

Licensing in Academic Research: Why It Matters, Where It Fails, and How to Get It Right

Disclaimer:

This is not legal advice.

It is a collection of good practices around open science and licensing.

If in doubt ask the division “Innovation” (I) or our legal department (R).

**Are you sure, you don't want to patent
or commercialize your results and
knowledge?**

Who decides this?

This depends on:

- You (do you plan to do this)
- Your supervisor
- (possible) regulations in the research project you work on (and get paid by)
- FZJ (as your employer)

I may be interested: What should I do?

On the [intranet-page](#) of the team “Innovation”, you can find more information.

There you can also find [frequently asked questions](#) and a [contact for your specific institute](#).

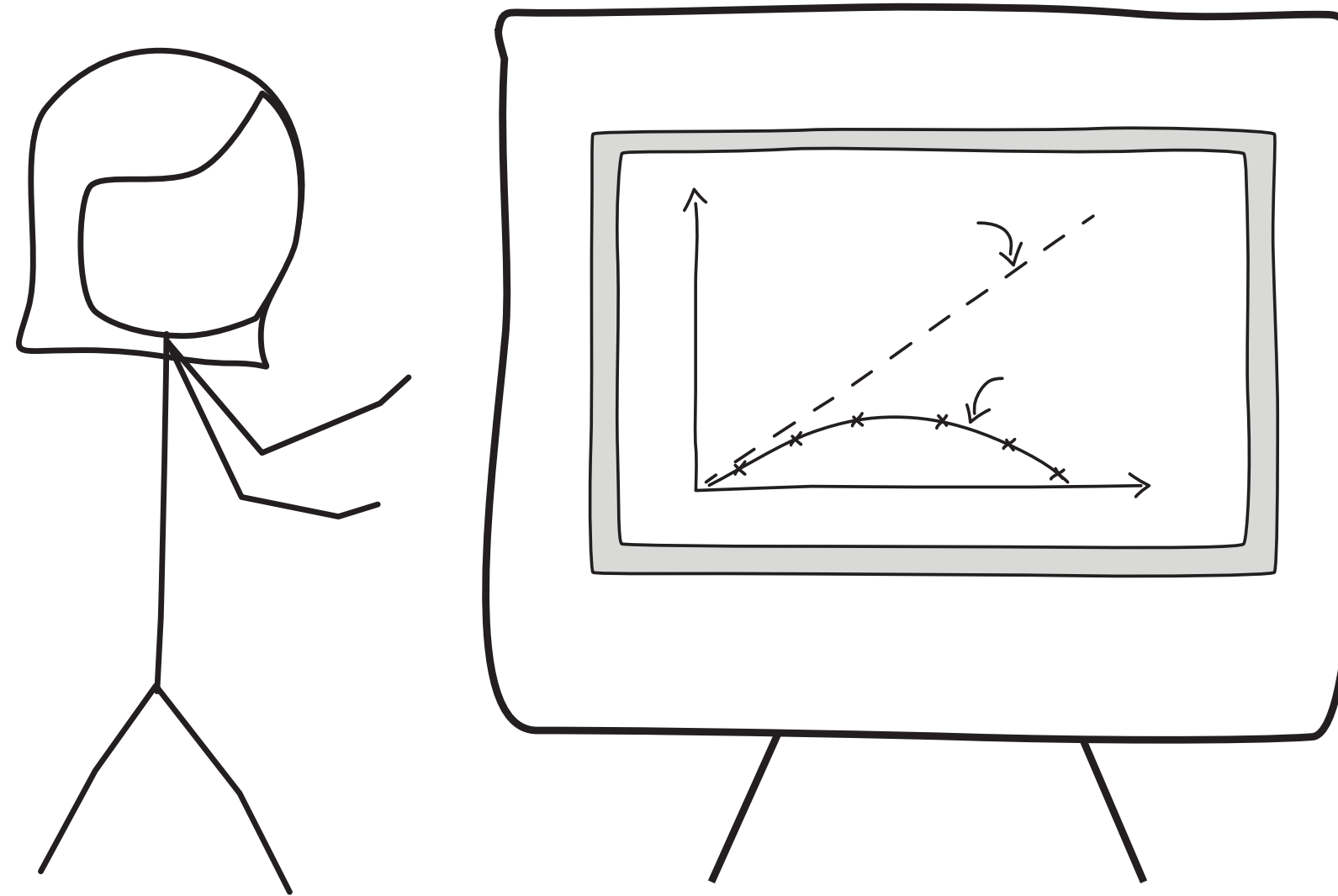
Furthermore, there is a guideline for patenting and licensing (*[Patenting and Licensing Guideline for Forschungszentrum Jülich 2021](#)*) with more information on the process of creating a patent and using a proprietary license.

Get in touch with them, they can help you.

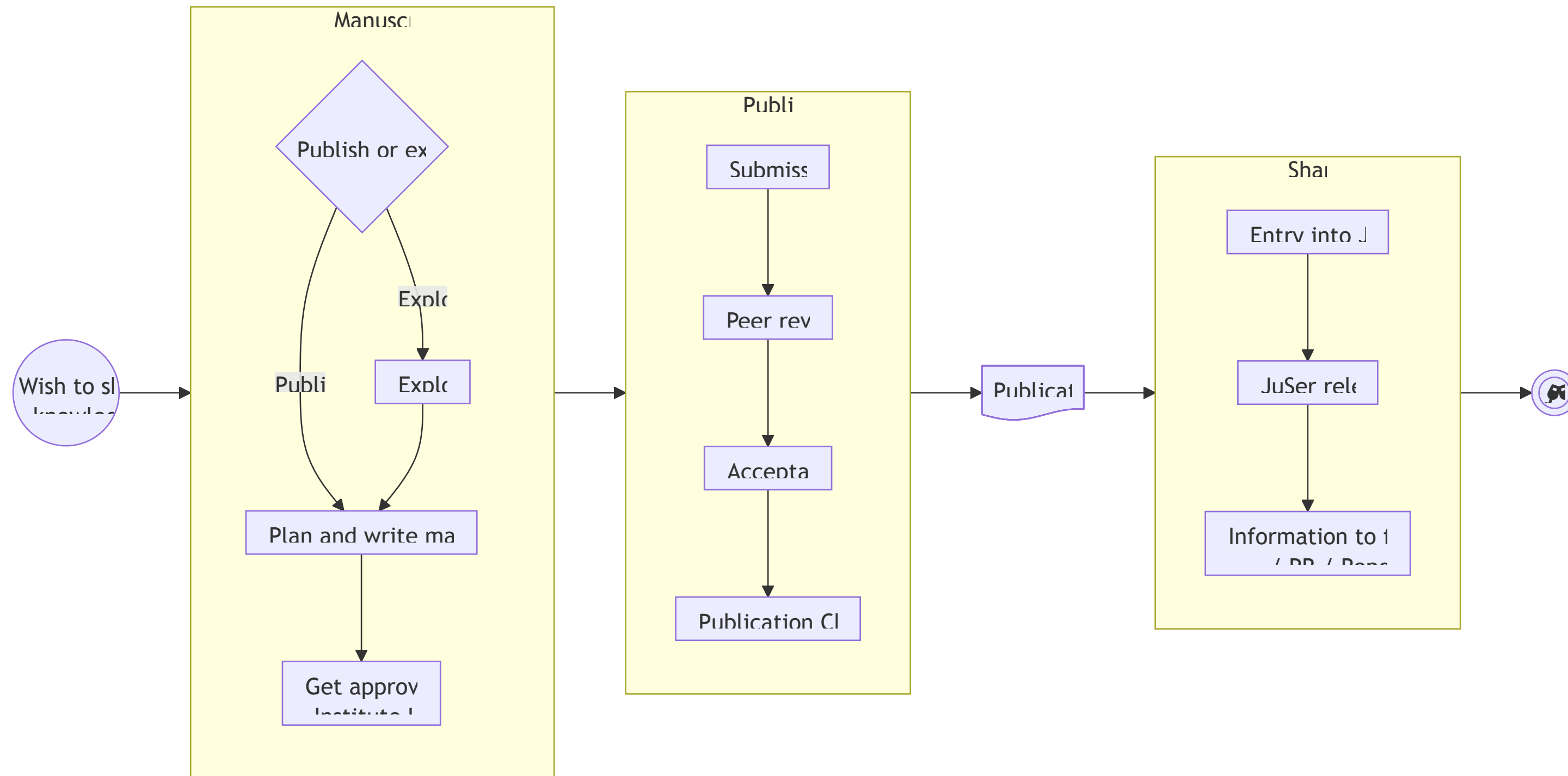
Ok, sounds good. But do I need to found a company for this?

No, you don't. There are many different ways to commercialize your research results.

Wow, I got nice graphs to show my results. Now I want to publish my results!



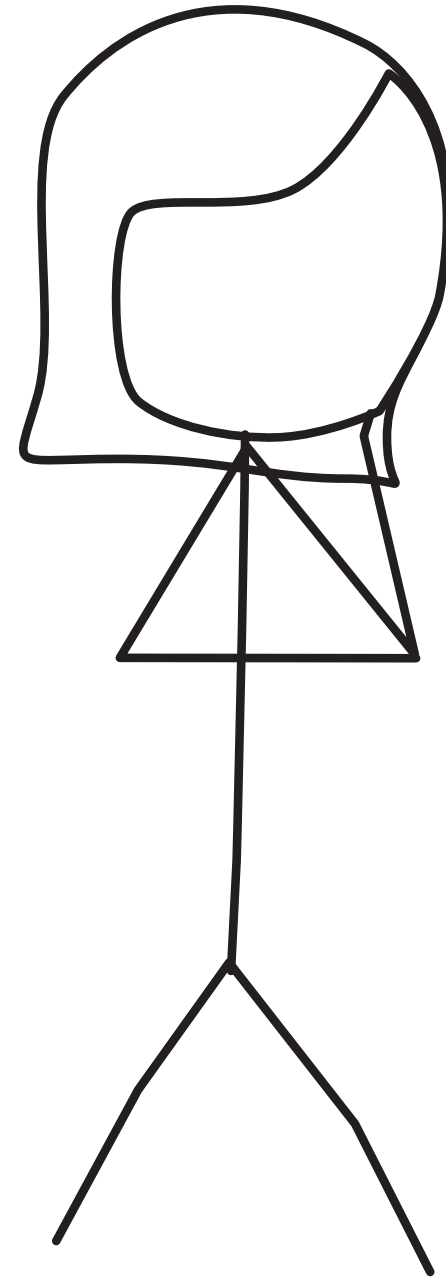
Workflow for text publications



This is not fine-grained enough

More details can be found on the [website](#) of the central library

Let's put a step back and ask: What is a license acutally?



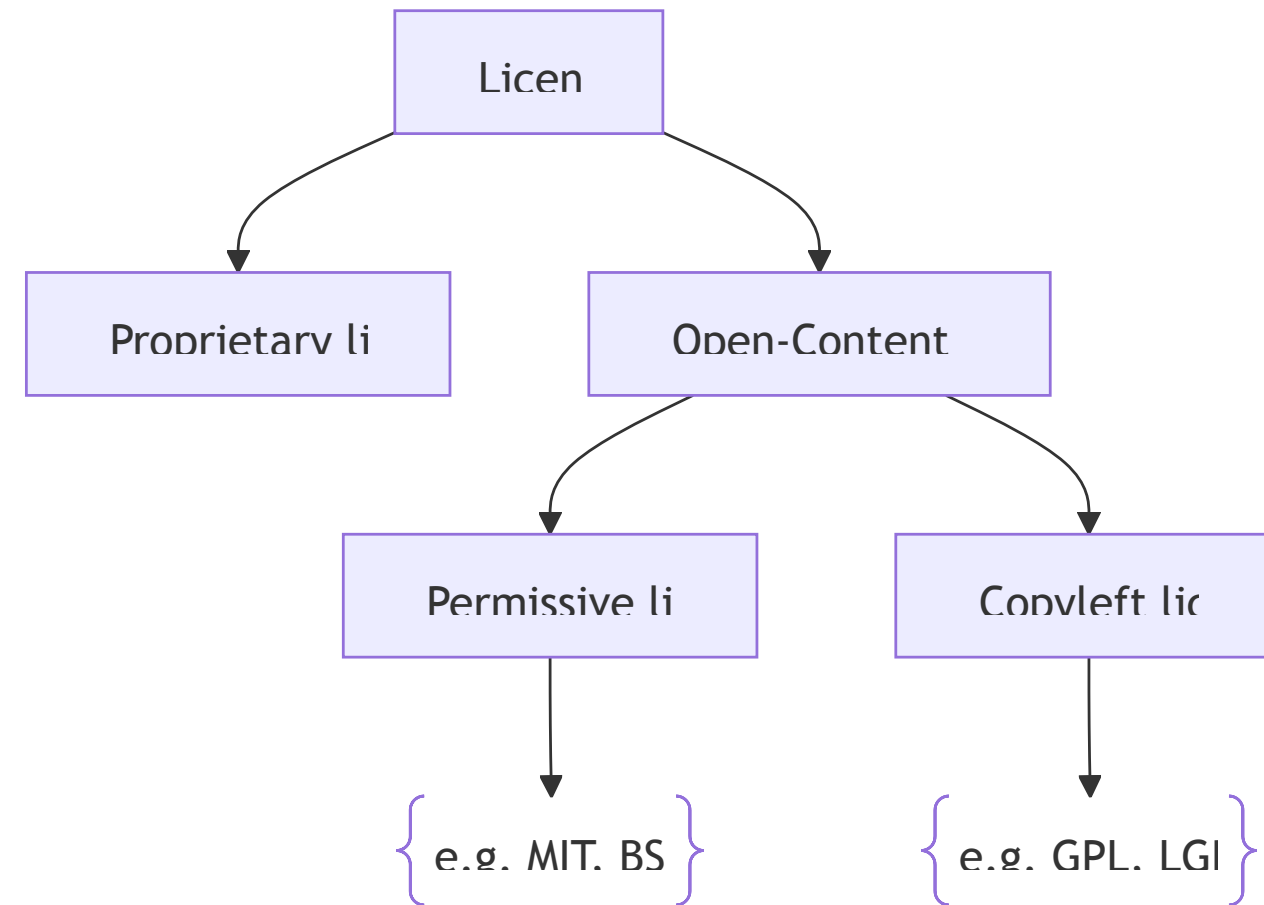
What can (and should) have a license?

Everything, that you want to share with others (external of your own working group) should have a license.

The license clearly states, what is not allowed to be done with the artifact and what is allowed to be done (e.g. distribution, monetarization, adaption,) under which conditions (e.g. naming you as creator).

This includes graphics/figures, data, code, documents, presentations, ...

Types of license



Commonly used licenses for ...

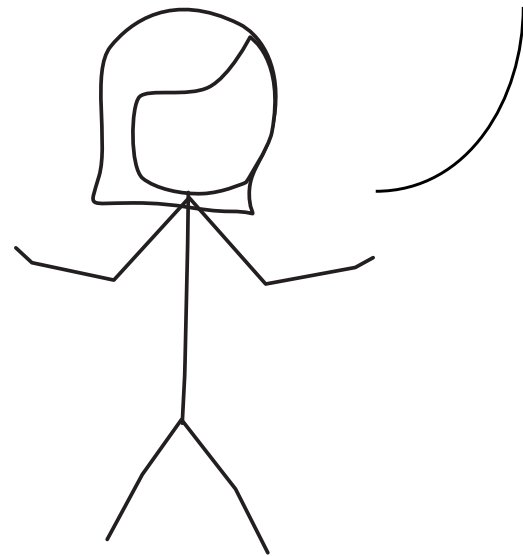
- software
- data
- text/documents
- images
- databases

⇒ We'll talk about those in the later parts of this presentation

Why should I have a license at all?

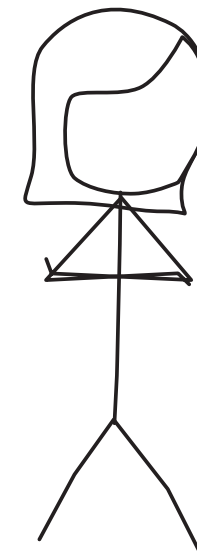
Assumption

Anyone can use, what I provide



Reality

Noone has a permission to use it!



What is legally copyrightable?

Good question.

Free Software Foundation Europe wrote a good blog post with an overview (“[Copyrightability of LLM-Generated Code](#)” 2026) (in the context of vide-coded software) of what is needed for something to be copyrightable.

What is legally copyrightable?

Main points of this blog-post (own summary):

- Copyright/Authorship can be attributed only to a human
- If a piece of software is not copyrightable, it is public domain
- For copyright, a certain creativity (“Schöpfungshöhe”) is needed
- Only implementation (not ideas) can be copyrighted

When in doubt

Ask Legal (R)!

How to publish data

Why should I publish data?

- Maybe you are required to publish your data:
 - by the journal, where you publish a paper
 - by the founding body of your project
 - your boss asks you to
- It is good scientific practice
 - It also enables researchers to reanalyze data for new findings

What data should I publish?

- Best: As much is needed to repeat your research
- At least: What is required by the journal, that you want to publish your paper in
- In many journals, there is a section about “Data availability” towards the end, that you need to answer on if/how your data is available to others. This includes the license under which your data is available.

Where do I publish data?

Different data repositories

- Repository of your discipline (see sections in [NFDI](#) and/or [re3data](#))
- Repository of your institute ([JülichData](#))
- [Zenodo](#)
- [OpenScienceFramework](#) of the [center for open science](#)

Github/Gitlab and others are NO data repositories!

How do I publish data?

Different per platform, but some things are common:

- You can decide on a license
- Often you can set an embargo-period, only after which the data is available publicly
- You can (and should) add Metadata to give context to your data and make your data more FAIR
- You get an identifier

What are commonly used licenses for research data?

For (meta)data, [Creative Commons](#) licenses can be used. They are a widely used and known set of licenses that work in many different legislations.

- For metadata you should use [CC-0](#)
- For data, CC-0 or [CC-BY 4.0](#) is recommended.

[OpenAire](#) recommends CC-0 for databases and datasets.

I cannot or do not want to publish my data. What shall I do?

- You should at least publish your metadata on [ülichData](#).
- Depending on the type of data you might need additional permission or anonymization to publish your data.

I want to know more

Here are some links for further reading:

- RDM portal on [Data Publication](#)
- On [licenses for research data](#)
- On [requirements when publishing your data](#) (in German)

How to publish (Research) software

What is research software?

Software that is used to generate, process or analyse results that you intend to appear in a publication (either in a journal, conference paper, monograph, book or thesis). Research software can be anything from a few lines of code written by yourself, to a professionally developed software package. (Hettrick et al. 2014)

How about vibe-coding or AI-assisted coding?

Still not 100% clear. No generally applicable law and/or precedent cases.

In August 2026, the Free Software Foundation Europe wrote a good blog post with an overview (“[Copyrightability of LLM-Generated Code](#)” 2026).

From that blog post: Three legal principles relating to copyright, useful for an analysis of the use of LLMs:

From FSFE Blog post (shortened):

1. The use of a machine to create a work does not negate copyrightability for the human creator
2. When a person hires someone to execute their established expression, the executing party has no claim to authorship if they use a process that does not require intellectual modification or highly technical enhancement of the expression.
3. Providing detailed instructions for the creation of the work by someone else constitutes only the provision of non-copyrightable ideas, and that person providing instructions has not done enough to warrant authorship

From FSFE Blog post (shortened):

This case is particularly relevant for copyright over software, as software elements produced as output are often used as building blocks in the assembly of a larger work[...] Developers often reuse elements that they do not own copyright over, and their use of such elements does not mean that they can claim authorship over them. They nevertheless are entitled to general authorship and copyright over the overall program that they have created. [...] The compilation, arrangement, and internal structure of a software program can be considered creative enough if it was implemented by a human developer.

In practical terms, this means that developers should structure their programs independently if they wish to claim authorship over such programs and license them as Free Software. They should not be using LLMs to produce a complete program for them, but rather only use such tools to support them in the implementation of their ideas.

Personal main points

- Prompting is not enough own creativity to claim copyright on output.
- If a large fraction (unclear where the border is) is LLM-generated, the code is likely non-copyrightable (public domain).
- Difference between code-snippets and whole codebase (arrangement of code): Code-snippets may, by themselves, not reach the needed novelty/creativity to claim copyright regardless of author.

Risk of Copyright-Infringement by LLM-Use

- As many LLMs are trained also with copyrighted material, it may happen, that they reproduce copyrighted material
- Saying “I didn’t know that and didn’t intend to infringe copyright” is not enough
- Still ongoing debate on whether generated code would count as “derived work” or as replication

Recommendations by the SFC

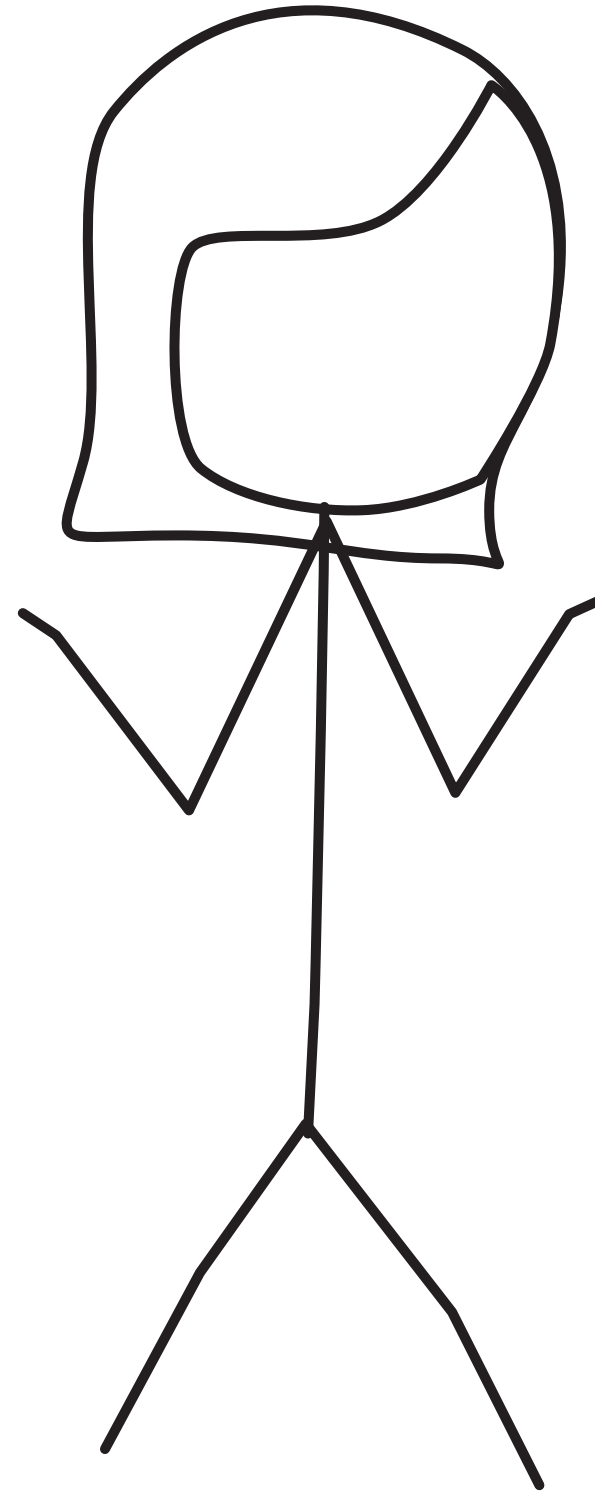
Software Freedom Conservancy gave “Recommendations When Using LLM-Backed Generative AI Systems for FOSS Contributions” (2026).

Subjectively the main ones (shortened) for this context are:

- Keep track of your interaction that lead to a generated piece of code. This is a meta-artifact, that is worth archiving.
- W.r.t. licenses there is still much going on and in the flow. Many unanswered legal questions with experts working on solutions.

- Inputs impact license of the artifacts. If you input code with a copyleft-license, the changes must as well be under that license.
- Copyleft as license to be used for generated code may be safer, as many LLMs are trained on copyleft code. SFC works on a license, that is copyleft and compatible with other copylefts and suitable for LLM-gen-AI outputs.
- Disclose how, when, what (incl. version), and how an LLM-gen-AI was used. Keep this also machine-readable in the commit logs.

Oh I see, this is difficult



Yes, and no

There are practical tips and files, that make it easier to keep track Especially usefull to recall later and when you share it.

Useful files for machine-readable information

- Citation File Format (CFF)
- `LICENSE.txt` (or `LICENSE.md`) for source code (simple version)
- `SPDX-Headers` / `REUSE Specification` (complex version)
- `aidecl.yaml`: File to specify if AI was used, where and which model. More information in ([Diblen 2026](#)).

Ok, I understood, but what about my specific case?

Go, ask “Innovation” (I in UE)

They are happy to help with your specific case. You can contact them at innovation@fz-juelich.de or on their website

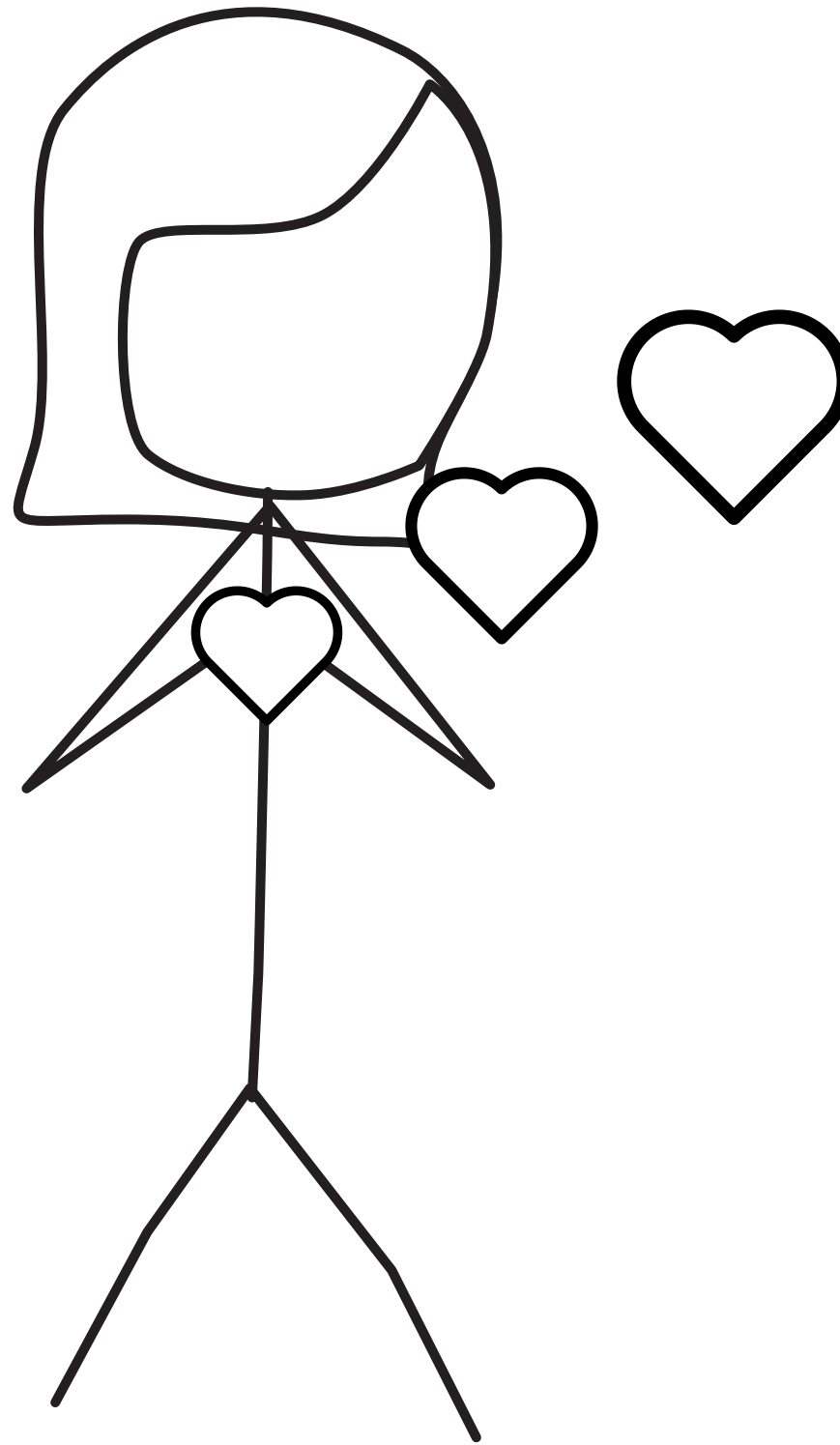
My case is more difficult than this

Go, ask legal (R)!

They are happy to help with more complicated cases.

Contact [Alexander Storm](#) or [Felicitas Jung](#)

Thank you



Acknowledgements

Thank you for support and advice in creating this presentation:

- Thomas Arndt
- Dirk Brömmel
- Dorit Jerger
- Felicitas Jung
- Janne Philine Krahleher
- Anna Lüscher
- Dennis Oliveira
- Sven Rank
- Ute Scheelhaas
- Robert Speck
- Amber Sprenger
- Alexander Storm

Acknowledgements

Thank you to Sonja Habbinga for providing the images of stickfigures and choosing a license, that allows my to adapt them.

This deck of slides is licensed under

CC BY-SA 4.0 for the content of the slides

For more detailed information, see the [public repository](#). It follows the [REUSE-Specification](#) and explicitly states a license per file.

Thank you for your attention

If you have questions, feel
free to get in touch.

References

“Copyrightability of LLM-Generated Code: Can We License ‘Vibe Code’ into Free Software?” 2026. News-Item. Free Software Foundation Europe; FSFE - Free Software Foundation Europe, August 25. <https://fsfe.org/news/2026/news-20260825-01.html>.

Diblen, Faruk. 2026. “Provenance in the Age of AI: Why We Need AI Declarations for More Transparency.” eScience Center Blog, April 23. <https://blog.esciencecenter.nl/posts/2026-04-23---provenance-in-the-age-of-ai-why-we-need-ai-declarations-for-/>.

Hettrick, Simon, Mario Antonioletti, Les Carr, et al. 2014. “Uk Research Software Survey 2014.” Zenodo, December 4. <https://doi.org/10.5281/ZENODO.14809>.

Patenting and Licensing Guideline for Forschungszentrum Jülich. 2021. Forschungszentrum Jülich GmbH. https://intranet.fz-juelich.de/en/organization/ue/departments/innovation-strategy/downloads/2021_06_patenting_licensing_guideline_fzj.pdf/@@downl

“Recommendations When Using LLM-Backed Generative AI Systems for FOSS Contributions.” 2026. Software Freedom Conservancy, June. <https://sfconservancy.org/llm-gen-ai/llm-backed-generative-ai->