Software Licensing

why it actually does matter

Stephen Worley

Brief Overview of Property Rights

- Copyright
- Patents
- Trademarks
- Licenses

Copyright

- All rights reserved
- For Creative and expressive works
- Automatically given to the creator when they make the work
- Theoretically it expires, but this has not happened since the 70s
    - Laws keep changing to make the coverage period longer

Patents

- Explain how it works and prove that it is different enough to be unique
- Given exclusive rights to use the patent for a limited time
- Yes, you can patent a software technique! Not necessarily a specific code implementation though.

Trademarks

- Protects against fakes/imitators
- Automatic like copyright, but registering a trademark can give it more power in court
- Names, logos, etc.

TM

Licenses

- Give people/companies rights to use, modify, and share your work
  - Without a license, such things are infringement!
- Also, can be used to set conditions/obligations of the licensee:
  - Proprietary code examples:
    - Prevents sharing, copying, or reverse engineering
    - Prevents benchmarking performance and sharing that data (Databases typically)
  - FOSS examples:
    - Attribution
    - Copyleft (explained soon!)

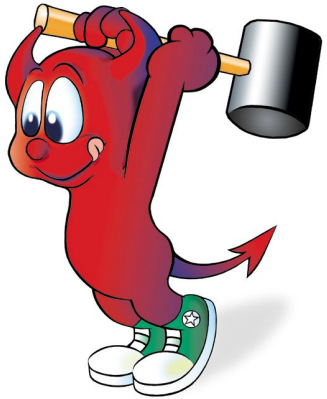
Prototypical Licenses

- Permissive
- Copyleft

Permissive Licenses (the world is a good place)

- Typically, lets developers do whatever they want with the code
- Can make the code proprietary
- Usually does require attribution
- Has a warranty disclaimer
    - Author is not responsible for f**k ups happening due to their code
- Use when you want your code to be used as widely as possible and make it easier for developers to integrate it into their projects.
- “Business friendly”
- Open Source is about working efficiently, not transparency
- Gotta be ok with someone making money off your code and giving nothing back in return

Permissive Licenses



- MIT/Expat/X11, BSD
- Apache 2.0 (special case):
  - Has a “Grant of Patent License” clause
  - Each contributor to the code is also granting anyone who uses the code a license to use any other patents that would otherwise be infringed upon by the code that was contributed.
  - If you do sue over these patents, you also lose any patent licenses granted to you via this clause
  - Prevents patent wars through “mutually assured destruction”

Case Study #1

- **The Community/Apache vs. Facebook**
- ReactJS (along with other web software) was under the BSD + Patents license which basically had the clause:
  - “You lose the patent license to use this software if you sue Facebook for patent infringement.”
- The community was furious; the Apache Foundation went so far as to ban all BSD + Patents licensed code from its projects (including React).
- FB eventually relented and released with an MIT license

Copyleft Licenses (the world is an evil place)

- Principle of granting users freedom to:
  - Use the program as they wish
  - Study the program and change it
  - Share the program
  - Redistribute their changes
- Prevents developers from locking-in users
  - You bought the software, use it how you want
- Requires developers to share the changes they make under the same license terms
  - Even after multiple iterations and forks of your code, the end users will still maintain the 4 original freedoms
- Prevents proprietary software but the code can still be used commercially
- Permissive licenses can be incorporated into copyleft, but not the other way around!
- Considered a “viral” infection, where anything copy left infects, becomes copyleft

Copyleft Licenses

- GNU General Public License : (GPL):
  - The “goto” copyleft license
  - “Viral” clause in the copyleft license only kicks in when you re-distribute the code in some way
    - Therefore, local changes don’t matter
  - GPLv2:
    - Cannot add or reduce restrictions of the GPL
    - Makes it incompatible with Apache 2.0 due to its patent clause addition
  - GPLv3:
    - Fixed Apache 2.0 compatibility by adding the patent clause explicitly
  - Typically you will allow your software to be licensed as GPLvX or higher in the agreement

Copyleft Licenses

- Lesser GNU General Public License (LGPL):
  - Applies the “viral” clause only to other code it sits with, not anything that’s just linked to it in a larger software application.
  - Typically, you see these as libraries
    - You incorporate an LGPL licensed library to build a larger proprietary application
  - Additionally, forces developers to allow users to “hotswap” whatever version of the LGPL licensed code they want:
    - Lets users use newer versions of the library or make their custom versions
    - Helps maintain user freedom (at least for the LGPL code)

Copyleft Licenses

- Affero GNU General Public License (AGPL):
  - Closes a loophole in GPL for networked applications like servers/webapps:
    - GPL forces code to be given to users when the software is distributed to them, but with these the software is never technically “distributed” to them, so it does not have to be shared.
  - Its sharing requirement kicks in even for users who only access it over the internet
  - "If you run a modified program on a server and let other users communicate with it there, your server must also allow them to download the source code corresponding to the modified version running there."

Case Study #2

- **Conical Ltd. (Ubuntu) vs Conservancy (Protects GPL and others) & Free Software Foundation**
- ZFS is licensed under the Common Development and Distribution License (CDDL) which is incompatible with the GPL license that Linux is under (maybe purposefully?).
    - CDDL is “Weak” copyleft like LGPL but also adds a requirement that the CDDL code remain ONLY licensed under itself.
- Canonical released Ubuntu 16.04 with a binary distribution of ZFS (OpenZFS) as a linux kernel module to adapt ZFS natively for linux
- Ultimately, after legal review from a team of lawyers, it was determined that Canonical was not in violation of GPL, but took a whole year!

Yea, but like... I’m lazy

“Whatever”

- You need a license!
- No one will use your code without a license of some kind with it since it is copyright
- “I promise I won’t sue you” is not good enough
- Visible on github != FOSS … (its copyright by default)
- Public Domain:
    - Forfeit your copyright
    - STILL NEED A WARRANTY TO PROTECT YOU
    - In some places this is not even legal (Europe?). They will not allow you to place work in the public domain because they believe its detrimental to you. Kinda like licensing someone to kill you. They will just throw out your Public Domain declaration and treat it as copyright.
    - Creative Commons License:
        - Tries Public Domain, if it fails, defaults to permissive

Custom Licenses

- Nope, just don’t do it
- I assure you whatever you want can be found in one of the already used licenses

Case Study #3

- **JSLINT vs. Everyone**
- Douglas Crockford (co-creator of JS) and other JS tools
- JSLINT is MIT but adds the clause:
  - “The software shall be used for good, not evil”
- Debian, Fedora, etc. can’t distribute in their official repos since they cannot guarantee the goodness of its use, and it conflicts with the GPL for letting users use software for any purpose (evil or not).
- Except IBM!
  - Douglas granted “IBM, its customers, partners, and minions ‘permission to use JSLint for evil’”
- I’m not making this stuff up

So What About Contributed Code?

- Copyright Assignment:
  - You forfeit copyright of your code over to the repo owner (probably not good)
- Contributor License Agreement (CLA):
  - Document you may have to sign before code will be merged in. Varies by project.
- Developer's Certificate of Origin (DCO):
  - What the Linux kernel uses
  - Minimal, no CLA required, retain copyright
  - Asserts that you have permission to submit the code and that it can be distributed under the same license as the rest of the project.
  - Protects the project from illegally submitted code by a contributor
  - When you sign a code commit in git with "-s" command, this is why

Takeaway

- Don’t just slap a license on your code; think about how and why you want people to use your software.
- Licenses are tools and should be used for different purposes.

https://choosealicense.com/

Questions?

Works Cited

https://www.youtube.com/watch?v=9kGrKBOytYM&t=351s
https://adtmag.com/articles/2017/09/25/react-relicensed.aspx
https://en.wikipedia.org/wiki/ZFS#Ubuntu_integration
https://sfconservancy.org/blog/2016/feb/25/zfs-and-linux/
https://en.wikipedia.org/wiki/Douglas_Crockford
https://en.wikipedia.org/wiki/Affero_General_Public_License#/media/File:AGPLv3_Logo.svg
https://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License#/media/File:LGPLv3_Logo.svg
https://commons.wikimedia.org/wiki/File:GPLv3_Logo.svg
https://www.freebsd.org/art.html
https://en.wikipedia.org/wiki/Apache_License#/media/File:Apache_Software_Foundation_Logo_(2016).svg