ion/ https://www.lithiumion-batteries.com/lithium-marine-batteries.php
Version:	1.0

Req ID:	AS-BATTERY-ADAPTIVITY-02
Short name:	Reconfiguration of battery topology
Aspect:	Composability
Category:	Reconfigurability; Adaptivity; Energy
Priority:	2, 3
Use case / scenario:	Reconfiguration of battery module in runtime
Description:	The battery should provide fairly constant energy with a certain voltage and supply energy to marine vehicle's components for the maximum amount of time. It should also minimise the risk of thermal runaway which can potentially present a fire hazard. Hence, it is recommended that the battery automatically reconfigures its topology by connecting the cells in series or in parallel to overcome the "weak link" and to prolong its lifetime when the fluctuations of voltage are detected or a certain voltage is needed, e.g. one of the cells is almost empty or is faulty.
Verification:	The battery can automatically reconfigure its topology in order to adapt to changing conditions

Conflicts:	N/A
Additional info:	Emerging technologies, evolving hazards: http://www.boatus.com/seaworthy/magazine/2012/january/hazards.asp Topology optimization of lithium ion batteries: https://repository.tudelft.nl/islandora/object/uuid:f59b08e4-c3e9-400c-9403-f8129cd5b578?collection=education Building battery arrays with lithium ion cells: http://americansolarchallenge.org/ASC/wp-content/uploads/2013/01/high_cell_count.pdf
Version:	1.0

Req ID:	AS-BATTERY-CELLS-MONITORING-03
Short name:	Monitoring of the battery cells
Aspect:	Composability
Category:	Mechanical/ physical design; Sensors, actuators; Energy
Priority:	2, 3
Use case / scenario:	Reconfiguration of battery module in runtime
Description:	The battery may monitor its status by monitoring the temperature, the power levels, and the output voltage of each cell. The monitoring of individual cells allows for detection of irregularities/faulty cells and potential prevention of thermal runaway or provision of voltage equalization. The detected issues can be overcome by switching to backup battery or by more sophisticated battery topology reconfiguration.
Verification:	The battery cells are being monitored in real time
Conflicts:	N/A
Additional info:	Temperature monitoring for battery management system: http://www.electronicdesign.com/electromechanical/add-simple-

	temperature-monitoring-battery-management-systems Battery monitoring system: http://www.eepowersolutions.com/categories/battery-monitoring-systems-bms-series/
Version:	1.0

Req ID:	AS-BATTERY-INFORMATION-04
Short name:	Communication of battery information to the user
Aspect:	Functional
Category:	System task
Priority:	2, 3
Use case / scenario:	Reconfiguration of battery module in runtime
Description:	The battery must communicate battery information to the user concerning the remaining power. This will allow the user to strategize the robot's operations to preserve the remaining energy. The battery may obtain the information from build-in battery management system or may perform the advanced calculations itself using various predictive/adaptive methods.
Verification:	The user is presented with the information about the remaining energy levels
Conflicts:	N/A
Additional info:	Methods for lithium ion battery remaining energy prediction: http://econpapers.repec.org/article/eeeappene/v_3a149_3ay_3a2015_3ai_3ac_3ap_3a297-314.htm https://www.semanticscholar.org/paper/Remaining-useful-life-prediction-of-lithium-ion-ba-Miao-Xie/c09f58f7e0d8118ebf0dfd8f498739f43c12ecfa
Version:	1.0

Req ID:	AS-BATTERY-MONITORING-THERMAL-05
Short name:	External battery temperature monitoring with thermal camera
Aspect:	Composability
Category:	Sensors, actuators
Priority:	2, 3
Use case / scenario:	Reconfiguration of battery module in runtime
Description:	The battery may monitor its temperature by using a thermal camera pointed at the battery to detect and prevent potential thermal runaway. Such thermal runaway presents a serious fire hazard that can be avoided by battery topology reconfiguration or switching to a back up battery.
Verification:	The thermal camera monitors the temperature of the battery
Conflicts:	N/A
Additional info:	Detectable thermal anomalies in batteries: http://www.battcon.com/PapersFinal2014/5%20Sirmans%20Paper%202014%20Final.pdf Thermal monitoring: http://systemswithintelligence.com/Solutions-ThermMon.html
Version:	1.0

8.6. Requirements for adaptive camera system

Req ID:	AS-ADAPTIVE-CAMERA-APP-01
Short name:	Web application for adaptive camera system
Aspect:	Functional
Category:	System task

Priority:	2, 3
Use case / scenario:	Adaptive camera system
Description:	The user must be able to start the web application for the adaptive camera system on his/her laptop, smartphone, or tablet. The app will allow the user to specify the task/context for the adaptive camera system which will combine images from multiple lenses/cameras in order to e.g. enhance image resolution, de-noise images, enhance the foreground, measure perception depth, etc. The app must also allow the user to view previously captured images/videos and the live previews of the marine environment.
Verification:	The web app allows the user to: specify the context/task for the adaptive camera system and view the previously captured images
Conflicts:	N/A
Additional info:	16 lens mobile camera system: https://www.wired.com/2015/10/light-l16-camera/
Version:	1.0

Req ID:	AS-ADAPTIVE-CAMERA-02
Short name:	Adaptive camera
Aspect:	Composability
Category:	Sensors, actuators; Adaptivity
Priority:	2, 3
Use case / scenario:	Adaptive camera system
Description:	The adaptive camera system must consist of at least two synchronized cameras working in tandem, or a camera with two lenses. The camera may optionally incorporate actuators for rotating the lenses. The cameras or lenses should point in the same direction with images overlapping to a

	different extent. It is recommended to use mobile cameras that can be easily programmed and controlled in Java. Multiple cameras/lenses can perform tasks that could be difficult or impossible for single cameras. Existing camera arrays are usually designed for a specific task only.
Verification:	The adaptive camera system comprises at least two cameras working in tandem or a camera with at least two lenses
Conflicts:	N/A
Additional info:	Dual camera phones are the future of mobile photography: https://www.theverge.com/2016/4/11/11406098/lg-g5-huawei-p9-two-camera-smartphone-trend-apple
Version:	1.0

Req ID:	AS-ADAPTIVE-CAMERA-ADAPTIVITY-03
Short name:	Adaptivity of the camera system
Aspect:	Functional
Category:	System task
Priority:	2, 3
Use case / scenario:	Adaptive camera system
Description:	The adaptive smart camera system must be able to create super-resolution images from multiple cameras or lenses and de-noise images taken by multiple cameras or lenses. It must adaptively produce higher resolution images or remove the noise, depending on the context specified by the user. In general, the adaptivity is manifested by the same camera system's ability to perform two or more different tasks depending on the user's selection, and the changing number of activated cameras to progressively enhance the image quality, remove the noise, etc. Existing camera arrays are usually designed for a specific task only.

Verification:	The camera system adapts to the user context by performing a specific task requested by the user
Conflicts:	N/A
Additional info:	Multiple view image de-noising: https://pdfs.semanticscholar.org/03ac/e3dab28ed00f2e96006aaae155824486cb8b.pdf Multi-camera imaging system using super-resolution: http://www.ieice.org/proceedings/ITC-CSCC2008/pdf/p465_A4-5.pdf
Version:	1.0

Req ID:	AS-ADAPTIVE-CAMERA-ADDITIONAL-FEATURES-04
Short name:	Optional context for the adaptive camera system
Aspect:	Functional
Category:	System task
Priority:	2, 3
Use case / scenario:	Adaptive camera system
Description:	It is recommended for the adaptive camera system to be able to adjust and enhance the videos and images from multiple cameras or lenses by: Real-time edge detection and edge enhancement Object detection and tracking Foreground and background detection Foreground enhancement Depth perception Image stabilisation All the aforementioned tasks can significantly benefit from the multi-lenses/multi-camera system design. Moreover, the additional functionalities can greatly enhance user's perception of the surroundings and his/her ocean

	monitoring experience.
Verification:	The additional optional context can be requested by the user
Conflicts:	N/A
Additional info:	3D reconstruction from multi-camera: http://www-hagen.cs.uni-kl.de/wp-content/uploads/publication/791.pdf Multi-camera tracking: http://imagelab.ing.unimore.it/imagelab2015/pubblicazioni/avss2005_calderara.pdf Multi-camera system for depth estimation: https://www.osapublishing.org/abstract.cfm?uri=isa-2015-IT3A.2
Version:	1.0

Req ID:	AS-ADAPTIVE-CAMERA-IMMERSION-05
Short name:	Adaptive camera system providing immersive experience for the user
Aspect:	User
Category:	Human Factors; Look and Feel
Priority:	2, 3
Use case / scenario:	Adaptive camera system
Description:	It is recommended for the adaptive camera system to be able to provide immersive videos and images to the user by means of: Stereoscopic images 360 degrees spherical/ panoramic images and videos Virtual reality (VR) goggles The 360 degrees/panoramic images can be produced from the images taken by the multi-camera/multi-lenses system by fusing them together, or utilizing the camera geometry. In the case of 360 degrees images, the

	additional immersion and enhanced ocean monitoring capabilities can be provided by the use of Virtual Reality (VR) goggles. The VR goggles will allow the user to monitor the ocean environment in a natural way in real time as well as view the previously taken images/videos in VR.
Verification:	The camera system creates the 360/wide angle images and videos and the user may optionally use VR goggles for even more immersive experience
Conflicts:	N/A
Additional info:	Camera array for wide angle pictures: http://www.isprs.org/proceedings/XXXV/congress/comm1/papers/90.pdf Using VR glasses with a drone: http://www.dreamflights.pro/how.html
Version:	1.0

Req ID:	AS-ADAPTIVE-CAMERA-IMAGE-TRANSMISSION-06
Short name:	Transmission of images and videos from the camera system
Aspect:	Functional
Category:	System task
Priority:	2, 3
Use case / scenario:	Adaptive camera system
Description:	Due to the potentially high file sizes of the data and real-time monitoring requirements it should be possible to transmit videos and images wirelessly in parallel, taking into consideration image and video compression and decompression techniques to further reduce the transmission burden. The videos and images must be transmitted via HyperText Transfer Protocol (HTTP).
Verification:	The images and videos can be efficiently transmitted wirelessly

Conflicts:	N/A
Additional info:	Fractal image compression: http://www.ti.com/lit/an/bpra065/bpra065.pdf
Version:	1.0

Conflicts:	The use of gyroscope is related to the robot's balance requirement
Additional info:	Gyros for boat stabilisation: http://www.marlinmag.com/new-normal-gyro-stabilization-for-boats#page-2
Version:	1.0

Req ID:	AS-AUTOPILOT-GPS-06
Short name:	Global Positioning System for autopilot
Aspect:	Composability
Category:	Sensors, actuators
Priority:	2, 3
Use case / scenario:	Autopilot - navigation
Description:	When it comes to auto-piloting, the marine robot must be able to keep track of its current location using the gps data from the gps sensor. A nautical chart, e.g. the free OpenSeaMap can be used in connection with gps to provide the autopilot with geographical location information.
Verification:	The marine robot can keep track of its current location
Conflicts:	N/A
Additional	OpenSeaMap:

info:	http://map.openseamap.org/
Version:	1.0

9. Conclusions

In this document we described all the CERBERO assessment scenarios, their challenges, demonstration activities planned within them, and respective detailed requirements. We have used a common strategy for requirements classification. This will facilitate the definition of the CERBERO cross-domain common set of requirements in the next iteration.

Please note that, this document provides preliminary descriptions that are meant to be updated during the project lifetime. Updates will take place yearly (M13 and M25 right after the first annual General Assembly meeting) and also in correspondence to the mid-term review (M19). 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.

10. 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

11. Annex: Interview script – Smart Travelling focus group sessions

11.1. Participants selection

Selected participants are at least 6 people per group. Selection criteria were (with a bit more slack later on):

- Able to read and write (either with or without glasses)
- Driver license for at least 10 years
- Qualification: a degree (university)
- NO journalists, public relations, OEM and TIER 1 employees, vehicle dealers, market research, taxi drivers
- Age between 30 to 50 years
- No gender segmentation
- There are two groups (electric/hybrid plug-in or hybrid (no plug-in) in which there is no segmentation on km/year, compared to the third group (gasoline/diesel) in which there is a segmentation on < 10.000 km/year vs. > 20.000 km/year.
- Use of the vehicle in all scenarios (urban, extra urban, highway).
- 5 days a week use of the vehicle
- Division in electric or hybrid plug-in, hybrid vehicle (no plugin) and gasoline/diesel is very clear.
- The last criterion, at least 1 year of use or a vehicle no older than 3 years

Each participant will receive a remuneration for his/her participation. Language during the sessions will be Italian in Italy and English in The Netherlands.

11.2. Introduction (10 min)

Welcome to this focus group session and first of all thank you for participating! My name is… (names of the focus group moderator) we work at… (name of the company) we'll be your moderators for the coming session. This is … (name of the assistant) also a colleague of us (brief personal intro from all of us and our roles).

We are conducting a study in driver's opinions on different kinds of cars, electric, hybrid and gasoline, gas, diesel and we are specifically interested in your needs and desires related to cars and driving in general. In that sense you cannot give us any irrelevant information or do anything wrong, we are interested in everything you could tell us even if it might sound simple or obvious to you. It will be relevant to us.

Before we start I would kindly request you to put your phones off or in silent mode. This to prevent any interruptions. If you have to make or answer a call, please walk outside to do so.

We'll start with filling out consent forms first. If you have any questions for us about this session or your participation, please ask us (explain what is in the consent forms briefly and hand them out).

At the end of this session or in the following days you'll receive a remuneration for your participation and you can fill out a form for traveling expenses.

Opening question (40 min)

Let's start with a brief introduction by everyone. Could you each state your first name, the kind of vehicle you drive and 3 words that characterize your way of driving? (pick a person to start with).

Follow-up question for all:

- What kinds of trips do you make by car? (separate them in long distance, short distance, urban/city, rural, national, international, etc).
- What are your goals for these trips?
- Could you explain why you make them and why you use a car instead of other transportation?
- What are your positive and negative experiences during these trips? (if you don't get an answer feed them with: think of habits, challenges, information needs, possible choices you have to make, things you do in your car (change radio stations, make phone calls, things you need to pay attention to, road quality).
- Looking at this overview: Which characteristics of your car are important and positive when you make trips? And which are negative or better arranged in other transport?
- Transition question: What do you think if you think of a different type of car? E.g. electric or hybrid or gasoline, would the positive and negative aspects change?

11.3. Slides (30 min)

Show the following slides one by one (see below) and hand them out when there is no beamer available.

Please read these slides and take your time, and if anything is not clear to you, ask us to clarify what we mean with the text on the slide.

1) If you read these description, what pros and cons do you see for these vehicle descriptions?
2) In what scenarios would these cars be useful, and why?
3) In what scenarios wouldn't you want to use these cars or one of them, and why?
4) If you compare these cars to your own car (or one you use often) would it answer your needs, habits and experiences the same way? Lesser? Or even better? And why?
5) What kinds of things would you need to be able to drive these cars? (e.g. think of information, services/communication, charging services, etc)? (if it doesn't take of: would you need information about the car's functioning via a dashboard? What would you want to know? What about an on-board information system? What should it show you? Are the primary driving controls the same?)
6) What about energy consumption and ecological aspects? How important are they for you and why?

7) What about safety aspects? What is important for you in this and why?

HYBRID CARS

- Hybrid cars utilize a **double engine (electric** and **conventional)**: an **electric engine** is **associated** with a **conventional engine** (e.g. petrol or diesel)
- During vehicle operation, therefore, you can use **only** the **electric engine**, only the **internal combustion engine** or **both engines**
- The **vehicle can travel short distances** (e.g. 5 km) in **electrical mode only** at low speed
- The **total number of km which can be covered** is determined by the **available fuel** and does not depend from the battery
- The **battery** is **recharged**:
  - **Recovering the energy** during braking / downhill and **accumulating it in** the **battery** as electric energy
  - **Through** the **energy generated by the conventional engine**, when the level of battery charge falls below a certain threshold

PLUG-IN HYBRID CARS

- Hybrid cars utilize a **double engine (electric and conventional):** an **electric engine** is **associated** with a **conventional engine** (e.g. petrol or diesel)
- During vehicle operation, therefore, you can use **only the electric engine**, only the **internal combustion engine** or **both engines**
- The vehicle can travel **up to about 50 km** in **electric mode only**, even at typical highway speed
- The **total number of km** which can be **covered is** determined by **the sum** of the **available fuel and battery power**
- The **batteries** of the **electric engine** can be recharged also by connecting the vehicle with a cable to a **household power outlet** or through a **public charging point**

ELECTRIC CARS

- **Electric cars** only utilize an **electric engine**
- The **total number of km** which can be **covered is determined the available battery power**
- The **batteries** of the **electric engine** can be recharged also by connecting the vehicle with a cable to a **household power outlet** or through a **public charging point**

11.4.Final assignment (20 min)

We're going to split up now in two groups.

Imagine it's the year 2030, July 5th, and you've taken part in a market survey in which you drove an electric vehicle for three weeks just before this date. Briefly discuss how you would have used your vehicle during these three weeks, what you've done, what your experiences have been in your group. Don't forget we're living in 2030 now! Make notes on the flip-over/paper we have provided.

Now imagine you're a director of a TV commercial that's about the car going to be launched and that you have tested, which aspects of your 2030 experience would you highlight in a commercial that describes a trip before departure from home to your final destination? Think for example of the features and functionality the car has, what information you received from the on-board instruments, in what scenario you drove and with what kinds of goals and the information you need to reach these goals. Use the past tense in your story, write your story down on the flip-over and highlight the most important aspects in detail! What would the camera zoom into? What would you want to emphasize?

Hand out assignment.

Let one participant of each group present their story and camera zoomed details on functionality or interactions and have a brief discussion if others recognize this.

11.5.End the session (10 min)

Thank participants for their participation. Let them fill out the traveling expenses forms and voucher forms and hand out the vouchers as well. Guide them out of the building.