Lesson 5:
What is this licensing madness and why you might care (a little)

Stephen R. Walli

Objectives

- Understand the (very) simple basics of intellectual property protection
- Be able to explain the basic license groups in the free and open source software space
- Know the basic licenses, and where to find more data
- Have some understanding of current controversies and the implications in an open source community

Notes

https://github.com/jhu-ospo-courses/JHU-EN.601.210/tree/main/lessons/4#notes

IANAL

Everything you need to know about open source licensing in (almost) 2 minutes

- Software is protected by copyright law in the U.S. and many other countries. [There are treaties between countries.]
- **USE A LICENSE.** Anyone that wrote the software, owns the copyright and can say how it is used. [**NOT** saying how it can be used does **NOT** remove that responsibility as the owner, even if you don't care.]
- People sometimes give up their copyright ownership to the software they write in their employment agreements. [You need to learn to read your employment agreement.]
- Different countries have subtly different rules about some things (e.g., public domain, moral rights)

Everything you need to know about open source licensing in (almost) 2 minutes

- If you want to liberally share your software with others so they can use it for anything, there are a few well-accepted SIMPLE licenses (Apache 2.0, BSD, MIT). These essentially say:
  - Do whatever you want with the software.
  - Don't claim my work as your own. (Maintain my copyrights.)
  - THIS SOFTWARE COMES WITH NO WARRANTIES, USE AT YOUR OWN RISK, [ALL IN CAPITAL LETTERS BECAUSE WE REALLY, REALLY MEAN IT].

MIT License

Copyright <YEAR> <COPYRIGHT HOLDER>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

If you care about …

- The ethics of software freedom over liberally licensed, collaboratively developed software, you probably already have opinions that involve the GNU General Public License family
- Making money, you likely need to consult a lawyer (or read Van Lindberg)
- Patents, you really need to consult a lawyer (and read Van Lindberg)

Other Federally Protected Intellectual Property

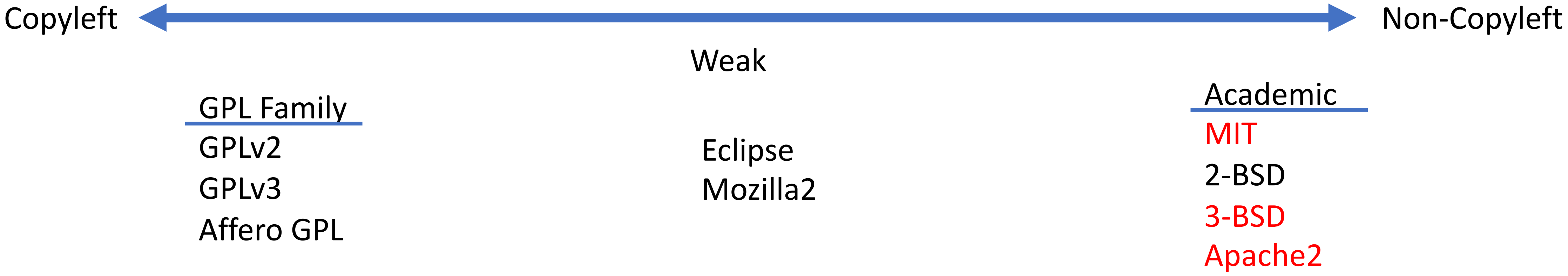
- Patents – regardless of how you feel about software patents, they are the law. They are also expensive to file properly.
- Trademarks – When you give your project a clever name, DO spend a few minutes in a GitHub search, a USPTO search (under trademarks), and a general search engine search for that name. Try not to clash in the same name space. And trademarks are also expensive to file.

A slightly longer set of license ideas

- Do use an OSI-approved license – license invention is expensive
- There are several big levers:
  - How much license reciprocity is required with respect to the software, modifications, and any derivatives someone develops?
  - What is said about patent licensing and litigation?
  - What legal jurisdiction covers the license?
- Other considerations in license choice include:
  - Are there project specific affinities? (e.g., language family)
  - History of the license and non-profit/corporate/commercial involvement? (e.g., Apache)
  - Beware of license proliferation and getting clever with licenses that don’t mix
  - Good discussion of license mixing at the FSF site

Reciprocity (aka Copyleft)

With a strong copyleft license, the license attaches to all derivatives.
With a weak copyleft license, the license attaches to just the project code.
With non-copyleft licenses, there is no attachment.



Copyleft

Non-Copyleft

Weak

GPL Family
GPLv2
GPLv3
Affero GPL

Eclipse
Mozilla2

Academic
MIT
2-BSD
3-BSD
Apache2

Other oddities and interesting ideas

- Dual licensing: the owner of a piece of software can license it as many ways to as many people as they choose – this is an artifact of copyright law and has nothing to do with open source
- Contribution License Agreements & Developer Certificate or Origin
  - There is a difference between licensing your work or assigning your work in a CLA and you should understand the implications
- Be thoughtful – Don't use a ”license picker”