

# Practical software licensing

Radovan Bast

UiT The Arctic University of Norway



Text: CC-BY 4.0

Slides: <https://doi.org/10.5281/zenodo.11554000>

## About me



- Theoretical chemist turned research software engineer.
- I write research software and teach programming to researchers and lead the [CodeRefinery\\_project](#).
- I lead the [high-performance computing group](#) and the [research software engineering.group](#) at UiT.
- Drafted a [Research software licensing guide](#)

*I am not a lawyer. This is not a legal advice ...*

# Copyright and derivative work: Sampling/remixing

- Copyright controls whether and how we can distribute the original work or the derivative work
- In the **context of software** it is more about being able to change and distribute changes
- Changing and distributing software is similar to changing and distributing music
- You can do almost anything if you don't distribute it

## Often we don't have the choice:

- We are expected to publish software
- Sharing can be good insurance against being locked out



Can we distribute our changes with the research community or our future selves?



[Midjourney, CC-BY-NC 4.0]

# Why software licenses matter

You find some great code that you want to reuse for your own publication.

- This is good for the original author - you will cite them. Maybe other people who cite you will cite them.
- You modify and remix the code.
- Two years later ... 
- Time to publish: You realize **there is no license to the original work** 

# Now we have a problem

 "Best" case: You manage to publish the paper without the software/data

Others cannot build on your software and data.

 Worst case: You cannot publish it at all

Journal requires that papers should come with data and software so that they are reproducible.

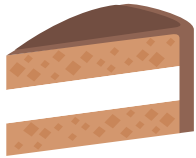
# Derivative work?

- Download some code from a website and add on to it:  yes  no
- Download some code and use one of the functions in your code:  yes  no
- Rewriting code you got from somewhere to a different language:  yes  no
- Linking to libraries (static or dynamic), plug-ins, and drivers:  yes  no
- You read a paper, understand algorithm, write own code:  yes  no
- Clean room design (somebody explains you the code but you have never seen it):  yes  no



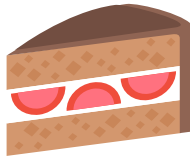
[Midjourney, CC-BY-NC 4.0]

# Software license types (metaphor: cake)



## Your recipe (code)

- Imagine you (original author) compose a recipe for a really tasty cake.
- In regular intervals you distribute cakes (release binaries).
- You share it on GitHub under the OpenCake organization.



## Famous restaurant ...

- Finds your cake recipe on GitHub
- The chef suggests improvements (derivative work).
- They wish to put it on their menu (distribution).

[Images: [EmojiOne](#), CC-BY-SA 4.0]

# Possible outcomes 1/4: custom/proprietary

## No license or custom license

- No restaurant chef will touch it: too much hassle to employ a lawyer to be sure that the cake can be served to customers.
- But maybe they will bake it and eat it and not distribute it and that is OK ("fair use" provision permits the making of copies for own use).



# Possible outcomes 2/4: permissive

## License: MIT or Apache or BSD-2

- It is OK to use the recipe and sell the cake.
- It is OK to not share the improved recipe.
- They have to indicate where the original recipe came from (preserve copyright notice) and possibly summarize their modifications.
- If somebody becomes sick, it is not the fault of the OpenCake organization (limit of liability).
- You may not get the improvements back to use yourself.

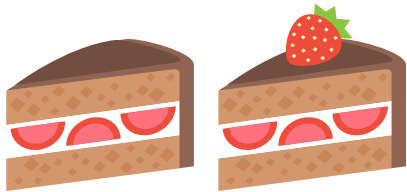
## License: BSD-3

- In addition to the above it is understood that the updated recipe are not endorsed by the OpenCake organization.

# Possible outcomes 3/4: share-alike, weak copyleft

## License: GNU Lesser GPL (LGPL)

- The restaurant has to share only the improved cake recipe but can keep other recipes closed.
- You can use their improved recipe and improve it further:



- The restaurant guests have to be able to exchange the cake from the menu by improved cakes from other restaurants (dynamic relinking).

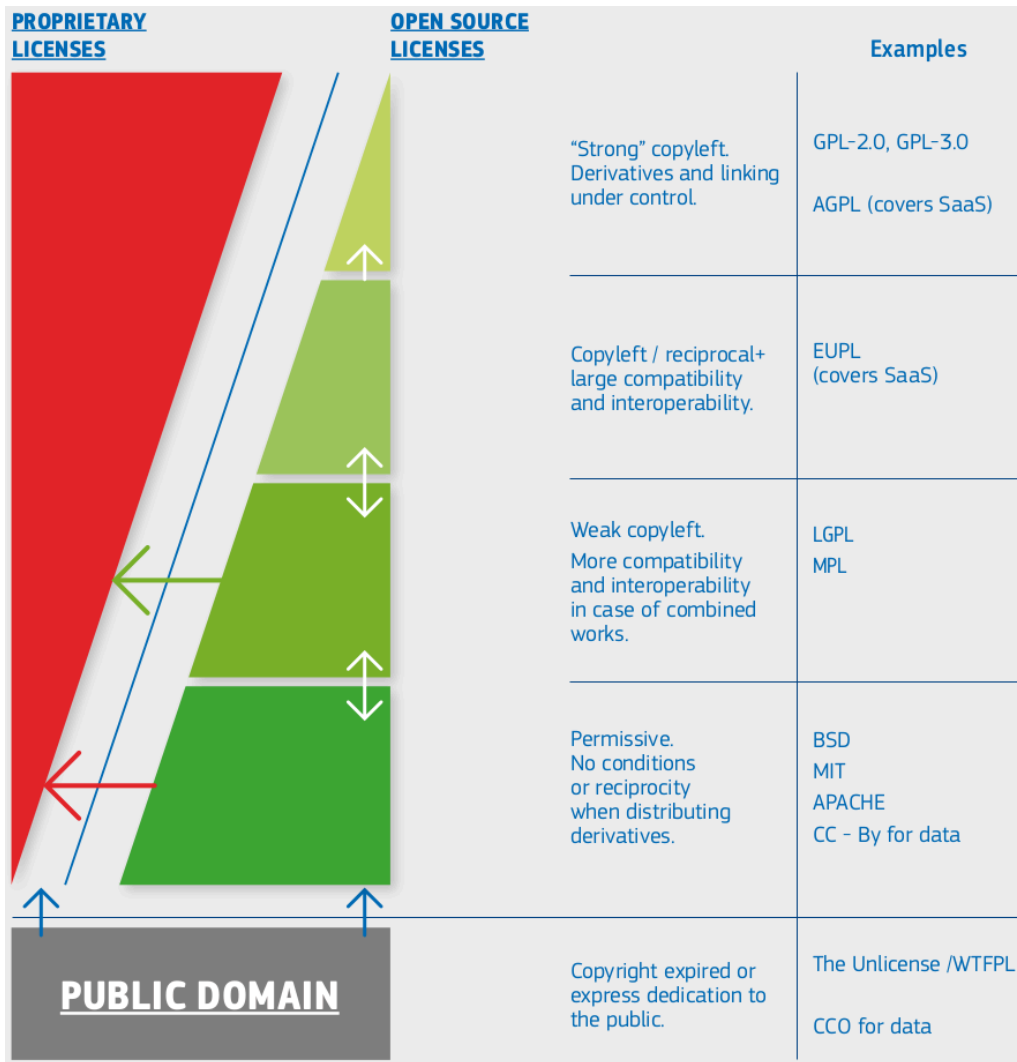
## License: Mozilla Public License v2.0

- Like LGPL but do not require that the modified cake can be exchanged by the restaurant guest.

# Possible outcomes 4/4: strong copyleft

License: GNU GPL or GNU Affero GPL

- If the cake is a part of the menu, the famous restaurant has to share the recipes of the entire menu.



[European Commission, Directorate-General for Informatics, Schmitz, P., European Union Public Licence (EUPL): guidelines July 2021, Publications Office, 2021,

<https://data.europa.eu/doi/10.2799/77160>]

## Beginning of a project



[Midjourney, CC-BY-NC 4.0]

- License does not seem important
- Easy to change (\*)
- **Start with a LICENSE file**

## Later in the project



[C.Stadler/Bwag, CC-BY-SA 4.0]

- Can be important
- Especially when combining codes or organizations
- Difficult to change
- Authors change affiliation

# If your work is not derivative work - you started from scratch

Clarify:

- Does your work contract, grant, or collaboration agreement dictate a specific license?
- Is there an intent to commercialize the code?
- When there is unknown or mixed ownership: If there are multiple persons or organizations as owners of the code, all must agree to the license.

Do not invent your own license. Choose one of the standard licenses, otherwise compatibility is not clear:

- [Joinup Licensing Assistant - Find and compare software licenses](#)
- [Joinup Licensing Assistant - Compatibility Checker](#)

If no overriding guidelines exist, we recommend:

- **European Union Public Licence, Version 1.2 or later (EUPL)**
- **MIT License**

# If your work is derivative work

**Check the license of the original code.** Depending on the license, your choices might be limited:

- [Joinup Licensing Assistant - Find and compare software licenses](#)
- [Joinup Licensing Assistant - Compatibility Checker](#)

If the original code does not have a license, you may not be able to distribute your derivative code.

- Try to contact the authors and ask them to clarify the license of their code.

# Receiving contributions

- Keep track of where things come from. Clarify their license.
- **Work as if the project is public even though it is still private:** This is to avoid surprises about code in the history with incompatible license years later when we decide to open the project.



Slides: <https://doi.org/10.5281/zenodo.11554000>

- [More about licenses](#)
- [More about software citation](#)
- [Research software licensing guide](#)



<https://coderefinery.org>



<https://research-software.uit.no>