



Licenses and Copyright

Free and Open Source Software (FOSS) Licenses

19/20 November 2019 | Guido Trens (JSC, SimLab Neuroscience)

Motivation

Terminology

Licenses

How to chose a License

Compatibility

Copyright and License Notice

Conclusion

Motivation

Terminology

Licenses

How to chose a License

Compatibility

Copyright and License Notice

Conclusion

License and copyright issues are a complicated matter and usually only an afterthought for software developers.

Motivation

- An important part of the scientific process is to make your source code and software available to others.
- Unlicensed code is closed code.
- Without a license agreement, software may be left in a state of legal uncertainty.
- Potential users may not know which limitations owners may want to enforce.
- Owners may leave themselves vulnerable to legal claims or have difficulty controlling how their work is used.

Any license is better than none!

Motivation

Terminology

Licenses

How to chose a License

Compatibility

Copyright and License Notice

Conclusion

Free and Open Source Software (FOSS) licenses terminology:

- *Copyright*
- *Copyleft*
- *Copyleft-type license*
- *Strong copyleft, weak and none copyleft*
- *Permissive, BSD-style license*
- *Viral license*

Copyright

- Protects the “*fixed form*“ of an idea, not the idea itself.
- Gives the author(s) exclusive economic and moral rights over the copyrighted material, i.e.:
 - Making copies
 - Issuing copies to the public
 - Adapting the work
- Copyrights are considered territorial rights and vary by country.

[Rowan Wilson (OSS Watch) (2009), “Copyright in Software and Open Source Licensing”
<http://de.slideshare.net/crmwilson/copyright-in-software-and-open-source-licensing>]

Copyleft

- Copyleft is a concept of using of the copyright and plays on the word “*copyright*”.
- It **doesn't** mean abandoning the copyright!
- To “*copyleft*” a program it is first stated that it is copyrighted.
- Distribution terms are added, that gives everyone the rights to use, modify, and redistribute the program's code, or any program derived from it, but only if the distribution terms are unchanged.

[GNU Operating System, “What is copyleft?”
<https://www.gnu.org/licenses/copyleft.en.html>]

Copyleft-type Licenses

- Copyleft-type licenses are a novel use of existing copyright law to ensure a work remains freely available.
- The ‘GNU General Public License’ (GPL), originally written by Richard Stallman, was the first software copyleft license to see extensive use, and continues to dominate in that area.

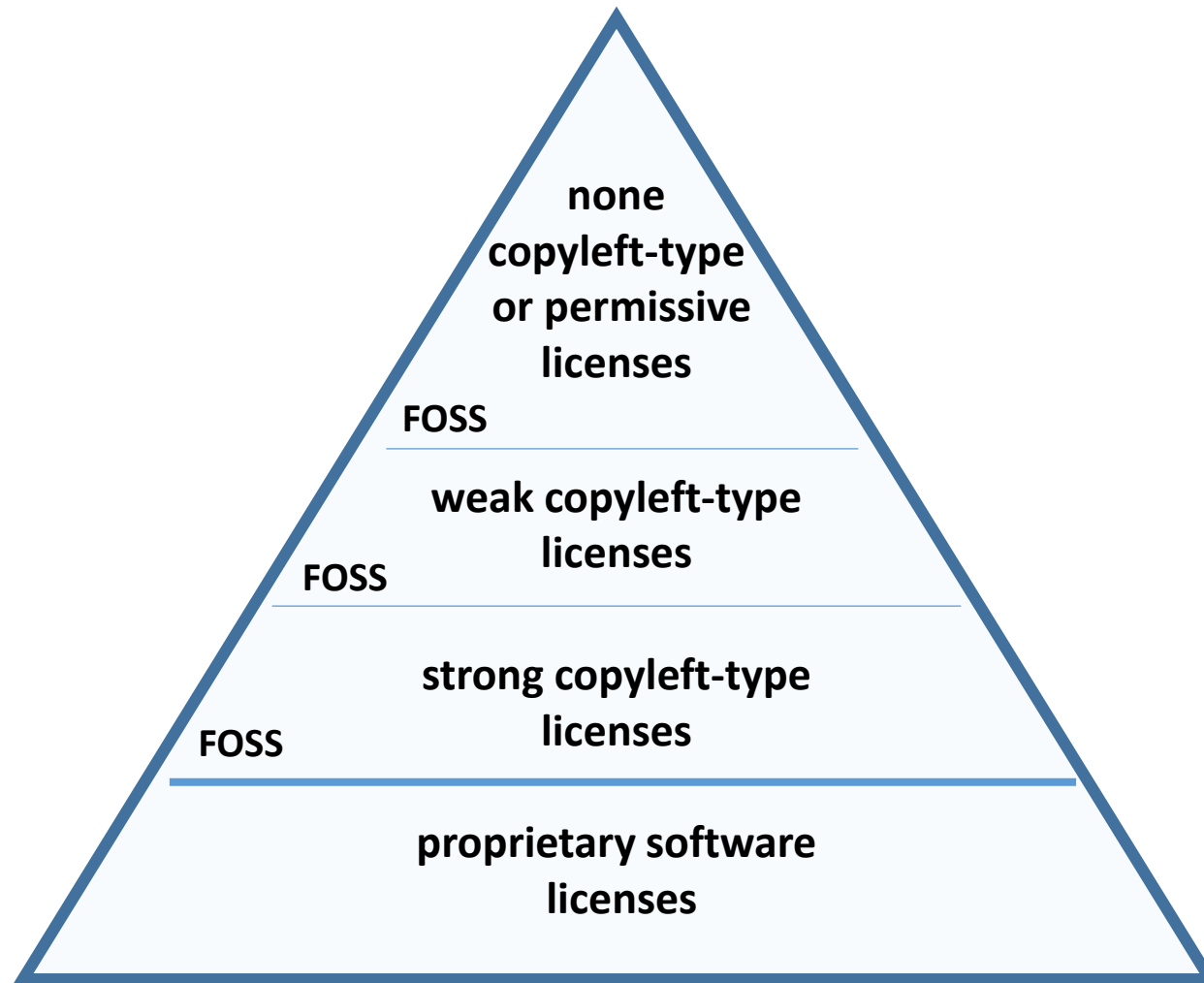
[Wikipedia, “Copyleft”
<https://en.wikipedia.org/wiki/Copyleft>]

Terminology

Strong, Weak and None Copyleft-type Licenses

- Copyleft advocates distinguish between:
 - *Strong copyleft & strong copyleft-type licenses*
 - *Weak copyleft & weak copyleft-type licenses*
 - *None copyleft & none copyleft-type or permissive licenses*
- This **represents a spectrum** where the copyleft gets weaker making trade-offs that might impede software freedom.

[Software Freedom Law Center, "Copyleft and the GNU General Public License: A Comprehensive Tutorial and Guide"
<https://copyleft.org/guide/comprehensive-gpl-guide.pdf>]



Motivation

Terminology

Licenses

How to chose a License

Compatibility

Copyright and License Notice

Conclusion

Strong Copyleft-type Licenses

- Full copyleft
 - Strong copyleft licenses exercise their scope fully.
 - Anything which is “*work*” or a “*work based on a work*” licensed under a strong copyleft is subject to its requirements.
- They are viral.

Also known as the “*copyleft effect*”.

[Software Freedom Law Center, “Copyleft and the GNU General Public License: A Comprehensive Tutorial and Guide”
<https://copyleft.org/guide/comprehensive-gpl-guide.pdf>]

Licenses

Strong Copyleft-type Licenses

- The most popular strong copyleft-type licenses are:
 - *GNU General Public License Version 2 (GPLv2)*
 - *GNU General Public License Version 3 (GPLv3)*
 - *Eclipse Public License Version 1.0 (EPL)*

Weak Copyleft-type Licenses

- In particular situations full copyleft may not best serve.

An example is free software which should be combined with code under a non-free license. The Lesser General Public License would allow this and is sometimes used in that fashion.

[Software Freedom Law Center, “Copyleft and the GNU General Public License: A Comprehensive Tutorial and Guide”
<https://copyleft.org/guide/comprehensive-gpl-guide.pdf>]

Licenses

Weak Copyleft-type Licenses

- The most popular weak copyleft-type licenses are:
 - *GNU Lesser General Public License Version 2 (LGPLv2)*
 - *GNU Lesser General Public License Version 3 (LGPLv3)*
 - *Mozilla Public License 2.0 (MPLv2)*

None Copyleft-type Licenses

- They are **permissive** licenses and do not have a copyleft.
 - *Most of them allow sub-licensing.*
 - *The creation and distribution of derivative works without restrictions.*
 - *They are very convenient for developers and cause no license compatibility issues.*
 - *Disadvantages are that derivative works **can become proprietary**.*

[FOSS Lawyers, “non copyleft licenses”
<http://fosslawyers.org/permissive-foss-licenses-bsd-apache-mit/>]

None Copyleft-type Licenses

- The most popular none copyleft-type licenses are:
 - *Apache License 2.0*
 - *MIT License*
 - *BSD 2-Clause and BSD 3-Clause License*
 - *Python License Version 2.0 (Python 2.0)*

Motivation

Terminology

Licenses

How to chose a License

Compatibility

Copyright and License Notice

Conclusion

How to chose a License

Contributing to an existing project

- Release your modified versions under the same license as the original work.
- Using a different license for your modifications often makes that cooperation difficult.
- You should only chose a different license when there is a strong reason to justify it.

One case where using a different license can be justified is when you make major changes to a work under a non-copyleft, permissive license. If the version you've created is considerably more useful than the original, then it's worth copylefting your work.

[GNU, "How to choose a license for your own work"
<https://www.gnu.org/licenses/license-recommendations.html>]

How to chose a License

Contributing to an existing project

- If you have chosen a different license:
 - *Make sure that the original license allows use of the material under your chosen license.*
 - *For honesty's sake, show explicitly which parts of the work are under which license.*

[GNU, "How to choose a license for your own work"
<https://www.gnu.org/licenses/license-recommendations.html>]

How to chose a License

Setting up a new project

- Depending on the software's purpose, different licenses for different projects are recommended.

The following often is subject of controversial debate:

- In general, the strongest copyleft license that doesn't interfere with that purpose should be used.
- For most programs, the most recent version of the GNU General Public License (GPL) is appropriate for your project.
- The strong copyleft of GPL is appropriate for all kinds of software, and includes numerous protections for users' freedom.

[GNU, "How to choose a license for your own work"
<https://www.gnu.org/licenses/license-recommendations.html>]

How to chose a License

Small Programs

- It is not worth the trouble to use copyleft for most small programs.

300 lines is a good benchmark: when a software package's source code is shorter than that, the benefits provided by copyleft are usually too small to justify the inconvenience of making sure a copy of the license always accompanies the software.

- *For small programs the non-copyleft, permissive Apache License 2.0 is a good choice.*

It has terms to prevent contributors and distributors from suing for patent infringement. This doesn't make the software immune to threats from patents (a software license can't do that), but it does prevent patent holders from setting up a “bait and switch” where they release the software under free terms then require recipients to agree to non-free terms in a patent license.

[GNU, “How to choose a license for your own work”
<https://www.gnu.org/licenses/license-recommendations.html>]

Motivation

Terminology

Licenses

How to chose a License

Compatibility

Copyright and License Notice

Conclusion

Compatibility

	AGPLv3	GPLv2	GPLv3	EPL	QPL	LGPLv2.1	LGPLv3	CDDL	MPLv2	Apache 2.0	BSD 2/3 clause	MIT	Zlib	ZPLv2	Python 2.0
AGPLv3		-	+	-	-	+	+	-	+	+	+	+	+	+	+
GPLv2	-		-	-	-	+	-	-	+	-	+	+	+	+	+
GPLv3	+	-		-	-	+	+	-	+	+	+	+	+	+	+
EPL	-	-	-		-	-	-	-	-	X	+	+	+	+	+
QPL	-	-	-	-		-	-	-	-	+	+	+	+	+	+
LGPLv2.1	+	+	+	-	-		+	-	+	-	+	+	+	+	+
LGPLv3	+	-	+	-	-	+		-	+	+	+	+	+	+	+
CDDL	-	-	-	-	-	-	-		+	+	+	+	+	+	+
MPLv2	+	+	+	+	-	+	+	-		+	+	+	+	+	+
Apache 2.0	+	-	+	-	+	-	+	+	+		+	+	+	+	+
BSD 2/3 clause	+	+	+	+	+	+	+	-	+	+		+	+	+	+
MIT	+	+	+	+	+	+	+	+	+	+	+		+	+	+
Zlib	+	+	+	+	+	+	+	+	+	+	+	+		+	+
ZPLv2	+	+	+	+	+	+	+	+	+	+	+	+	+		+
Python 2.0	+	+	+	+	+	+	+	+	+	+	+	+	+	+	

+ compatible, - incompatible, X unclear

[Deutsches Zentrum für Luft- und Raumfahrt e.V., „Nutzung von Open Source Software im DLR“]

Content

Motivation

Terminology

Licenses

How to chose a License

Compatibility

Copyright and License Notice

Conclusion

Copyright and License Notice

- Every nontrivial file needs a copyright notice as well as the license notice.

Any file more than ten lines long is nontrivial for this purpose.

- Copyright notices should look like this:

```
Copyright (C) year_1, year_2, year_3 copyright-holder
```

```
Copyright (C) year_1 - year_n copyright-holder
```

- The word “*Copyright*” must always be in English, by international convention.

[GNU, “Copyright notices”
https://www.gnu.org/prep/maintain/html_node/Copyright-Notices.html]

Copyright and License Notice

- The license notice depends on the license chosen.

Example: GPL license notice:

```
This file is part of Foobar.
```

```
Foobar is free software: you can redistribute it and/or modify it  
under the terms of the GNU General Public License as published by  
the Free Software Foundation, either version 3 of the License, or  
(at your option) any later version.
```

```
Foobar is distributed in the hope that it will be useful, but  
WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
General Public License for more details.
```

```
You should have received a copy of the GNU General Public License  
along with Foobar. If not, see <http://www.gnu.org/licenses/>.
```

[GNU, "How to use GNU licenses for your own software"
<https://www.gnu.org/licenses/gpl-howto.en.html>]

Motivation

Terminology

Licenses

How to chose a License

Compatibility

Copyright and License Notice

Conclusion

Conclusion

- Always license your code: unlicensed code is closed code!
- Use a GPL-compatible license for broad compatibility.
- Depending on the software's purpose different licenses might be appropriate.
- The type of license preferably to be used, permissive, BSD-style license or viral, GPL-style license is controversial discussed in the open source community.
- Every non-trivial file needs a copyright notice.