www.openswarm.eu

Call: HORIZON-CL4-2022-DATA-01

Type of action: RIA

Grant agreement: 101093046

Deliverable 1.1: Consolidated Requirement Database

Work Package 1: Requirements and System Architecture

Task Lead: SIA

WP Lead: INRIA

This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101093046.

Funded by the European Union

Document information	
Author(s)	Franz Zeilinger, Thomas Watteyne
Reviewers	Brian Coffey
Submission date	
Due date	30-Jun-2023
Type	DEC
Dissemination level	PU

Document history			
Date	Version	Author(s)	Comments
22-May-2023	1	Thomas Watteyne	Initial version
25-Mar-2023	2	Franz Zeilinger	Adding PoC Inputs, Consolidation
31-May-2023	3	Micael Couceiro	Revised PoC2-related requirements
15-Jun-2023	4	Thomas Watteyne, Franz Zeilinger	Added Introduction, Methodology, Revised whole document
22-Jun-2023	5	Kate O'Riordan, Brian Coffey, Dara Osullivan	Review
29-Jun-2023	6	Franz Zeilinger	Integration of Reviewers Feedback

DISCLAIMER

This technical report is an official deliverable of the OpenSwarm project that has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No.101093046. Contents in this document reflects the views of the authors (i.e. researchers) of the project and not necessarily of the funding source—the European Commission. The report is marked as PUBLIC RELEASE. Reproduction and distribution is limited to OpenSwarm Consortium members and the European Commission.

Table of contents

Introduction

This document specifies the requirements for the final "product" of the OpenSwarm project. By "product", we refer to the technical innovations generated by the three scientific pillars, developed in WP2 "*Orchestration of Collaborative Smart Nodes*", WP3 "*Collaborative Every-Aware AI*" and WP4 "*Energy-Aware Swarm Programming*".

Collecting and formalizing requirements is of utmost importance for the following two main reasons.

First, it serves as a guiding direction for conducting the research. Clearly, research is a non-deterministic activity and, as such, the research path is not necessarily a straight one. Yet, we have collectively gathered and formally written up the requirements, to serve as a general compass to lead our research towards satisfying them.

Second, it serves as a validation vehicle for the results of our research. That is, once an element has been designed (e.g. a communication protocol, an AI algorithm, a swarm compiler), we use the requirements to verify that element answers the requirements.

As a result, for the consortium, this deliverable is really more a reference. It contains a set of requirements which have been gathered and categorized. We gave each requirement a clear name and description, to be able to refer to them throughout the project. In addition, each of the requirements is also linked to one or more specific Proof of Concepts (PoCs) or demos developed within the project. This assures that the requirements are strongly linked to their later real-world application.

Throughout the project, we will show we use the requirements collected in this document both to drive the innovation in WP2, WP3 and WP4, and verify the appropriateness of the results. Moreover, the OpenSwarm implementation (WP5) will be a key element for qualifying the performance of the solution and show how it matches up to the requirements.

Methodology

We have followed a strict and rigorous requirement gathering methodology. This methodology is borrowed from the one used by the JAMA framework[1], arguably the best-in-class requirement gathering tool, used by several partners of the consortium.

We have used the five defined Proof of Concepts (PoCs) as initiators of the requirements gathering and have then complemented these with requirements that go beyond the specific PoCs, in an attempt to be as general as possible.

The methodology is composed of the following steps:

First, we identify the stakeholders. We define as stakeholder any entity, person or function which will interact with the solution, at any point during the lifetime of the solution. Different stakeholders may interact at different stages through the lifetime. Similarly, different stakeholders approach the solution from a different angle, therefore putting different types of requirements onto the solution. By thinking in terms of stakeholders, we force ourselves to collect a set of requirements which offers the widest possible coverage.

Second, we detail the setup of the PoC. This is a general description of the environment in which the solution is used. While the setup is not necessarily a requirement, it paints a picture and allows us -- as we gather the requirements -- and the reader -- as they read through this deliverable – to build up an intuition that colors the requirements.

Third, we write the user stories. This is a typical step in software design and consist of writing down what each for the stakeholders expects from the solution. The format of the user story is typically "as X, I expect Y", where X refers to a stakeholder and Y to a requirements. User stories are an intermediary step between the stakeholders/setup (which provides the "story" around the solution), and the formal requirements.

[1] https://www.jamasoftware.com/

Fourth, we write down the requirements one by one, in such a way that they don't necessarily refer to a specific PoC but keep the input of each PoC still present in the requirements description.

To create the requirements, we started by asking each PoC lead to write up a list of stakeholders, a setup, user stories and list of requirements. We then had a series of meeting where each PoC and its requirements was discussed. After that, we looked at the union of all requirements and added ones which we believe were missed. The next step was to categorize the requirements, which necessarily involved merging duplicates and modifying their description so they are as general as possible.

The remainder of this deliverable is organized as follows. It starts by giving an overview of each PoC. This includes a description of the PoC, a list of stakeholders, details about the setup and finally the user stories that go with it. We then detailed the generalized requirements that were not necessarily captured by the union of the PoCs. The section that follows are the results of this work, a series of categorized requirements.

Overview PoCs

PoC1 - Cities & Community: Renewable Energy Community

The object of consideration in this PoC is a secondary transformer station in an electrical distribution system with its associated grid. Various players are active in this section of the distribution network, such as (active or passive) prosumers, sensors, central jointly operated components (such as storage systems or PV plants) and the distribution system operator (DSO). Individual prosumers may also group themselves together to form one or more energy communities. In this case an energy community operator may be needed.

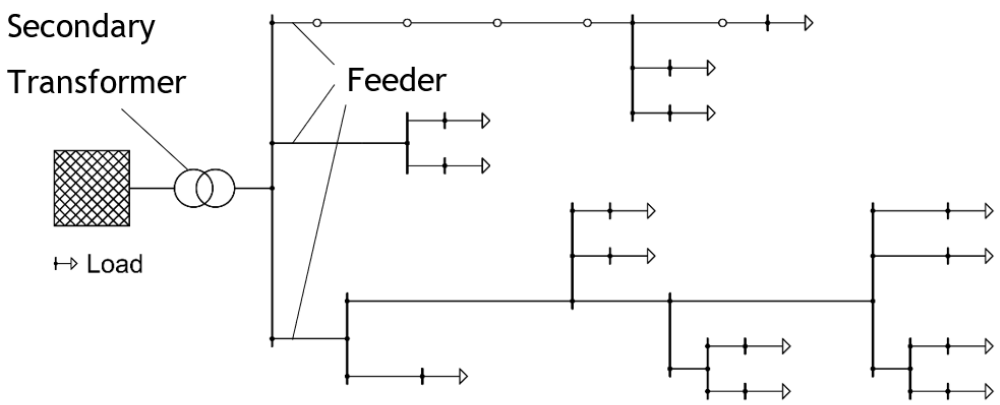

Figure 1: Example of an electrical distribution grid

The objective is to establish an interactive network between all the active actors in the grid section, such that they demonstrate cooperative behavior. In addition, for the optimal use of locally generated energy, it is very important to comply with power grid restrictions such as transformers or transmission line capacity limits or voltage band limits. The idea is to realize this by applying OpenSwarm technologies i.e., via an interactive network, where the interaction between the actors can be achieved without additional configuration effort and the network (swarm) is able to react and adapt dynamically to changes in the environment. Accordingly new actors will be integrated into the system immediately and effortlessly. The same holds for actors leaving the system, which should be possible at any time with the remaining system reconfiguring itself accordingly.

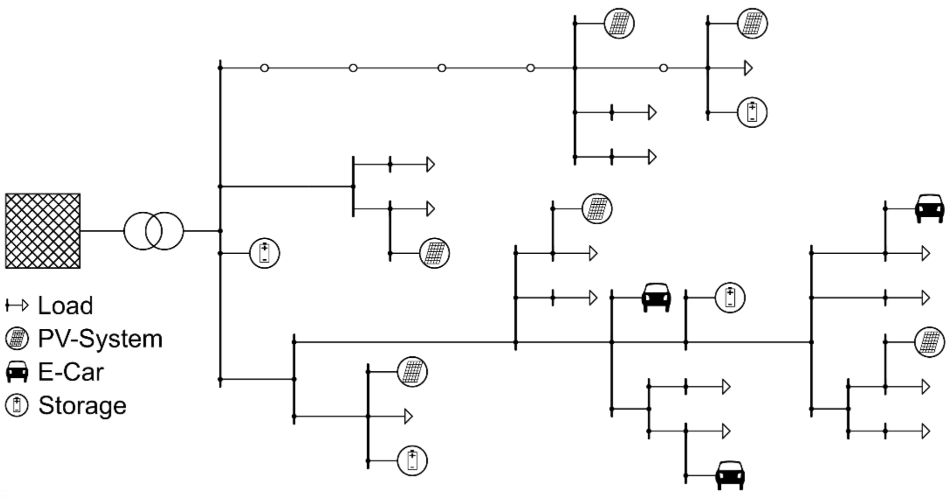

Figure 2: Adding new stakeholder and components to the system

Stakeholder / Components

In the following section the different categories of stakeholders / components interacting with the system are introduced and outlined, beginning with the stakeholders in the system:

- A normal **consumer** is a customer of the DSO which is connected to the power grid. A consumer has load characteristics and a certain guaranteed connection power value. The consumer may have additional components installed in the future, such as a charging station, which, if used at full power, would exceed the originally requested power limit at his grid connection point. In the future consumers may have a suitable communication interface at their system boundary to the DSO for controlling these additional components. This system boundary is in the context of the OpenSwarm project the (smart) meter (swarm connector) of each consumer.
- A **prosumer** is an end-user household that, in addition to its normal electrical load, also has potential for feeding electrical energy back into the grid. For example, a house that has an additional PV system installed could be a prosumer, if it feeds back energy into the system. Active prosumers have the possibility to actively control their power demand or to store energy locally, e.g., via a home automation and/or storage system. In the future active prosumers may have a

suitable communication interface at their system boundary to the DSO. In the following, this system boundary is the (smart) meter (swarm connector) of each prosumer.

- In a future scenario, individual prosumers in the considered part of the distribution grid will also group themselves together to form one or more **energy communities (ECs)**, to share energy locally. There may also be a role here in the form of an **operator** of this **energy community (ECO)**.
- The **distribution system operator (DSO)** is responsible for building, maintaining, and operating the distribution network within its regulatory limits. All entities connected to the distribution network are its customers. To ensure operation, it evaluates, for example, power requests from existing or new network customers (e.g., prosumers) regarding possible grid violations and grants approval or rejects installation of a new system. In future, however, the DSO will be able to influence the power demand of its customers actively to prevent overloading of components or violation of limits in times of high grid utilization. In times of low utilization, customers can then use higher power potential. Thus, the DSO will in the future have the ability to communicate certain control instructions to consumers / prosumers connected to the distribution system (e.g., via their smart meter or a standardized interface to their home management system) to manage power bottlenecks on the grid.

For the proposed system operation the following components must be taken into consideration:

- To enable the DSO to estimate the utilization and monitor the condition of its distribution network, measurement data from the respective network is necessary. This data can be provided by distributed **sensors** in the grid, by the customers' smart meters or by measuring devices that are operated by someone else (e.g., prosumers) but whose data is made available to the DSO. Ideally, the data would be made available to the DSO without much configuration or engineering effort.
- Another future player in the system under consideration are **jointly operated assets,** such as a community owned PV system or a neighborhood battery storage system operated by an EC or the DSO. Another example is a PV system,

which is built and operated by several parties. These components can provide different services to the different participants in the distribution grid, depending on the design of their operation. For example, a central storage facility can support the distribution grid in operation, offer balancing power, and temporarily store locally generated energy for an EC.

- A suitable **communication infrastructure** is necessary to enable the interaction of the DSO with other grid connected entities as well as the swarm-based interaction of individuals. Therefore, every single entity, which should be controlled because it could provide considerable flexibility, has installed a **swarm gateway**.

Setup

It is assumed that the following system description will be covered by an appropriate regulatory regime in the future.

The individual participants in the energy system under consideration are connected in terms of communication or have the possibility to share information via a common network / protocol. This network / protocol makes it possible, to address individual participants directly. It is also possible to define, via a central entity, which participants belong to a certain area (such as a common feeder).

Each participant has an interface to this network / protocol. It is also possible that different protocols exist in parallel for different tasks, such as one for billing and one for exchanging measurement data.

User Stories

Story “Discovery / Onboarding”

- Participants are available / getting online / active / are connected to the grid.
- Everyone can share what "he" can deliver. Maybe like a “Broadcast”
  - Measurement data
  - Technical capabilities
  - Services
  - …

- Information is processed and is made available to the system.
- Participants can also ask for certain needs.
    - E.g., storage capacity
- System provides needed Information.
    - Like, who can provide the needed storage service.

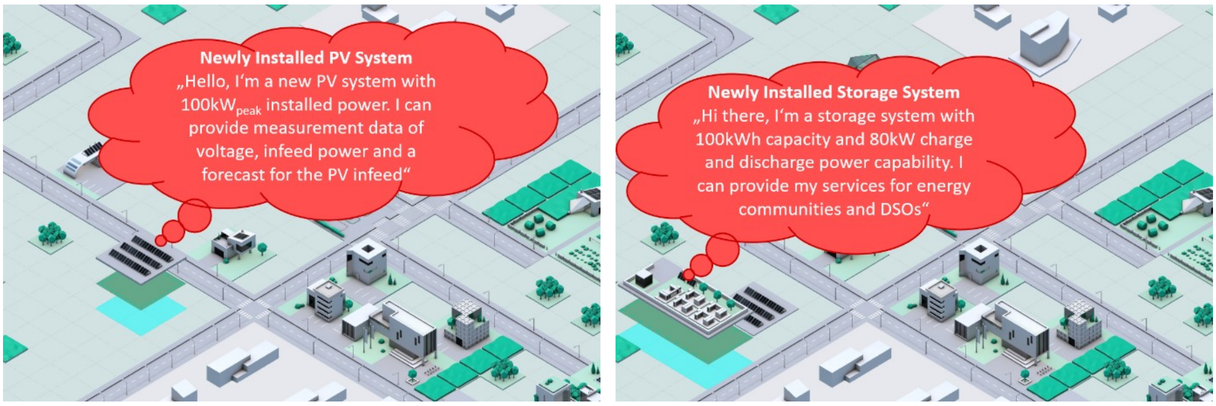

Figure 3: Visual for Story "Discovery / Onboarding"

Story "Operation"

- Participants are linked together to achieve the desired behavior.
    - E.g., PV and storage system together with suitable loads to increase self-sufficiency.
- Based on capabilities and possible services, the updated linked system is operated.
    - The system behavior is negotiated between the participants automatically.
    - All necessary terms are fixed automatically.
- If changes occur (e.g., storage system is shut down due to maintenance) the combination is built up newly with the remaining or additional participants.

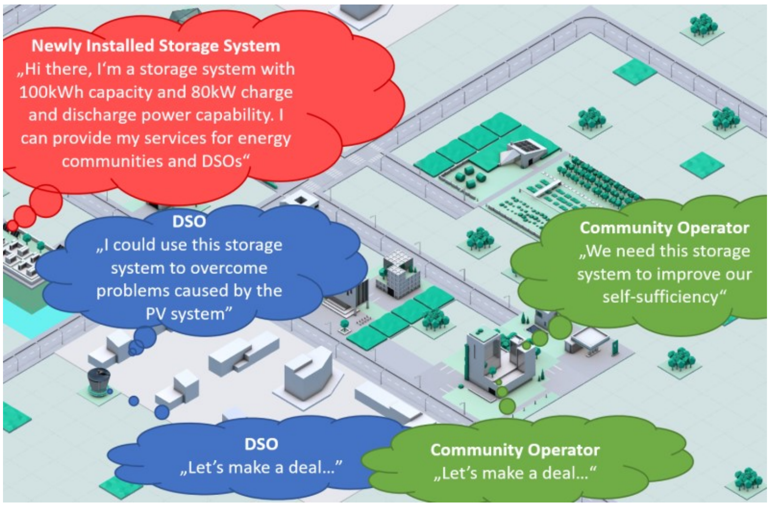

Figure 4: Visual for Story "Operation"

A concrete user story for the above described procedure, is the case of "New PV plant is installed"

- A new PV system has been installed.
- Problem: Grid comes close to load or operational limits as a result (e.g., potential overvoltage situations).
- DSO allows operation of PV plant but needs voltage values.
- DSO requests measured values from the affected string.
- DSO can now estimate more accurately, based on the available data, whether the PV system will lead to the estimated grid problems.
- If problems occur, countermeasures can be taken (e.g., down-regulation of the PV system).

Story "Measured Value"

- Request from a participant to system: "Give me a measured value from location XY (coordinates / grid location / address)"
- The system returns a response with "sensor contact information" if a sensor is nearby.
- If no sensor is nearby, system tries to provide measured value at this location based on available data (via interpolation / cumulation / …) from other sensors.
- If this is also not possible, response is "no sensor available" or "data delivery is not possible".

- In a grid area an Energy Community (EC) is formed.
- Normally each single customers would get an "operating envelope" from the DSO to ensure that all grid limitations are met (reduce power in certain time periods).
- This envelope must be met by everyone on its own, but is not optimal for the EC.
- The EC asks the DSO if they can operate together.
- The DSO can check if the members can reduce potential grid limit violations.
- The DSO tells the EC-Members, who contributes how much to a potential grid limit violation.
- The DSO gives EC members the possibility to access measurement data.
- The DSO tells the EC members which limitations they must meet.
- The EC members together ensure that restrictions are complied with.
- The DSO can verify this based on the available measurement data.

PoC2 - Environmental: Supporting Human Workers in Harvesting Wild Food

Bilberries and cowberries are among the most important non-wood forest products harvested in Europe. Bilberries are used in the treatment or prevention of conditions associated with inflammation, dyslipidemia, cardiovascular disease (CVD), cancer, diabetes, dementia, and other age-related diseases. This attractive, zero-maintenance resource has encouraged an increase in commercial picking. To match the growing demand for flavorful and nutrition-packed wild berries, companies invite many pickers to help with the harvest. Because these are wild berries, pickers spend most of their time roaming through the forest searching for them. Earnings are directly related to berry type, quality, and weight, so pickers typically work 12 to 14-hour days to ensure they find a sizable patch of berries. Working conditions are extremely challenging. Any tool that would map out where there are berries, guiding pickers to the right location, would significantly improve working conditions, which is what this PoC is all about – delivering under-canopy swarms of unmanned aerial vehicles (UAVs) for forestry mapping.

The use of under-canopy swarms of UAVs for forest measurements is a relatively new technology that has yet to be fully explored. Vandapel et al. [1] proposed the use of a single UAV flying under the canopy, computing a network of free space bubbles to form safe paths in foliage-cluttered environments that were earlier measured from above the canopy. Vian and Przybylko [2] developed a patented remote sensing system associated with the UAV that generates tree measurement information to reduce operator dependency. Chisholm et al. [3] implemented a UAV LiDAR for forest surveys along the roadside to model trees that surround the roadside. Hyyppä et al. [4] demonstrated under-canopy UAV laser scanning for accurate forest field measurements. Despite these accomplishments, to our knowledge, there have been no studies or patents addressing the use of UAV swarms for wild berry mapping.

Personas / Stakeholders

These are different personas interacting with the system:

- The **field worker** is a person that works as a berry picker. Without this, field workers would typically work between 12 to 14 hours a day, though the expectation is that, with the proposed technology, 8h shifts can be achieved. Although they do not directly control the UAVs, they benefit from the maps these generate.

- The **swarm operator** is the technician in charge of the correct operation of the swarm of UAVs. They will be the ones deploying the UAVs in the field, switching/charging batteries when needed, monitoring the performance of the swarm, and take corrective action if maintenance is required.
- The **robotics developer** is the person who develops the ROS (Robot Operating System) software which runs on the UAVs and interprets the datasets acquired by the swarm.

Setup

The elements in the PoC2 are as follows:

- The system is deployed in **forestry scenario** with the typical fauna and flora, preferably with berry yields, though not mandatory for the concept addressed in PoC2.
- A **single GNSS** (Global Navigation Satellite System) **base station** is deployed on site, preferably 30 minutes prior to the start of the system, offering real-time kinematics (RTK) corrections to the UAVs up to several kilometers within the forest using LoRa.
- Typically, there is a **single control station** overlooking the swarm system (laptop), managed by the **swarm operator**. Updates are received via cellular communication, when available, or over the same mobile ad-hoc network (MANET) technology to be adopted by UAVs when in close proximity to the system.

Up to **12 UAVs** equipped with stereo cameras, GNSS-RTK (Real Time Kinematic), IMU (Inertial Measurement Unit) and other sensors are deployed on site for single-robot SLAM (Simultaneous Localization and Mapping) and subsequent collaborative mapping.

Figure 5: A single OpenSwarm UAV performing SLAM in the forest.

User Stories

- As a **field worker**, I want to use my own smartphone to quickly access detailed forest maps using an app, like a Google Maps for forests, which shall include multiple layers, such as traversability, elevation and berry yields.
- As a **field worker**, I do not want to worry about the swarm of UAVs nor do I want these to impede with my job; quite the other way around, I just need the collaborative map generated by these, which can even be provided to me after they UAVs have fully explored a given region.
- As a **swarm operator**, I want to assess the status of the swarm and each UAV individually through one quick look at a monitor.
- As a **swarm operator**, I want the system to warn me of battery depletion with, at least, 5 minutes notice, to either organize the landing for battery replacement and/or recharging.
- As a **swarm operator**, I need all the devices in the network to mutually authenticate, and for all the communication to be encrypted (i.e. it cannot be understood by an attacker carrying a sniffer equipment).

- As a **robotics developer**, I want the system to run, either autonomously or not, and produce collaborative maps in an efficient way and on-the-fly, using their in-board processor and state-of-the-art map representations, such as octrees.
- As a **robotics developer**, I want the system to offer a clear Application Programming Interface (API) to the **field worker** app, using web technologies, such as JSON, for them to easily observe the forest maps generated by the system.

PoC3 - Environmental: Ocean Noise Pollution Monitoring

We want to create a system that monitors how many boats sail through a Protected Marine Environment (PMA), and ideally record their speed. PMAs are home to a very diverse ecosystem which is severely impacted by noise pollution, which is mainly coming from motor boats. Today, counting boats is done occasionally by PMA staff on a kayak. Boaters do not intentionally go too fast though PMAs. The goal of continuous monitoring is therefore not to fine boaters, rather to keep track accurately about the pollution, and launch targeted communication campaigns.

PMAs are very pleasant and peaceful places; naturally, boaters want to visit them. This is why there are always buoys in PMAs, as attaching a buoy has no impact on the seafloor, whereas using the anchor does (see below).

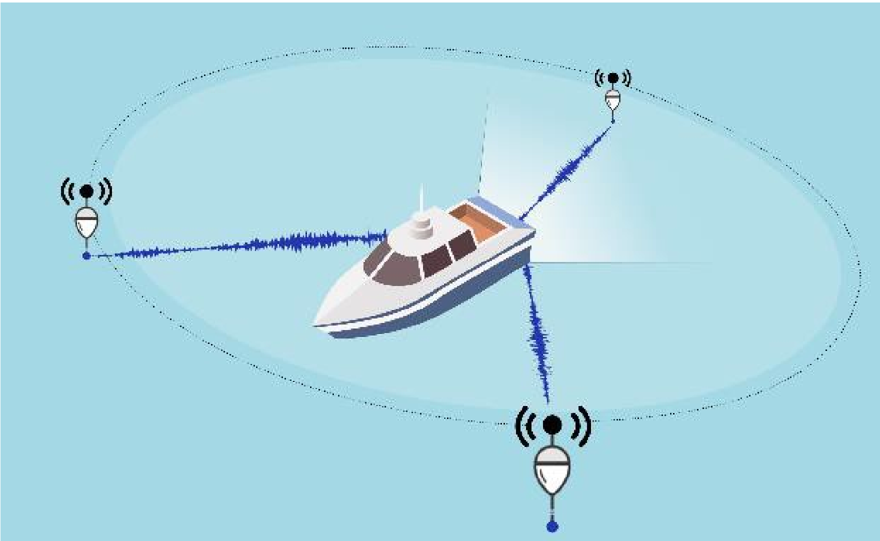

Figure 6: Illustration of the PoC3 Concept

The technical goal of this PoC is to add a hydrophone to some of these buoys, and equip them with advanced electronics to

(1) recognize the sounds of a motor boat,
(2) coordinate between buoys that a sound was recognized
(3) collectively locate the boat through trilateration and
(4) communicate that information to the cloud.

Figure 6 helps illustrate the concept.

While there are thousands of PMAs in Europe, we focus on the PMA in Cap d'Agde, in the South of France. We have excellent work relationships with the staff of that PMA and have had the opportunity of testing other technology there. It is very representative of other PMAs in Europe. The PMA itself is a rectangular piece of water 900m by 500m in size, conveniently located right outside of the entrance of the Cap d'Agde marina, approx. 1.2km from the headquarters of the marina (where we could install hardware) and around the "Fort de Brescou", a tiny island with a medieval looking fort on it (very useful to be able to land there during expeditions).

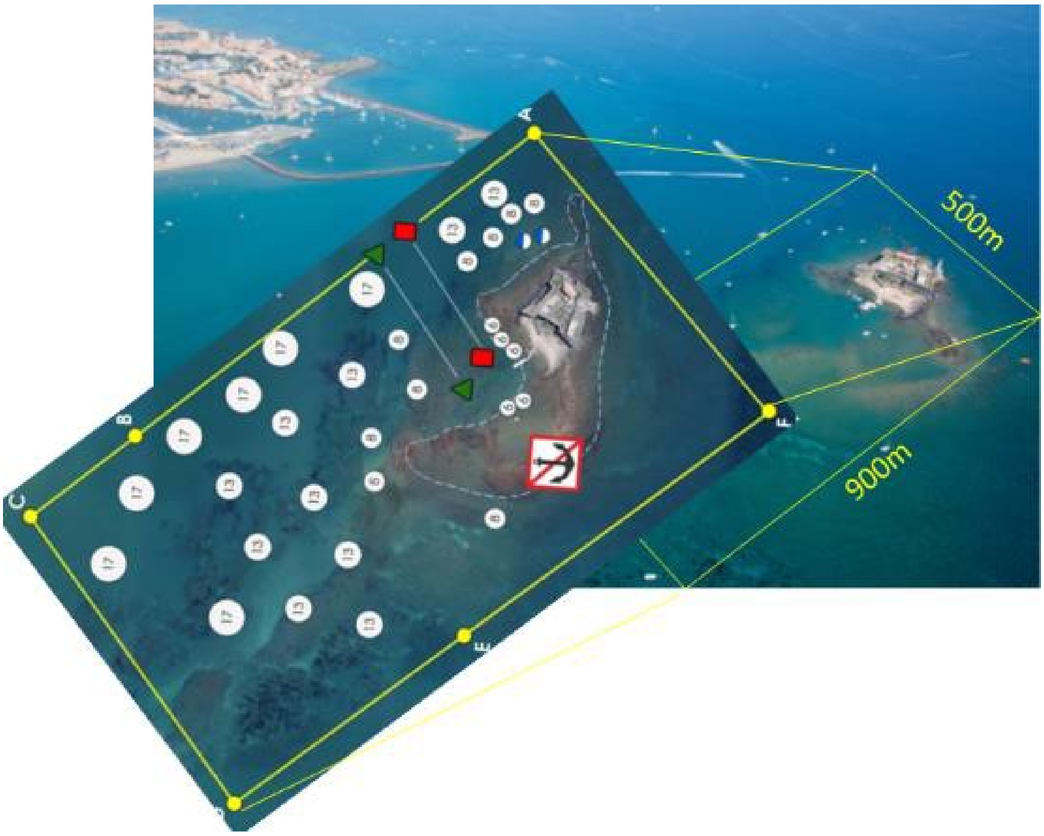

Figure 7: The Cap d'Agde PMA

Stakeholder / Components

- The **marina staff** are the people who are using the final system to monitor the PMA. They will receive the alerts and will have access to the reporting.
- The **network operator** is the technician in charge of the correct operation of the network. They will be the ones installing the equipment, monitoring the performance of the network, and taking corrective action if the network needs maintenance.

- The **application developer** is the person who develops the software which interprets the data received from the machines through the network, and programs the alerting and reporting routines.

Figure 8: The smart buoy

- The **bouy**: Falco has developed a smart buoy. The buoy is a plastic ball filled with air, with a stick sticking out of the water to ease the mooring process for the boat owners. To keep the stick upright, there is a stick-shaped counterweight inside the water. The orange part in Figure 8 is the opening to a little chamber inside the ball, to host the electronics. The plastics used ensure both the ball and the chamber are water-tight. The buoy itself is installed using either a rope or a chain to a block of cement on the sea floor. The rope is preferred as its lighter and less prone to damaging the sea floor. The rule of thumb is to have a rope with a length that corresponds to three times the depth of the water. This means that, as winds and currents shift, the buoy can rotate 360 degrees around its mooring point, with a radius that can be tens of meters. There is an ingenious second ball attached halfway on the rope to keep it from touching the sea floor. The impact of such a buoy on the seafloor ecosystem is limited to the size of the concrete block (which itself typically hosts lots of wildlife). This is much better for the environment than letting boats use their anchor, as this destroys lots of the seafloor as the boat is pushed around its anchor point.

Setup

The system is composed of:

- At least 3 buoys (you need 3 for 2D trilateration) fully equipped with the electronics for the test. There are several options for this:
  - A full buoy as depicture above, with dedicated hardware installed in the chamber
  - An add-on box (or second small buoy) to an existing mooring buoy
- A Marina Web interface
  - Live view of the Buoys status
  - Reporting
- A Monitoring Interface
  - Network Health
  - Buoy Devices Health

User Stories

- As a **marina staff**, I want to know how many boats sailed through the PMA on a per-hour basis.
- As a **marina staff**, I want to know if the buoys in my PMA are connected to the network.
- As a **network operator**, I want to assess the health of the network and the battery level of devices I am supervising through one quick look at a monitor.

- As a **network operator**, I need all the devices in the network to mutually authenticate, and for all the communication to be encrypted (i.e. it cannot be understood by an attacker carrying a sniffer equipment).
- As a **network operator**, I want to receive an alert if the network health degrades below a given limit or if a device disconnects from the network.
- As a **network operator**, I want to be able to change the configuration of a device remotely.
- As an **application developer**, I want the network gateway to produce the data using a standard computer data format such as JSON.
- As an **application developer**, I want the system to offer a clear Application Programming Interface (API) using web technologies such as REST for me to write a program that controls the buoy device configuration (e.g. sampling frequency, detection algorithm parameters…).

PoC4 - Industrial/Health: EHS in industrial production sites

Protecting the environment, conserving our natural resources, fostering the health and performance of employees as well as safeguarding their working conditions are core of social and business commitment at many companies and often even part of the corporate strategy. EHS combines Environmental Protection, Health Management and Safety aspects. This POC focus on safeguarding and addresses: Occupational Health and Safety.

In this OpenSwarm PoC we aim to increase the EHS capabilities of a factory site and the corresponding situational awareness for all humans and machines involved on a factory site. This section outlines the requirements for OpenSwarm to apply a smart sensor network in combination with consumer devices, IT and OT infrastructure. The aim is to seamlessly realize information and application network integrating the operating machines, the digitally empowered factory environment, and finally the digitally empowered human. The specific objective of the network is to continuously collect information of the mentioned systems, identify contexts and change indicators to evaluate whether safety policies and requirements are potentially violated and to communicate them to the human in a timely and context related manner. In parallel humans shall be able to re-configure, change, extend the EHS applications running in the network to adapt to updated EHS scenarios and policies.

Personas

These are different personas interacting with the system:

- The **factory worker** is a person that works on the factory site. They typically work 8h shifts, often moving around the floor, between buildings, and different zones with different EHS requirements, policies, and measures. They are equipped with mobile devices like smartphones, smartwatches, tablets or any other wearable to receive information regarding safety issues, recommendations, etc. or to issue EHS requests and incident information. These personalized smart devices are connected to the network and digitally empower the factory worker to be part of the network both by providing information (directly through input, indirectly through sensor data) and by receiving information from the network.

- The **EHS manager** is a person who defines, implements, and monitors the policies and measures related to EHS on a factory site. Depending on the specific environmental conditions, level of automation, and EHS risks he/she performs risk analysis in different zones of the factory sites, defines EHS policies for these zones, and implements measures to ensure that all policies are operational. He/she monitors the EHS state of the overall factory site and takes responsibility for incidents and implements countermeasures for the future. He/she is also responsible for training all personas operating on the factory site and will be asked if changes in the environment will be implemented.
- The **forklift driver** is a factory worker who drives mobile machines like fork-lifts used for intra-logistics on the factory site.
- The **AGV/AMR operator** is responsible for fleet management and operation of autonomous mobile transport systems on the factory site.
- The **IT & sensor network operator** is the technician in charge of the correct operation of the IT and sensor network. They will be the ones installing the equipment, monitoring the performance of the network, and taking corrective action if the network needs maintenance.
- The **application developer** is the person who develops the software which interprets the data received from the machines through the network, defines the needed business logic of the applications executed on the network. This logic implements the agreed EHS policies and measures.
- The **factory owner** defines the overall Zero-harm and sustainability KPIs for the factory site.

Setup

A factory site has multiple buildings. Each building has heterogeneous production machines on various levels of automation. In addition, there are areas for buffering of material. In-between the production machines and storage areas the AGVs/AMRs are sharing movement areas, where they need to coordinate their interactions to avoid accidents. From a human perspective a factory site is crowded with objects and looks messy. You might have different noise sources with different volume levels. There might be areas with different temperature profiles. The factory workers are equipped with digital connected devices.

The elements in the network are as follows:

- Sensors tags for information sharing, environmental monitoring and localization.
- AGVs/AMRs which have a self-localization and a certain computational power.
- Fork-lifts either equipped with a sensor-tag or a mobile consumer device.
- Mobile Consumer Devices of human operating in the factory site.
- Decentralized Edge Cloud for analytics.

User Stories

Focus Use Case: Safety – "Virtual fences and proximity alerts"

- As a **factory worker**, I want to get context and situation related hints and warnings automatically, like a warning, when I'm entering a dangerous area in the factory.
- As a **factory worker**, I want to get alarms addressing multiple senses of me (visual, acoustical, …) scaled with the severity risk on my mobile devices, when I am directing towards a dangerous situation.
- As a **factory worker**, I want to get visibility (notifications, warnings, etc.) to vehicles that are not in line of sight, to avoid dangerous situations and accidents; (Typically, the machine systems are directly adjacent to the paths and crossings cannot be seen).
- As a **EHS manager** I want to be able to integrate mobile moving equipment (e.g., AGVs / AMRs) and static zones with defined safety zones (static or moving) in order to enable safety warnings with virtual fences for people on the shopfloor.

Use Case: Device and Person Onboarding

- As an **IT & sensor network operator**, I want to easily add any kind of new systems, and network participants. This must be efficient and automized to avoid configuration failures.
- As a **factory worker**, I want to use devices (e.g., smartphone, tablet) to interact with the OpenSwarm network.

Use Case: EHS Configuration

- As a **factory worker**, I want to configure my personal levels of when I'm receiving warnings to get individualized hints and warnings.
- As a **EHS manager** I want to configure policies based which spatial context parameters trigger which behavior in proximity of humans.

Use Case: Localization "Use position of mobile units (AGVs/AMRs) to configure position of mobile sensor tags"

- As a **EHS manager** I want to have an overview where all my sensor nodes are, where my mobile units are and where the humans in the network are.
- As an **IT & sensor network operator** I would like to have independent attachable sensor tags, which can be configured (e.g., their position) based on mobile units with known position configuration.

Use Case: Information Queries "Digital empowered worker can request information from the network"

- As **factory-owner** I want to have the opportunity to get an overview of which of my employees are in a safety-critical area, in order to assess the risks.
- As **application developer** I want a statement on the quality of the data provided by the machines and the network, in order to be able to interpret it correctly and derive meaningful measures.

Use Case: Data Collection "Collect environmental data of the factory site and identify anomalies, potential hazards, and potential areas of EHS improvement"

- As an **EHS manager** I would like the network to perform automatic detection of anomalies and potential hazards.
- As an **EHS manager** I would like to get an overview of the current EHS state and valid regulations in the area of my responsibility.

Use Case: Navigation and Guidance “AGVs/AMRs are re-routed based on human coverage inside the factory site”

- As a **EHS engineer** I would like the network and the AMR/AGVs to take information of the network into account, to adapt navigation planning of the AMR/AGVs based on the human coverage in the proximity and the overall factory site to increase the overall safety level on-site.

PoC5 - Mobility: Moving Networks in Trains

In the mobility sector, the transport of goods plays a crucial role, not least because of climate change and the associated measures to shift the transport of goods from road to rail in order to save CO2. Therefore, not only the availability of trains crucial, but also their reliability. However, freight wagons today are much less digitized than, for example, long-distance passenger trains.. These wagons are purely passive transport elements without power supply or communication with the locomotive. Additional sensors required to monitor critical components such as bogies, bearings or damping systems must therefore be energy self-sufficient and without any dependence on a local energy supply or communication infrastructure.

Since freight trains in particular are often reconfigured depending on the transport order, which changes the composition of the wagons on the one hand and their order on the other, it is difficult to determine both the current global location of the wagons and to ensure communication between the self-sufficient sensors and a central data sink. The process of re-arranging freight trains in shunting yards is a complex and intricate task that requires careful planning, coordination, and precise execution. Shunting yards serve as critical junctions where multiple freight wagons are rearranged to form new train compositions or facilitate the sorting and distribution of cargo. The complexity arises from various factors such as the size of the yard, the number of tracks, the different types of wagons, and the specific requirements of each train's destination. Shunting operations involve intricate maneuvering of locomotives, coupling and decoupling of wagons, and ensuring proper weight distribution and load balancing. It requires expertise in train handling, knowledge of yard layouts, and adherence to safety protocols to prevent accidents or damage to equipment.

Personas

- The **train operator** is a person or a group that must ensure the operational readiness of the trains and the safe operation of these trains. This includes, in particular, the availability of locomotives and wagons, the planning of maintenance work, the provision of replacements in the event of a failure of locomotives or wagons, and the repair of the components in the event of damage.

- The **infrastructure operator** is a person or a legal entity that ensures the provision of the rail infrastructure as well as the safe operation of trains on this infrastructure. This includes capacity planning on public lines, the maintenance of rails, switches, signaling systems and other critical components, as well as the expansion of rail infrastructure depending on capacity utilization forecasts.
- The **train manufacturer** is a legal entity that manufactures current train models and drives the development of future models based on new standards as well as changing customer requirements and usage behavior.
- The **logistics provider** is a person or a legal entity that transports goods by rail. It is dependent on being provided with a sufficient number of functional freight wagons and locomotives in order to be able to fulfil transport orders on time and on budget.

Setup

A freight train consists of several freight cars and at least one locomotive, which is the traction engine. The freight cars are connected to each other via couplers, which usually provide a purely passive connection. A freight car usually has two bogies, which in turn have two axles. The average freight train on the German rail network runs with 25 to 30 freight cars. In the future, freight trains will comprise around 35 wagons, which corresponds to a length of 740 meters. In the U.S., freight trains could reach a total length of several kilometers, corresponding to more than 100 wagons. The locomotive is typically equipped with a gateway, which is connected via cellular data connection to the train operation center and may be used as a central node for additional sensors attached to the freight wagons.

The technical equipment on the land-side of a railway plays a crucial role in ensuring the safe, efficient, and reliable operation of the railway system. This equipment encompasses a diverse range of systems and infrastructure that support various functions, such as train control, signaling and communication. For train control and signaling, the land-side equipment typically includes trackside signaling devices, such as signals, switches, axle counters and track circuits, which regulate the movement of trains and ensure proper routing at junctions. Additionally, centralized control systems, such as interlocking systems, are employed to coordinate and manage train movements, ensuring safe and efficient operation. Communication systems, such as radio-based train-to-ground communication networks, enable communication between trains and the control center, facilitating information exchange and operational coordination.

User Stories

Focus Use Case: Localization – "Where is my freight wagon?"

- As a **train operator**, I want to have quasi real-time visibility of the location of all my freight wagons during operation to ensure efficient management and provide tracking of the cargo to the carrier.
- As a **logistics provider** responsible for managing freight transportation, I want a localization system that enables me to track the location of freight wagons and provides visibility to my customers. This system will allow me to offer real-time updates, improve customer satisfaction, and effectively manage the transportation process.
- As an **infrastructure operator** responsible for maintaining railway networks, I want to have a localization system that helps me monitor the location of freight wagons passing through my infrastructure. This system will enable me to ensure the safety of the tracks, plan maintenance activities, and manage any disruptions effectively.
- As a **train manufacturer**, I want to provide a localization system as part of the train's onboard systems to enhance the operational capabilities of my freight wagons. This system will allow operators and maintenance teams to monitor wagon locations, perform diagnostics, and improve overall performance.

Focus Use Case: Monitoring – "Status of critical components"

- As a **train operator**, I want a comprehensive monitoringsystem for critical components of freight wagons to ensure safe and reliable operations. This system will enable me to proactively identify any issues or abnormalities, minimize downtime, and maintain efficient freight transport.

- As an **infrastructure operator** responsible for maintaining railway networks, I want a monitoring system for critical components of rail infrastructure to ensure safety, efficiency, and timely maintenance. This system will help me detect and address potential issues before they lead to failures or disruptions.
- As a **logistics provider** responsible for freight transportation, I want a monitoring system that allows me to track the condition of critical components in freight wagons or rail infrastructure to ensure the safety and reliability of the cargo being transported.
- As a **train manufacturer**, I want to provide a robust component monitoring system as part of my freight wagons or infrastructure equipment to ensure their optimal performance, enhance safety, and minimize maintenance costs.
- As a train manufacturer, I want to provide a robust component monitoring system as part of my freight wagons or infrastructure equipment to learn from the usage and behavior of components during operation to improve the development of the next generation of equipment.

Focus Use Case: Composition– "Joining freight wagons to form a train"

- As a **train operator**, I want to dynamically form a self-reconfigurable train-wide sensor network by intelligently deploying and reconfiguring sensor-equipped freight wagons based on the specific needs of each journey. This network will provide real-time data and enable flexible monitoring and optimization of train operations.
- As an **infrastructure operator** responsible for maintaining railway networks, I want to leverage a train-wide sensor network formed by equipping freight wagons with sensors. This network will enable me to monitor the condition of the rail infrastructure, identify potential failures, and schedule maintenance activities efficiently.
- As a **train manufacturer**, I want to design and provide freight wagons equipped with sensors to enable the dynamic formation of a train-wide self-reconfigurable sensor network. This network will allow train operators, infrastructure operators, and carriers to flexibly configure and adapt the sensor deployment based on their specific requirements.

Generalized Requirements

Of course, the above listed use cases of each outlined PoC only cover a portion of all possible use cases of OpenSwarm technology. Therefore, efforts were made to complement also more generalized requirements, both with additional use cases and with an attempt at providing a first level of consolidation. These following generalized requirements are later merged with the use case requirements in section "Consolidated Requirements".

User Experience

[impacts WP2, WP3, WP4, WP5] The ambition of OpenSwarm is to create the building blocks that can be integrated directly into an end product. As such, the OpenSwarm implementation should be treated as a product, and its development should be done accordingly.

In particular, user experience should be considered when developing any "executable". We define executable as a solution that an end user interacts with. We call end user a person who is reasonably comfortable with technology but has not participated in the development of the solution.

In particular, we want an end user to be able to get the solution running in less than 15min. This timespan goes from unboxing to running and involves any installation and configuration. Getting to that low limit involves several aspects, including creating a very simple and to-the-point quick start guide, streamlining the onboarding/startup process, and creating simple and intuitive user interfaces.

Test Coverage

[impacts WP2, WP3, WP4, WP5] The ambition of OpenSwarm is to create a large body of source code. One of the main challenges is to verify and validate that the different building blocks of that solution work, including when operating together. The universally accepted technique to achieving this is Continuous Integration (CI), where tests are run each time new code is pushed.

The OpenSwarm source code consists of several repositories, hosted by development environments such as GitHub. Developers introduce changes to the code hosted on them using utilities such as Git. Within this development environment, we want 80% of the code to be Continuously Integrated.

This may involve developing test cases using a testing framework such as pytest and having test running such as GitHub Action execute the tests at each Git push and pull-request. The environment may be set in such a way that merging of the new code is disabled if the tests fail. In many/most cases, tests can run on any computer, including the developer's computer, or a server in the GitHub farm of runners. In some cases, the code runs on dedicated hardware. In that case, the runner may be a local machine with hardware attached to it.

Security

[impacts WP2, WP5] Security is a term that encompasses several aspects, including the traditional Confidentiality, Integrity, Authentication, but also more subtle aspects such as secure joining or the validation of a firmware image.

T2.2 and T2.3 are dedicated to security; it is hence not the intent of this document to replicate that work. That being said, OpenSwarm must thrive to develop solutions in which security cannot be turned off, and in which security does not impact the overall performance by more than 10%.

Secure Joining

[impacts WP2, WP5] Secure joining is a subset of security, which focuses specifically on the process by which devices integrate into a swarm. OpenSwarm must offer a true "zero-touch" experience: the devices must come loaded with whatever is necessary to allow for them to mutually authenticate with the swarm, and join that securely without requiring the end user to "touch" them to e.g. install a pre-shared key.

Consolidated Requirements

In this section consists of a List of consolidated requirements, which were derived out of the inputs from each PoC as well as the generalized needs for the system. Each of the defined requirements has its own identifier in the form "Req.X. Name". Each requirement is also associated with the PoCs or the general requirements (list in each paragraph "Related to PoCs: X"). The more detailed requirements description has stated its source PoC in a paragraph in the form

> (PoCX) It is kept in this document, to have a clear view on the needs of each single PoC for future reference. The corresponding input is held in this style.

Also, it identifies the "requirements owner", who can then be contacted during the development process to clear questions, make more detailed specifications and give additional information.

During the consolidation process some remarks and inputs were added to this document, to get a clearer understanding of the specific requirements or the link of it to a PoC. This kind of Input is marked at the beginning of the paragraph in the form

> *(PoCX Remark)* The remark itself is held in this style, to distinguish it more clearly between what is the requirement and what is the context being provided for it.

The listed Requirements are in no specific order, the numbering indicates no priority of the requirement.

Additionally, the identified requirements are grouped together into the following top-level requirements categories:

- **Communication –** related to the communication between the components of the system
- **Components** - related to the realization of single system components, especially in terms of hardware
- **Energy Consumption** – related to power, power supply and energy needs of single components
- **Configuration –** related to configuration of components and system parts as well as the whole system

- **System Management** – related to tasks and functions needed during system operation
- **Data Processing** – related to data streams, how these are processed, stored, distributed
- **Use Cases** – related to special needs and functions out of the user stories / use cases / PoCs. These are naturally very specific and should be seen as "on top" of the more general requirements. But these functionalities are needed to make the demonstration of the PoCs possible.
- **System** – related to more general system relevant needs

Later in the course of the project, a way to track the requirements hast to be decided later. E.g. there could be the possibility that the requirements are tracked in JAMA or Confluence by INRIA. This must be decided out of the needs of WP2, WP3 and WP4.

Also, there are very few acceptance criteria formulated for each of the requirements. This was not done in this document because of the early stage of development process. These acceptance criteria must be added later. Also are some of the requirements not formulated very specific. This indicates that there is still

Communication Requirements

Req.1. Regulatory Compliance

Any wireless communication solution developed by OpenSwarm must comply with the wireless comms regulations.,. This includes regulations related to maximum transmit power, radio duty-cycle and coexistence-related elements.

Related to PoCs: 1, 2, 3, 4, 5, General

(PoC1) The system must comply with relevant regulations, standards, and policies, such as those related to data privacy, cybersecurity, and energy market rules.

(PoC4) The system must comply with relevant regulations, standards, and policies, such as those related to data privacy, cybersecurity, and EHS.

(PoC5) Compliance with Privacy Regulations: The network should comply with privacy regulations and guidelines, ensuring the protection of personal data and user privacy rights. Compliance with privacy regulations safeguards the privacy and confidentiality of sensitive information collected by the sensor network, maintaining the trust and compliance of stakeholders.

Req.2. General Communication Needs

A reliable and efficient communication between the OpenSwarm components is required to share and receive information, and to coordinate with each other via a common network / protocol. It must be possible that different protocols exist in parallel.

Related to PoCs: 1, 2

(PoC1) The individual participants are connected in terms of communication or have the possibility to share information via a common network / protocol. Each participant has an interface to this network / protocol. It must be possible that different protocols exist in parallel for different tasks, such as one for billing and one for exchanging measurement data.

(PoC2) The unmanned aerial vehicles (UAVs) should have reliable and efficient communication to share map data and coordinate with each other, benefiting from technologies preferably well-suited to implement mobile ad-hoc networks (MANETs), with multi-hop routing if possible.

(General Remark) This requirement must be specified and defined in more detail later during the project.

Req.3. Sensor Network Integration

OpenSwarm components must be able to integrate into an infrastructure based on IP communication.

Related to PoCs: 4

(PoC4) The sensor network shall be able to integrate into an infrastructure based on IP communication as all required higher-level services and components base their communication on IP standards. This may be achieved by defining a reference IP application. In optimal case it is completely supporting IP features and there are no limitations. If there are limitation it should be possible to notify them to the applications and other devices.

Req.4. Communication patterns

The communication must support multiple communication patterns including one-way patterns and two-way patterns to support information push (broadcast and point-to-point), information query, and action calls.

Related to PoCs: 1, 4

(PoC1) This communication makes it possible to address individual participants directly. It is also possible to define, via a central entity, which participants belong to a certain area.

(PoC4) The middleware and API for the network shall support multiple communication patterns including one-way patterns and two-way patterns to support information push, information query, and action calls.

Req.5. Network Self-Optimization

The communication network between the system components must possess self-optimization capabilities, continuously adapting and optimizing its performance, resource allocation, or communication parameters.

Related to PoCs: 1, 4, 5

(PoC5) The network should possess self-optimization capabilities, continuously adapting and optimizing its performance, resource allocation, or communication parameters. Network self-optimization enhances the network's efficiency, adaptability, and resource utilization, ensuring optimal network performance under changing conditions or requirements.

(PoC4 Remark) This will also be relevant for PoC4. There is of course a definition of acceptance criteria needed. E.g. which optimization is expected after one minute after a change? These criteria must be defined before the development process in WP2, WP3 and WP4 is started.

(PoC1 Remark) The above requirement is also needed for the communication within PoC1.

Req.6. Interference Mitigation

The communication must include mechanisms to mitigate interference from external sources or coexisting networks, ensuring reliable and accurate data transmission.

Related to PoCs: 5

(PoC5) The network should include mechanisms to mitigate interference from external sources or coexisting networks, ensuring reliable and accurate data transmission. Interference mitigation minimizes the impact of signal interference on the network's performance and data quality, enhancing its reliability and robustness.

(General Remark) Later in the project a particular standard for compliance must be defined to give some guidance as to what is being referred to here.

Req.7. Ad-Hoc Communication

The components must support ad-hoc communication enabling direct communication between single system components without relying on centralized infrastructure.

Related to PoCs: 2, 4, 5

(PoC5) The sensor network should support ad-hoc communication between sensors or objects, enabling direct communication without relying on centralized infrastructure. Ad-hoc communication ensures robust and flexible communication within the network, especially in scenarios where infrastructure-based communication is limited or unavailable.

(PoC2 Remark) The system of PoC2 also falls in this requirement, though there is some redundancy with "Req.2 General Communication".

(PoC4 Remark) This will also be relevant for PoC4 and relates to „Focus Use Case: Safety – "Virtual fences and proximity alerts"".

Req.8. Addressing

Addressing must be data-centric and not device-centric.

Related to PoCs: 4

(PoC4) On middleware and application layer addressing should be data-centric and not device-centric.

(PoC4 Remark) Req.3 "Sensor Network Integration" and Req.8 fit together, as Req.3 relates to network layer 3 (&4) and Req.8 to a network layer 7 or even application. So, from this perspective Req.8 is valid and can be fulfilled within the same system on top of Req.3.

Req.9. Quality of Service (QoS)

The communication must provide reliable and consistent QoS, ensuring sufficient bandwidth, low packet loss, minimal delay, and other performance metrics.

Related to PoCs: 5

> (PoC5) The sensor network should provide reliable and consistent QoS, ensuring sufficient bandwidth, low packet loss, minimal delay, and other performance metrics. QoS guarantees the delivery of sensor data and communication requirements, meeting the expectations of applications and stakeholders relying on the network.

> *(General Remark)* This requirement must be specified and defined in more detail later during the project.

Components and their Requirements

Req.10. Swarm Gateway / Swarm Connector

Every component in the system must be connected via a swarm connector. A swarm gateway is a central unit for controlling the system and for centralized tasks, like data processing (see section “Data Processing Requirements”).

Related to PoCs: 1, 3, 5

> (PoC1) A suitable communication infrastructure is necessary to enable the interaction of the DSO with grid connected entities as well as the swarm-based interaction of individuals. Therefore, every single entity with considerable flexibility has installed a swarm connector.

> (PoC3) A single swarm gateway MUST be able to service the entire Protected Marine Environment (PMA).

> (PoC5) The locomotive of a freight train is equipped with a gateway, which is connected via cellular data connection to the train operation center and is used as a central node for additional sensors attached to the freight wagons.

Req.11. Sensor Requirements

Sensors can be installed in the physical system and must be able to perform certain measurement tasks. The measurements are provided to the OpenSwarm system over a swarm connector. The QoS of the transferred data can be adjusted depending on monitoring needs (see section "Data Processing Requirements") and energy usage needs (see "Energy Consumption Related Requirements").

Related to PoCs: 1, 5

> (PoC1) To enable the DSO to estimate the utilization and monitor the condition of its distribution network at all, measurement data (voltages, currents, phase angles) from the respective network is necessary. On the one hand, this data can be provided by distributed sensors in the grid, by the customers' smart meters or by measuring devices that are operated by someone else (e.g., prosumers) but whose data is made available to the DSO. Ideally, the data would be made available to the DSO without much configuration or engineering effort.

> (PoC5) A train-wide sensor network formed by equipping freight wagons with sensors. This network will enable to monitor the condition of the rail infrastructure, identify potential failures, and schedule maintenance activities efficiently.

Req.12. Mobile Devices

Mobile devices like wearables, smart phones or VR-headsets must be part of the OpenSwarm system or must be integratable into the OpenSwarm system. They act as sensors, interface to the humans in the system and receive configuration.

Related to PoCs: 4

> (PoC4) Mobile consumer devices (e.g., smartphones, tablets, wearables) need to be part of EHS sensor network as they represent the digital empowered human. They are both a data source for the network (e.g., information from attached sensors, configurations), but also the interface to the human to provide information, warnings, etc. and to receive configuration and expected network behavior, policy, programming updates from the human.

Energy Consumption Related Requirements

Req.13. Power / Energy Consumption

The developed components must have a low power / energy consumption to support longer battery lifetime, energy harvesting and energy efficiency.

Related to PoCs: 2, 3, 4

(PoC2) Low power communication technologies should be considered since unmanned aerial vehicles (UAVs) mustbe battery powered. But most of the power consumption comes from the motors. So, even if using technology with higher power consumption like WiFi, it should have only the effect of reducing the flight time by one minute.

(PoC3) The buoy devices MUST be battery powered. The gateway MAY be powered by plugging it into a standard 115V/220V battery socket in the control station.

(PoC4) Energy efficiency: Sensor tags shall be able to run more than one year and independent from any fixed power source.

(PoC 1 Remark) The components under consideration are mainly installed in the electrical distribution grid and are connected therefore to a constant power source, so no limitations are present. But in terms of energy efficiency a low power consumption of constantly connected devices ("Stand-by-Power") is desirable.

(PoC5 Remark) Leveraging energy harvesting a low power consumption is needed (see "Req.15 Energy Harvesting and Efficiency".

Req.14. Battery Lifetime

The OpenSwarm solution must allow to operate without battery replacement for at least 150% of its target operating time.

Related to PoCs: 2, 3, 4, General

(PoC2) The unmanned aerial vehicles (UAVs) MUST operate for, at least, 15 minutes in the normal operation conditions.

(PoC4) Energy efficiency: Sensor tags shall be able to run more than one year and independent from any fixed power source.

(PoC3) All the battery powered elements MUST operate for at least 8 months in the normal operation conditions (as buoy are removed from the water and stored on land during winter).

(General) We define as "low-power" a solution which can operate without battery replacement for at least 150% of it target operating time. That is, if a device is installed in an environment where batteries can be replaced every year, it should be able to operate for 1.5 years on those batteries.

Req.15. Energy Harvesting and Efficiency

OpenSwarm should be able to support also energy harvesting mechanisms.

Related to PoCs: 5

(PoC5) The network should support energy harvesting mechanisms and employ energy-efficient techniques to reduce reliance on external power sources and optimize overall energy consumption. Energy harvesting and efficiency promote sustainable operation and reduce the environmental impact of the sensor network, prolonging battery life and minimizing energy costs.

(General Remark) PoC5 is the only one to require energy harvesting. This has to be investigated in more detail later during the project.

Configuration Requirements

Req.16. Discovery

The OpenSwarm system must support component discovery and registration, allowing newly introduced objects to join the system seamlessly.

Related to PoCs: 1, 2, 4, 5

(PoC1) There must be the possibility that the different participants with their strongly differing characteristics and abilities can find each other.

(PoC4) There must be the possibility that the different participants with their strongly differing characteristics and abilities can find each other.

(PoC5) The network should support object discovery and registration, allowing newly introduced objects to join the network seamlessly. Object discovery and registration enable the network to identify and incorporate new movable objects, expanding the coverage and capabilities of the sensor network.

(PoC2 Remark) PoC5 is here very similar to PoC2. As a MANET, it is likely that some nodes (UAVs) will get disconnected from time to time due to their mobile nature. Therefore, the network is expected to be constantly on the look-out for UAVs and incorporate them dynamically.

Req.17. Auto Configuration / Negotiation

There must be mechanisms for the OpenSwarm components to receive context based configuration and shared information in the system automatically. The components must have the ability to self-configure, allowing the components to automatically discover and connect with neighboring components.

Related to PoCs: 1, 2, 4, 5, General

(PoC1) There must be an opportunity for participants to make independent "arrangements" to achieve a specific goal or goals. This process should be automated, within a rough framework defined beforehand.

(PoC4) **Auto-configuration:** New nodes in the network should receive context based configuration and shared information in the system automatically. If a node is replaced by a new node and takes over its full role, there should be a way to copy the configuration from one node to another.

(PoC5) **Self-Configuration**: The sensor network should be capable of self-configuration, allowing the sensors to automatically discover and connect with neighboring sensors or objects. Self-configuration enables the network to adapt to changes in the environment and the movement of objects without manual intervention.

(General) **Secure joining** is a subset of security, which focuses specifically on the process by which devices integrate a swarm. OpenSwarm must offer a true "zero-touch" experience: the devices must come loaded with whatever is necessary to allow for them to mutually authenticate with the swarm, and join that securely without requiring the end user to "touch" them to e.g. install a pre-shared key.

(PoC2 Remark) PoC5 is here very similar to PoC2.

Req.18. Adaptability / Change detection

The OpenSwarm system detects when changes have occurred. The system must adapt to dynamic changes in the communication and components topology. These changes may be caused by the movement of components, adding or removing of components, automatically establishing and maintaining communication and functional links between the components.

Related to PoCs: 1, 2, 4, 5

(PoC1) There must be a way for the system to recognize when changes have occurred. If this is the case, the previously created contracts must be redetermined. Either by including new participants or removing those that have fallen out of the system.

(PoC5) The sensor network should be adaptable, capable of adjusting to changing network conditions, sensor deployments, and operational requirements. Adaptability allows the network to respond to variations in object movement, dynamic environments, or evolving application needs, ensuring continued effectiveness and relevance.

(PoC5, PoC2) **Dynamic Network Topology**: The network should adapt to dynamic changes in the network topology caused by the movement of objects, automatically establishing and maintaining communication links. Dynamic network topology allows the network to remain connected and operational even when objects move or new objects join the network.

(PoC5, PoC4) **Self-Healing and Fault Tolerance**: The network should possess self-healing capabilities to recover from communication failures or sensor malfunctions, ensuring continuous operation and fault tolerance. Self-healing and fault tolerance enhance the reliability and resilience of the sensor network, minimizing downtime and maintaining data collection and communication.

(PoC4 Remark) **Self-Healing and Fault Tolerance**: We could also play with fault-tolerance in PoC2 by increasing the connectivity of the MANET. For instance, ensuring biconnectivity would also ensure that the communication is not disrupted even when one UAV fails / gets out of reach. It is not a mandatory requirement for PoC2 but could be considered.

(PoC5) **Scalability and Flexibility**: The network should be scalable to accommodate varying numbers of sensors and movable objects, allowing the network to grow or shrink dynamically. Scalability and flexibility ensure that the sensor network can adapt to changing deployment requirements and accommodate the addition or removal of sensors or objects.

Req.19. Self-description / Awareness of own needs

All system entities must provide mechanisms to be able to describe themselves including at minimum their identity, their type, their location. In addition, each component must have some sort of description or definition of what its needs / goals are.

Related to PoCs: 1, 4

(PoC1) Each participant must have some sort of description or definition of what its needs / goals are. This is a major input into the negotiation process to find an optimal solution (maybe a local optimal solution) for each participant.

(PoC4) **Self-description**: All system entities need to provide mechanisms to be able to describe themselves including at minimum their identity, their type, their location.

Req.20. Localization

Components must have at least a rudimentary information about their localization.

Related to PoCs: 1, 2, 4

(PoC1) Components should have at least a rudimentary information about their localization (e.g. Geo-Coordinates, Address)

(PoC2) UAVs need to be self-aware of their localization, combining SLAM with GNSS-.RTK to maintain a single common reference (GNSS base station).

(PoC4) Swarm gateways, sensors or sensor tags in the network shall be able to determine their position relative to each other.

System Management Requirements

Req.21. Application Deployment

The Open Swarm System must support the usage of the whole computational continuum from the components to the cloud to deploy and manage applications on each component.

Related to PoCs: 1, 4, General

(PoC4) The system shall support the usage of the whole computational continuum provided by the system from the energy efficient sensor tag to mobile moving computing resources (on board computers, smart phones), to mobile applications, to on-site edge-cloud systems. It should be possible to update applications on all network nodes in the system.

(PoC1 Remark) This function is necessary for a continuous integration and save system operation (to perform updates).

Req.22. System and Device Management / Application Performance Monitoring

The system must support monitoring, reconfiguration, and error management of each single node but also the overall system.

Related to PoCs: 1, 2, 3, 4, 5, General

(PoC4) The system shall support monitoring, reconfiguration, and error management of each single node but also the overall system. It should be possible to update applications on all network nodes in the system.

(All) "Application Performance Monitoring" (APM) refers to the routines that inform the end user of the overall health of the swarm. Health can include aspects such as predicted lifetime, RAM usage, etc. We need the APM modules to generate performance indicators which are with 20% of the real values.

(PoC1 Remark) This function is necessary for a continuous integration and save system operation (to perform updates).

Data Processing Requirements

Req.23. GDPR Compliance

Any data collected and processed by technology developed by the OpenSwarm project must do this in a way to be in full compliance of the General Data Protection Regulation (GDPR). A general description of the regulation can be found at https://gdpr.eu/what-is-gdpr/.

Related to PoCs: 1, 2, 3, 4, 5, General

Req.24. Information Caching

All nodes must be able to cache local information (e.g., sensor data), which can be queried by other nodes in the system.

Related to PoCs: 4

Req.25. External Information Storage

Nodes of the network (e.g., sensor tags) must be able to save information from and share saved information to all other entities in the proximity who are part of the network.

Related to PoCs: 4

Req.26. Distributed Data Fusion

The sensor network or a central component (like die OpenSwarm gateway) must facilitate distributed data fusion, allowing sensors to collaboratively process and aggregate data collected from multiple sources. Distributed data fusion enhances the accuracy and reliability of sensor data by combining information from multiple sensors, improving decision-making and system performance.

Related to PoCs: 5

Req.27. Context Identification

The Open Swarm system must be able to define and identify grouping contexts either based on application demands or proximity.

Related to PoCs: 4

(PoC4) The network shall be able to define and identify group contexts either based on application demands or proximity.This must be done in a way that system nodes are part of the network, soa specific group can be identified.. They form for a certain period of time an envelope of interest for a specific purpose.

(PoC1 Remark) This is also a requirement for PoC1.

Req.28. State Synchronization

The system must support a mechanism for state synchronization with consistency keeping. Especially must this mechanism work with respect to short connectivity drop-outs of mobile nodes.

Related to PoCs: 4

Req.29. Latency / Real-time Data Processing

Depending on the use case, the Open Swarm System must be able to allow low latency data exchange time synchronization between the components and almost real time data processing.

Related to PoCs: 1, 2, 3, 4, 5

(PoC1) The system must be able to process data in real-time to provide timely information to the actors and enable them to react to changes dynamically.

(PoC2) The data produced by a robot much reach another robot in less than 5s.

(PoC3) The data produced by a buoy must reach the gateway in less than 10s.

(PoC4) The system must be able to process data which is safety critical in real-time to provide timely information to the actors and enable them to react to changes dynamically.

(PoC5) The sensor network should exhibit low latency, minimizing the delay in data transmission and communication between sensors and objects. Low latency enables timely data collection, event detection, and response, which is crucial for applications requiring real-time monitoring or control.

(PoC5) **Time Synchronization**: The network should support time synchronization among sensors and objects to ensure accurate data coordination, event correlation, and temporal analysis. Time synchronization enables accurate time-stamping of sensor data, facilitates data fusion, and supports synchronization-based applications, such as coordinated control or event sequencing.

(General Remark) This requirement must be specified and defined in more detail later during the project.

Use Case Functions

Req.30. Boat Detection

The system must be able to detect the presence of a boat within 5 minutes.

Related to PoCs: 3

Req.31. Reporting of Boat Detection

Detection data reports of a boat detection must be updated at least every hour.

Related to PoCs: 3

(General Remark) This requirement is a needed functionality for the source PoC. This functionality must be built up on the basic requirements of the OpenSwarm system.

Req.32. Dynamic Object Tracking

The network must support dynamic object tracking, continuously monitoring the location and movement of objects within the network. Dynamic object tracking enables real-time monitoring and data collection for applications such as asset tracking, surveillance, or environmental monitoring.

Related to PoCs: 5

Req.33. Proximity Detection

The network must be able to detect proximity between all entities of the system and provide information regarding distance and direction between each entity. Accuracy requirements depend on the use case.

Related to PoCs: 4

System Requirements

Req.34. Interoperability

The system must be able to support multiple types of components of different technologies or different vendors.

Related to PoCs: 1, 4, 5

(PoC1) The system must be able to support multiple types of devices and technologies to enable the participation of all actors, such as prosumers, sensors, storage systems, PV plants, and energy communities.

(PoC4) The system must be able to support multiple types of devices and technologies to enable the participation of all actors, this included consumer devices, integration with enterprise IT, and OT systems e.g. from production machines.

(PoC5) The sensor network should support interoperability, allowing seamless integration and communication with sensors and systems from different vendors or technologies. Interoperability facilitates the integration of sensors and objects from diverse sources, enabling the network to benefit from a broader range of capabilities and data sources.

Req.35. Scalability

The Open Swarm System must be scalable and able to accommodate the integration of new actors at any time without disrupting the operation of the system.

Related to PoCs: 1, 4, 5, General

(PoC1) The system must be scalable and able to accommodate the integration of new actors at any time without disrupting the operation of the system.

(PoC4) The system must be scalable and able to accommodate the integration of new actors at any time without disrupting the operation of the system.

(PoC5) The sensor network should be scalable, capable of accommodating many sensors and movable objects without significant performance degradation. Scalability ensures that the network can handle increasing sensor deployments and support the addition of new objects while maintaining efficient data transmission and processing.

Req.36. Maintainability

The sensor network must be designed for ease of maintenance, allowing efficient troubleshooting, repair, and replacement of sensors, objects, or network components. Maintainability minimizes downtime, reduces operational costs, and ensures the longevity and sustainability of the sensor network.

Related to PoCs: 1, 4, 5, General

Req.37. Security

The system must be secure and able to protect against unauthorized access, data breaches, and cyberattacks.

Related to PoCs: 1, 2, 3, 4, 5, General

(PoC1) The system must be secure and able to protect against unauthorized access, data breaches, and cyberattacks.

(PoC2) The security solution in place **MUST offer mutual authentication**. Initial keying material **MAY be provided** by the network operator during deployments or happen automatically using some sort of Public Key Infrastructure (PKI).

(PoC3) The security solution in place MUST offer mutual authentication. Initial keying material MAY be provided by the network operator during deployments or happen automatically using some sort of Public Key Infrastructure (PKI).

(PoC4) The system must be secure and able to protect against unauthorized access, data breaches, and cyberattacks.

(PoC5) The sensor network should incorporate strong security measures to protect against unauthorized access, data breaches, and malicious attacks. Security is essential to ensure the integrity, confidentiality, and privacy of sensor data and communications, preventing unauthorized manipulation or exposure of sensitive information.

Req.38. Flexibility

The system must be flexible and able to adapt to changing requirements, to adapt to changing regulatory standards, market conditions or additional user needs.

Related to PoCs: 1, 4

(PoC1) The system must be flexible and able to adapt to changing requirements and conditions, such as changes in energy demand, supply, and market conditions.

(PoC4) The system must be flexible and able to adapt to changing requirements and conditions, such as changes in EHS policies, integration of new applications and external system components as network information producer or consumer.

Req.39. Robustness

The Open Swarm System must be resilient and able to operate even if some actors fail or leave the system.

Related to PoCs: 1, 4, 5, General

(PoC1) The system must be resilient and able to operate even if some actors fail or leave the system.

(PoC4) The system must be resilient and able to operate even if some actors fail or leave the system.

(PoC5) The sensor network should be robust, capable of operating effectively and adapting to changing conditions, such as signal interference, object movement, or environmental variations. Robustness ensures the network's resilience in the face of external factors and enables it to maintain reliable communication and data collection capabilities.

Req.40. Reliability / Availability

The system must be highly available and able to operate 24/7 to ensure the wanted quality of the on the system relying services.

Related to PoCs: 1, 2, 3, 4, 5, General

(PoC1) The system must be highly available and able to operate 24/7 to ensure the reliable supply of energy to the grid.

(PoC2) The peer-to-peer reliability, i.e., the portion of data generated by a robot that reaches another robot within the same mobile ad-hoc network (MANET), MUST be, at least, 90% .

(PoC3) The end-to-end reliability, i.e. the portion of data generated by the machines that reach the back-end, MUST be at least 99.99%.

(PoC4) The system must be highly available and able to operate 24/7 to ensure reliable EHS measures. Unexpected disturbances need to be detected and communicate via respective alarming mechanisms.

(PoC5) The sensor network should operate reliably, ensuring consistent data collection, communication, and connectivity even in challenging environments or during object movement. Reliability is crucial for maintaining continuous monitoring, tracking, and communication capabilities within the network.

(General) When operating in an environment it was designed for, a deterministic network must offer at least 99.99% end-to-end reliability.

Req.41. Ease of use / Usability / User Interface

The Open Swarm System must be user-friendly and easy to use for all actors. The user interface must be as universally usable as possible on all types of devices a user has. This should be achieved by utilizing web technologies.

Related to PoCs: 1, 4, 5, General

(PoC1) The system must be user-friendly and easy to use for all actors, including prosumers, energy communities, and the distribution system operator.

(PoC4) The system must be user-friendly and easy to use for all actors.

(PoC5) Usability and User Interface: The network should provide a user-friendly interface and intuitive controls, enabling easy configuration, monitoring, and management of the sensor network. Usability and a well-designed user interface simplify the operation and maintenance of the sensor network, reducing the learning curve and enhancing user satisfaction.

(General) We need the user interface of a solution to be as universally usable as possible. Today, web technology satisfies that requirements, as all types of devices a user has, has some sort of web browser, and that a UI based on web technology can easily be exercised programmatically. We hence encourage OpenSwarm UIs to be web based.

Req.42. Cost-Effectiveness

The system must be cost-effective, and the benefits of using the system must outweigh the costs of implementing and operating it.

Related to PoCs: 1, 4, General

Req.43. Test Coverage

Developers introduce changes to the code hosted by them using utilities such as Git. Within this development environment, we must have 80% of the code to be continuously integrated and tested.

Related to PoCs: General

Req.44. Documentation and Support

The Open Swarm System must be accompanied by comprehensive documentation, guidelines, and support resources to assist in installation, configuration, troubleshooting, and maintenance. Documentation and support resources facilitate the deployment, operation, and maintenance of the sensor network, ensuring that users have access to the necessary information and assistance.

Related to PoCs: General

Summary

This document is the first specification of the requirements for the final “product” of the OpenSwarm project. By “product”, we refer to the technical innovations generated by the three scientific pillars, developed in WP2 “*Orchestration of Collaborative Smart Nodes*”, WP3 “*Collaborative Every-Aware AI*” and WP4 “*Energy-Aware Swarm Programming*”.

Originally, a prioritization of the requirements was also planned. The result of this process would have been some sort of ranking of the requirements, to see, which requirements are “more” needed for the OpenSwarm solution.

Due to the effective collection from the PoCs, with their real-world approaches, as well as the grouping, consolidating and ordering of the requirements, it turned out that this prioritization was not really necessary. Rather, the collected requirements reflect a good cross-section of the final solution, which may differ in their weighting depending on the PoC, but in principle already represent a good cross-section.

Also, there are very few acceptance criteria formulated for each of the requirements. This was not done in this document because of the early stage of development process. These acceptance criteria must be added later. Rather, the requirements collected in this document represent an initial starting point for the collaborative development of the OpenSwarm product. The requirements defined here allow a systematic processing of the necessary developments, such as tracking, defining sub-requirements and documentation.

How the database created here will be integrated into tools commonly used for software development will be defined in a later phase of the project. While this deliverable is completed with Task 1.1, it continues to provide a basis for further development and additions that may arise during the course of the project. It will therefore continue in the project with the appropriate additions as a living document either in its existing or in another form.

In conclusion, the identified requirements will be summarized in tabular form with the corresponding links to the descriptions. This is done firstly as a requirements list and secondly from the point of view of the PoCs on the requirements.

List of Requirements (with Link to corresponding PoC)

Requirement ID	Title	Page	PoC1	PoC2	PoC3	PoC4	PoC5	General
Req.1	Regulatory Compliance	36	X	X	X	X	X	X
Req.2	General Communication Needs	37	X	X				
Req.3	Sensor Network Integration	37				X		
Req.4	Communication patterns	38	X			X		
Req.5	Network Self-Optimization	38	X			X	X	
Req.6	Interference Mitigation	38					X	
Req.7	Ad-Hoc Communication	39		X		X	X	
Req.8	Addressing	39				X		
Req.9	Quality of Service (QoS)	40					X	
Req.10	Swarm Gateway / Swarm Connector	40	X		X		X	
Req.11	Sensor Requirements	40	X				X	
Req.12	Mobile Devices	41				X		
Req.13	Power / Energy Consumption	41		X	X	X		
Req.14	Battery Lifetime	42		X	X	X		X
Req.15	Energy Harvesting and Efficiency	43					X	
Req.16	Discovery	43	X	X		X	X	
Req.17	Auto Configuration / Negotiation	44	X	X		X	X	X
Req.18	Adaptability / Change detection	44	X	X		X	X	
Req.19	Self-description / Awareness of own needs	45	X			X		
Req.20	Localization	46	X	X		X		
Req.21	Application Deployment	46	X			X		X
Req.22	System and Device Management / Application Performance Monitoring	47	X	X	X	X	X	X
Req.23	GDPR Compliance	47	X	X	X	X	X	X
Req.24	Information Caching	47				X		
Req.25	External Information Storage	48				X		
Req.26	Distributed Data Fusion	48					X	
Req.27	Context Identification	48				X		
Req.28	State Synchronization	48				X		
Req.29	Latency / Real-time Data Processing	48	X	X	X	X	X	
Req.30	Boat Detection	49		X				
Req.31	Reporting of Boat Detection	49		X				
Req.32	Dynamic Object Tracking	50					X	
Req.33	Proximity Detection	50					X	
Req.34	Interoperability	50	X			X	X	
Req.35	Scalability	51	X			X	X	X
Req.36	Maintainability	51	X			X	X	X
Req.37	Security	51	X	X	X	X	X	X

Requirement ID	Title	Page	PoC1	PoC2	PoC3	PoC4	PoC5	General
Req.38	Flexibility	52	X			X		
Req.39	Robustness	52	X			X	X	X
Req.40	Reliability / Availability	53	X	X	X	X	X	X
Req.41	Ease of use / Usability / User Interface	54	X			X	X	X
Req.42	Cost-Effectiveness	54	X			X		X
Req.43	Test Coverage	54						X
Req.44	Documentation and Support	55						X

List of PoCs (with Link to corresponding Requirements)

	Regulatory Compliance	General Communication Needs	Sensor Network Integration	Communication patterns	Network Self-Optimization	Interference Mitigation	Ad-Hoc Communication	Addressing	Quality of Service (QoS)	Swarm Gateway / Swarm Connector	Sensor Requirements	Mobile Devices	Power / Energy Consumption	Battery Lifetime	Energy Harvesting and Efficiency	Discovery	Auto Configuration / Negotiation	Adaptability / Change detection	Self-description / Awareness of own needs	Localization	Application Deployment	System and Device Management / Application Performance Monitoring	GDPR Compliance	Information Caching	External Information Storage
	Req.1	Req.2	Req.3	Req.4	Req.5	Req.6	Req.7	Req.8	Req.9	Req.10	Req.11	Req.12	Req.13	Req.14	Req.15	Req.16	Req.17	Req.18	Req.19	Req.20	Req.21	Req.22	Req.23	Req.24	Req.25
PoC1	X	X		X	X					X	X					X	X		X	X	X	X	X		
PoC2	X	X					X						X	X		X	X			X		X	X		
PoC3	X									X			X	X								X	X		
PoC4	X		X	X	X		X	X				X	X	X		X	X	X	X	X	X	X	X	X	X
PoC5	X				X	X	X		X	X	X				X	X	X	X				X	X		
General	X													X			X				X	X	X		

	Distributed Data Fusion	Context Identification	State Synchronization	Latency / Real-time Data Processing	Boat Detection	Reporting of Boat Detection	Dynamic Object Tracking	Proximity Detection	Interoperability	Scalability	Maintainability	Security	Flexibility	Robustness	Reliability / Availability	Ease of use / Usability / User Interface	Cost-Effectiveness	Test Coverage	Documentation and Support
	Req.26	Req.27	Req.28	Req.29	Req.30	Req.31	Req.32	Req.33	Req.34	Req.35	Req.36	Req.37	Req.38	Req.39	Req.40	Req.41	Req.42	Req.43	Req.44
PoC1				X					X	X	X	X	X	X	X	X	X		
PoC2				X	X	X						X			X				
PoC3				X								X			X				
PoC4		X	X	X					X	X	X	X	X	X	X	X	X		
PoC5	X			X			X	X	X	X	X	X		X	X	X			
General										X	X	X		X	X	X	X	X	X

Glossary

AGV	Automated Guided Vehicle
AMR	Autonomous Mobile Robot
API	Application Programming Interface
CI	Continuous Integration
DSO	Distribution System Operator
EC	Energy Community
ECO	Energy Community Operator
EHS	Environmental protection, Health management and Safety aspects
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
JSON	JavaScript Object Notation – An open standard file format and data interchange format that uses human-readable text to store and transmit data objects.
LiDAR	Light Detection and Ranging
MANET	Mobile Ad Hoc Network
PMA	Protected Marine Environment
PoC	Proof of Concept
QoS	Quality of Service
REST	Representational State Transfer – A software architectural style
RTK	Real-Time Kinematics
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping
UAV	Unmanned Aerial Vehicle

References

[1] N. Vandapel, J. Kuffner and O. Amidi, "Planning 3D Path Networks in Unstructured Environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[2] J. Vian and J. Przybylko, "Tree Metrology System". United States of America Patent 9,198,363, 2012.

[3] R. Chisholm, J. Cui, S. K. Y. Lum and B. M. Chen, "UAV Lidar for Below-canopy Forest Surveys," *Journal of Unmanned Vehicle Systems,* vol. 1, no. 1, pp. 61-68, 2013.

[4] E. Hyyppä, J. Hyyppä, T. Hakala, A. Kukko, M. A. Wulder, J. C. White, J. Pyörälä and X. Yu, "Under-canopy UAV laser scanning for accurate forest field measurements," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 164, pp. 41-60, 2020.