D7.6 Annual Report on Collaboration - Year 3

Grant agreement no.	780785
Project acronym	OFERA (micro-ROS)
Project full title	Open Framework for Embedded Robot Applications
Deliverable number	D7.6
Deliverable name	Annual Report on Collaboration - Year 3
Date	December 2020
Dissemination level	Public
Work package and task	7.2
Author	FIWARE
Contributors	eProsima, Bosch, PIAP
Keywords	micro-ROS, robotics, ROS, microcontrollers, collaboration
Abstract	This annual report on collaboration activities performed under the umbrella of the OFERA project and in relation to the micro-ROS platform.

Abbreviations

Term	Definition
DDS	Data Distribution Service
DDS XRCE	DDS for extremely resource-constrained environments
IMU	inertial measurement unit
MCU	microcontroller
rcl	ROS 2 client support library
rmw	ROS 2 middleware interface
ROS	Robot Operating System
RTOS	real-time operating system

1 Introduction

In this reporting period, previous collaborations of micro-ROS with the ROS and FIWARE Communities as well as with other EU funded Projects (e.g., ROSIN and Robmosys) have been strengthened. Regarding ROS, the close relationship between micro-ROS and the ROS community includes different lines of collaboration, most of them under the umbrella of official ROS 2 activities. The presence of two micro-ROS partners, Bosch and eProsima, in the ROS Technical Steering Committee is particularly noteworthy and relevant to the interests of micro-ROS. Both initiated the formation of an Embedded Special Interest Group (SIG) for ROS 2 during the ROSCon event in 2018, which have turned into the official ROS Embedded Working Group (WG). Since the very beginning of the ROS Embedded WG, eProsima took the coordination role while Bosch is one of its main caretakers. Besides, Bosch also participates in the ROS 2 Real-Time WG, which is also key for the visibility and collaboration opportunities of micro-ROS within the ROS 2 community. These WGs have regular meetings on a monthly basis, some bilateral cooperations with organizations like Robotis and RENESAS. To that aim, project status updates, as well as interesting news, will be posted not only to the micro-ROS site but also to the ROS Discourse embedded category.

On the side of FIWARE, the progress of the Micro XRCE-DDS component continues according to its roadmap as incubated FIWARE GE and during 2020 a new component, the SOSS-FIWARE, has been prepared and will be contributed to the FIWARE Catalogue in the first quarter of 2021. The activities associated with the regular report of micro-ROS results at the level of the FIWARE Technical Steering Committee (TSC) continued as usual during 2020 and, since the overall technical direction of the FIWARE Open Source Community is driven there, new opportunities for collaboration are always emerging. Furthermore, micro-ROS is linked to the activities of the FIWARE Chapter focused on interfaces with IoT, Robotics and 3rd-party systems not only through the Micro XRCE-DDS but also through the Fast DDS component (SOSS-FIWARE will join them soon). Last but not least, micro-ROS is one of the pillars within the framework of FIWARE Technical Roadmap activities in robotics, which accounts for a specific working group with monthly meetings. In it, members from the eProsima and FIWARE technical teams collaborate with members from other organizations like Atos, Fraunhofer IML, NEC, Engineering, and the Japanese TIS Inc. The evolution of this working group and its activities make micro-ROS accessible to new organizations and set a promising context for initiating new collaborations.

Regarding the collaboration with EU projects such as ROSIN and RobMoSys, they remain as active as they were and the communities behind these three projects continuously find collaboration spaces to develop concrete specifications and software features. Besides ROSIN and Robmosys, new collaborations are being initiated within the framework of another European project, the DIH^2 Project. During 2020 and 2021, this project is financing 26 experiments in which robotic systems will be applied in manufacturing environments where the type of demand includes changing lot sizes and highly customized products. Since two partners from the Ofera consortium (eProsima and FIWARE) are participating in this project, the use of miro-ROS is being promoted in DIH^2 experiments. The goal is clear, we aim to showcase micro-ROS in a TRL6-7 scenario for a real manufacturing company.

Last but no least, in this year 2020, micro-ROS has opened new promising lines for collaboration which include, among others, the membership of micro-ROS in the Zephyr Project.

2 Collaboration with the ROS Community

The ROS ecosystem undoubtedly defines the most strategic axis for establishing collaborations around the micro-ROS framework. Since the beginning of the project, the OFERA consortium has worked very hard for micro-ROS to find its place in this ecosystem. Today, different lines of collaboration can be considered established and opportunities to create new ones emerge every month. The following subsections summarize the main collaboration activities around the ROS ecosystem during 2020.

2.1 Contributions to ROS 2 Core Packages

As a key aspect for the long-term maintenance of micro-ROS, the OFERA consortium tries to contribute as many micro-ROS features as possible into the ROS 2 core packages at https://github.com/ros2/. To that aim, multiple pull requests have to be coordinated with the community very thoroughly as they affect all ROS developers. Besides, the Open Source Robotics Foundation also runs extensive tests on these pull requests and, therefore, most of the pull requests require weeks before they are accepted. Particularly relevant are the contributions of micro-ROS that gravitate around the core ROS 2 repositories, rcl, rcutils, and rclcpp along with the ones that are continuously made to several middleware-related repositories.

In this direction, a particularly relevant milestone for micro-ROS in 2020 was the announcement of the micro-ROS Foxy release and its compatibility with ROS 2 Foxy and Fast DDS 2.x. This release of micro-ROS supports new hardware platforms in addition to the reference Olimex STM32-E407 evaluation board, namely the Crazyflie 2.1 Drone and ST Discovery IoT kit (B-L475E-IOT01A), and new Real-Time Operating Systems (RTOSes) in addition to the already supported NuttX, specifically FreeRTOS and Zephyr v2.3.0. It uses the eProsima middleware Micro XRCE-DDS 1.3.0, fully compatible with the latest version of eProsima Fast DDS 2.0 for ROS 2 Foxy.

Besides, micro-ROS Foxy features the implementation of the rclc, an additional package that complements the ROS Client Support Library (rcl) to make it a feature-complete client library in the C programming language. The rclc package implements functionalities of the standard ROS 2 rclcpp layer, adapting them to the capabilities and needs of the low-resource devices targeted by micro-ROS. Among them, the rclc Executor and the Lifecycle. The rclc Executor provides a C API to manage the execution of subscription and timer callbacks, similar to the Executors of the rclcpp layer. It is optimized for resource-constrained devices and provides additional features and mechanisms that allow the manual implementation of deterministic schedules with bounded end-to-end latencies. The rclc_lifecycle package provides convenience functions in C to bundle an rcl node with the ROS 2 Node Lifecycle state machine, similar to the ManagedNode class of the rclcpp.

Furthermore, the Callback-group-level Executor developed in the first year of OFERA has been brought mainline to rclcpp in September 2020 by developers of Open Robotics. The implemented the proposed refinement for all Executor variants of the rclcpp package.

Finally, new demos and examples, combining the different hardware platforms and RTOSes supported and written in the new RCLC API, have been added to the already very complete pool of out-of-the-box and easy to use micro-ROS examples.

The complete list of improvements and bug fixes in micro-ROS Foxy is accessible here: release notes.

2.2 Interaction with ROS Developers

The vibrant community of developers that is behind ROS is one of the main pillars of this open source framework for robotics developments. Thus, a good interaction between the OFERA consortium and the community of ROS developers plays a vital role in the success of micro-ROS as well as in its sustainability after the project. The OFERA consortium interacts directly with ROS developers in different ways:

A. OFERA members are active in existing ROS developers oriented communication channels including ROS Discourse and ROS Answers in particular.

B. OFERA consortium members are also active contributors to ROS GitHub repositories. This collaboration is done via pull request and issues in which discussions and design are open to the community, including OFERA members.

C. Aside from the previous “official” ROS channels, OFERA has launched a micro-ROS dedicated Slack workspace: https://micro-ros.slack.com/.

D. A fourth way of interaction with the community of ROS developers is the participance of Consortium members in community oriented events like: ROSCon, ROS-Industrial Conferences and Workshops… more detail can be found in D7.3 Annual Report on Communication and Dissemination - Year 3.

In 2020, a particularly relevant activity aimed at enabling new collaboration opportunities between micro-ROS and ROS/ROS 2 developers was the eProsima sponsorship of the ROS Developers Day 2020. The event took place on June 27th and offered hands-on presentations about ROS developments (including examples and simulations). The eProsima team was the lecturer in two sessions, one of them regarding Server-Client Discovery in Fast DDS as an alternative node discovery mechanism for ROS 2, the second one was fully focused on micro-ROS. The micro-ROS talk was given by the eProsima software engineer Pablo Garrido Sánchez, who showed with a very practical hands-on tutorial how to run micro-ROS on the Zephyr Real-Time Operating System (RTOS). This way the community of ROS/ROS 2 developers had the opportunity to know more about micro-ROS and identify possible synergies it has with their technology.

Figure 2.2.1. This link takes to the series of talks in which the advances of micro-ROS were presented to the community of ROS/ROS 2 developers.

2.3 ROS 2 Technical Steering Committee

The ROS 2 Technical Steering Committee (TSC), initiated by the Open Source Robotics Foundation (OSRF) in September 2018, is in charge of broadening the participation to accelerate ROS 2 delivery. It defines the ROS 2 development roadmap and its members contribute to the development of ROS 2 core libraries and tools. The OFERA consortium accounts for two members of the ROS 2 TSC Bosch (one of the ROS 2 TSC founders) and eProsima. Other TSC members are Amazon, Apex.AI, Canonical, Intel, LG Electronics, Microsoft, ROBOTIS, Samsung, GVSC, Tier IV, the Toyota Research Institute and Wind River. The exhaustive list of ROS 2 TSC members is provided at https://index.ros.org/doc/ros2/Governance/.

The TSC membership of Bosch and eProsima allows to align the micro-ROS developments well with the roadmap for each ROS 2 release. In addition, it is a great opportunity to stay at the forefront in technical discussions about the ROS 2 core layers. Besides, since the ROS 2 TSC aims at conducting its technical discussions as publicly as possible, it makes it possible to establish Working Groups (WG) as collaborative spaces in which specific topics can be discussed in greater detail. The list of ROS 2 WGs is also provided at https://index.ros.org/doc/ros2/Governance/, just below the list of TSC members. The ROS 2 WGs in which the OFERA consortium is involved are mainly two: i) the Embedded Systems WG which is leaded by eProsima and accounts for the regular participation of the OFERA consortium and ii) the Real-time WG and the Middleware WG in which Bosch is an active participant.

2.4 ROS 2 Embedded Working Group

In July 2020, the OFERA consortium decided to retake the effort to organize a ROS Working Group dedicated to the embedded world with 4-week periodicity. Starting on July the 22th, Embedded Working Group (EWG) meetings have been organized and held every four Wednesdays at 5 pm CET, with the exceptions of holidays (e.g, the August meeting was skipped due to the fact that most of the organizers and many participants were unavailable due to summer vacations). The persons in charge of the organization have been members of eProsima, with strong support from the Bosch partners and valuable contributions from both the PIAP and FIWARE partners, who have participated in most of the encounters and have delivered short presentations focused on their role in the project. The iter followed for each meeting has been the following:

- About one week early, the agenda is prepared and released, together with a post on the ROS discourse as part of a dedicated thread. At the same time, a public announcement is made on the micro-ROS Slack channel and a mail to a list of interested members of the community is sent.
- In general, the agenda consists of discussing the new features, recent developments and technical advancements of the project that happened during the 4 weeks preceding the meeting. A space of ~ 15-20 is also dedicated to a presentation given by some participant external to the team of the organizers. Finally, a Miscellaneous section is dedicated to opening the debate to the community and/or discussing less technical aspects of the project (e.g., advertise conferences or demos) or, finally, to comment on some open issue or Pull Request on GitHub.
- The meeting happens on Google Meet, and in general the participants number oscillates between 17 and 23 persons, which is quite a good objective for this kind of meetings, especially if compared with the rest of ROS working groups. It lasts around 1 hour and it is recorded, when possible. The video is then made available on the ROS discourse.

During 2020, there have been 5 EWG meetings, publicized on this ROS discourse thread: https://discourse.ros.org/t/ros-2-embedded-wg-meetings/15460.

After some of the meetings, a post has been published on the blog held by the consortium members on the micro-ROS web page:

- The ROS 2 Embedded Working Group is back on July 17, 2020 by francesca-finocchiaro
- 2nd ROS 2 Embedded Working Group on September 9, 2020 by francesca-finocchiaro

2.5 ROS 2 Real-time Working Group and Middleware Working Group

The Real-time Working Group was founded in May 2019 and the Middleware Working Group one year later in June 2020. Members of the OFERA project (in particular from Bosch) are participating actively in both working groups since their beginnings. In general, there is a large overlap in the participants of the two working groups, which is a consequence of the fact that execution management and communication are closely connected. In the current design of the ROS 2 stack - and the micro-ROS stack - the decision on the processing order of messages and other events is distributed to the middleware layer and the client library. This set of slides visualize and explain this design in detail. The slides were presented by Ralph Lange, from Bosch, in his talk at the ROS-Industrial Conference in December 2020.

A focus of the Real-time Working Group is to improve this design to better meet the requirements for processing with bounded latencies and determinism. In this context, for the ROS 2 Foxy release in June 2020, OFERA members from Bosch contributed an extension to the ROS 2 middleware interface which introduced a take_with_info function throughout the whole stack to get messages at the client library layer together with their middleware meta-data. Furthermore, in September 2020, they supported Open Robotics in bringing the Callback-group-level Executor concept proposed by OFERA in 2018 mainline in the ROS 2 stack.

In the next few months, a whitepaper by the Real-time Working Group which discusses alternative designs as foundation for improved execution management shall be published.

Figure 2.5.1. Real-time and Middleware Working Groups deal with common aspects of ROS. This is mainly due to the fact that execution management and communication are closely connected. The set of slides available here elaborates more on this connection.

3 Collaboration with the FIWARE Community

FIWARE brings a curated framework of open source software platform components (referred to as FIWARE Generic Enablers - GEs) which can be assembled together and with other third-party components to build platforms that support the development of Smart Solutions faster, easier and cheaper. More technical descriptions of FIWARE can be found on the FIWARE website or GitHub.

The FIWARE NGSI API provides a simple yet powerful API for solving a basic need in any smart solution: how to gather, manage and provide access to context information. There is a core Context Broker component in every “Powered by FIWARE” platform which supports this API and enables the integration of the rest of platform components. The key contribution of the harmonised FIWARE NGSI API and the central Context Broker is the provision of a basis for interoperability of smart solutions/services that run on top of “Powered by FIWARE” platforms which significantly eases their portability and replicability.

A rich suite of complementary FIWARE components can be combined with the FIWARE Context Broker in a modular architecture and integrated as part of a “Powered by FIWARE” platform. These components are referred to as FIWARE Generic Enablers (GEs). The complete set of FIWARE GEs are structured in the following four main chapters:

- *Core Context Management chapter*, comprising the core FIWARE Context Broker component as well as components enabling integration with multiple alternative data sinks for storage and further processing of historic context data
- Chapter comprising FIWARE GEs helping to implement the *interface with IoT devices, robots and third-party systems*, capturing context information updates and translating required actuations;
- Chapter comprising FIWARE GEs for *advanced monitoring, using dashboard and analytical support tools, as well as processing and analyzing current and historic context data* using event rules, advanced Big Data and AI algorithms, targeted to support smart decisions or the smart automation of processes;
- Chapter comprising FIWARE GEs dealing with *management, publication and monetization of context data and services*, preserving defined access and usage control policies.

The results of micro-ROS are being integrated into the chapter of “*Interfaces with IoT devices, robots and third-party systems”*. The component Micro XRCE-DDS was the first of the micro-ROS contributions to the FIWARE Catalogue and during this year a new micro-ROS component, the SOSS-FIWARE, has emerged in 2020 and will be contributed soon. These two components along with Fast DDS, an eProsima product that is the default middleware for ROS 2, form the ecosystem of FIWARE Enablers that gravitates around ROS and DDS based robotics systems.

As this stack of components is one of the main pillars of the FIWARE for robotics roadmap, the OFERA partner eProsima plays a major role within the FIWARE community. The role of eProsima goes beyond the maintenance of its enablers, they actively participate in the FIWARE Robotics WG by attending the regular meetings and collaborating in the development of the FIWARE strategic roadmap for robotics. Besides the aforementioned aspects, the collaboration between microROS and FIWARE in 2020 includes a series of webinars developed by eProsima and the specific FIWARE track at ROS World 2020 FIWARE as key activities to promote the collaboration between micro-ROS, ROS and FIWARE ecosystems.

3.1 Integration of micro-ROS results in FIWARE

The FIWARE chapter associated with robotics developments is the "*Interfaces with IoT devices, robots and third-party systems*" chapter. In it, the integration between FIWARE NGSI and robotics systems is being materialized in three main axes ROS-, DDS-, and OPC UA-based robotics integration. In particular, the evolution of the FIWARE NGSI integration with ROS 2 and DDS goes hand in hand. And this is mainly due to the extraordinary link that the pioneer FIWARE enabler for DDS, Fast DDS, has established between both worlds. While Fast DDS is the pioneer FIWARE enabler within the FIWARE DDS family, the Micro XRCE-DDS enabler is the first contribution of micro-ROS to the FIWARE catalogue. During 2020, the progress of both enablers has been periodically reported at the FIWARE TSC level and their repository and documentation has been continuously maintained. Besides, the new features and advances made in these components as well as the specific evolution of micro-ROS have been continuously discussed under the umbrella of the FIWARE Robotics Working Group. The good news in 2020 is that a new component, the SOSS-FIWARE component, has emerged to cope with some integration requirements that have been identified for new use cases. This component will become the second FIWARE enabler contributed by micro-ROS in early 2021. The following sections 3.2 and 3.3 contribute the necessary links to check the activities that have gravitated around Micro XRCE-DDS and SOSS-FIWARE during 2020.

3.2 Micro XRCE-DDS

As a FIWARE enabler, Micro XRCE-DDS accounts for a dedicated section within the FIWARE Catalogue, which is accessible in this link. Besides this documentation, the status of Micro-XRCE-DDS is regularly reported by eProsima in the FIWARE TSC. The committee certified that the maintenance, supervision, and integration of Micro-XRCE DDS evolved successfully during 2020 according to the set of best practices, requirements, and guidelines that FIWARE has set for the contribution and maintenance of enablers. These specifications are accessible here.

A public summary of the Micro XRCE-DDS evolution can be found in the micro-ROS blog. Below is the series of micro-ROS posts in which Francesca Finocchiaro, from eProsima, reported on Micro XRCE-DDS.

- Release of eProsima Micro XRCE-DDS 1.2.0 on May 12, 2020.
- Release of eProsima Micro XRCE-DDS 1.3.0 on May 22, 2020.
- micro-ROS and Micro XRCE-DDS as Fiware enablers on June 17, 2020.
- Memory Profiling of Micro XRCE-DDS on September 3, 2020.

3.3 SOSS-FIWARE

SOSS stands for System-Of-Systems Synthesizer. This component is in charge of enabling seamless communications between heterogeneous systems for a number of protocols and languages. It behaves as a message-passing tool that, by speaking a common language, centralizes and mediates the integration and the communication between the different protocols is made possible by system-specific plugins, or System-Handles. The current list of built-in System-Handles includes: DDS, NGSI, ROS, ROS2, and WebSocket. The SOSS-FIWARE component, which gravitates around the SOSS NGSI System-Handle, is the component that will enable the seamless integration of other SOSS System-Handles with the Orion Context Broker and, therefore, it plays a major role for the integration between NGSI and micro-ROS architectures. In the first quarter of 2021, this component will become a new FIWARE Generic Enabler (see the FIWARE enabler contribution requisites here). Besides, a webinar in which a preliminary version of the SOSS-FIWARE component was demonstrated in May 2020 is accessible here.

3.4 Collaboration Activities in the FIWARE Robotics Working Group

In 2020, the activities of the FIWARE Robotics Working Group have been strengthened, the working group has grown and a series of monthly meetings has been organized and regularly attended by all the participants. Each of the meetings had a clear focus on the work of one single participant in order to understand its key application scenarios and use cases as well as the status and roadmap established for its current developments. exposed their lines of work to identify possible collaborations and define a strategic agenda that aims to provide tangible results in 2021.

The list of members in the FIWARE Robotics Working Group includes the following organizations:

- Atos, Engineering, eProsima, FIWARE Foundation, Fraunhofer IML, NEC and TIS Inc

Among others, the ideas and work streams set in 2020, which will be further developed in 2021, are:

- Definition of Smart Data Models for FIWARE Ready Robotics Applications (e.g., data models for FIWARE Ready AGVs).
- Definition of a Multi-tier Reference Architecture for Robotics that clearly defines the components that must run in the robot, close to the robot (edge) or in the cloud and what communication requirements may be demanded (e.g., 5G) as well as the different deployments and required interfaces (e.g., robot ⟵ ⟶ edge, robot ⟵ ⟶ local datacenter, edge ⟵ ⟶ cloud…)
- Integration of platform tiers with microcontrollers based on micro-ROS
- Design of a RTPS(DDS) binding for the NGSI context broker (e.g., how to map RTPS(DDS) native concepts into NGSI-LD, Support of RTPS(DDS) events as raw data streaming notifications in NGSI-LD Brokers, full RTPS(DDS) binding for context information management)
- 3D representation and visualization of robot digital twins in open source frameworks

3.5 FIWARE Wednesday Webinar Series

Another pillar of the FIWARE Technical Roadmap is the series of FIWARE Wednesday webinars in which specific aspects of the FIWARE technology is presented to the vast community that gravitates around the use of standardized context information in a number of domains. Within the FIWARE Wednesday webinars a specific series with a clear focus on FIWARE for robotics applications took place in 2020 and micro-ROS played a major role in them (particularly relevant are the ones in the list below). Most of the contacts and collaboration opportunities that emerged are now integrated as part of the work streams identified for the FIWARE Robotics WG.

- Micro XRCE-DDS & micro-ROS: Bringing DDS and ROS into microcontrollers on June the 17th, 2020 - FIWARE Foundation webinar - Jaime Martin Losa (eProsima CEO)
- micro-ROS and FIWARE - Enabling Robotics Systems on Microcontrollers on September 24, 2020 - FIWARE Wednesday Webinar- Francisco Melendez (FIWARE Foundation, Robotics Expert)

3.6 FIWARE Track in the ROS World 2020

In 2020, the FIWARE Foundation, in collaboration with eProsima, and Algebraic.ai, organized the FIWARE track in the ROS World 2020. The details of this track can be checked here. The micro-ROS project, FIWARE for Robotics, and the ALMA project were the essential topics of discussion in this track.

4 Collaboration with other EU-funded Projects in Robotics

4.1 ITP MROS in RobMoSys

The EU project RobMoSys, which ended in December 2020, has worked on an integrated approach for robotic software platforms by applying model-driven methods and tools on the top of code-centric platforms like ROS. The model-based approach for runtime system (re-)configuration by System Modes, developed in Task 4.3 of OFERA, can be considered as one contribution towards this vision.

An academic consortium by TU Delft, Universidad Politécnica de Madrid, Universidad Rey Juan Carlos in Madrid, and IT University Copenhagen approached Bosch Research for joining forces towards the development of a model-based approach for even more advanced reconfiguration and metacontrol methods based on architectural models, system modes, and ontological reasoning.

For this endeavour, the five partners applied successfully for an Integrated Technical Project (ITP) named Metacontrol for ROS 2 (MROS) in the second open call of RobMoSys. MROS ran from October 2019 to November 2020. The tasks by Bosch, mainly conducted by members from OFERA, were (1.) joint application of the System Modes concept and technology, (2.) support on ROS 2 diagnostics, which has been ported from ROS 1 by Bosch, and (3.) the evaluation of the Metacontrol approach in an industrial pilot by a Bosch consumer robot prototype. For a second pilot by Rey Juan Carlos University in Madrid, researchers from the same university implemented the System Modes technology in the ROS 2 Navigation stack with the goal to bring this mainline in 2021. The following architectural diagram depicts the overall MROS approach. The System Modes are shown in the middle. Diagnostics as important input and trigger for mode transitions is not depicted for the sake of clarity.

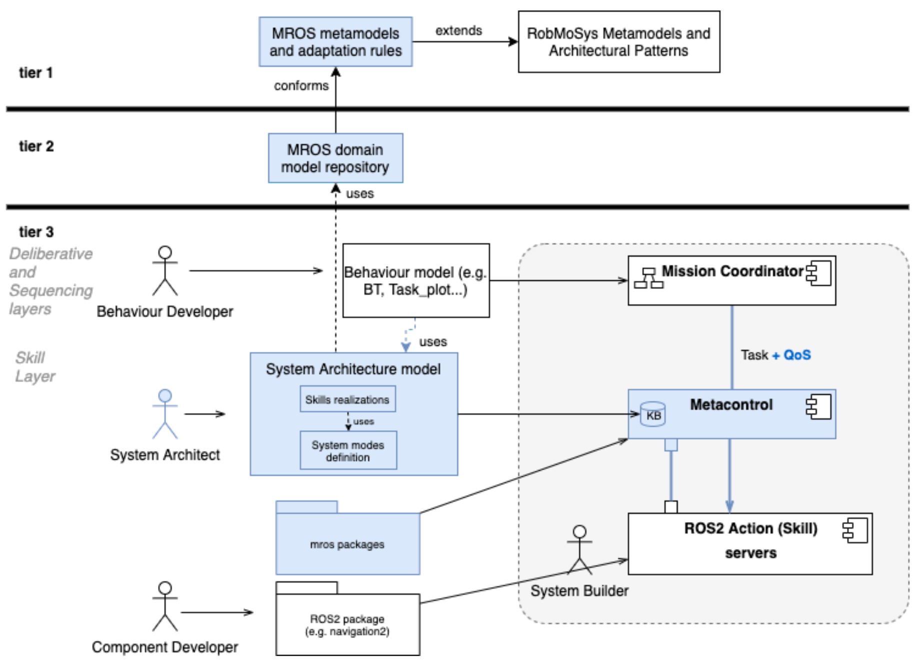

Figure 4.1.1. The MROS website can be found at https://robmosys.eu/mros/. Most of the code is open-sourced at https://github.com/MROS-RobMoSys-ITP/.

4.2 Collaboration with ROSIN

ROSIN has already been a good source of FTP for OFERA consortium members adding value to the Robotics ecosystem in Europe. With ROSIN there has been good and fluent collaboration in the past and further collaboration is envisioned, not only with FTPs but with the OFERA consortium members being invited to ROSIN conferences. OFERA consortium members were invited at the three most important ROSIN events in 2019 and this tendency continues in 2020. The list below includes the names, dates, and locations of these events:

- The ROS-Industrial EU Tech Workshop held in Stuttgart in May 2019.
- The ROS-Industrial Conference in Stuttgart in December 2019 where eProsima and BOSCH gave respective talks.
- The ROS-Industrial virtual Conference, held online in December 2020, in which both eProsima and BOSCH gave presentations.

This framework of collaboration with the ROSIN ecosystem is a space of great relevance to the introduction of micro-ROS in industrial environments. Therefore, we hope to maintain collaborative activities between ROS Industrial and micro-ROS that go beyond the ROSIN and OFERA projects.

4.3 SOSS developed in FTP ROSIS with OSRF

Born as the product of a joint effort between the OSRF (now Open Robotics) and eProsima, SOSS stands for System-Of-Systems Synthesizer and is a tool for communicating an arbitrary number of protocols that speak different languages.

The communication is made possible by system-specific plugins, or System-Handles, which provide the necessary conversion between the target protocols and the specific language spoken by SOSS and are discovered at runtime after having been installed. This is mediated by the SOSS core, which acts as an intermediate message-passing solution centralizing the communication thanks to its common language representation, based on an implementation by eProsima of the DDS-XTYPES specification by the OMG.

Once a system is communicated with SOSS, it enters the SOSS world and can straightforwardly reach out to any other system that already exists in this world.

SOSS is configured by means of a YAML text file, through which the user can provide a mapping between the topics and services on the middlewares of the systems involved.

eProsima has further developed two specialized tools based on SOSS: DDS Integration Service (DDS IS), and ROS 2 Integration Service (ROSIS), respectively allowing intercommunicating any DDS (or DDS-based) system, and any ROS 2 system with any other protocol which is already part of the SOSS ecosystem.

ROSIS enriches the functionalities of SOSS by offering a built-in DDS specific System-Handle that automatically implements integration of any protocol with the ROS 2 world. Specific System-Handles are already available for ROS 1, FIWARE Orion ContextBroker, WebSocket and more, allowing seamless integration of ROS 2 with any of these systems.

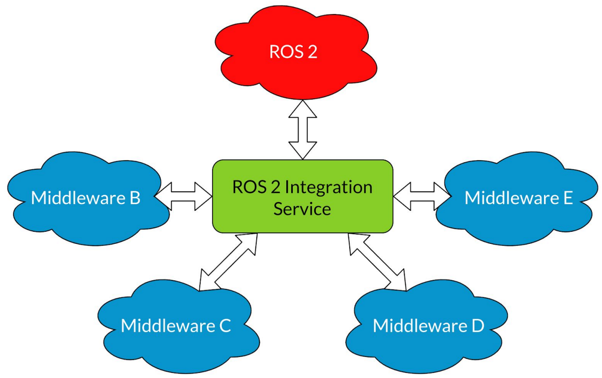

Figure 4.3.1. ROSIS allows seamless integration of ROS 2 with heterogeneous systems. Specific System-Handles like ROS 1, FIWARE NGSI and WebSockets are already available.

The main goal of the ROSIS implementation within this project is to offer a fully open source solution for two specific goals:

1. **Integration:** Connect ROS 2 to other protocols (MQTT, zeroMQ, a file/DB, etc): There are some existing bridges, such as ROS 2 to ROS 1 Bridge, or FIROS2 (FIWARE to ROS 2), but this project will offer a common framework, standard interfaces, and useful services to create general Bridges and transformations.
2. **Routing:** Enable ROS 2 over Internet/WAN. Currently, ROS 2 just allows LAN comms because of the underlying protocol of ROS 2. The project will offer routing services for ROS 2 traffic in several scenarios.

The following list summarizes the main features of ROSIS as a complete tool to integrate other technologies with ROS 2 and enable ROS 2 over WAN/Internet:

- It is fully integrable with Colcon. ROSIS comes integrated into ROS 2 workspaces as a package.
- The configuration is done using YAML format which increases the readability compared to XML configurations.
- It comes provided with a set of Docker images, so that it's ready to be used out of the box.
- It defines and provides a well-defined API to create new system integrations.
- Developed Bridges: FIWARE , DDS, ROS 2, ROS 1, WebSocket, REST.
- It provides usage examples and tests: TCP tunnel using DDS, ROS 2 domain change and a connection ROS 2 – FIWARE .
- It provides user documentation:
    - SOSS: https://soss.docs.eprosima.com/en/latest/index.html
    - DDS IS: https://integration-service.docs.eprosima.com/en/latest/

The outcome of this first Milestone can be found in a set of publicly available repositories:

- SOSS core (soss_v2) and ROSIS: Configuration and libraries loader. Includes additional System Handles (ROS 2, WebSocket): https://github.com/eProsima/soss_v2/tree/feature/xtypes-dds
- SOSS-FIWARE: Integration of FIWARE with any other System Handle: https://github.com/eProsima/SOSS-FIWARE/tree/feature/xtypes-support
- SOSS-DDS (base for DDS IS): Integration of native DDS (Fast DDS used as implementation): https://github.com/eProsima/SOSS-DDS/tree/feature/xtypes-dds

4.4 Collaboration with the DIH^2 Project

The OFERA members FIWARE and eProsima also participate in DIH^2 a project that is aimed at establishing a large European network of Digital Innovation Hubs (DIHs) with clear focus on robotics. From the technical point of view, DIH^2 is materializing the concept of an Open Digital Platform for Robotics Based Agile Production and FIWARE has been selected as the underlying technology for this platform.

The Technology Transfer Program of DIH^2 considers 26 experiments in real manufacturing companies that will develop and validate robotics based solutions for the agility problems they are facing. Both, eProsima and FIWARE are leading and mentoring these technical developments and DDS, ROS, and micro-ROS along with OPC UA are the default technologies offered for the field level middlewares that run below the core platform layer, which is based on the FIWARE NGSI standard. The first round of DIH^2 experiments started in 2020 and the welcome camp for the selected consortia included a training session on the aforementioned technologies. Some projects are exploring the use of micro-ROS and, therefore, the Technology Transfer Program in DIH^2 is becoming a great opportunity to showcase its performance in a real manufacturing environment.

5 Other Collaboration Activities

5.1 The Zephyr Project

The Zephyr Project is an open source project supported by the Linux Foundation. In turn, Zephyr is a scalable open source Real-Time Operating System (RTOS) designed to fit resource constrained and embedded systems, with safety and security in mind. A number of synergies exist between Zephyr and micro-ROS, they target a similar range of systems and applications with deterministic and time-critical constraints including networks of sensors and actuators, microcontrollers and small IoT devices of all kinds. Thus, micro-ROS and Zephyr have joined their forces to easily create heterogeneous distributed robotic systems in which these devices can be seamlessly integrated to benefit from and/or contribute to the ROS 2 ecosystem. By the end of September 2020, micro-ROS was announced an Associate Partner of the Zephyr Project. The official press release created for this announcement is accessible in this link[1].

[1] https://www.zephyrproject.org/google-and-facebook-select-zephyr-rtos-for-next-generation-products/

5.2 FreeRTOS

FreeRTOS is a market-leading and widely used Real Time Operating System (RTOS) for microcontrollers and embedded systems, built with an emphasis on reliability and ease of use. It has been for long integrated into micro-ROS through its build system, which enables the reuse of the majority of tools and features provided by the FreeRTOS community and partners. In September 2020, a micro-ROS post was published on the FreeRTOS blog, the post offers a technical and detailed recount on why FreeRTOS makes a lightweight and ideal RTOS over which running micro-ROS. Two successful cases of integration of FreeRTOS with hardware relevant for micro-ROS applications already exist the Crazyflie 2.1 drone and the ESP32 MCU. The Crazyflie software makes profitable use of several FreeRTOS tools and functions, and a demo example of a micro-ROS application working with FreeRTOS on the Crazyflie can be found here. As for the second hardware, natively integrated with FreeRTOS and offering a ready-to-use Wi-Fi antenna and Bluetooth function, a very recent port of micro-ROS has been carried out and practical demos are in line. Since the community of users of both micro-ROS and FreeRTOS is rapidly expanding and compelling use-cases are emerging, further integration of micro-ROS with the libraries offered by FreeRTOS and FreeRTOS+ is foreseen in the very near future.

5.3 PX4 Ecosystem

The collaboration between micro-ROS and the PX4 Ecosystem has been strengthened in this period and the presentation of micro-ROS in the PX4 Developer Summit was one of the key milestones for this achievement. The PX4 Developer Summit is a 2-day international event in which professionals from all over the world exchange technological insights of the drone sector. A webinar entitled “Bringing micro-ROS to PX4-based flying systems”, was presented by Jaime Martin, the CEO of eProsima and coordinator of the micro-ROS project, in conjunction with Nuno Marques, founder and lead software engineer of Drone Solutions. It covered the basic concepts around the micro-RTPS bridge used to integrate PX4 into the DDS world, and the proposal of migrating to a more mature, flexible and secure interface with DDS and ROS 2 based on micro-ROS and its middleware Micro XRCE-DDS. The large visibility gained through this event significantly activated the interaction between micro-ROS and community[2] behind the PX4 Developer Summit, which values the intensive activity and the quality of the results in micro-ROS. The year 2021 looks promising for the new connections established to generate new results through collaborative projects.

The talk that was given by Jaime Martin (eProsima) and Nuno Marques (Drone Solutions) is available here: https://youtu.be/8JCH_Yg8eX4 .

[2] The event accounts for approximately 50 speakers from companies like NXP, Microsoft, Open Robotics, Auterion, Dronecode and of course PX4. A wide range of topics are covered and the audience of the PX4 Summit includes developers with multidisciplinary skills whose interests include both software and hardware related aspects.

5.4 Canonical

In collaboration with Canonical, the Micro XRCE-DDS Agent has been wrapped as a snap package, which sets an important milestone for easing the usability of Micro XRCE-DDS and, therefore, widens the scope of potential micro-ROS users.

Snap is a package manager designed to bundle and handle applications and their dependencies on several Linux distros, among which Ubuntu. The snap packaging of the Agent comes with two ways of running it: as a simple executable or by means of a Linux service. The first implies the usage of the Agent's built-in CLI, where you can specify the standard configuration parameters (such as transport, port...) directly. Besides the specification of the Agent's launch parameters via the CLI, users can configure them thanks to the snap services interface, using the snap set micro-xrce-dds-agent <param>=<value> command. To have a look at the full list of configurable parameters, click here.

5.5 Tutorials Section at micro-ROS.github.io

The micro-ROS documentation and especially the sections in micro-ros.github.io dedicated to concepts and tutorials and demos have improved significantly in 2020. The OFERA consortium understands that this section is essential for attracting new users and thus creating new opportunities for collaboration.

6. Conclusion

This document reports on collaboration activities performed during 2020 under the umbrella of the OFERA project in relation to the micro-ROS platform. Some of these activities are associated with the reinforcement of existing collaboration frameworks while others are totally new and have resulted from the good evolution of micro-ROS in this particular reporting period. The activities reported have been organized in 4 axes:

- Collaboration activities between micro-ROS and the ROS ecosystem
- Collaboration activities between micro-ROS and the FIWARE ecosystem
- Collaboration activities between micro-ROS and EU-funded Projects
- Other Collaboration Activities (i.e., Zephyr, FreeRTOS, PX4 and Canonical)

As the OFERA project has been extended by one year, the OFERA consortium will do its best to establish new lines of collaboration and to further strengthen those that are active. A new report will be produced in December 2021 which will extend this document to include the collaboration activities associated with this 1-year extension of the project.