EU RISE
ROBOTICS IN SPACE ECOSYSTEMS

Open-Source Strategy and Implementation Plan for Future Space Ecosystem
D 2.7

This project has received funding from the European Union's Horizon Research and Innovation Programme under grant agreement No. 101134934.

Funded by
the European Union

Project Information

Information type	Details
Grant Agreement Number	101134934
Project name	EUropean RobotIcs for Space Ecosystems
Project acronym	EU-RISE
Call Identifier	HORIZON-CL4-2023-SPACE-01
Topic	HORIZON-CL4-2023-SPACE-01-12
Funding Scheme	HORIZON Research and Innovation Actions
Granting authority	European Health and Digital Executive Agency
Project start date	01.12.2024
Project duration	24 months
Project coordinator	Airbus Defence and Space GmbH
Deliverable reference number and full name	D2.7 – Open-source strategy and implementation plan for future Space Ecosystem
Delivery Date	22/03/2024
Issue	**v1.0**
Document produced by	Libre Space Foundation
Document verified by WP Leader	Airbus Defence and Space SAS
Document authorised by Project Coordinator	Airbus Defence and Space GmbH
Dissemination Level	PU/PP/RE/CO *

* Please indicate the dissemination level using one of the following codes:
- **PU** = Public,
- **PP** = Restricted to other programme participants (including the Commission Services).
- **RE** = Restricted to a group specified by the consortium (including the Commission Services).
- **CO** = Confidential, only for members of the consortium (including the Commission Services).

Funded by the European Union

Foreword

In this document, we will introduce open-source software and hardware in the field of robotics. We will start by thoroughly analyzing the open-source software and hardware, both those developed in the past and those playing an integral part in the present. Thus, we will highlight the power of open-source and showcase established open-source projects. We will walk the reader through the open-source landscape, illustrating both the challenges addressed as well as the suggested solutions, addressing concerns, limitations, and issues, navigating through the proposed solutions. Furthermore, we will dive into monetisation strategies and ways that various business models can be explored to leverage the power of open-source in the field of robotics. Highlighting the potential defined in the field and the types of entities that can dabble into it. Expanding on the impact open-source robotics can have not just in space but in education, research and other fields, and emphasizing the transformative potential of open-source robotics in various industries. It is an attempt to include and portray the benefits, challenges, solutions, and potential business opportunities within this dynamic and constantly expanding field.

List of participating organizations

Participant No.	Participant Organization Name	Country
1 (Coordinator)	AIRBUS DEFENCE AND SPACE GMBH	DE
2	DEUTSCHES FORSCHUNGSZENTRUM FÜR KÜNSTLICHE INTELLIGENZ GMBH	DE
3	MAGELLIUM SAS	FR
4	AIRBUS DEFENCE AND SPACE SAS	FR
5	AIRBUS DEFENCE AND SPACE LTD	UK
6	SENER AEROESPACIAL SOCIEDAD ANONIMA	ES
7	The Exploration Company GmbH	DE
8	OIKOPLUS GMBH	AT
9	LIBRE SPACE FOUNDATION	EL

Change Log

Version	Date	Description
v1.0	22.03.2024	First document release

Export Control Information

This document contains national, EU or/and US export controlled technology (data):

Yes ☐ No ☒

If No:

This document has been assessed against all applicable export control regulations and is "Not Listed".

If Yes, please complete the following as applicable:

1/ National Military / Dual-use export controlled content

(other than US and UK)

Country/countries of origin for technology contained in this document:

☐ France ☒ Germany ☐ Spain ☐ other: Canada

☐ This document contains technology which is controlled by military export control regulation, classification […]. Transmission abroad requires an export licence.

☐ This document contains technology which is controlled by national dual-use regulation, classification [CAN 5504.2.a.ii]. Transmission abroad requires an export licence.

☐ This document contains technology which is controlled by European Union dual-use regulation 428/2009 Annex I as amended, classification [9E001 (9D005 / 9A004e3 / 9A004f2)]. Transmission within the EU does not require an export licence. Export from the EU requires an export licence.

2/ UK Export Controlled content

☐ This document contains technology which has been assessed against the UK Export Control list and is rated as [e.g. MLXXX / XEXXX]. Transfer from the UK requires an export licence.

3/ US (ITAR / EAR) export controlled content

☐ This document contains technology which is controlled by the U.S. government under [USML category number / ECCN] and which has been received by [legal entity] under authority of [licence number / ITAR exemption / EAR licence exception / NLR]. Any re-export or re-transfer of this document in part or in whole must be made in accordance with the appropriate regulation (ITAR/EAR) and authorization (e.g. DSP 5, TAA, ITAR exemption, BIS licence or licence exception, NLR).

☐ This document contains technology which is designated as EAR99 (subject to EAR and not listed on the CCL).

Table of Contents

Definitions and Abbreviations

Abbreviation	Meaning
3D	Three-dimensional
API	Application Programming Interface
BSD License	Berkeley Software Distribution
cFS	Core Flight System
CMS	Content Management System
DOF	Degrees of Freedom
ESROCOS	European Space Robotics Control Operating System
GEO	Geostationary Earth Orbit
GPL	GNU General Public License
HIL	Hardware-in-the-Loop
H/W	Hardware
LiDAR	Light Detection and Ranging
MIT License	Massachusetts Institute of Technology
MOOS	Mission Oriented Operating Suite
NASA	National Aeronautics and Space Administration
OpenCV	Open Source Computer Vision Library
OpenRAVE	Open Robotics Automation Virtual Environment
OSH	Open-Source Hardware
RAMS	Reliability, Availability, Maintainability, and Safety
ROS	Robot Operating System
SpaceROS	Space Robot Operating System
STEM	Science, technology, engineering, and mathematics
S/W	Software
TASTE	The ASSERT Set of Tools for Engineering

URDF	Universal Robot Description Format
XML	Extensible Markup Language

1. Introduction

The commercial and institutional roadmaps from ESA, EU, DLR, CNES and UKSA emphasize the necessity of developing in-orbit capabilities to facilitate the growth of a forthcoming space ecosystem, while also fostering European self-reliance in the creation, deployment, and utilization of global space-based infrastructures, services, applications, as outlined in the EC Destination (HORIZON-CL4-2023-SPACE-01).
The upcoming space ecosystem is expected to undergo significant growth, with over 40,000 satellites and 100 lunar missions anticipated to be launched in the next ten years. This extensive infrastructure will subsequently drive the demand for capabilities to transport, assemble in-orbit and maintain this infrastructure for future use. As such, it is essential to develop in-orbit assembly and servicing capabilities combined with a logistic architecture at this time.
The EU-RISE project aims to advance robotic and autonomous technologies required for self-governing manufacturing, assembly and servicing (e.g. refueling, payload exchange). Therefore the project is composed of two interconnected workstreams: The first workstream will focus on the assessment of the market needs and define commonalities and services needed for a future space ecosystem. These services and capabilities will be used to compose a modular architecture based on the existing building blocks and capable to address the different markets. The second workstream is to adapt these building blocks from previous EU activities and industrial developments to realize a system which is able to provide these services in a complete end to end system.

This document is part of the first workstream of the EU-RISE project. This document explores the past and present of open-source software and hardware and how they can be used to promote space robotics and help shape the field's future. As this is an emerging field, the purpose is to examine the existing technologies and the ways the field can establish synergies that will push the boundaries of space exploration. Thus, this chapter not only does it provide a glimpse into the past software and hardware, but also takes a closer look at the current state and suggests ways to move the field forward. Its objective is to provide a detailed analysis of the past, present, and future of the field, identifying the limitations, the solutions, and the opportunities.

More specifically, it explores the way open-source communities contribute and iterate, focusing on the benefits of the collaborative approach, which significantly expedites development compared to traditional, closed-source methods. Fostering a more inclusive and dynamic space robotics ecosystem and a culture of knowledge sharing and collective problem-solving. This collaborative spirit leads to rapid innovation and the development of more robust and adaptable robotic systems. Taking a closer look at how open-source projects can lay the groundwork for industry-wide standards. And how this standardization streamlines development and ensures compatibility between diverse robotic systems, fostering seamless collaboration and mission effectiveness. Lastly, this chapter will focus on the business opportunities that can be fostered following an open-source approach, and it will be wrapped up with a detailed analysis of the limitations faced and the solutions suggested so that space robotics will be able to utilize open-source to move the field forward.

This document will evolve during the workstream 2 of the EU-RISE project.

2. Past and ongoing open-source S/W and H/W activity analysis in the space robotics community

2.1 Past Open-Source S/W and H/W Activity Analysis Efforts

We will begin the analysis of the open-source software and hardware by taking a closer look at the hardware that has already been developed. Numerous open-source projects have contributed to the advancement of robotics by providing tools and frameworks for S/W and H/W activity analysis. Some notable examples include:

- **ROS (Robot Operating System)**[1] is a popular modular and open-source S/W framework designed specifically for robot software development. It is a set of software tools and libraries providing a standardized way to connect and coordinate different components of robotic systems. By providing common functionalities and reusable code, developers can focus on their robotic system's unique requirements and aspects, picking and choosing the components needed. ROS includes tools for logging and analyzing robot activity.

- **OctoMap**[2] is an open-source H/W mapping library used to create 3D representations of the environment around a robot. It provides efficient algorithms for storing and updating occupancy grids, facilitating robot navigation and obstacle avoidance. OctoMap can be integrated with ROS, making it easier to integrate with other robot functionalities and applications.

- **URDF (Universal Robot Description Format)**[3] is an open-source XML-based file format that specifies robots' kinematic and dynamic properties. It is widely used in simulation and control applications and enables the creation of standardized robot models that can be exchanged between different software tools. More specifically, a URDF file can specify a robot's structure (both the individual parts that comprise it as well as how those parts connect together), defining, thus, its physical capabilities. URDF also provides information to visualization tools so the robot can be accurately represented within a simulated environment.

- **OpenCV**[4] stands for Open Source Computer Vision Library, and it is a powerful and widely used open-source library for real-time computer vision applications. It offers a wide collection of algorithms to be used for a number of computer vision tasks. It supports real-time processing of images and videos, locating and classifying objects within an image or video frame, face recognition, motion tracking, and building 3D models of objects from images or sensor data.

[1] https://www.ros.org/
[2] https://octomap.github.io/
[3] https://www.mathworks.com/help/sm/ug/urdf-model-import.html
[4] https://opencv.org/

- **Rock**[5] is an open-source framework with a set of libraries focused on 3D mapping applications through innovative LiDAR (Light Detection and Ranging). This is why this is a versatile tool that is very popular in certain industries. It offers a range of hardware and software solutions to streamline the 3D mapping workflow, enabling real-time 3D mapping and visualization, ensuring map accuracy, and real-time kinematic correction.

- **cFS**[6] is NASA's Core Flight System, an open-source reusable software framework designed for spacecraft flight software. It is a pre-built toolkit that aims to simplify and speed up the development process for various space missions. It works on a wide range of spacecraft and consists of different layers and components that can be mixed and matched based on the specific mission requirements.

- **TASTE**[7] stands for The ASSERT Set of Tools for Engineering. It's an open-source suite of tools specifically designed for developing embedded real-time systems. With TASTE, a system is described using models instead of directly writing code. This allows for easier visualization, analysis, and verification before the actual coding begins. TASTE can handle systems that have a mix of different hardware and software components. It can also automate the generation of code from the models, reducing manual work and potential errors, and it is well-suited for developing systems where safety is of primary significance.

- **SpaceROS**[8] stands for Space Robot Operating System, an open-source framework designed specifically for developing flight-quality robotic systems and autonomous applications for space missions. Built upon the foundation of ROS, SpaceROS has a strong base for development while addressing and overcoming ROS's limitations. It ensures compliance with rigorous verification and validation practices required for spacecraft flight software. It meets critical safety standards, and it provides functionalities for areas like guidance, navigation, control, and robotics alongside supporting technologies like simulation and software modeling.

- **ESROCOS**[9] stands for European Space Robotics Control Operating System. It was an initiative funded by the European Commission to develop an open-source framework for building flight software for space robots. Its aim was to reach the high standards of Reliability, Availability, Maintainability, and Safety (RAMS)

---

5 https://www.rock-robotics.org/
6 https://cfs.gsfc.nasa.gov/
7 https://taste.tools/
8 https://space.ros.org/
9 https://www.h2020-esrocos.eu/

required for space robotics applications. The project is stalled and no longer under development.

2.2 Ongoing Open-Source S/W and H/W Activity Analysis Efforts

Open-source S/W and H/W activity analysis efforts continue to evolve, addressing new challenges. Here are some examples of ongoing projects:

- **MOOS (Mission Oriented Operating Suite[10])** is a real-time S/W framework for managing and controlling complex systems in challenging environments. Its use has been extended to space robotics missions. It can facilitate the planning and scheduling of robot activities, dividing tasks amongst various robots, monitoring their operational health, providing tools for managing distributed teams of robots, coordinating data acquisition and analysis, and handling mission-critical events.

- **OpenRAVE (Open Robotics Automation Virtual Environment)[11]** is an open-source 3D robotics simulation toolkit that allows the simulation of various robotic platforms and environments. This framework supports the evaluation of diverse control algorithms that dictate robot movement and environmental interaction. It can also analyze robot performance across a number of different scenarios.

- **Gazebo[12]** is a popular open-source 3D robotics simulation environment extended for simulating space robotics applications. With Gazebo, movements and interactions between robots and objects are illustrated accurately within a simulated environment, reflecting space-based scenarios. Gazebo also supports different robotic platforms and seamless integration with various other S/W frameworks like ROS.

3. Benefits of Open-Source S/W and H/W Activity Analysis

The adoption of open-source S/W and H/W activity analysis in space robotics offers several benefits:

- **Increased Collaboration**: Open-source projects bring together researchers, engineers, and developers with diverse skill sets. This allows for a more holistic approach to problem-solving where each expert contributes to the project, adding their expertise and knowledge and coming up with unique and diverse solutions. This also fosters the inclusion of fresh perspectives and the utilization of new innovative approaches that might not have been readily available within a single organization. Collaboration thus expands not only the horizons of the contributors

[10] https://www.robots.ox.ac.uk/~pnewman/papers/MOOS.pdf
[11] https://github.com/rdiankov/openrave
[12] https://gazebosim.org/home

but also the number of solutions created. In addition, following open-source methodologies, collaborative development leads to more well-documented projects and codebases that follow best practices. Not only does this improve the quality of the software that is created, but it also makes it easier for new developers and contributors to get involved actively.

- **Reduced Development Costs**: Open-source tools and frameworks eliminate the need for costly proprietary software, allowing teams to focus on their core research and development goals. By leveraging the use of existing open-source tools, teams don't have to spend time and resources developing functionalities that already exist. Instead, they invest their valuable time into researching and developing other aspects of their project, such as novel algorithms, advanced robotics functionalities, or mission-specific adaptations. As many open-source projects are focused on modularity and scalability, this allows the teams to easily expand the functionalities of existing tools to meet the specific needs of a project. This ensures that the software under development can easily evolve alongside the project and saves the team from the need for costly replacements down the line.

- **Rapid Innovation**: Open-source development fosters a culture of continuous improvement and innovation, allowing for rapid adaptation to new challenges and emerging technologies. As is usually the case with most open-source projects, a larger community revolving around those projects scrutinizes the code. It handles bugs and vulnerabilities, which are usually identified and addressed early on, so open-source projects show a much quicker response time in handling vulnerabilities when compared to closed-source projects with limited reviewers. This constant iteration process ensures that the software remains stable, robust and reliable, allowing for faster advancements and the incorporation of new features.
  In general, open-source projects and communities foster a culture of open communication and knowledge sharing. This allows developers to readily discuss potential improvements, propose new functionalities, and experiment with innovative approaches. This free flow of ideas accelerates the development cycle and leads to the creation of more advanced and reliable solutions. In addition, open-source projects allow for experimentation and the integration of cutting-edge technologies. Developers can quickly test and adapt new advancements into existing codebases, fostering rapid innovation and pushing the boundaries of what can be created.

- **Accessibility to Resources**: Open-source S/W and H/W are freely available to anyone interested, enabling researchers and enthusiasts worldwide to participate in the development of a range of technologies, including space robotics. By providing free access to powerful tools and frameworks, open-source projects allow a wider range of participants to contribute to the field, fostering a more diverse and inclusive space robotics community. Talented individuals from across the globe contribute their skills and expertise regardless of their institutional affiliation or geographical location. Often, the availability of codebases and documentation provides a platform for students, researchers, and aspiring engineers to learn about technologies and space robotics, experiment with real-world code, and hone their skills.

- **Open-source and Open-science:** Open-source S/W and H/W have an impact on a wide range of fields and industries. It is not technology alone that can benefit and utilize those; there are more fields that can take advantage of open-source S/W and H/W. Open science is one of these fields, as it can greatly benefit and achieve greater transparency, collaboration, and accessibility in research. Open-source tools allow researchers to share and scrutinize the code used to analyze data and generate results. This transparency fosters trust in research and allows other scientists to replicate the findings, a cornerstone of scientific progress. What is more, open-source platforms facilitate collaboration between researchers across institutions and disciplines. Scientists can share code, build upon existing tools, and develop new functionalities together, accelerating scientific progress. Moving away from expensive commercial licenses can make research tools more accessible to scientists worldwide, particularly those in under-resourced institutions. This can help foster a more equitable research landscape.

  Open-source S/W and H/W activity analyses provide a platform for collaboration, reduce development costs, promote rapid innovation, and ensure accessibility to resources. As space robotics continues to evolve, open-source initiatives will remain crucial in driving the development of sophisticated and efficient robotic systems for exploration and scientific missions.

4. Different business models based on open-source for the space robotics industry

Several business models have emerged that leverage open-source principles to generate revenue and support the development of advanced space robotics technologies.

1. Software Licensing and Support: Many open-source software projects offer commercial licensing and support packages to organizations that require customized features, extended maintenance, or dedicated technical assistance. This model allows developers to monetize their work while maintaining the software's open-source nature. This way, the core functionality of the software remains freely available under an open-source license (e.g., MIT, GPL), allowing anyone to use, modify, and distribute the code. On that open-source core, the developers can offer additional services for which they can be charging. These services can include tailoring the software to the specific needs of an organization, for instance, adding functionalities for a particular space mission. Providing extended maintenance services with ongoing bug fixes, security updates, and compatibility patches for the open-source codebase. Offering dedicated technical support to developers for troubleshooting, consultation, and even training.

Within this business model, smaller companies and even startups can focus on providing commercial support, customization, and integration services for various open-source software projects. They can employ a team of developers with deep knowledge of specific open-source tools and cater to organizations seeking tailored solutions. Larger companies with existing software solutions may offer integrations or complementary products that work with popular open-source software platforms. They can provide commercial support packages for both their proprietary software and the integrated open-source components.

2. Hardware Development and Sales: Open-source hardware (OSH) projects often involve designing and producing hardware components for space robotics applications. Companies can leverage open-source hardware designs to manufacture and sell specialized hardware modules, robots, or systems, generating revenue while contributing to the open-source community. What is more, the open-source model fosters an active and vibrant developer community that continuously improves upon existing designs. Companies can benefit from these advancements by incorporating community-driven optimizations into their manufactured products. In addition, the open-source community can serve as a valuable testing ground for new hardware concepts. By releasing designs for open evaluation and feedback, companies can assess market demand and refine their products before full-scale production. Whether by manufacturing and selling open-source hardware directly to customers or by offering customisation options based on open-source designs, this is a business model that has potential.

Within this business model, both smaller companies and larger ones can opt to leverage the power of open-source space robotics hardware designs to produce components, modules, or entire robots for companies or research institutions that lack in-house manufacturing capabilities. Whether implementing specialized functionalities based on the open-source design or integrating the open-source hardware with other systems. Pre-assembling and testing the open-source hardware for easier customer use, offering additional features or functionalities on top of the core open-source design, or even providing warranties, support services, and user manuals for their products. These are all business ideas on which companies can expand and thrive.

3. Customization and Integration Services: Companies can offer customization and integration services for open-source software and hardware components, tailoring them to specific mission requirements or integrating them with existing systems. This model caters to organizations seeking customized solutions without the need for in-house development expertise. The companies that follow this business model become specialists in open-source space robotics software and hardware. They possess deep knowledge of the available tools, functionalities, and potential limitations. And when the need for modifications arises, they jump in and customize the services, adapting the software or hardware to address the unique requirements of each customer. Integrating different components and seamlessly combining different elements into a cohesive and functional robotic system. Charging for customization and integration, providing ongoing support and maintenance to customized solutions, and even offering training on utilizing and customizing open-source tools for space robotics applications.

This is a business model that small companies, larger companies and even Universities and research institutions can follow to monetise their efforts. Whether by providing customization and creating tailor-made solutions or catering to different functionalities, mission requirements, and designs. Supporting integration can be achieved by seamlessly combining multiple open-source components or integrating them with existing proprietary systems used by an organization. Smaller or larger companies and startups with a strong engineering background focusing on customized software elements to interact with or bridge functionalities between open-source software and other systems. Research Institutions and Universities with expertise in space robotics can offer customization and integration services as a form of technology transfer. Offering training workshops or consulting services to organizations seeking to utilize open-source technologies in their space robotics projects.

4. Training and Consulting Services: As open-source space robotics technologies become increasingly complex, demand for training and consulting services is growing. Companies can provide training courses, workshops, and consulting services to help organizations effectively utilize open-source tools and platforms for their space robotics projects. More specifically, companies can develop structured training programs that cover different aspects of open-source space robotics, from core functionalities to advanced customization techniques. They can organize intensive workshops focused on specific tools or applications, providing hands-on experience and addressing user questions. They can even offer personalized consulting services to guide organizations in selecting suitable open-source tools, implementing them in their projects, and troubleshooting any challenges encountered.

Relevant to this business model are companies, startups, and universities that can develop, organize, and deliver training courses, workshops, and consultations. For example, a company specializing in open-source robotics software might offer online courses on operating open-source robot operating systems and practical workshops on developing robot behaviors. An engineering services company might conduct training sessions on integrating open-source computer vision libraries with existing robotic manipulators used for spacecraft assembly tasks. A university space robotics research group might develop and deliver an online course on utilizing open-source tools for satellite data analysis. A company offering consultations with expertise in open-source spacecraft simulation software might offer training and on-site support to an organization developing a new satellite mission.

5. Data Analytics and Services: Open-source data analysis tools and techniques are widely adopted in space robotics. Companies can offer data analytics services, providing insights from space robotics data collected during missions or experiments. This model leverages open-source tools to generate value for clients. As raw space robotics data often needs cleaning, formatting, and organization before analysis, companies can offer services to prepare the data for analysis using open-source tools. Creating clear and compelling visualizations helps communicate complex data insights. Companies can leverage open-source tools to create interactive dashboards and reports that tell the story within the data. Companies can offer services like anomaly detection, predictive maintenance, or robot behavior optimization based on space robotics data, even by utilizing open-source machine learning libraries.

The companies that can be relevant to this business model are open-source service companies that, for example, can pre-process and clean space robotics data using open-source tools. Then, they can create interactive dashboards by visualizing key performance indicators for a robotic arm on the International Space Station. Data Analytics service companies might develop a custom data pipeline using a combination of open-source and proprietary tools to analyze data from a Mars rover mission, identifying potential anomalies in sensor readings that could indicate equipment issues. Another example can be a university research group that might collaborate with a space exploration company to develop an open-source machine learning algorithm for anomaly detection in satellite imagery using data from past space missions. Independent Data Analysts and Consultation agencies, a small company or a startup, can be hired to analyze data collected by a student team's CubeSat mission, using open-source tools to assess the performance of the onboard attitude control system.

6. Education and Outreach Initiatives: Open-source can significantly promote STEM education and foster interest in space robotics among young people.

Companies can support educational initiatives, workshops, and competitions that utilize open-source tools and platforms to inspire future generations of space robotics engineers. Developing open-source learning materials like tutorials, documentation, and online courses focused on using open-source tools for space robotics projects. Sponsoring robotics clubs and competitions that utilize open-source platforms, providing financial aid, mentorship, or workshop opportunities. Designing and releasing open-source hardware kits specifically for educational purposes, allowing students to build and experiment with basic space robotics concepts.

This business model is relevant to a number of companies, startups, universities, and research institutes. For example, a university or research institute can create a comprehensive online course teaching students how to use their open-source robot simulation software or develop a summer camp program where students build and program their own open-source rovers to navigate obstacle courses. A startup in edTech can create a series of educational workshops, material, and tutorials demonstrating how to use open-source tools for data analysis of real-world space mission data.

7. Controlled ecosystem: in this community-driven model, one entity (a project founder, a major contributor, an institution, an organization, or a group) guides the community. The community is expanded around active members who drive it forward by meeting new needs and actively contributing to creating new solutions. These are turned into additional services, and customisation options are added to the core product. The benefits of the plethora of services added are strong, and what is more, the community bestows the company/organization with strong influence, reputation, and power. This community-driven business model is supported by continuous innovation and constant expansion through services and customization.

WordPress and Automattic have adopted this business model, and it has had a massive impact on WordPress's success and proliferation. WordPress is the world's most popular and most-used content management system (CMS). It is free and open-source, and ever since its first release in 2005, it has expanded and grown. Nowadays, WordPress powers 43.3% of all websites and enjoys a 60.8% market share in the CMS market. The power of WordPress lies in the fact that its base project is free and open-source. At the same time, around it orbits a robust community of highly engaged users and a massive industry of additional services. From Themes and Plugins to managed WordPress hosting, security and website optimisation. WordPress now entails free and paid services that provide an end-to-end solution.

Following this model for the space robotics field, a university research lab could release an open-source robotic arm design, fostering a community that develops tools for simulation, control, and mission planning. The lab might offer advanced control algorithms or custom manipulator configurations as paid services. A startup could develop an open-source spaceflight software framework, attracting developers to contribute functionalities for navigation, communication, and autonomy. The startup could offer training, technical support, and integration services for organizations building space missions.

4.1 Airbus and the Controlled Ecosystem

After reviewing the different business models and ideas that can thrive using open-source software, let's examine the Controlled ecosystem business model and how Airbus can use it.

To start with, Airbus can act as the core orchestrator in the open-source space robotics field, bringing together a network of features, services and stakeholders in the field. By positioning itself at the center of a collaborative ecosystem, Airbus can solidify its market leadership in the open-source robotics industry. By creating a free, open-source core, any additional functionalities, features, modifications, and services added by the community will be charged. Sharing the research and development burden across the ecosystem can lessen the financial risks associated with bringing new technologies to market. Collaboration will allow Airbus to leverage the expertise and resources of its partners, spreading the costs and accelerating the development process.

By embracing the controlled ecosystem business model, Airbus can strengthen its position in the competitive robotics industry, fostering innovation, efficiency, and stronger relationships with stakeholders.

4.2 Takeaway

All the business models analyzed in this section demonstrate the versatility of open source in the space robotics industry. By leveraging open-source principles, companies can generate revenue, contribute to innovation, and expand their reach within the space community. As space robotics continues to grow, open source will likely play an increasingly prominent role in driving technological advancements and supporting space exploration.

A more extensive and detailed analysis of the different models can be found in the OpenSatCom Report[13] compiled by Libre Space Foundation[14].

[13] https://opensatcom.org/2020/11/12/open-source-models-satcom-release/
[14] https://libre.space/

5. Challenges and Potential Solutions for open-source in Space

The space business is a highly regulated industry with strict software quality and reliability requirements. Open-source software can be a valuable resource for space companies, but it also presents some challenges that need to be addressed.

5.1 Challenges

- **Qualification:** Open-source software is not typically qualified to the same level as proprietary software used in space applications. This means it may not meet the rigorous standards required for missions, which could have catastrophic consequences if something goes wrong. Strict qualification processes ensure software meets rigorous standards for reliability, performance, and safety. Open-source projects, often lacking formal qualification, can introduce an element of risk.

- **Software quality:** The quality of open-source software can vary widely. Some projects are very well maintained and have undergone rigorous testing, while others may be less well-tested and contain bugs that could compromise the safety of a space mission. Since open-source thrives on community contributions, not all projects follow the same quality standards. While some projects boast rigorous testing and code review practices, others might have looser quality control.

- **Maintenance:** Open-source software is typically maintained by a community of volunteers. This can lead to issues with bug fixes and updates being delayed or not being released at all. This can be a major problem for space companies that rely on software to function reliably for years or even decades. More specifically, as open-source projects often rely on volunteers, this does not guarantee the long-term maintenance of an open-source project. This raises concerns about bug fixes, security updates, and continued support for critical software used in spacecraft. In addition, open-source projects often have unpredictable release cycles for bug fixes and updates. For space applications, a predictable and well-tested update process is essential to ensure compatibility and maintain mission-critical functionality.

- **Licensing:** Open-source software is often licensed under open-source licenses, which can have implications for intellectual property and commercialization. Space companies need to carefully review the licensing terms of any open-source software they use to ensure they comply with their own licensing policies and the requirements of their customers. Often, Space agencies and companies have intellectual property (IP) concerns. Some open-source licenses, like the GNU General Public License (GPL)[15], require derivative works to be open-source as well. This might not align with a space company's desire to keep certain technologies proprietary. What is more, a project might use a combination of open-source licenses with varying terms. This can create a complex web of restrictions and obligations, making it difficult for space companies to integrate the software into their systems while complying with all licensing requirements. Lastly, space companies often work with customers who have specific licensing requirements. Open-source software might not meet those requirements, creating a hurdle for adoption.

- **Export Controls**: can limit the sharing of sensitive technical data between companies and research institutions in different countries, hinder collaboration on international space projects and slow down innovation. In addition, the licensing process for exporting space technology can be lengthy and complex. This can add time and cost to projects, hindering the pace of development and commercialization in the space sector.

5.2 Potential Solutions

Qualification

- Develop a set of qualification standards for open-source software specifically tailored to space missions' unique risks and requirements.
- Use a third-party organization that specializes in qualifying open-source software.
- Extensive verification and validation procedures ensure the software performs as intended and identify potential bugs before launch.
- Complete traceability of every line of code allows engineers to identify the origin and purpose of each component, aiding in troubleshooting and risk mitigation.

Software quality

- Utilize tools and techniques to assess the quality of open-source software, identifying potential bugs and security vulnerabilities.
- Implement rigorous code reviews and testing to evaluate open-source software quality before integration.
- Establishing community-driven certification programs can help assess open-source software for spaceflight. These programs can define benchmarks for testing, code reviews, and quality control processes.

[15] https://www.gnu.org/licenses/gpl-3.0.en.html

- A hybrid model can be followed where core functionalities come from a qualified open-source project and critical components are developed and maintained under stricter control. This approach can leverage the benefits of open-source while mitigating quality concerns.
- Encouraging the development of open-source projects specifically aimed at space applications can promote a culture of rigorous testing, documentation, and quality control from the ground up.

Maintenance

- Establish in-house expertise in maintaining open-source software or collaborate with third-party companies specializing in such services.
- Establish a formal process for tracking and managing open-source software dependencies, ensuring prompt updates with the latest bug fixes and security patches.
- Space-specific open-source projects can be designed with sustainability in mind. This could involve funding models to support long-term maintenance or fostering a core developer team responsible for ongoing updates and support.
- Space agencies can maintain control over updates and ensure long-term support by leveraging a qualified open-source project for core functionalities and developing critical components with a dedicated team.
- Strong community governance structures within open-source projects can help ensure continued development and maintenance. This could involve clear leadership roles, contribution guidelines, and a focus on long-term project health.

Licensing

- Consult with legal counsel to review licensing terms of open-source software, ensuring compliance with internal policies and customer requirements.
- In certain cases, negotiate with copyright holders to obtain additional rights or permissions as needed.
- Encouraging the use of permissive open-source licenses, like the MIT License[16] or the BSD licenses[17], can be beneficial. These licenses grant users more freedom to modify and distribute the software without requiring them to make their derivative works open-source.
- Some projects offer dual licensing options, with a permissive license for general use and a commercial license for companies needing more flexibility. This approach can cater to the needs of both open-source enthusiasts and space companies with commercial interests.
- A careful legal review of any open-source license is crucial before integration into a space project. This ensures compliance with the company's own licensing policies and the requirements of its customers.

[16] https://opensource.org/license/mit
[17] https://en.wikipedia.org/wiki/Berkeley_Software_Distribution

Export Controls

- Countries can work together to harmonize export control regulations on sensitive technologies. This would create a more level playing field for companies around the globe and simplify the export process.
- Expanding existing agreements or creating new ones specifically focused on space technologies could facilitate responsible trade while addressing security concerns.
- Improved communication and information sharing between countries can foster trust and collaboration on space exploration projects while maintaining export control objectives.

By addressing these challenges, space companies can successfully leverage open-source software to develop innovative, cost-effective solutions for their missions. Open-source software can be a valuable asset for the space industry, but it must be used responsibly and in accordance with the unique requirements of space missions.