Paper OS16

Open-Source practicalities and the legal side

Katja Glaß, Katja Glass Consulting, Berlin, Germany

ABSTRACT

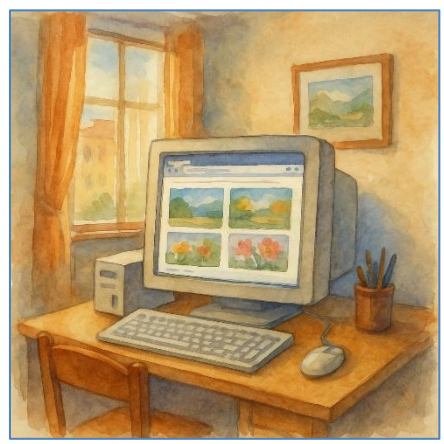
The rising popularity of open-source within the pharmaceutical sector presents a multitude of opportunities. However, next to its benefits there are uncertainties surrounding permissible usage. Did you know that source code in a paper default to copyright protection? Moreover, the prevalence of copy-left licenses in R raises questions about the obligation for R-based software solutions to also adopt copy-left open-source practices.

This presentation aims to demystify the legal aspects of leveraging open-source in pharmaceutical solutions while ensuring compliance with the respective open-source licenses. It will offer insights, shedding light on how to appropriately utilize open-source solution through deriving or enhancing to a commercial or open-source solution. Attendees will gain a comprehensive understanding of how to effectively harness open-source tools while adhering to legal frameworks, fostering innovation and maximizing the potential of open-source.

DISCLAIMER

This paper is intended for educational and informational purposes only and does not constitute legal advice. It should not be relied upon as a substitute for obtaining professional legal counsel or conducting independent legal assessment. The examples and interpretations provided are illustrative and may not apply to all jurisdictions or specific cases. Statements of fact and opinions expressed are solely those of the author and do not necessarily reflect the views of any organization or institution. While care has been taken to ensure accuracy, no guarantee is made as to the completeness or correctness of the information presented.

INTRODUCTION



Once upon a time, in the 1990s, the Internet was born - a time of astonishment and excitement. Suddenly, information was everywhere. Knowledge, articles, and, most fascinatingly, images became accessible at the click of a button. What once required cutting out pictures from magazines could now be achieved through a simple save.

But with this new freedom came new conflicts. Copyright still applies, even in the digital world. As users eagerly shared and reused content, questions arose and so did legal claims. Early lawsuits highlighted the uncertainty around what was allowed, and where the boundaries lay. Even today, image copying remains a major challenge. A 2019 report estimated that 2.5 billion images are stolen each day[1].

At the core of this issue lies a simple gap: missing education. Many users were never taught how copyright works in the digital age. How can I use content? How should I share it? While copyright awareness is increasingly part of school education today, older generations and even many professionals still lack understanding of how intellectual property applies online. And when it comes to open source, confusion grows even further: what can be used, modified, or shared, and under which conditions?

This paper aims to contribute to that missing education. It seeks to raise awareness of the challenges, rights, and opportunities that arise when using open source in practice. Through real-world examples and practical insights, it will illustrate how legal frameworks intersect with everyday development activities in the pharmaceutical and life science context. The goal is not only to clarify potential risks but also to empower readers to make informed decisions—transforming uncertainty into confident, compliant, and collaborative open-source engagement.

RIGHTS AND LIMITATIONS

KNOWING THE LIMITS

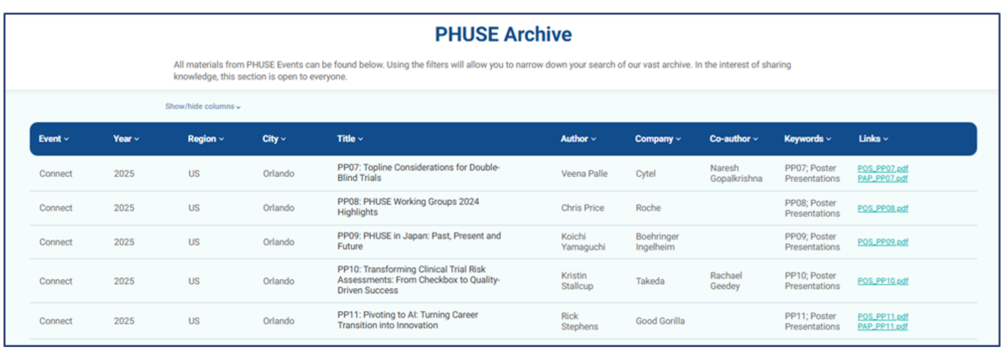
When we talk about rights and limitations, we often think first about written text or arts. But the same principles apply to everything we create and share, from images to source code. The default is copyright – everything we create is automatically protected by copyright. It does not require any registration. Whether it is a figure in a publication, a photograph on a website, or a script, the author owns the rights by default.

This means that, in most cases, copying text, code, or images without permission is not allowed. However, learning from existing work, referencing it, or drawing inspiration from it is generally acceptable. The key distinction lies between using and reusing. We can study and understand materials, but we cannot simply duplicate or redistribute them as our own.

To share work more freely, creators can apply explicit licenses that define what others may do with their content. If no license is given, the default remains full copyright. This often causes confusion, especially in academic or technical contexts where sharing and collaboration are encouraged but legal clarity is missing.

Consider PHUSE conferences and papers, for example. PHUSE is an organization specifically designed to connect, share and advance. But when papers are created, we can read and learn from them, cite their results, and build upon the ideas they present. But copying parts of their text, figures or contained source code is not allowed unless the author explicitly permits it.

Copyright per default



PHUSE Archive

All materials from PHUSE Events can be found below. Using the filters will allow you to narrow down your search of our vast archive. In the interest of sharing knowledge, this section is open to everyone.

Show/hide columns

Event	Year	Region	City	Title	Author	Company	Co-author	Keywords	Links
Connect	2025	US	Orlando	PP07: Topline Considerations for Double-Blind Trials	Veena Palle	Cytel	Naresh Gopalkrishna	PP07; Poster Presentations	POS_PP07.pdf PAP_PP07.pdf
Connect	2025	US	Orlando	PP08: PHUSE Working Groups 2024 Highlights	Chris Price	Roche		PP08; Poster Presentations	POS_PP08.pdf
Connect	2025	US	Orlando	PP09: PHUSE in Japan: Past, Present and Future	Koichi Yamaguchi	Boehringer Ingelheim		PP09; Poster Presentations	POS_PP09.pdf
Connect	2025	US	Orlando	PP10: Transforming Clinical Trial Risk Assessments: From Checkbox to Quality-Driven Success	Kristin Stallcup	Takeda	Rachael Geedey	PP10; Poster Presentations	POS_PP10.pdf
Connect	2025	US	Orlando	PP11: Pivoting to AI: Turning Career Transition into Innovation	Rick Stephens	Good Gorilla		PP11; Poster Presentations	POS_PP11.pdf PAP_PP11.pdf

Awareness of these rights and limitations helps us act responsibly. It encourages us to respect the work of others while also protecting our own. Copyright is not meant to restrict learning; it exists to recognize authorship and ensure fair use. But awareness also brings power to decide how openly we want to share our creations. As creators, we can actively enable the use, sharing, and further development of our work by attaching a license to it. It is easy to tag it with a Creative Commons license for images or articles, or an open-source license for software and code. We can define what others are allowed to do.

Remember, also for source code copyright applies by default. No license means no reuse. Copying unlicensed code can lead to legal issues. Open-source licenses can explicitly grant permissions if their terms are followed. In case you are not sure if a license is available, the safe approach is to learn from the idea, do not copy the code itself. You can re-create it in your own words and logic.

FROM COPYRIGHT TO LICENSE

Copyright is the starting point for everything we create. Licenses provide a practical way to change that default: they define what others can do, under which conditions, and with what level of attribution or openness.

There are many types of licenses depending on the nature of the content.

- **Data and data models** may use open data licenses such as the Open Data Commons (ODC) suite or Creative Commons for data, which define how datasets and data structures can be reused, redistributed, or remixed. These licenses are increasingly relevant in the context of **AI model training**, where large datasets serve as input for machine learning. Clear licensing ensures that both data providers and AI developers understand the legal boundaries of data use, promoting transparency, reproducibility, and ethical AI development.
- **Images, documents, and training materials** often rely on Creative Commons (CC) licenses, which provide a clear, standardized way to indicate what level of openness, modification, or attribution is allowed. These licenses are also increasingly applied to **AI-generated content**, helping clarify whether derivative or machine-assisted works can be reused or redistributed.
- **Source code and AI models** are typically governed by Open Source Initiative (OSI). Approved licenses such as MIT, Apache 2.0, or GPL. These licenses meet the OSI's Open Source Definition, ensuring that software and by extension, AI models whose weights or architectures are released under similar terms, can be freely used, modified, and distributed.

Creative Commons (CC) licenses offer a clear and flexible way for creators to decide how openly their work can be shared. As shown in the Creative Commons License Spectrum, the range spans from fully open - Public Domain (PD) or CC0, allowing unrestricted use - to more structured forms like CC BY, which requires attribution, and by-attribution and share alike CC BY-SA, which ensures that derivatives remain under the same open terms. This spectrum empowers creators to balance openness and control, making it easy to share knowledge and creativity while clearly defining the rights and responsibilities of users.

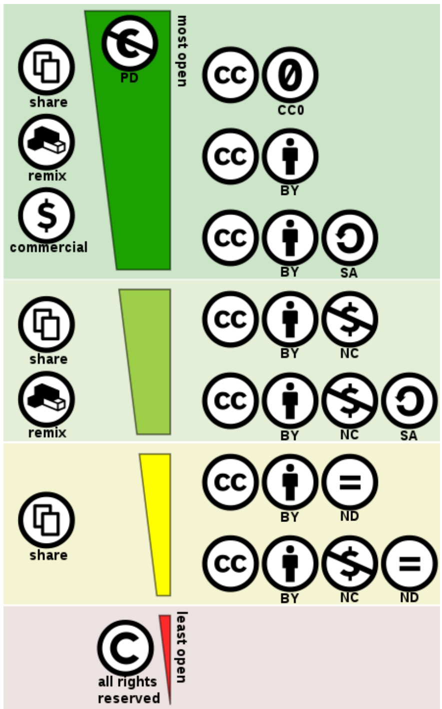


Figure 1: Creative Commons License

Spectrum[2]

Within the open-source licenses, two main categories dominate: permissive and copyleft licenses. Permissive licenses such as Apache 2.0, MIT, or BSD offer maximum flexibility, allowing derivative works to be re-licensed, including for commercial or proprietary use. They are comparable the CC BY model in Creative Commons. In contrast, copyleft licenses such as GPLv3 or Microsoft Public License require that any derivative work remain under the same license terms. This “share alike” principle preserves openness across generations of reuse, ensuring that improvements to open-source software continue to benefit the community.

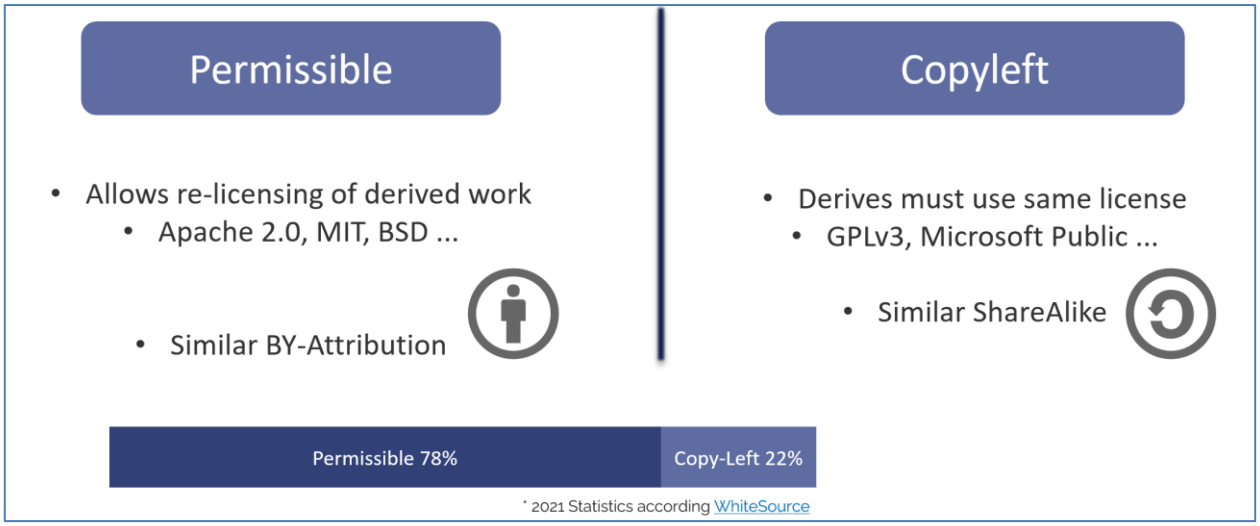


Figure 2: Open-Source License Types

According to 2021 statistics from WhiteSource[3], about 78% of open-source projects use permissive licenses, while 22% follow copyleft models, reflecting a strong trend toward flexible yet responsible sharing.

In practice, reality often follows the principle of “no plaintiff, no judge.” Countless examples of unlicensed or misused material exist online - not necessarily out of bad intent, but because people simply do not know what is legally allowed. This lack of awareness leads to widespread accidental infringement, especially when sharing code or figures. A practical recommendation for cases where PHUSE paper content might be reused: do not assume it is open source - ask the author for permission. In many cases, the author originally intended to make the work reusable and will gladly provide or clarify the license. A short email can turn an uncertain situation into a compliant and respectful collaboration.

FROM PRINCIPLES TO PRACTICE

When it comes real examples, things are not always as clear as they appear. Take the example of SAS documentation, which is publicly available online but accompanied by detailed Terms of Use.

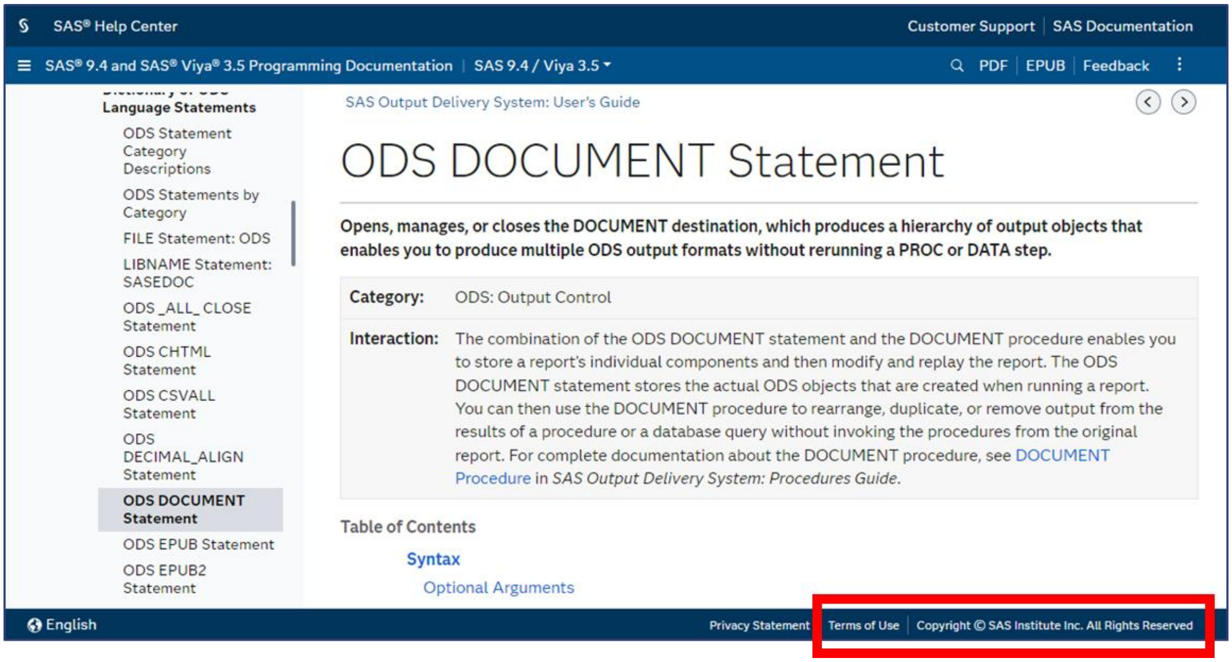


Figure 3: Screenshot from the SAS documentation highlighting the Terms of Use

The "Terms of Use"[4] for the website which contains the documentation is the following:

1) […] own personal informational, educational or other **non-commercial** purposes unless otherwise expressly indicated;
2) all copies you make must properly **attribute** […]
3) you may **not modify** any of the materials […]
4) …

In theory, that means even a simple copy-paste from an online code example could be restricted. In practice, however, very few users ever read these terms. The expectation is that documentation examples, especially code snippets, are meant to be reused for learning and problem-solving. The main purpose of such terms is to protect the provider's content from being republished as a book, integrated into competing products, or redistributed at scale and not to stop customers from experimenting.

Legally, though, these examples are not freely reusable. The license explicitly limits their use to personal or educational purposes and prohibits modification or redistribution. That means incorporating SAS documentation code into a public GitHub project, for example, would formally violate the stated terms. It's a good reminder that even openly accessible content is not necessarily open in the legal sense.

Additionally, for those who have purchased the software, there are separate license terms that govern usage. A typical SAS Software License Agreement might state that "the customer may use any documentation only to support the customer's use of the related software."[5] Even with such a clause, the restriction remains - legally, the documentation and its code examples may not be reused or shared beyond that specific purpose. Posting them publicly or embedding them in other tools would still breach the license, regardless of intent.

LESSONS LEARNED AND RECOMMENDATIONS

The SAS documentation example illustrates how easily assumptions about openness can lead to legal uncertainty. Just because material is accessible online does not mean it is free to reuse. Even when intentions are good, copying or sharing documentation examples without an explicit license can create compliance risks.

SO WHAT CAN YOU DO?

When creating documentation, always include a license statement for any code examples, figures, or templates. This makes your intent clear and enables others to reuse your material safely. When writing posts, blogs, or papers, also include a short license note for any source code you share - especially if you intend others to learn from or build upon it.

If you wish to share code openly, consider simple and recognized options such as Unlicense or Public Domain (CC0) for complete freedom, or MIT if you prefer to keep attribution while still allowing broad reuse. These small additions make a big difference: they turn uncertainty into clarity, enable collaboration, and demonstrate professional responsibility when contributing to the open knowledge ecosystem.

OPEN-SOURCE LICENSE EXAMPLES

Open source allows the use, modification, and free distribution of software and code - typically under two main license types: permissive, which enables broad reuse with minimal conditions such as attribution, and copyleft, which requires derivatives to remain under the same open terms.

FINDING OPEN-SOURCE SOLUTIONS

However, one of the practical challenges today is finding open-source solutions that are relevant and compliant within the pharmaceutical and life sciences industry. While open-source development has grown rapidly, resources are still scattered across multiple platforms and initiatives. To address this, several dedicated repositories and collections have emerged that make it easier to discover, evaluate, and adopt open tools for regulated environments.

Some key examples include:

- Open Source Portal for Clinical Study Evaluations – https://www.glacon.eu/portal
  A curated overview of open-source projects, tools, scripts and educational material supporting clinical study evaluations.
- COSA Repository – https://cosa.cdisc.org/
  The CDISC Open Source Alliance (COSA) repository, hosting community-driven projects aligned with CDISC standards and interoperability goals.
- Pharmaverse – https://pharmaverse.org/
  A collaborative ecosystem of R packages and open tools designed for clinical reporting, analysis, and data standardization.

These resources provide starting points for identifying trustworthy, well-maintained open-source solutions, along with insights into their licensing, scope, and community support.

MIT EXAMPLE – CORE

A good example of a permissive license in practice is CORE (CDISC Open Rules Engine) - an open-source engine designed to validate study data against CDISC rules and compliance checks. Released under the MIT License, CORE allows for maximum flexibility in reuse and adaptation.

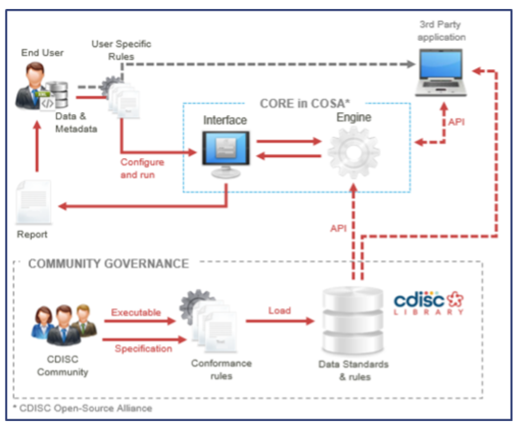


Figure 4: CDISC CORE using MIT

Because of this license, organizations and developers can freely copy the source, modify it, and use it in their own applications, as long as the original MIT license notice is retained. This enables derivative works such as a CustomCORE, where the engine is adapted or extended for specific internal workflows, while the resulting solution may remain proprietary.

The same applies to tools such as a Dataset Processing Tool, which might reuse or integrate selected code snippets from CORE or include this as hole or modified. As long as the MIT license reference is kept, the new solution could be open-source or commercial and can carry its own copyright. Real-world examples include the open-source Smart Submission Dataset Viewer and the commercial Pinnacle 21 (P21), both of which utilize CORE as part of their validation capabilities.

GPLV3 EXAMPLE - OVERVIEW

The GNU General Public License version 3 (GPLv3) is a strong copyleft license that defines not only how open-source software can be used but also how openness must be maintained. The accompanying graphic illustrates the key principle: rights are granted to users, not automatically to the public.

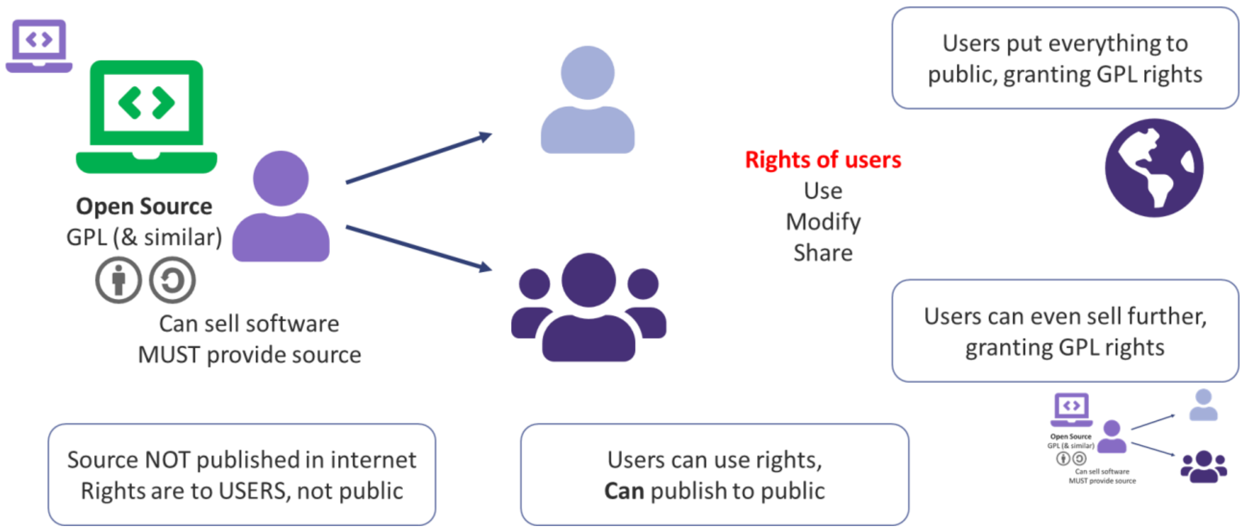


Figure 5: Rights and Limitations of GPLv3

Under GPLv3, anyone can use, modify, and share the software freely. The code can even be sold, but the seller must always provide access to the corresponding source code. This ensures transparency while allowing both commercial and non-commercial use.

Importantly, GPLv3 does not require that modified versions be made public by default - only that the users who receive the software have the same rights to use, modify, and share it. If users choose to make their version public, they continue granting the same GPLv3 rights downstream, creating a snowball effect where openness accumulates over time. Each new recipient inherits the same freedoms, ensuring that transparency and collaboration expand naturally with every reuse and redistribution.

GPLV3 EXAMPLE - OPENSTUDYBUILDER

A practical example of the GPLv3 model can be seen with OpenStudyBuilder. Imagine a scenario where a vendor (Vendor X) develops a customized version of OpenStudyBuilder on behalf of a pharmaceutical company (Pharma Y). The customized version as OpenStudyBuilderModified is deployed within Pharma Y and used by internal staff, external partners, and academic collaborators. The code may be managed in a private GitHub repository, accessible only to defined user groups.

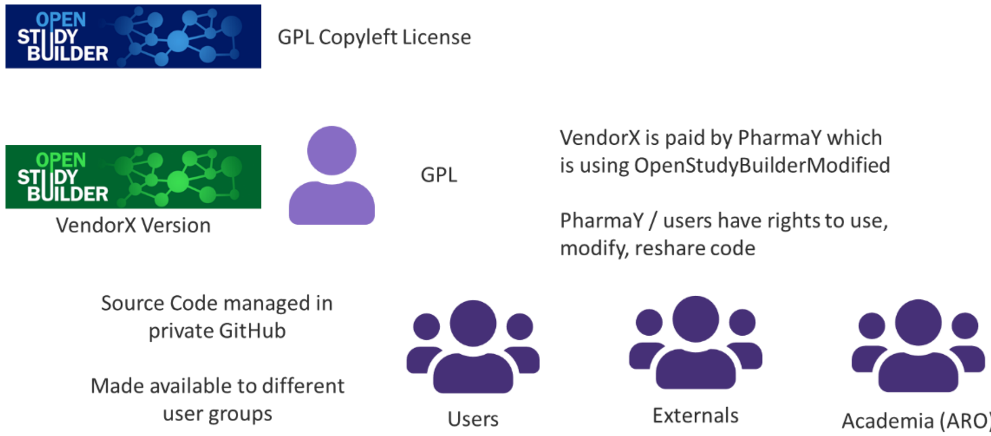


Figure 6: OpenStudyBuilder using GPLv3 with Modification Example

Even though the repository is private, the GPLv3 license ensures that every user who receives access to the software also receives the right to the source code. These users, in turn, inherit the rights to use, modify, and re-share the software under the same GPLv3 terms. This creates a transparent environment where collaboration across organizations becomes possible, while still protecting the openness of the original codebase.

One of the most common **concerns for vendors** working under a GPLv3 license is the fear that their custom development could be easily redistributed, potentially diminishing their commercial advantage. For example, if John Doe, a researcher working in academia, requests access to the software, the vendor is legally obligated to provide him with the source code—because the GPLv3 license guarantees this right to all users who receive the software.

However, what John receives is more limited than it might seem. He obtains only the source code for the GPL-covered components. He does not automatically receive:

- Documentation and user guides
- Detailed change logs or development notes
- Database content or production data
- Configuration files tailored to specific environments
- Plugins or extensions that are separately licensed or developed under different terms

This distinction is crucial. While GPLv3 enforces openness for the licensed software itself, it does not extend to the surrounding ecosystem of materials, data, or proprietary add-ons. In most real-world cases, the GPL-covered code represents only a small portion of a complete solution. The broader system comprising integrations, configurations, and domain-specific content remains under the vendor's control.

Thus, while GPLv3 ensures user rights to transparency and reuse, it does not undermine a vendor's business model. The core value often lies not in the code alone, but in the know-how, maintenance, and accompanying services that make the solution truly functional.

EXAMPLE – MIXING LICENSES (COPYLEFT AND PERMISSIVE)

License compatibility becomes particularly relevant when software components with different open-source licenses are combined. A good illustration of this interaction can be seen with R and the admiral package. The R programming language itself is licensed under GPL, while admiral - a framework for ADaM dataset creation - is released under the Apache 2.0 license, which is permissive.

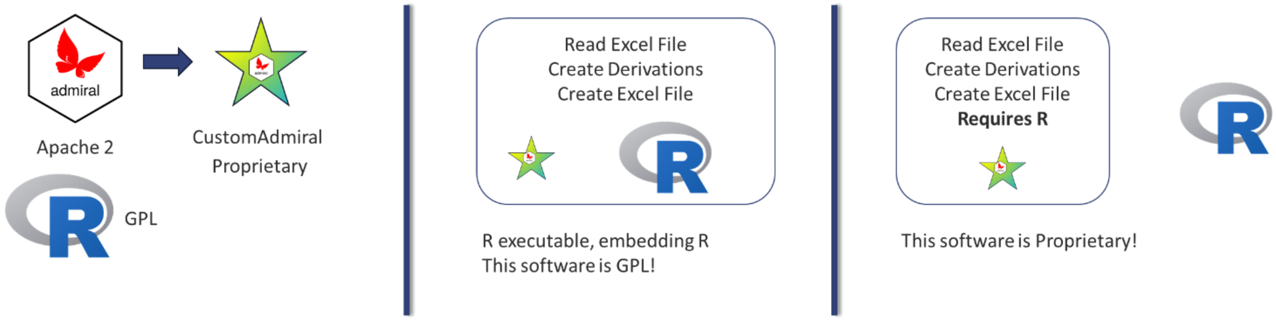


Figure 7: Illustrations of combination options for GPL and Apache 2.0

Imagine a company develops a custom version called CustomAdmiral, which reads Excel inputs, creates derivations, and writes results back to Excel. If the company decides to distribute CustomAdmiral as a single executable that embeds the R engine directly, the result becomes a combined work under GPL. In this case, because R's copyleft license applies to everything that is linked and distributed together, the entire executable must be released under GPL.

However, if CustomAdmiral is provided only as an R package that runs within an existing R installation, it can remain proprietary. Here, R is merely an external dependency rather than part of the distributed software, and the Apache 2.0 license allows this flexibility.

This example highlights a key compliance principle: when open-source components are linked or embedded, the strongest license (in this case, GPL) governs the entire combined product. When they are separated by usage or installation, each retains its own licensing terms allowing a permissive and proprietary coexistence.

LICENSE CHANGES

Another important aspect of open-source licensing is the possibility of license changes - especially for projects with multiple contributors and public repositories. A recent example is CDISC's CORE (CDISC Open Rules Engine), which is currently released under the MIT license and hosted on a public GitHub repository with 20 contributors. CDISC has announced upcoming changes for CORE, which might also include a change of license.

When developers contribute to such a repository, they implicitly agree to the project's existing license - in this case, MIT. This means that each contribution is also licensed under MIT, and the project as a whole remains permissively licensed. As a result, any company or individual can legally take the public CORE source code and release a modified version under a new license, even a proprietary one.

While this flexibility is a hallmark of permissive licensing, it can lead to fragmentation. Multiple parallel versions of CORE could emerge - some open, some closed - each evolving along different development paths. This risks confusion among users and the community, complicates maintenance, and weakens collaboration, as no single canonical version would represent the "official" project anymore.

Hypothetically, if CORE had been released under GPLv3 instead of MIT, the situation would be different. In that case, all contributors (unless a Contributor License Agreement (CLA) had been in place) would automatically contribute their changes under GPLv3 as well. Consequently, any redistributed version would have to remain open-source, preventing full proprietary forks but also limiting flexibility for commercial adaptations.

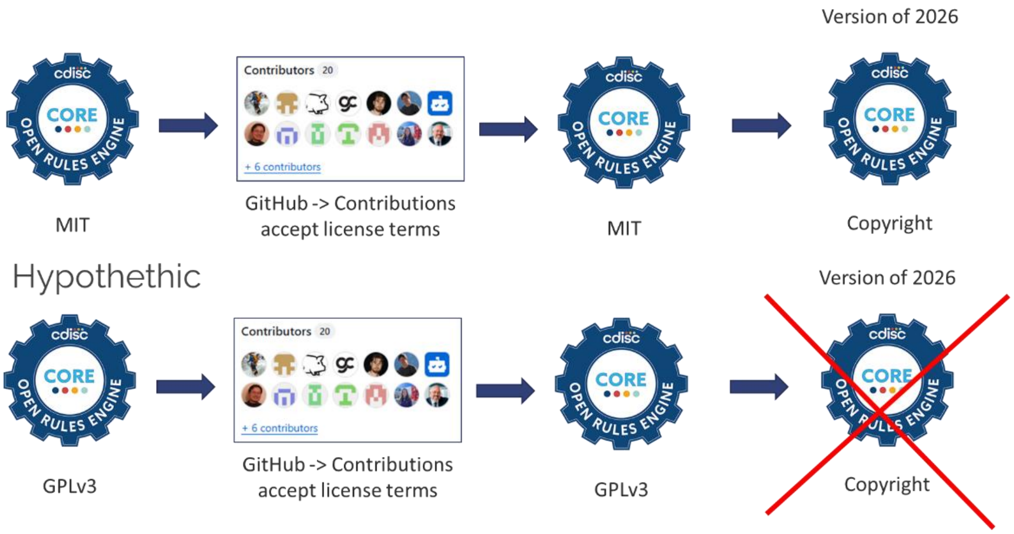


Figure 8: Example License Changes after Contributions

This example highlights the delicate balance between freedom and governance in open source: permissive licenses like MIT encourage adoption and innovation, while copyleft licenses like GPLv3 protect long-term openness and preserve community integrity.

CONCLUSION

Open source has become a cornerstone of digital transformation in the pharmaceutical and life sciences industry. It enables collaboration, accelerates innovation, and supports transparency across organizations and disciplines. Yet, much uncertainty remains, especially around what is legally allowed and how to act responsibly when using or sharing open-source software.

By default, copyright applies to everything we create, including code, documentation, and data. Without an explicit license, reuse is not permitted. The safest approach when encountering unlicensed material is to learn from the idea and re-program the code independently. Always check the license or terms of use before reusing anything. Also consider when sharing your own work, apply a clear license so others know what they can and cannot do.

Understanding license types is key. Permissive licenses like MIT or Apache 2.0 encourage broad reuse and flexibility. Copyleft licenses like GPLv3 maintain openness and transparency across generations of reuse, ensuring that improvements stay within the community. Developers should not be afraid of copyleft - the real value lies typically not in the code itself, but in the data, validation logic, and open collaboration.

To summarize a few guiding principles:

- Default is copyright - no license means no reuse.
- Safest route - use the idea, not the code. Reprogramming is safest.
- Check licenses and terms of use - before using or distributing materials.
- Apply a license to your own work - to enable reuse.
- Don’t fear copyleft - it protects openness, not competition.

Remember: the code is rarely the diamond; the real value is in the data, collaboration, and shared knowledge it enables.

Legal compliance is not a barrier - it is the foundation for sustainable, trustworthy, and community-driven innovation. By understanding and respecting these principles, the industry can continue to harness the full potential of open source while ensuring fairness, clarity, and long-term progress.

REFERENCES

[1] How Big a Problem Is Copyright Infringement? https://nanpa.org/2019/04/24/how-big-a-problem-is-copyright-infringement/ - accessed on 05.10.2025

[2] Image - “Creative commons license spectrum” by Shaddim; original CC license symbols by Creative Commons - https://creativecommons.org/about/downloads/https://creativecommons.org/policies/Original CC license icons licensed under CC BY 4.0, CC BY 4.0, https://commons.wikimedia.org/w/index.php?curid=47247325

[3] WhiteSource – Statistic for open-source-license distribution - https://www.mend.io/blog/open-source-licenses-trends-and-predictions/ - accessed on 05.10.2025

[4] Terms of Use – SAS homepage stating the legal terms - https://www.sas.com/en_us/legal/copyright.html - accessed on 11.10.2025

[5] Email information from SAS Support – received on 23.05.2024

AI REFERENCES

[1] Paint style images are created with ChatGPT 5 on 05.10.2025 (https://chatgpt.com/)

[2] Paper text review and enhancement suggestions has been done with ChatGPT 5 on 05.10.2025 (https://chatgpt.com/)

ACKNOWLEDGMENTS

I would like to thank all open-source contributors whose dedication, creativity, and collaboration make shared progress possible. Special thanks go to the companies and organizations that actively support and enable open-source development within the pharmaceutical and life sciences industry, fostering transparency and innovation across the field.

Acknowledgment is also due to CDISC and PHUSE for their continued efforts in driving open-source initiatives forward, building bridges between standards, technology, and community, and helping shape a more connected and collaborative future for our industry.

RECOMMENDED READING

Recommended reading lists go after your acknowledgments. This section is not required.

[1] Article on licenses (https://www.glacon.eu/portal/articles/2022-03-01-licenses.html)

[2] Choose a license – good overview (https://choosealicense.com/licenses/)

[3] The legal side of open source (https://opensource.guide/legal/)

[4] CDISC Open Source Alliance (https://cosa.cdisc.org/)

[5] Open Source Technology in Clinical Data Analysis (https://phuse-org.github.io/OSTCDA/)

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Author Name:	Katja Glaß
Company:	Katja Glass Consulting
Email:	katja.glass@glacon.eu
Website:	www.glacon.eu