Open Source Development Models in Satellite Communications

A report from the OpenSatCom.org activity of the European Space Agency implemented by Libre Space Foundation and inno[3]

1. Intro

1.1 Open source models

Originating from specific software development methodologies under a license approved by the Open Source Initiative, **open source models** have grown in definition to expand to more broadly practices associated with technology development (in all parts of its lifecycle: ideation - design - development - iteration - marketing - usage). A model is more complete and complex than a methodology since it encompasses legal, revenue and community aspects, which are vital to understand in order to sufficiently analyze, cross-compare and decide upon an open source development methodology. Thus for the purposes of this report open source models and open source development methodologies will be used interchangeably, since we will be tracking all traits of the methodologies and treat them as models.

1.2 Main licenses and their differences

Free Software and Open Source are initially legal concepts (whose official definitions are respectively stewarded by the Free Software Foundation and the Open Source Initiative), that are in practice embodied in Licences, that happen to be standardized legal texts: so, even if there are some proliferation problems, the number of Open Source licence texts is still limited, compared to the amount of projects published under those[1].
The main criteria to classify the different licences is related to the notion of ***copyleft***, a pun on the work “copyright” that refers to an obligation of reciprocity regarding rights conceded by the original author. The copyleft was designed to ensure that derivative works of an Open Source piece of software would also remain Open Source. Copyleft is instantiated with different significant variations:

- Variations of the scope (a **strong copyleft** licence will extend its scope as far as permitted by the applicable copyright law, to derivative works or collective works based on the software[2] ; a **weak copyleft** licence will limit this scope, generally to the initial software itself, allowing certain combinations with proprietary software[3] ; a **permissive** licence will have no copyleft clause at all[4]: there is an ESA Open Source for each of these categories[5]).
- Variations of the triggering element: most of the time, it’s distribution but it can also be network access (the most common licence for this being the GNU Affero GPL)

This aspect has a strong impact on the possible business model related to the development of the software (like selling proprietary software derived from or combined with the Open Source software).
The other main criteria is the presence of a clause protecting the user against software

1The SPDX project provides a good reference on the different Open Source licences. https://spdx.org/licenses/
2For example : The GNU General Public License, the CeCILL Free Software License Agreement.
3For example : The GNU Lesser General Public License, the Mozilla Public License
4For example : The MIT license, the Apache License
5https://essr.esa.int/esa-open-source-policy

patents aggressions: those clauses are absent in early academic Open Source licences (like MIT or BSD-3-Clause) but present in more “industrial” Open Source licences (like the Apache License 2.0 or the Eclipse Public License 1.0) and most modern licences (Mozilla Public License 2.0, GPL-3.0).

1.3 Open Standards are not Open Source methodologies

While Open Standards are key to the success of an Open Ecosystem, they are only the first step towards it. Other necessary elements, are, as expected, on the legal level, with the adoption of an actual Free and Open Source licence but also on the organisational level (Open Governance and collaborative methods) ; it’s only when all of these aspects are covered that an efficient collaborative ecosystem, able to deliver on the promises of Open Source, can be expected.
In a similar way, Open APIs (also called public APIs) are a step toward greater interoperability, and, as such, can foster the opening of an ecosystem, but they are not equivalent to actual Open Source projects, in terms of leveling the playing field for new players and enabling innovation.

1.4 Open Source pervasiveness

Open Source solutions now constitute the base of most technical stacks in the vast majority of technical fields, on three levels:

- Open Source nowadays provides the baseline for **software development methodology**: a growing number of companies officially adopt an “Inner source”[6] approach, and most software developers are familiar with Open source basic practices, as shown by the success of platforms like Github or Gitlab.com.
- As permissive licenses allow the reuse of Open Source libraries in proprietary software, the vast majority of software development include **third party Open Source libraries**.
- Modern **infrastructure solutions** heavily rely on Open Source projects be it for traditional components (like databases) or more innovating ones (e.g. the field of containers with projects like Docker and Kubernetes), where Open Source has been able to combine disruptive and incremental innovation, while allowing rapid mass adoption.

Beyond these generic successes, a number of ecosystems have been able to leverage Open Source models in order to sustain their evolution (See section “Assessing Open Source models in the context of specific industries/ecosystems”)

But despite all the successes, it must be kept in mind that, on each of three aforementioned levels, Open Source is also a risk factor that has to be anticipated to be mitigated:

6Companies like PayPal or Comcast were among the first to largely communicate about their inner source practices (http://innersourcecommons.org/getting-started/). In Europe, it is adopted by corporations in various fields (SNCF, Engie, Société Générale, Veolia, Thales, PSA, etc.) https://static.sched.com/hosted_files/osseu19/3a/Innersource%20ExperiencesENGIExSNCF%40OSSummitEurope.pdf

- Open Source software development methodology implies new management methods, adapted to the greater autonomy given to individuals inside the company and a kind of loss of direct control, which is incompatible with strategies like micro-management.
- Inclusion of third party libraries in developpements requires a rigorous compliance program: such practices are being standardized by projects like OpenChain[7].
- Even successful infrastructure components can rely on vulnerable business models (see section 1.5 Business models in Open Source, a non settled dispute)

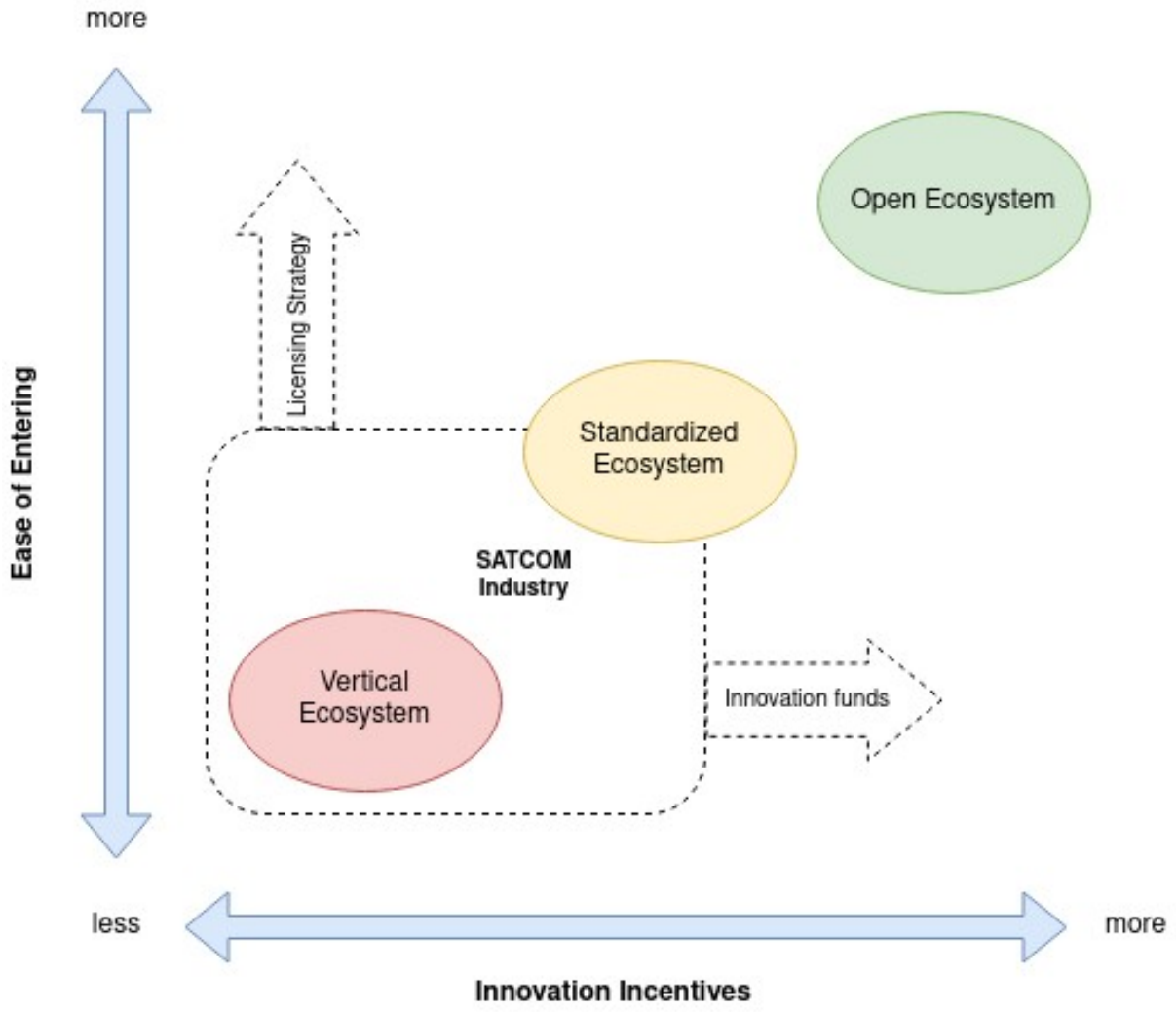


Figure 1: Mapping ecosystems with innovation and ease of entering attributes

An ecosystem can be characterized based on two core attributes: 1. The innovation incentives it presents to the participants and 2. The ease of entering for new participants. As seen in Figure 1 there are 3 different categories of ecosystems simplified. A vertical, a standardized and an open one. A vertical ecosystem is usually controlled by a single vendor controlling much of it (e.g. Salesforce AppExchange ecosystem). A standardized ecosystem is brought together by key players trying to achieve interoperability for the benefit of their

7https://www.openchainproject.org/

share in the industry (e.g. the 3GPP[8] initiative). An open ecosystem not only promotes the standardization of its technologies, but also promotes open implementations and is much more approachable for new entrants (e.g. the IETF[9] has many of the characteristics of an open ecosystem).

Based on all the information we could survey and our view of the broader ecosystem, the SATCOM industry has been traditionally multiple vertical ecosystems and moved towards some standardization (through efforts like CCSDS, ECSS, DVB etc) on various of its parts. Yet is far from an Open Ecosystem and specific actions should be taken to explore this direction for the benefit of the SATCOM industry.

1.5 Business models in Open Source, a non settled dispute

In a way, Open Source has won ("If Software Is Eating The World, Then Open Source Will Chew It Up (And Swallow)" Forbes 2015). But the business models of Open Source are still evolving and the sustainability of Open Source projects is still a complex and actively discussed topic.
The spectacular lack of global investment in critical Open Source components have been brought to light by the security vulnerability “Heartbleed” affecting OpenSSL and following studies[10] showed that it was more of a systemic problem than an isolated incident. In reaction to this, some projects like the Core Infrastructure Initiative have been started and global reflections on the topic are getting more attention, as shown by the creation of dedicated events like Sustain Open Source Summit[11].
In parallel but related ways, recent years have seen a significant trend of software vendors who have considered Open Source licences to be a too weak protection for their business models against what they consider unfair competition and have switched to new pseudo-open source licences, like :

- The Server Side Public License[12] extends the scope of its copyleft clause to such an extent outside the covered application that many consider it as not compatible with the Open Source Definition ; debates for its validation by the OSI were so fierce that the steward of the license eventually withdrew it[13].
- The Business Source License[14] or the Redis Source Available License[15] impose limitations on the usage of the software they cover (for instance they forbid the usage by potential competitors), so are undoubtedly not compatible with the OSD, and are globally acknowledged as such.

8 https://en.wikipedia.org/wiki/3GPP
9 https://en.wikipedia.org/wiki/Internet_Engineering_Task_Force
10 See “Roads and Bridges: The Unseen Labor Behind Our Digital Infrastructure”, Nadia Eghbal, The Ford Foundation, 2016 https://www.fordfoundation.org/about/library/reports-and-studies/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure/
11 https://sustainoss.org/
12 https://www.mongodb.com/licensing/server-side-public-license/faq
13 http://lists.opensource.org/pipermail/license-review_lists.opensource.org/2019-March/003989.html
14 https://blog.sentry.io/2019/11/06/relicensing-sentry
15 https://redislabs.com/legal/licenses/

1.6 Open Source Business models are not only vendor-centric

Single-vendor centric or open governance

In the specific context of Open Source, the question of business models requires first to determine the perspective used, whether it's - classically - single-company-centric or if it's more project-centric. This distinction directly relates to the two main governance models categories that apply to Open Source projects: those with a shared, open governance and those with a single-vendor centric one. In a single-vendor centric project, a single entity typically controls the whole IP attached to the project, whether by being its only contributor or by specific arrangements with other contributors : a Contributor Licence Agreement (CLA), in which the contributors grants only to the owner of the project very large rights, enabling him to relicense these contributions under the licence of its choice (including a proprietary one).
Projects with an open governance, on the contrary, allow different independent entities to be involved, often in the context of a formal neutral third party (generally a foundation), but not necessarily (major industrial projects like PostgreSQL and Samba show that an informal open governance is also a viable option). In such projects, no participant centralizes the whole IP on the project, but the foundation can do so, pursuing various goals: the Apache Foundation, for instance, has a CLA[16] in order to have the flexibility to change the licence without having to ask the different contributors (this CLA that was later adapted by companies like Facebook), and the Free Software Foundation asks for Copyright Assignments[17] to be able to enforce the GPL.
But such practices are often seen negatively because many consider decentralization of Intellectual Property a good thing. This opinion is shared by many different Open Source players, be they companies like Red Hat[18] or non-profit like the Software Freedom Conservancy[19]. Even Gitlab switched away from its CLA to a DCO[20] ("Developer Certificate of Origin"), in order to meet the expectations of major Open source communities like Debian and Gnome. A DCO allows to properly track IP in a project, but with no additional IP transfers. Successful foundations like the Linux Foundation or the Eclipse Foundation have adopted a DCO. Jean-Baptiste Kempf, leader of the VLC project, insists that a CLA is not needed to change a project's licence[21], based on its experience of relicensing of parts of VLC, with the agreement of the hundreds of contributors to the project.
It is important to note that the governance model of a project can be a key element of its viability. Some projects have proven not being successful nor even viable in a (proprietary) vendor centric model, but achieved a very significant development with an open governance: Blender 3D, which failed to be a successful proprietary solution, reached a remarkable success within the community-centric context of the Blender Foundation[22].

16 http://www.apache.org/licenses/contributor-agreements.html
17 https://www.gnu.org/licenses/why-assign.en.html
18 https://opensource.com/article/19/2/cla-problems
19 http://ebb.org/bkuhn/blog/2014/06/09/do-not-need-cla.html
20 https://about.gitlab.com/blog/2017/11/01/gitlab-switches-to-dco-license/
21 https://inno3.fr/actualite/entretien-avec-jean-baptiste-kempf-contributeur-majeur-vlc
22 https://www.blender.org/

Open Source as opposed to proprietary or as opposed to Custom software development

Another key point to understand business models of Open Source project is that they can be a substitute not to a proprietary third party vendor solution but to an internal ad-hoc development: in such a context, the development dynamic is quite different, because the focus is not market centric but rather centered around the optimization of internal development resources and practices.

1.7 Qualification of success for an Open Source business model

Intrinsic limitations of financial gains

In an Open Source context, the measure of the success of a business model is not restricted to the ability to maximise the generated profits. It is even likely that the very nature of Open Source bears some intrinsic limitations, related to its lockin-averse nature: those limitations can seem negative from an individual point of view, but may be an advantage from a more global point of view.

For example, the financial success of Red Hat, even though it is more than substantial, is actually limited in view of its technological impact.[23] The situation is similar for Automattic, the company founded by WordPress' creator, whose revenue can appear limited compared to some competitors, for a solution that powers more than 25 % of the whole Web. We generally rarely see monopolistic market-shares in the open source development ecosystem, but this is not inherently a drawback of such ecosystems; on the contrary that can be considered as a strong indicator of healthy dynamics: Open source is a model associated with a more mature, equitable market.

Distribution of the generated value

It should be noted on this example, that Automattic captures only a fraction of the revenue generated by WordPress, and that the solution is the source of revenue for a vast quantity of companies all over the world ; this effect is all the more important that, beyond direct monetisation, it is also a source of value that contributes to the business success of companies that monetise elsewhere, as explained below). Because of this distribution, the measurement of the total value is complex, but from a society point of view (at least in a liberal economy), this distributed aspect of the value generated can be accounted as an advantage.

Also, from the perspective of an entity whose goal is to promote innovation in a given domain or ecosystem, this distribution also appears as a positive aspect. For example, such considerations were key in the decision for RTE to open source some of its business applications and create the LF Energy initiative[24] ; outside the software field, it is the same logic that motivated Tesla to open a significant part of its patent portfolio : “We believe that applying the open source philosophy to our patents will strengthen rather than diminish

23See https://techcrunch.com/2014/02/13/please-dont-tell-me-you-want-to-be-the-next-red-hat/
24https://www.lfenergy.org/members/

Tesla's position in this regard."[25]
Given the ESA's mission ("to shape the development of Europe's space capability and ensure that investment in space continues to deliver benefits to the citizens of Europe and the world"), this dimension should be given specific attention.

Creation of value other than direct financial value

One specificity of Open Source software that must also be taken into account is its ability to create value beyond the financial streams it can generate. For instance, in an educational context, the technological and legal ability to study its inner mechanisms is most valuable, and in a scientific context, the ability to allow peer researchers to reproduce and validate experiments is fundamental.
These are especially relevant for the ESA, considering its context and goals.

25https://www.tesla.com/blog/all-our-patent-are-belong-you

2. Open Source Development Models

In order to assess the applicability of various Open Source models in SATCOM as applied to other industries we need to first define the core attributes of the most commonly used Open Source models. Although there are various definitions and analysis of open source models in the global bibliography [26] [27] [28] [29] [30] [31] there is a need to produce a more concise cataloging and attributes definition on the most common models (measured by adoption rates on other industries) which we are attempting here:

26https://opensource.com/article/17/12/open-source-business-models
27Zeena S., Pijkerman. (2018). A Comparative Case Analysis of Open Source Software Business Models.
28Krishnamurthy, Sandeep. (2005). An Analysis of Open Source Business Models.
29Raymond, Eric S. (2001). The cathedral and the bazaar
30Karl Fogel. Producing Open Source Software
31Lerner, J.; Tirole, J. (2005). The Scope of Open Source Licensing

Model 1	**Business to Business open source**
Description	This model is very prevalent. By design it pushes OEM adoption by partners and competitors across sectors. It has the potential to squeeze out competition by, for example, providing zero priced solutions to markets and therefore disrupting new similar products (such as when browsers commonly became free of cost). Highly strategic for Business Development purposes to be used as a leverage in the market. This model has multiple parallels with previously prevalent business practices of zero-priced solutions including the benefits and drawbacks, becoming a strategy for gaining market share as a revenue opportunity. E.g. Android as a platform that defaults to Google services, driving revenue through other products rather than being a revenue stream by itself.
Revenue	Enlarged market share (absolute and relative) for paid additional services due to interoperability and market growth and complementary products pricing
Tools	Internal versioning systems with gated releases
Legal	Non-copyleft most of the time
Effort	Medium to High
Community	The lead company does not put much emphasis on welcoming or nurturing contributors; exceptions may be made for strategically important organizational partners.
Benefits	Can drive industry adoption of a technology that is strategically important to your organization.
Drawbacks	Requires heavy investment on Business Relationship and commitment on maintenance
Examples	Android, Chromium
Examples in SATCOM	Capella, SNS3

Model 2	**Controlled Ecosystem**
Description	A community with a leader, often the project founder or other major contributor (institution/company/group or rarely an individual). Often with a broad community with a wide range of drivers for members who agree and align under the principle of the leader leading. The core provides base value, but the varied contributions across a healthy plugin ecosystem allow the project to address a much larger and diverse set of needs than any one project could tackle alone.
Revenue	Additional services on top of the core product and customization options
Tools	Public versioning systems
Legal	Mostly copyleft licenses
Effort	High especially on community management
Community	Welcoming, often with structures designed to promote participation and introduce new contributors
Benefits	Builds a sustainable ecosystem in which the founding organization or company retains strong influence.
Drawbacks	Requires heavy investment and expertise on community management.
Examples	WordPress, Drupal, Joomla
Examples in SATCOM	Lesser used model (SatNOGS might be applicable)

Model 3	Common Upstream
Description	Usually a block or module for integration or common usage into other products or software, therefore users tend to be developers. Common upstream module projects can carry huge influence in downstream dependent projects and applications although this does not necessarily mean the Common Upstream project is massively visible to developers dependent on it.
Revenue	Restricted. Mainly support funds, targeted development and additional expert services.
Tools	Wide open versioning control and participation
Legal	Typically non-copyleft (for enabling wider adoption)
Effort	Low to Medium
Community	Welcoming, and specifically amenable to one-time contributors
Benefits	Connections to many downstream dependee projects offers insight into market and usage trends, and can provide openings to potential partners.
Drawbacks	Minimal revenue possibilities
Examples	OpenSSL, Bootstrap
Examples in SATCOM	Orekit, LibCSP, pepSAL

Model 4	**Ethically open-source**
Description	Highly defined and specified ethically open source projects usually have a small motivated team at their core that are tasked and focussed on delivery of quality outcomes. Often founders of these projects have a clear and vibrant vision and can often leverage this into a healthily funded roadmap. Their open source strategy is often rooted in a commitment to transparency and providing insurance: they want to instill confidence among developers and users in order to promote adoption, and being open source is one ingredient in doing that.
Revenue	Diverse. Targeted development (for bootstrapping) or services on top of the main product are the most common ones.
Tools	Tight, in order to ship one core product.
Legal	Usually non-copyleft, but may be copyleft under certain circumstances.
Effort	Low to Medium
Community	Difficult to enter; focused on the core group.
Benefits	Achieves a quick, focused effect on a specific area; if successful, can co-opt competition.
Drawbacks	Uncertainty for revenue models and might not scale well on long term projects.
Examples	Diaspora*, TOR
Examples in SATCOM	poliastro, QUCS

Model 5	**Open Hardware - Design Focus**
Description	Organisation is focussed on design and development. Prototyping may be in house with manufacture external and often decentralised. Whilst contributors may contribute to hardware development, often contributors are downstream developing applications for the platform, peripherals and or developing firmware upgrades. Often long fork-merge cycles.
Revenue	Both direct B2C and B2B, Partner retailer licensing.
Tools	Broad tooling for hardware development. Contributor and version control platforms are more often built for software and may present unusual challenges.
Legal	non-copyleft and copyleft
Effort	Medium
Community	Welcoming, hardware development communities have a long lineage of sharing expertise.
Benefits	Large opportunity for maximising usage and development of hardware platforms.
Drawbacks	Complexity of drawing hardware contributors due to technical gatekeeping, physical tooling and manufacture process not easily accessible. Hardware vulnerable to potential non licensed cloning. Decentralised hardware development can be costly due to multiple hardware fabrication requirements.
Examples	Lulzbot, Arduino
Examples in SATCOM	LimeSDR, HackRF SDR

Model 6	**Opportunistic open-source**
Description	An organization/company/group which publishes code under a free license but invests no follow up effort into building open source dynamics. Typically, the published code is not the organization's prized technology. In some cases it is a one-off publication and thus likely to become quickly outdated. Some driving factors for this model of open-source are: Contractual, license or even branding requirements (organization forced to publish open source as a minimum legal requirement) and wrong-interpretation or lack of knowledge about other open source models (education and awareness issue).[32]
Revenue	No specific revenue model (could be a future opportunity if effort is given)
Tools	No specific tools. Distribution of code generally happens in a dry and one-off way.
Legal	non-copyleft or copyleft
Effort	Minimal
Community	Community is dormant to non-existent.
Benefits	Minimal to none resources requirements. Might raise awareness about an initiative.
Drawbacks	Underutilization of opportunity to establish an intentional exploration of a revenue stream and sustainability model around an initiative. Bad reputation about the project, leading to skepticism from possible collaborators and contributors.
Examples	Massive code dumbs from organizations like NASA[33] and CERN[34] do include a lot of opportunistic open-source projects.
Examples in SATCOM	n/a

32 https://jolt.law.harvard.edu/assets/digestImages/8-Vetter_Round-5-Complete.pdf
33 https://code.nasa.gov
34 http://opendata.cern.ch/search?page=1&size=20&type=Software

Model 7	**Mass Market open-source**
Description	Mass Market open source projects are user-centric competitors to existing mass market proprietary offerings, differentiating on their development and user-promise approach. Mass Market open source projects have certain opportunities that come with economies of scale, and they should actively seek to take advantage of those opportunities. Translations, User testing at scale, Thought leadership can be some of them.
Revenue	Traditional mass market revenue streams including ads, premium subscriptions and data exploitation. Also direct contributions (donations) is a possible revenue stream.
Tools	Tight, in order to ship one core product.
Legal	Non-copyleft generally, but may be copyleft depending on the project's business strategy.
Effort	Medium
Community	Fully open, but relatively brusque for the vast majority of users. Because of the large number of users, these projects evolve toward a users-helping-users pattern, rather than the development team serving as user support. The team itself can often seem somewhat distant or unresponsive.
Benefits	Large user base can help the project be broadly influential, and get added value from user contributed effort.
Drawbacks	Later-stage model that cannot be applied to newcomers. Requires a dedicated User-focused strategy and catering.
Examples	LibreOffice, Firefox
Examples in SATCOM	FreeCAD, KiCAD

The seven open source models presented here are by no means an exhaustive list of all available open source models, neither do they represent the only compartmentalization and grouping of the available practices. They do represent, according to our experience, a wide range of possibilities with inherent embedded flexibility. This gives an analyst the ability to slightly adapt them to describe or propose models applicable to complex ecosystems with multi-role players like the SATCOM ecosystem.

2.1 Assessing Open Source models in the context of specific industries/ecosystems

Open Source has been present for long enough to spread in various contexts, giving the opportunity to observe the way Open Source models work, not in an isolated fashion but inside complex ecosystems.

Four different ecosystems will be studied here in order to illustrate the different business models used by the types of players involved, as well as the key external factors that fostered the success of Open Source in these contexts. The first two will provide generic lessons with regards to Open Source models, while the two others will also provide domain specific elements that can relate more specifically to Satcomms.

- The Linux Kernel project, as it's an iconic and well documented success ;
- the Open Source GIS ecosystem, that has the particularity to be closer to encompass a large part of business applications and is not restricted to lower infrastructure layers ;
- the automotive industry has had Open Source success for years and has a manufacturing dimension that offers interesting similarities to the Satcom industry ;
- the terrestrial communications industry has more in common with Satcom, as technical evolutions increase the convergence of the two.

2.1.1 A model Open Source ecosystem : the Linux kernel

The Linux kernel is an undeniable Open Source success, a landmark in the Free Software landscape: it's a single project but it compasses a large enough spectrum both in terms of duration and number and types of players involved, to be studied as an ecosystem on its own.
The linux kernel is an obvious success, in terms of usage first (from mobile phones - with a market share over 80 % to supercomputers - where is market share is now just 100 % for the top 500 machines) but also in terms of diversity of its community: its sustainability doesn't rely on a unique player or restricted set of players.
All these elements describe success from the project perspective. It would be interesting also to consider that the magnitude of this success is not reflected in financial terms, nor in terms of public fame for its initiator, Linus Torvalds[35] ; but it could be questioned that the maximisation of these two would be a desirable target.

Business models involved

The ecosystem of an Open Source project goes far beyond its code contributors, but for the purpose of this study, it will provide us a good first approximation.
The annual report of 2017 by the Linux Foundation provides a list of the top 30 contributors. Thoses 30 entities can be grouped into 7 categories :

1. **Chip manufacturers** (Intel, Linaro, AMD, Renesas Electronics, Broadcom, ARM, Texas Instruments, NXP Semiconductors, Imagination Technologies, Cavium, NVidia, Rockchip) 31,70 %
2. **Operating system vendors** (Red Hat, SUSE, Google, Oracle, Canonical) 15,90 %
3. **Hardware vendor** (IBM, Samsung, Mellanox, Huawei Technologies, Code Aurora Forum) 11,60 %
4. **Individuals** 8,20 %
5. **Service providers** (Freelance consultants, Free Electrons -now Bootlin, BayLibre, linutronix) 5,90 %
6. **Advertising / Private Data Seller** (Facebook) 0,90 %
7. **Non profit** (Outreachy) 0,80 %

(Note that those top 30 contributors amount to 75 % of the contributions)

35Although it still brought him undeniable financial success and fame among a technical community

To each of these categories, it is possible to associate main types of revenue models related to their contribution to the linux kernel.
Chip manufacturers, hardware vendors: The software helps them sell their products. They need to be involved in the making of the software to have a better adequacy between their products and the project.
Operating system vendors : They combine different types of business models:

- They sell professional services
- They provide access to privileged distribution channels
- They sell Certifications

Service providers: As expected, they get revenues from selling professional services.
Individuals: It is interesting to notice that a very significant amount of work is still performed by individuals. There is not necessarily any revenue stream directly associated with their contribution to the project.

Factors allowing the success

Low position in the technical stack/platform: The kernel is a generic component, it is a *platform* that offers the possibility to build projects on top of it in various contexts.
Usage of the GNU GPL and its reciprocal (copyleft) clause: According to Linus Torvalds himself, the choice of a free/Open Source licence with a copyleft clause was key to Linux' success:

> "FSF and I don't have a loving relationship, but I love GPL v2. I really think the license has been one of the defining factors in the success of Linux because it enforced that you have to give back, which meant that the fragmentation has never been something that has been viable from a technical standpoint."[36]

It's important to stress out that this example proves that the GNU GPL is not "anti-industrial", hence it can not be superficially dismissed when it comes to choosing the licence for an industrial project.

Integration with other Open Source projects: The GNU project and the Linux Kernel were both mutually instrumental to their respective successes. Also, the early availability of major applications running on top of them, like the Apache Web server was key to the globalisation of their adoption.

A pre-existing open standard : the POSIX standard: The POSIX standard played a key role in the creation of Linux - as it provided technical guidance, and in its adoption - as it facilitated interoperability and portage of application with other Unix and Unix-like operating systems. It is the base to the Linux Standard Base (LSB), along with other open standards.

Protection against software Patents: Software patents, which have different legal dynamics in the EU and UShttps://opensatcom.org/projects/, have caused harm to innovation especially to smaller players. To protect the Linux Kernel ecosystem, the Open

36https://www.cio.com/article/3112582/linus-torvalds-says-gpl-was-defining-factor-in-linuxs-success.html

Invention Network[37] has been created in order to improve legal safety.
Open Governance via the Linux Foundation: The need for a neutral third party hosting the governance of the Linux kernel led to the creation in 2000 of the Open Source Development Labs (OSDL), that merged in 2007 with the Free Standards Group to create the Linux Foundation. The Linux Foundation created a structure that is now the de facto reference (to a point that its success can even appear as problematic) for organising collaborative industrial projects with multiple stakeholders.

2.1.2 The Open Source GIS industry

The majority of Open Source successes arose in the lower layers of the technical stack, and especially the infrastructure area, but the GIS domain is a notable exception, as successful Open Source projects are present up to the business application layer. The GIS Open Source community really works as an ecosystem, with strong interactions and interoperability between projects and a common culture embodied by common Open Standards (by the Open Geospatial Consortium) and common open governance practices (embodied by the Open Source Geospatial Foundation).
Key projects include the Geospatial extension for the PostgreSQL Database, PostGIS, and the Desktop client QGIS.

Business models and revenue streams

The main business models and revenue streams involved are:

- Professional services
- Service/product requiring the existence of the Open Source project.

A detailed analysis of QGIS - probably the most iconic project of this ecosystem - shows that the code contributions come mostly from a numerous and diversified (in terms of nationality) group of SMEs with high level expertise. It should be noted that the end user entities are deeply involved in the Open Source project, even if they are not always directly developing the code, but delegating it to the aforementioned SMEs: this reflects in the release notes of the application, which mention for each feature both the entity that financed it and the author of the code ; for instance:

> This feature was funded by Arpa Piemonte (Dipartimento Tematico Geologia e Dissesto) within ERIKUS project.
> This feature was developed by Martin Dobias (Lutra Consulting) and Faunalia
>
> This feature was funded by Grundbuch- und Vermessungsamt des Kanton Zug
> This feature was developed by David Signer, OPENGIS.ch
> (https://qgis.org/en/site/forusers/visualchangelog32/)

Factors allowing the success

Strong Open Standards: The GIS industry relies in great part on Open standards, mainly produced by the Open Geospatial Consortium. There are tight links between the OGC and

37https://www.openinventionnetwork.com/about-us/ Its scope has then been broadened to more cover more Open Source projects and complementary initiatives have emerged, like http://lotnet.com

OSGeo (see below), materialized by a Memorandum of Understanding[38].
Open Governance: A neutral domain-specific foundation Open Source Geospatial Foundation (OSGeo) has a domain-wide action and also spreads the culture of Open Gouvernance to the individual projects, by making it a requirement during the incubation phase.
Integration between Open Source projects: Projects often have a modular architecture that allows integration with specific tools in a mutually beneficial way (e.g. QGIS with Orfeo Toolbox). From an organisational perspective, the existence of OSGEO creates a common culture that also materializes through specific conferences FOSS4G.
Involvement of end users: Even if they are not always directly developing code, end user entities are part of the Open Source projects.
Mostly reciprocal (copyleft) licenses for applications (GPL for QGIS and PostGIS) while libraries (like Open Layers, OTB, etc.) are covered by permissive licenses.

2.1.3 The automotive industry

The automotive industry gives a good example of successful adoption of Open Source in the context of a large scale manufacturing industry. It offers several interesting similarities with the Satcom industry, like the complexity of its supply chain or the safety critical aspects.
The automotive industry has had global collaboration initiatives for a long time, like AUTOSAR (AUTomotive Open System ARchitecture)[39] founded in 2003, but whose actual openness is restricted (at least initially) to architecture and to members, in a typical “Gated Community” way.
Since 2009, major projects have arisen providing actual Open Source software and hardware, at different levels of maturity and different positioning with regard to the core business.

The Genivi project

The Genivi alliance was started in 2009 and was centered around In-Vehicle Infotainment (IVI), to gradually extend its scope to easing integration of multiple Operating Systems in the context of a “connected cockpit”: its members are OEMs (BMW Group, PSA, etc.), automotive Tier 1 suppliers (Magneti Marelli, Visteon, etc.) and technology companies (Intel, Wind River, etc .)
Its IVI deliverables have been deployed in production across many brands like BMW, PSA, Hyundai, Jaguar / Land Rover, etc. and beyond these technical deployments, Genivi has played a key role in infusing an Open Source culture in the automotive industry.

Factors allowing the success

Open governance: The Genivi Alliance is a 501 c6 organisation[40].

38 https://wiki.osgeo.org/wiki/OSGeo_signs_Memorandum_of_Understanding_with_OGC
39 https://www.autosar.org/
40 In the USA, a 501 c6 is a non-profit organisation aiming at promoting a common business interest of its members. By contrast, a 501 c3 is a non-profit for charitable, educational, religious, literary, or scientific purposes: for instance, the Software Freedom Conservancy (https://sfconservancy.org/) is an important 501 c3 for Open Source.

Involvement of end users: The project is led by end user companies and has successfully involved suppliers from different tiers.
Focusing on non-differentiating aspects: The collaboration was possible between direct competitors, because the object of the collaboration was not at the heart of their business value.
Open Standards and specifications: The first step of the Genivi project was to create common specifications, to align with the needs of the different stakeholders. They developed a 3-levels compliance program, for: specifications, API and code. (see next item)
Integration with other projects with different levels of openness: the project relies heavily on other Open Source projects, like the Linux Kernel, but also works to integrate with more closed projects like AUTOSAR. They also work to bring a more open governance to technologies that are Open Source from a legal point of view but are organizationally controlled by a single player (Android/Google). The revised mission of Genivi is now articulated around the Multi-OS Integration.
File-scoped reciprocal (weak copyleft) license: The main licence chosen by the project was the MPL-2.0.

The OpenMDM project

OpenMDM is an Open Source project driven by an Eclipse Foundation Working group (https://www.openmdm.org/working-group/governance) to create applications for measured data management systems. It addresses industry specific constraints like decade long conservation of data. Its members are car manufacturers (OEMs like Audi, BMW, Daimler, etc.), first tier suppliers (like Siemens) and service providers. It has a limited scope but it is a good example of an Open organisation in an industrial setting for a specific tool.

Factors allowing the success

Open governance: Eclipse foundation Working Group
Involvement of end users: The project is led by end user companies and has successfully involved suppliers from different tiers.
A tool more than a part of the product itself: to some extent, it is a specific aspect of non-differentiating aspects.
Open Standards and specifications: The project describes itself as “Based on open standards and open interfaces” and stresses the role of ASAM ODS.
Weak copyleft license: The project the Eclipse Public Licence, typical of the “platform approach” of the Eclipse Foundation.

Automotive Grade Linux

The AGL project is hosted by the Linux Foundation and is built on top of the experience of Genivi and on some of its code base and relies also on other Linux Foundation projects like Tizen. Its scope started with IVI but expanded to a more general concept of Connected cockpit. While it started in 2012, its deployment in production is still rather limited, even though it is gaining strong traction, especially in Asia.

Baidu's Apollo

Baidu has launched an ambitious project of Open Autonomous Driving Platform, called Apollo[41].
It is still in early stages but is taken very seriously by the industry and could get into production in a rather short term. It is important to notice that this project is led by a pure technological player, while this is an industry specific project and more central to the core product (cars) than, for instance, IVI.

Open Source Vehicles (Open Hardware/Software)

Different projects around the idea of complete Open Source cars have emerged a few years ago, the most visible one being Open Source Vehicle[42].

Although these projects are quite low on a TRL scale, and that some of them have experienced some setback after a promising start (like the POM of Renault[43]), it must be taken into account that electric cars have a complexity relatively lower to traditional cars, so that this new technical paradigm makes the arrival of disruptive new players in the field much more likely. A parallel might be drawn with the arrival of constellations of smaller satellites in the Satcom industry.

Business models and revenue streams

In all these different projects, the dominating types of business models and revenue streams involved are:

- Service/product requiring the existence of the Open Source project.
- Professional services

2.1.4 Open Source in terrestrial communications industry

The terrestrial communications industry as known two major evolutions which were tightly related to Open Source :

- The embracing of Internet and Voice over IP, which happened in the 90's and 2000's
- The virtualisation of networks, currently happening

They provide examples of different types of ecosystems: one vendor centric and one project centric.

Asterisk, Open Source PBX

The impact of Open Source in the convergence of telephony and data/IP networks materializes in the generic lower technical layers - including at the operating system level, with the creation of the Carrier Grade Linux specifications - but Asterisk is an example of Open Source success in a limited, but very domain specific area.
The arrival of Asterisk in the PBX landscape was rather disruptive in a context locked by proprietary hardware solutions and allowed the Digium company to prosper and create a viable ecosystem around its product. But even with an Open Source product at its heart, its

41 https://github.com/ApolloAuto
42 https://github.com/OSVehicle
43 http://www.4erevolution.com/en/renault-pom-open-source/

mechanisms are quite similar to a classical vendor centric ecosystem because the Digium company centralises the intellectual property rights (through a Contributor Licence Agreement, CLA[44]) and the decisions for the project are also centralized.

Business models and revenue streams

The revenue streams and business models of Digium are very diverse but rather similar to the ones of a classic software vendor :

- The centralisation of IPR, which includes copyright, allows them to have a Dual licencing / Open Core scheme, so they actually sell proprietary licences..
- They sell dedicated hardware that embeds Asterisk.
- They provide Professional Services directly
- They sell Certifications related to Asterisk trademark (that they own)

The Asterisk ecosystem includes also a diversity of others companies, who get revenues from

- Professional services
- Selling proprietary software licences of larger solutions, that embed asterisk

Factors allowing the success

Pre-existing Open Standards : Asterisk was able to implement protocols like SIP, H.323, etc. and to integrate with other existing solutions.
Middle position in the technical stack/platform: even if it not as low as an operating system in the technical stack, Asterisk can still be considered as a platform on top of which different applications can be built. Its website advertises it as “a free and open source framework for building communications applications”.
The GNU GPL ? Although Asterisk is licenced under the GPL-2.0, like the Linux Kernel, the consequences of this choice of licence are very different, because of the different governance models (and especially the centralised IP control through a CLA). But, *a minima*, it’s another indication that the GNU GPL is not incompatible with business success.

Network virtualisation

Since 2010, many Open Source projects have emerged around the topic of network virtualisation. The vast majority of them share two strategic characteristics: they are foundation-governed projects and they receive a direct involvement of Telecom players like telecom operators and equipment providers, who, from hardware manufacturer, progressively shift to software development companies.
The great number of projects are developed under the governance of the Linux Foundation, which has created a dedicated initiative to foster interoperability between them : LF Networking "Harmonizing Open Source Networking”.
Its main projects are:

- OPNFV: https://www.opnfv.org/
- ONAP: https://www.onap.org/
- Open Daylight: https://www.opendaylight.org/

Two other interesting projects include Open Source Mano (NFV Management and Orchestration), which is hosted by the ETSI[45], and Neutron, which is part of the OpenStack

44https://issues.asterisk.org/jira/secure/DigiumLicense.jspa
45This kind of project is rather atypical for ETSI and its future evolutions should bring interesting

project (and as such hosted by the OpenStack Foundation)**.**
Contributors to these projects include a great number of telecom operators, from a wide range of countries across different continents (AT&T, Bell Canada, China Telecom, Deutsche Telekom, Orange, Turk Telekom, Verizon, Vodafone, etc.), equipment providers (Wind river, Nokia, Ericsson, etc.), along with service providers and hardware manufacturers.

Business models and revenue streams

Telecom operators: the revenues come from their regular activities, for which the Open Source software is a key asset, especially in terms of innovation, as they enable them to follow the pace of the radical technical evolutions at the core of their activities.
Equipment providers: the Open Source software is an enabler for them to sell hardware, but also to evolve and take into account the growing part of software in their activities related to the growing part of virtualisation.
Service providers: As expected, they get revenues from selling professional services.

Factors allowing the success

Open governance: The open governance of these projects is hosted by Open Source foundation, like the Linux Foundation or the OpenStack foundation, but also a standardisation organisation (ETSI).
Involvement of end users: Many companies are involved to develop software that is necessary to their core business.
Open Standards and specifications: The involvement of standardisation organisation is an additional sign of the tight relations between Open Source and Open Standards. Developing Open Source reference implementations could appear like the natural extension of creating open specifications, even though the standards industry may consider Open Source reference implementations problematic as they prevent the monetization of the FRAND[46] IP they have created.
Integration between Open Source projects: In addition to specific governance efforts (like LFN), some projects emphasize on their interaction with upstream projects (like OPNFV), co-organising events, etc.
Permissive/Weak copyleft licenses: most of the project are licenced under permissive (Apache-2.0) licences, some under weak copyleft (EPL-1.0).

Virtual Radio Access Networks: example of a different kind of openness

The virtualisation of Radio Access Networks has more recently been by a series of projects that claim some openness :

- The O-RAN Alliance (https://www.o-ran.org)
- The Telecom Infra Project's OpenRAN group (https://telecominfraproject.com/openran/)
- Cisco's Open vRAN

insights.
46Fair, Reasonable, And Non-Discriminatory (FRAND) is a common licencing policy in standards organisations, aiming at facilitating a wide adoption while still allowing direct monetization of the involved IP rights. FRAND is globally recognized as incompatible with Open Source legal and business models. Open Source oriented standards organisations generally adopt an RF (Royalty Free) IP policy.

The definition of openness in these contexts is very different from the Open Source Definition, even though some may also produce actual Open Source software, under a true, OSI-approved open source license.
The ORAN Alliance (which results from the merger of Xran Forum with C-Ran Alliance) creates specifications, whose implementations are subject to a FRAND policy, which is incompatible with Open Source models, and develop software, via two different subprojects:

- The O-RAN Specification Code Project is an O-RAN Alliance project. “Its objective is to allow contributions that will acknowledge essential patents.” It “leverages the O-RAN Software License.”, which is not an Open Source Licence.
- The O-RAN Software Community, which produces actual Open Source software, and is implemented as a series of LF Projects, LLC, like the subprojects of the LF Networking project mentioned above. The software is published under an Apache-2.0 licence, which include a royalty-free patent grant.

The OpenRAN group of the Telecom Infra Project has a similar IP policy (RAND https://telecominfraproject.com/wp-content/uploads/PG-Charter-OpenRan.pdf), incompatible with Open Source software.
There is little information publicly available regarding Cisco’s Open VRan, but their mentions of openness refers more to defining standard interface than to developing Open Source software.

Although those projects have already achieved significant involvement of various players, they are still relatively young to allow an evaluation of their success.

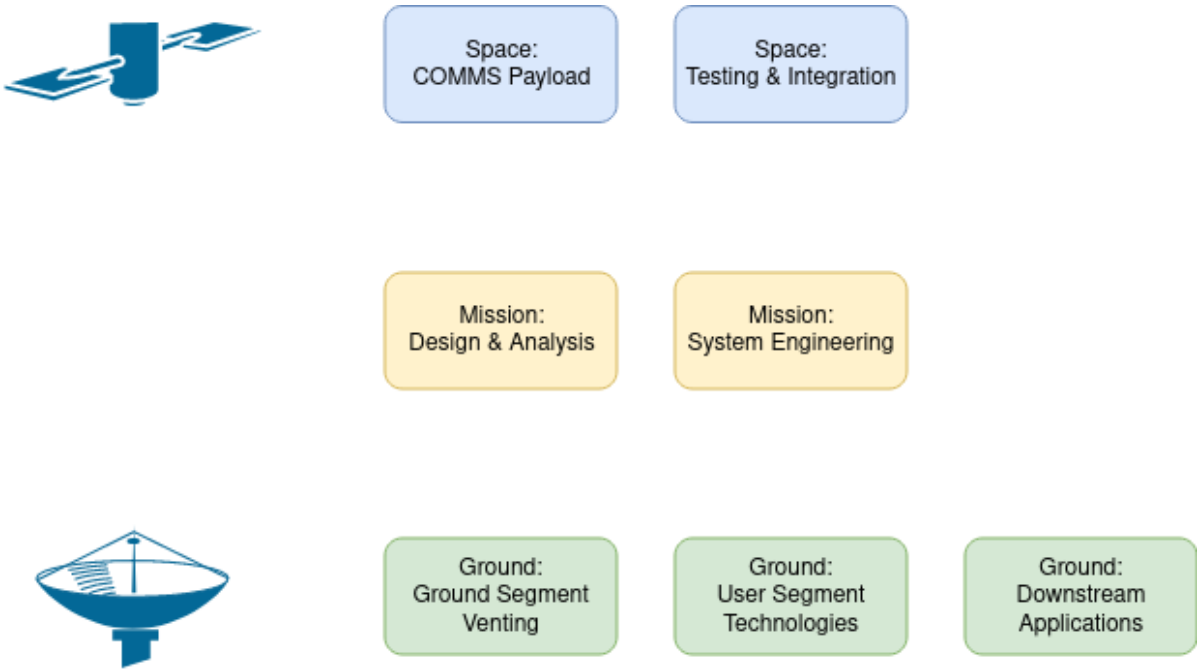
2.1.5 Open Hardware in telecom

Different projects of Open Hardware have emerged in the field of telecommunication. The most ambitious one is probably the Telecom Infra Project mentioned above: it was launched by Facebook, after the success of Open Compute[47] (an Open Hardware project for datacenters). TIP hasn’t yet reached the level of its predecessor, probably in relation to Intellectual Property issues, which are more present in the telecom sector. Some commercial products from working groups like CrowdCell have already been put on the market[48].
Some smaller projects have achieved a significant success from a technical and community point of view, like Myriad RF[49], a family of Open Source hardware and software projects for wireless communications.

47 https://www.opencompute.org/
48 https://limemicro.com/news/lime-microsystems-nextcloud-to-demonstrate-limenet-crowdcell-edge-computing-capabilities-at-tip-summit-2019/
49 https://myriadrf.org/

3. SATCOM Domains and applicability of Open Source

Since the inception of the Satellite Communications as a part of the broader Space Industry, there have been various attempts to identify distinct domains within it [50] [51] [52] [53].
Since there is no universal way to identify SATCOM domains, for the context of this report we chose to use a breakdown of the industry that is more relevant to the existing open source projects identified.



COMMS Payload

Domain	**COMMS Payload**
Statistics	8 Open source projects identified
Models	1. Business to Business 3. Common Upstream 5. Open Hardware - Design Focus

50Abe, Yuma & Tsuji, Hiroyuki & Miura, Amane & Adachi, Shuichi. (2018). Frequency Resource Management Based on Model Predictive Control for Satellite Communications System. IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences. E101-A. 2434-2445. 10.1587/transfun.E101.A.2434.
51Barbera, S & Pighetti, L & Fernández Piñas, David & Admella, M & Cano, J. (2015). SESAR SatCom System Identification and Verification Strategy. 10.1109/DASC.2015.7311489.
52Blisle, Claude & Andreadis, Peter & Bernier, Steve & Lvesque, Franois & Barbeau, Michel. (2000). Bandwidth Allocation for IP Traffic Over Satellite Links.
53Knopp, A. & Schwarz, Robert & Lankl, Berthold. (2011). MIMO system implementation with displaced ground antennas for broadband military SATCOM. 2069-2075. 10.1109/MILCOM.2011.6127624.

Applicable Support	a) Support common upstream projects b) Seed open hardware creation

On the Space segment a SATCOM domain identified is directly the COMMS payload including the hardware and software needed for it. A relevant development is that SATCOM has been extended into the Micro [54] (and in recent cases Nano [55]) Satellite manufacturing industry, with a steady growth in the last decade. For its hardware focused nature, applicability of open source models can be examined through the lenses of the (quite recent) open hardware initiatives. One model that could be applied is **5. Open Hardware - Design Focus** that could bring a differentiation factor for fabrication less initiatives focused on designs and overall architecture versus trying a more traditional vertical hardware vendor approach. This shift could signal a change in the business models traditionally associated with hardware manufacturing in the Satellite manufacturing industry, creating reusable designs distributed with open licenses freeing up resources to new entrants in the market and enabling innovation with a focus on state-of-the-art technologies (rather than reinventing the wheel). It is worth noting that such an open source approach (front-loading costs of R&D) is much more scalable and economically viable for the scale envisioned for the satellite manufacturing industry in the near future, with imminent deployment of multi-hundred LEO constellations for SATCOM purposes [56]. For the software side of the COMMS payload domain two models are relevant and specifically **1. Business to Business** open source and **3. Common Upstream** since both of those models imply controlled connections with partners and added value through collaborations with a common baseline.

Testing and Integration

Domain	**Testing and Integration**
Statistics	10 Open source projects identified
Models	3. Common Upstream 5. Open Hardware - Design Focus (for Environmental Testing)
Applicable Support	a) Support common upstream projects b) Seed open hardware creation

Testing and integration deserves a separate call-out (rather than including in the payload manufacturing industry) due to the high-level of reusability that can be achieved. We can identify two sub-domains, Functional and Environmental and for the purposes of this report the focus will be on the Functional one. Access to reliable testing and integration has been traditionally hard for new entrants in a market, and existing players in the ecosystem create their own vertical approaches many times missing opportunities for synergies with other players. An open source model applicable to this area of the industry is **3. Common Upstream** that could focus on creating reusable GSE, testing and integration facilities that could benefit a wide variety of new or old players in the industry lowering the barrier to entry.

54Scott C. Burleigh, Tomaso De Cola, Simone Morosi, Sara Jayousi, Ernestina Cianca, and Christian Fuchs, “From Connectivity to Advanced Internet Services: A Comprehensive Review of Small Satellites Communications and Networks,” Wireless Communications and Mobile Computing, vol. 2019, Article ID 6243505, 17 pages, 2019. https://doi.org/10.1155/2019/6243505.
55https://spacenews.com/startup-plans-space-based-cubesat-network/
56https://artes.esa.int/megaconstellations/overview

Standardization (not only on the bespoke standards but on their practical implementation too) and community creation/engagement can strengthen ties in the ecosystem, create opportunities for new synergies and collaboration and enhance the reliability and TRL status of those technologies. From a business model perspective, since services are the main driver for this area of the industry we should envision open source initiatives offering their services on top of their developed technologies, while their openness enabling easy integration and proliferation of them.

Mission: Design & Analysis

Domain	**Mission: Design & Analysis**
Statistics	30 Open source projects identified
Models	2. Controller Ecosystem 4. Ethically open-source
Applicable Support	a) Support controlled ecosystem explorations

Mission design and analysis is a highly specific area of the SATCOM industry with a strong focus on academic research and early R&D efforts, while serving the ultimate business needs of the implementer. Due to the careful balance needed between flexibility and serving specific needs the application of an open source model in this area needs to be examined through the lens of the existing projects. Model **4. Ethically open-source** seems to be the one applied currently by the major open source players in this space (GMAT, Orekit, Poliastro and others), and although it has yielded impressive results a cross comparison with existing proprietary competitors (e.g. AGI STK) it calls probably for a new perspective. One possible future avenue for this area is the exploration of **2. Controlled Ecosystem** where an implementer can create (or amend and existing) core mission design and analysis package and while releasing it open source, hosting it online as a freely accessible service with paid integrations and customizations, ensuring wide adoption and an emergence of a market around it.

System Engineering

Domain	**System Engineering**
Statistics	8 Open source projects identified
Models	3. Common Upstream
Applicable Support	a) Support common upstream projects

For system engineering software tools the most relevant open source model that could be applied is the **3. Common Upstream**. Although restricted in terms of revenue possibilities, this model offers the best strategy for wider adoption in this area of SATCOM. The upstream controller/creator will be able to steer the downstream usage of their technology while creating avenues and openings for potential collaborations and partnerships in other (downstream) parts of the technology stack. Support from R&D funds is crucial for this model

and agencies or major players are expected to invest and support such efforts in light of enabling the ecosystem and lowering the barrier to entry for new players thus creating space for innovation and growth of the market.

Ground Segment Vending

Domain	**Ground Segment Vending**
Statistics	9 Open source projects identified
Models	1. Business to Business 2. Controlled Ecosystem
Applicable Support	a) Incentivization for standardization b) Support controlled ecosystem explorations

The Ground Segment Vending domain is in the midst of a “transformation” of its own following the shift we are seeing in the upstream SATCOM sector with commercial adoption of miniaturized satellites. The opportunity for open source models in the area of the industry is dual and depends on the envisioned end-game for the potential implementer of that model. One approach is the application of the model **1. Business to Business** open source by having a ground segment vendor pushing for a wide OEM adoption of their core technology (possibly including terminals) disrupting existing proprietary offerings and increasing the market share, enabling for capitalization on additional services. Another model that can be applied is **2. Controlled Ecosystem** with a vendor taking a more vertical approach to their technology offering, controlling the development of the base technology components while allowing for a flourishing market to be developed on top of the core offering. Expanders in that ecosystem will be able to meet niche and diverse needs while the core developer/controller benefits from the wide adoption of their technology to offer premium services and customization options as service providers. The first model is applicable to the “Teleport” vendors more, while the second towards the “Owned Ground Segment” vendors. A critical approach that will enable flourishing of open source development models in this domain is the standardization that needs to happen through intervention (by the agency) to ensure incentivized cross-compatibility through application of standards.

User Segment technologies

Domain	**User Segment technologies**
Statistics	1 Open source projects identified
Models	2. Controlled Ecosystem 5. Open Hardware - Design Focus
Applicable Support	a) Support controlled ecosystem explorations

User segment technologies (user terminals, antennas and related technologies) can be considered a saturated technologically (and from a business model perspective) domain of the SATCOM industry, specifically when it comes to GEO related SATCOM offerings (e.g.

media broadcasting). What can be considered an interesting opportunity for a novel application of open source models are the new developments (and user/functional needs) of the LEO based SATCOM offerings (from SATCOM LEO constellations) on M2M and IOT related technologies. The model **5. Open Hardware - Design Focus** can be applied when it comes to hardware development (user segment terminals and antennas) since most of the designs will have to differentiate themselves from the current offerings of the GEO-based market. An additional model of **2. Controlled Ecosystem** can be relevant especially when such applications reach a critical mass and user adoption to enable enough exploitation of additional services and customization on top of a large market. Since there is a pace difference between design, development and deployment of technologies in space and on ground, backwards compatibility should be cultivated. Sticking to open standards (and away from proprietary solutions) will also ensure wider adoption from neighbouring ecosystems which is key for the success of user segment technologies.

Downstream Applications

Domain	**Downstream Applications**
Statistics	25 Open source projects identified
Models	3. Common Upstream 7. Mass market open-source
Applicable Support	a) Support common upstream projects

For downstream applications as a SATCOM domain, Open Data business models (coming from the EO industry) have been steadily explored as an option. Due to their nature (not being development models) they are not within the realms of this call and proposal, but are presenting interesting opportunities for exploration of open source development methodologies and models, by drawing parallels with other industries. Model **7. Mass Market open-source** can be applied in this area specifically due to its focus on user-facing products with a massive target audience. Such a model can enable the implementer to explore differentiation opportunities like localization, internationalization, mass user-testing and contributions/feedback from power and conscious users. Also with the lens of machine-learning on top of available big-data for downstream usage another applicable[57] open source development model is the **3. Common Upstream** since it provides opportunities for focusing only on the state-of-the-art development and re-using through collaborations datasets and code.

An overview of the link between the SATCOM domains identified and the suggested exploration of open source development models can be found in Figure 2. Notice that 4. Ethically Open-Source was not suggested as a specifically applicable open source development model since it could be universally applicable and it relevant to the driving forces behind the development rather than the domain applied. Also 6. Opportunistic is not suggested for any SATCOM domain since it can be considered a “fallback” or un-intentional open source development model without much potential for growth.

57Sonnenburg, SÃ¸ren, et al. "The need for open source software in machine learning." *Journal of Machine Learning Research* 8.Oct (2007): 2443-2466.

	B2B	Controlled Ecosystem	Common Upstream	Ethically Open-Source	Open Hardware	Opportunistic	Mass Market
Space: COMMS Payload							
Space: Testing and Integration							
Mission: Design & Analysis							
Mission: System Engineering							
Ground: Ground Segment Vending							
Ground: User Segment technologies							
Ground: Downstream Applications							

Figure 2

4. Existing Usage of Open Source in SATCOM

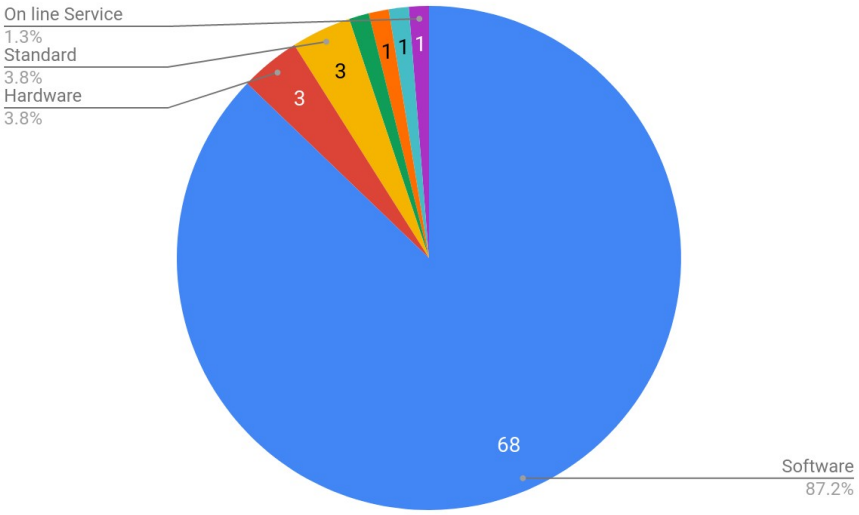
For the purposes of this activity a list of all existing open source projects in the SATCOM industry has been compiled (and delivered as an annex to this document). As input to this list we conducted original research in the domains identified, used existing relations of Libre Space Foundation and Inno3, and had input from ESA through a series of meetings exchanging lists and information around identified open source projects. The following attributes were compiled for each identified project:

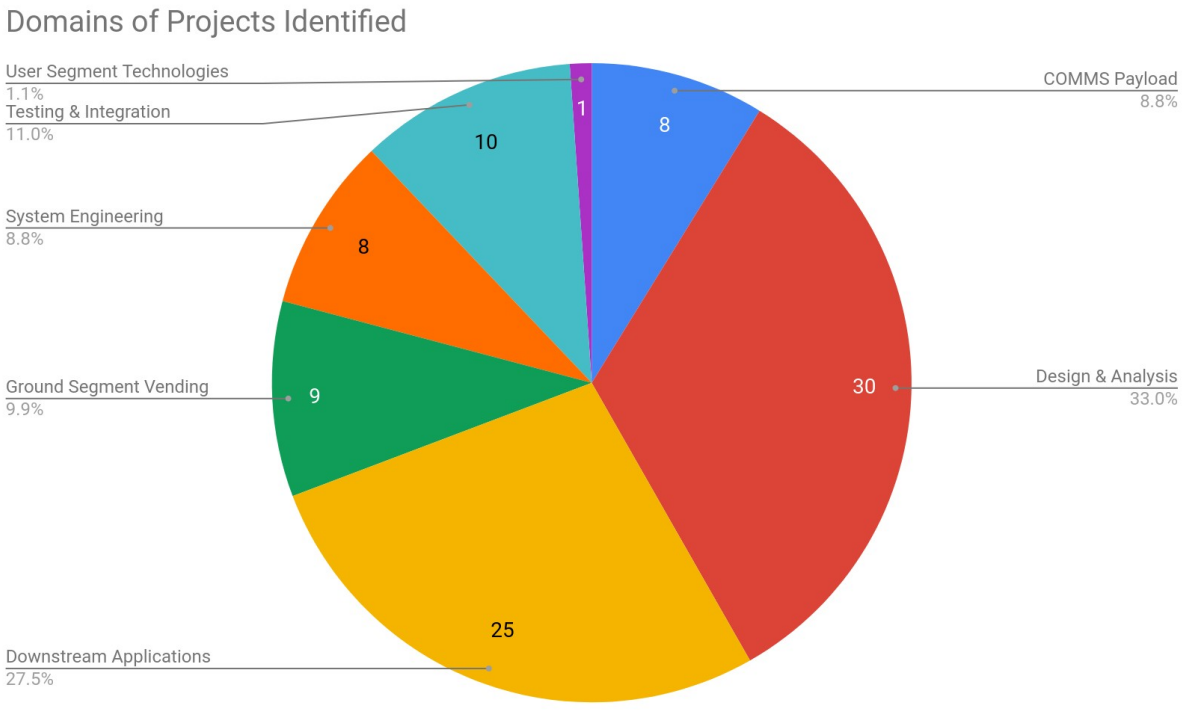
- Name
- Description
- Category
- License
- Specificity
- Type
- Link
- Format
- Type of maintaining entity

The list was completed with 79 entries with 27 of them identified as Priority 1 projects that should be contacted for further exploration. The list can be found online here:

https://opensatcom.org/projects/

The Projects identified are predominantly Software (~87%) which is to be expected given the maturity of the open source development models around the Software industry.





Maintaining such a list in a publicly documented way will have an important positive impact on any efforts of open source development ecosystem growth since mapping an ecosystem and identifying gaps for intended approach as well as learning from previous experiences of other projects is crucial for any newcomer project/initiative/company.

5. Reach out to identified Open Source Projects

In order to get insight and valuable information with various existing open source projects in the SATCOM industry we reached out with a survey request to them through the following communications:

*Hello **[NAME],***

My name is Manthos Papamatthaiou and I am a Technical Manager of Libre Space Foundation for OpenSatCom, an ESA funded activity to investigate the usage of open source methodologies in the Satellite Communications (SATCOM) Industry.

*You have been identified as a contact of an open source project that is within or related to the SATCOM industry and in specific **[PROJECT NAME]**. With this email we would like to invite you to complete a quick survey regarding your project and its usage.*

The survey only takes 5 minutes and can be found here:
[LINK]

Within this survey there is a possibility for you to opt-in for further communications about the project and a possible call-interview opportunity with your project as we dive deeper into example use-cases of open source in the broader SATCOM ecosystem.

Feel free to forward this email to another person within your project in case you are not responsible for such communications or not related to the project anymore.

The results of this survey will be published openly on the project website. We will update all participants when this happens.

Thanks a lot for your time and contribution towards enhancing the application of open source methodologies on the Satellite Communications Industry.

For OpenSatCom,

Manthos Papamatthaiou

We received responses from the following projects/people:

Full Name	Name of the project	Website	Repository	Main license
Daniele Lacamera	pepSAL	github.com/danielinux/pepsal	github.com/danielinux/pepsal	GNU General Public License version (GPL)
Luc Maisonobe	Orekit	https://ww.orekit.org	https://gitlab.orekit.org/orekit/orekit.git	Apache License
Artur Scholz	LibreCube	https://librecube.org/	https://gitlab.com/librecube	MIT License (MIT)
Juan Luis Cano Rodríguez	poliastro	https://docs.poliastro.space/	https://github.com/poliastro/poliastro/	MIT License (MIT)

Jani Puttonen	Development of an open-source, modular and flexible satellite network simulator	http://sns3.org	https://github.com/sns3/sns3-satellite	GNU General Public License version (GPL)
Johan De Claville Christiansen	Cubesat Space Protocol (libcsp)	libcsp.org	Https://GitHub.com/libcsp/libcsp	GNU Lesser General Public License (LGPL)
Stéphane LACRAMPE	Capella	https://polarsys.org/capella/	http://git.polarsys.org/c/capella/capella.git/	Eclipse Public License
Pierros Papadeas	SatNOGS	https://satnogs.org	https://gitlab.com/librespacefoundation/satnogs	GNU General Public License version (GPL)
Tom Henderson	ns-3	https://www.nsnam.org	https://gitlab.com/nsnam/ns-3-dev.git	GNU General Public License version (GPL)

Most of the project responded that the relevant open source models to them are either “Ethically-Open Source” or “Common Upstream” which are indicative of a relatively immature ecosystem, since more established and mature models (with higher chances of sustainability) like “B2B” or “Controlled Ecosystem” were less relevant for the surveyed projects.

On the question of “How your project can be helped” for further success in this emerging ecosystem the responses where:

- Reaching out to current users
- Funding, providing reference data for validation, contributing to the community
- Funding for prototypes and studies; help in outreach to a wider audience
- Funding is essential to secure full time contributors, so far achieved for 3 months a year with Summer of Code programs
- Testing and feedback give direction to the project and inform the developers of existing problems
- Technical writers could improve the status of documentation taking into account the different type of users reading it"
- Ratification of the protocol as a standard (or set of standards)
- Funding from large organisations to develop new features and extensions. Official support from very large/public organizations
- Funding of baseline development of underlying technologies. Outreach to a global community in order to achieve better coverage.
- More maintainers are needed.

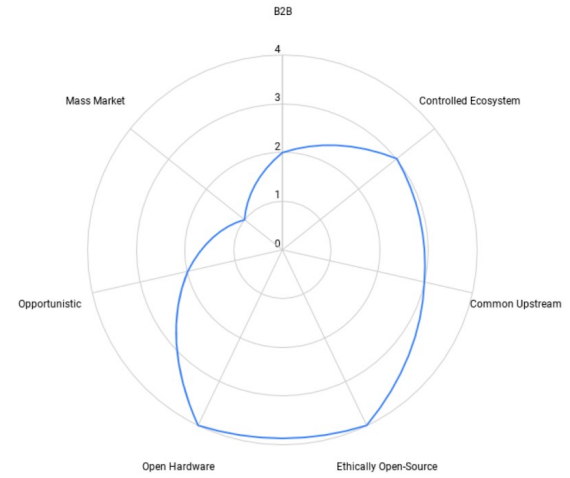
showcasing funding (and thus sustainability) as the critical aspect for them and a wider community and audience (through recognition in the industry) as a secondary worry.

All the surveyed projects where aware of their applied usage in the SATCOM industry and specifically for each one of them:

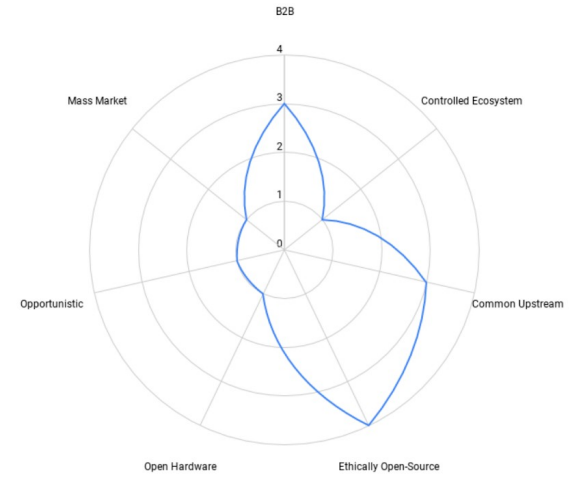
Project Name	What type of usage are you aware of?	Which sections of the SATCOM industry are targeted by your project?
pepSAL	Commercial, Academic, Research	Ground segment (terminals / networks)
Orekit	Commercial, Academic, Research	Other space segment electronics & hardware, Operation services, Ground segment (terminals / networks), End user applications, Modeling and mission simulation
LibreCube	Academic	RF Electronics & Hardware, Operation services, Ground segment (terminals / networks)
poliastro	Commercial, Academic, Research	Modeling and mission simulation
SNS3	Research	Modeling and mission simulation
Cubesat Space Protocol (libcsp)	Commercial, Academic, Research, Military	RF Electronics & Hardware, Other space segment electronics & hardware, Operation services, Ground segment (terminals / networks), End user applications, Modeling and mission simulation
Capella	Commercial, Academic, Research	RF Electronics & Hardware, Antennas, Operation services, Ground segment (terminals / networks), Modeling and mission simulation
SatNOGS	Academic, Research	RF Electronics & Hardware, Antennas, Mechanical, Operation services, Ground segment (terminals / networks)
ns-3	Commercial	Modeling and mission simulation

The survey results around the models can be seen as follows:

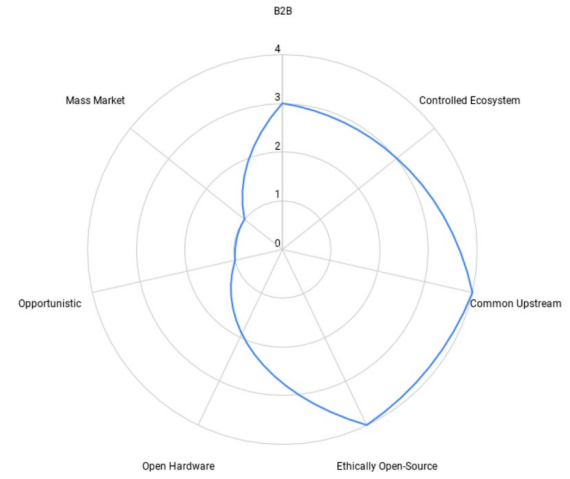
SatNOGS



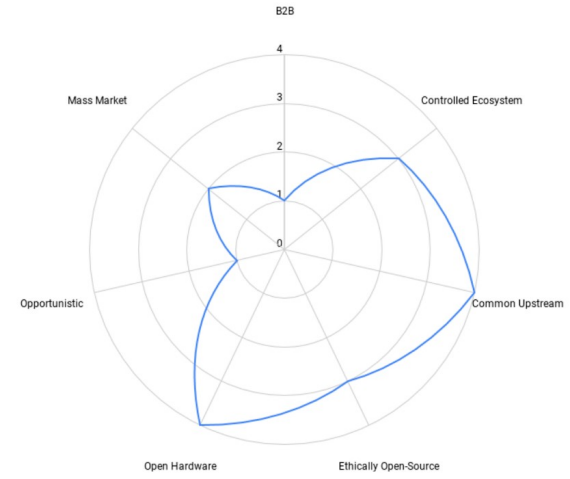
pepSAL



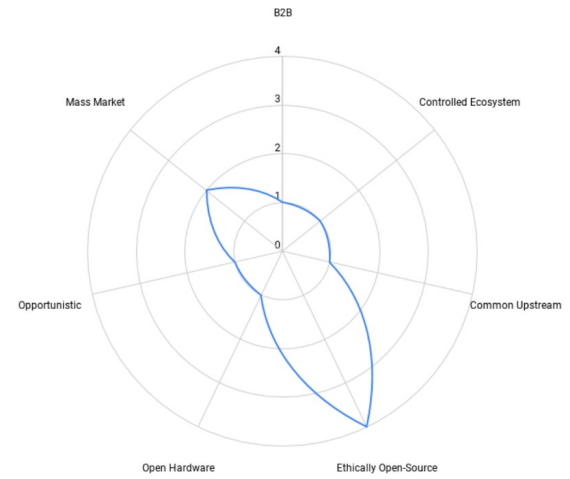
Orekit



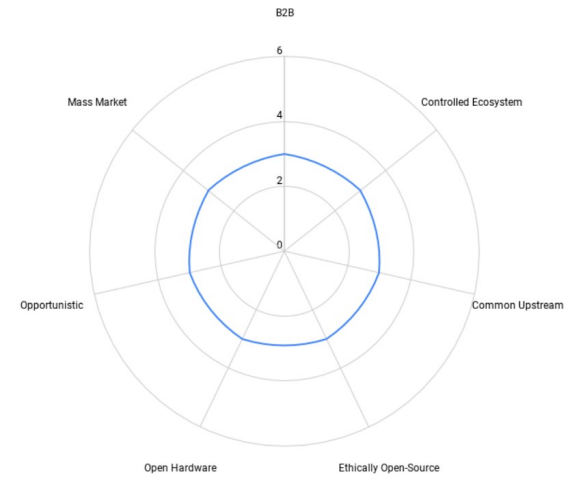
LibreCube



poliastro

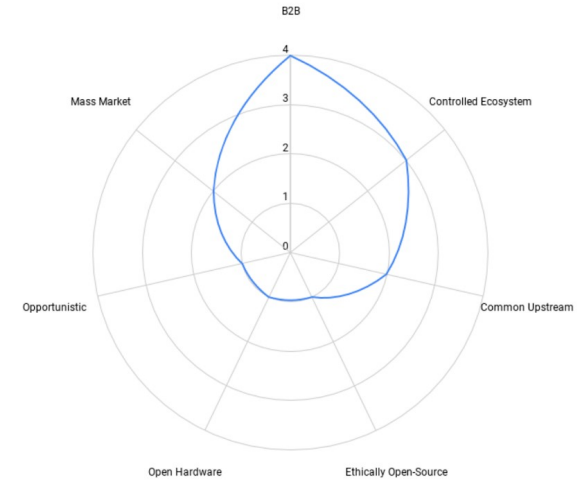


sns3

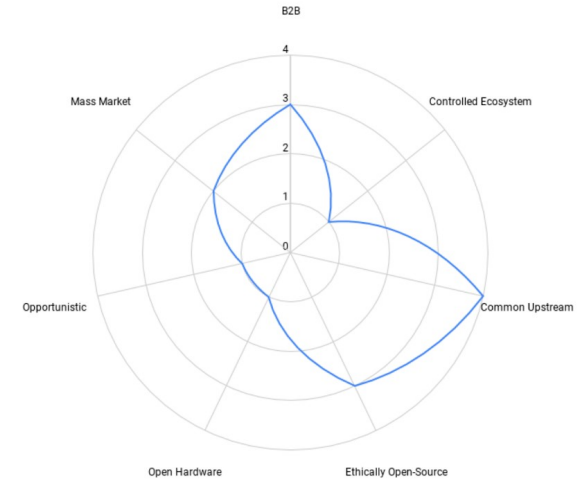


1. Not related at all, 2. Slightly unrelated, 3. Slightly related, 4. Highly related

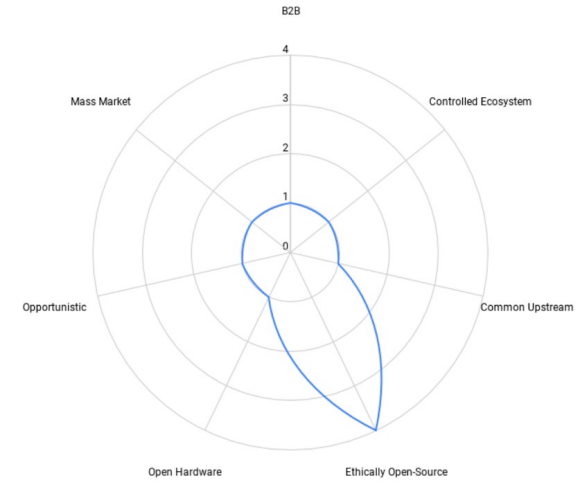
Capella



Cubesat Space Protocol (libcsp)



ns-3



1. Not related at all, 2. Slightly unrelated, 3. Slightly related, 4 Highly related