Logiciel Libre
Cours 4 — Licenses

Stefano Zacchiroli
zack@pps.univ-paris-diderot.fr

Laboratoire PPS, Université Paris Diderot

2014-2015

URL	http://upsilon.cc/zack/teaching/1415/loglib/
Copyright	© 2014-2015 Stefano Zacchiroli
	© 2008-2013 Miguel Vidal
License	Creative Commons Attribution-ShareAlike 4.0 International License
	http://creativecommons.org/licenses/by-sa/4.0/deed.en_US
	CC BY-SA

Outline

1. Free Software licensing

2. License bestiary
   - Academic licenses
   - Public Domain & Co.
   - Other permissive licenses
   - Weak copyleft licenses
   - Strong copyleft licenses

3. License popularity

Why Do I Need a License? (redux)

User point of view

- Copyright covers code.
- IP and Copyright is oriented toward preventing use of copyrighted material.
- If you don't license your code, it can't be used (legally) by other people.

No License Required?

Author point of view

- Copyright comes as soon as someone creates a “tangible” (expressible) work.
- In absence of any licensing declarations, don’t allow any uses (“all rights reserved”).
- Therefore, some declaration is necessary to allow sharing.
- One option is to declare no license is required to use the work (e.g., Public Domain Dedication, more on this later).

Licenses and Communities

- Software licenses are social contracts just as much as they are legal documents.
- When you choose a license, you are charting a course for the future
- You are often establishing a relationship to a larger community.
- Not purely about mechanical and legal choices.
- It can be difficult change later: it is worthwhile spending time to understand it.

The Legal Framework for Free Software Licenses

- Based on international copyright laws and provide the user with certain freedoms. These are granted as permissions which could not be exercised without the license (by default “all rights are reserved”).
- Legal Hacking: FOSS licenses behave as any other license except that they grant a number of rights to the user rather than restricting them.
- in particular: they grant enough rights to guarantee that all users of the software enjoy the 4 Freedoms
- that does *not* mean “free for all”; Free Software license can (and almost always do) impose specific limitations

The Legal Framework for FOSS Licenses (2)

Summing-up:

- FOSS are consistent with IP laws: it's incorrect to suggest FOSS licenses destroy IP.
- Legally, the only difference between proprietary and free software is the license (i.e., terms of use).
- Licenses (Free or not) are based on the *copyright* law of specific countries.

FOSS License Example

Implementing a basic Free Software license might be very easy:

Free License Example

Copyright (c) 2012 Foobar Developers. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the redistributions of source code must retain the above copyright notice.

FOSS License Example

Implementing a basic Free Software license might be very easy:

Free License Example

Copyright (c) 2012 Foobar Developers. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the redistributions of source code must retain the above copyright notice.

That’s all!!

Should I write my own license?

Should I write my own license?

Should I write my own license?

Should I write my own license?

NO.

Why You Should Not Write Your Own License

Many people have attempted to write their own FOSS licenses but:

- You will probably get it wrong (ex., Artistic License 1.0... too clever!).
- Your license will not be an OSI approved “open source” license, and (probably) not a Free Software license (approved by the FSF).
- License proliferation is harmful.
  - increase uncertainty about Free Software: which license should I choose to go FOSS?
  - entangles the license (in)compatibility matrix
  - it makes more difficult to understand what happens when code bases mix different kind of licenses

See, e.g., `http://opensource.org/proliferation`

Why You Should Not Write Your Own License

Many people have attempted to write their own FOSS licenses but:

- You will probably get it wrong (ex., Artistic License 1.0... too clever!).
- Your license will not be an OSI approved "open source" license, and (probably) not a Free Software license (approved by the FSF).
- License proliferation is harmful.
  - increase uncertainty about Free Software: which license should I choose to go FOSS?
  - entangles the license (in)compatibility matrix
  - it makes more difficult to understand what happens when code bases mix different kind of licenses

  See, e.g., `http://opensource.org/proliferation`

Just choose a popular Free Software license.

The Free Software Definition

FOSS licenses are the mechanism to legally implement the 4 freedoms:

When you receive a Free Software you get:

- Freedom 0 The freedom to use (run) the program, for any purpose.
- Freedom 1 The freedom to study how the program works, and adapt it to your needs.
- Freedom 2 The freedom to redistribute copies.
- Freedom 3 The freedom to improve the program, and release your improvements to the public, so that the whole community benefits.

The Free Software Definition

FOSS licenses are the mechanism to legally implement the 4 freedoms:

When you receive a Free Software you get:

- Freedom 0 The freedom to use (run) the program, for any purpose.
- Freedom 1 The freedom to study how the program works, and adapt it to your needs.
- Freedom 2 The freedom to redistribute copies.
- Freedom 3 The freedom to improve the program, and release your improvements to the public, so that the whole community benefits.

Freedoms 1 and 3 require access to a source code.
All four freedoms must be granted at the same time!

Concepts related to FOSS licenses

- Use: The right to use (run) the program, for any or some purposes.
- Redistribution: The act of copying the program and giving it to others.
- Derivative work: A program based on other programs, reusing its source code (more details later).
- Authorship attribution: The obligation of recognizing the authorship of a work when using it or applying any change, such as deriving or redistributing it.

The program remains “owned” (in the IP sense) by the copyright holder. With the license, the user only get specific usage rights.

Types of FOSS licenses

Every FOSS license, no matter the kind of work, must guarantee the four freedoms mentioned above for the case of software.

However, there are free software licenses more permissive and other more strict (the so called "copyleft" licenses are often considered to be the most strict).

Please note that the difference in "strictness" between two different FOSS licenses *doesn't imply* that one of them is "less free" than the other. The benchmark for being "Free" is always the 4 freedoms.

FOSS Licensing

A first look at FOSS licenses, along the spectrum of license complexity.

From least to greatest complexity:

- Academic Licenses
- Permissive Licenses
- Reciprocal Licenses
- Scope-Limited Reciprocal Licenses

Academic Licenses

- Relevant subset of permissive licenses
- The simplest licenses: very few restrictions (close to PD).
- Reserving only attribution (keep names and copyright notice).
- Available for all uses, including use in proprietary closed source products.
- Originally written for and popularized by universities.

Examples: MIT, BSD, ISC

Permissive Licenses

- Superset of academic licenses
- Include explicit grant of patent license
- Available for almost all uses, including use in proprietary closed source products

Examples: Apache

Patent Licenses

Grant of Patent Licenses

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable, patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work , where such license applies.

Reciprocal Licenses

- Requires that derivative works maintains the same license (or, in fact, a compatible one)
- In most case reciprocal licenses require binary distribution to also include full source code
- Also known as “strong copyleft” (or “copyleft”)
- Sometimes called, as denigration tactic, “viral licenses”. If reciprocally licensed code is incorporated, then the application is “infected”

Examples: GPL, AGPL
Examples: CC BY-SA (for non-software works, more on this later)

Scope-limited reciprocal licenses

- Like reciprocal licenses, but with limits on the scope of which parts of a derived work fall under the license terms
- Usually: changes to the main work falls under the license terms
- Usually: additional works that happen to be used with/added to/embedded with the main work do not
- According to the denigratory analogy: “virality” is limited to the main work
- Also known as: “weak copyleft”

Examples: MPL, CDDL, LGPL

What is copyleft?

> *Copyleft is a strategy of utilizing copyright law to pursue the policy goal of fostering and encouraging the equal and inalienable right to copy, share, modify and improve creative works of authorship.*
>
> *Copyleft (as a general term) describes any method that utilizes the copyright system to achieve the aforementioned goal. Copyleft as a concept is usually implemented in the details of a specific copyright license, such as the GNU General Public License (GPL) and the Creative Commons Attribution Share Alike License.*
>
> *Copyright holders of creative work can unilaterally implement these licenses for their own works to build communities that collaboratively share and improve those copylefted creative works.*
>
> — *http://copyleft.org/*

What is copyleft? (cont.)

- Granting the four freedoms is enough to guarantee users will get them only for a specific copy of the work
  - how about further downstream redistribution?
  - how about derived works?
  - how about future versions?
- Copyleft makes sure that all users receiving a copy of the program, no matter how modified, also enjoy the four freedoms.
- It is an active defense of user's freedoms.
- The copyleft clause might have diverse implementations but all of them (at least for software licenses) share the same concept: distribution of any version of this program must preserve user freedoms.
- on the other hand copyleft does preclude some business models, and for that reason it gets backlash (e.g., from big companies) — more on this later

Types of FOSS licenses

Along the axes of copyleft, free licenses can then be classified as:

- Permissive licenses: The author retains copyright solely to disclaim warranty and require proper attribution of modified works, but permits redistribution and modification in any work, even proprietary ones (CC-BY, BSD, Apache, MIT).
- [Weak copyleft]
- Copyleft licenses: The author retains copyright and permits redistribution and modification provided all such redistribution is licensed under the same license. Additions and modifications by others must also be licensed under a compatible copyleft license. (GPL, AGPL, CDDL, CC-BY-SA).

Types of FOSS licenses

Along the axes of copyleft, free licenses can then be classified as:

- Permissive licenses: The author retains copyright solely to disclaim warranty and require proper attribution of modified works, but permits redistribution and modification in any work, even proprietary ones (CC-BY, BSD, Apache, MIT).
- [Weak copyleft]
- Copyleft licenses: The author retains copyright and permits redistribution and modification provided all such redistribution is licensed under the same license. Additions and modifications by others must also be licensed under a compatible copyleft license. (GPL, AGPL, CDDL, CC-BY-SA).

Please note that they are all FOSS licenses.
But while permissive licenses (and partly weak-copyleft licenses) allow to produce proprietary derivative works, copyleft license don't.

Restrictions and FOSS

Are there permissible restrictions in FOSS licenses?

Restrictions and FOSS

Are there permissible restrictions in FOSS licenses?

Yes, of course: everything that does not get in the way of the 4 freedoms is acceptable.
In practice, deciding what is OK and what is not is not always clear cut, often not codified in guidelines, and the decisions may very across organizations (FSF/OSI/Debian/etc).

Macro-categories of commonly accepted restrictions are:

- attribution of authors (as long as attribution does not impede normal use of the work)
- transmission of freedoms (e.g., copyleft)
- detailed protection of user freedoms (access to source code or prohibition of "technical measures", DRM)

Warranty and disclaimer

- Software by itself is not a consumer product.
- When software is (combined into) a consumer product, disclaimers are ineffective.
- "As Is": we are accepting item in the actual state with all its faults.

Example — BSD warranty disclaimer

> *THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.*

(yes: it's all ALL CAPS)

Example — GPL 3

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

License compatibility

Intuition: Compatibility ↔ merge (source) code from different FOSS licenses.

- Two licenses are compatible if a joint derivative work (i.e., a work containing some code released under one license, some under the other) could be legally distributed
  - possibly as Free Software, although the notion of license compatibility is more general than that
- Compatibility is determined by comparing restrictions imposed by all involved licenses.
- A dependent variable, that however does not affect compatibility *per se*, is the resulting license under which the joint derivative work will be redistributed
  - e.g., GPL and BSD licenses are compatible, but the resulting joint work will be under the terms of GPL only
  - e.g., GPL and MPL version 1.1 are incompatible (i.e., it is impossible to integrate code released under the two licenses without violating the terms of at least one of them)

Dual- (or multi-) licensing

Distribute software under two (or more) different sets of licenses.

Variants:

- license segregation: different licenses apply to different copies of the same program (e.g., for proprietary relicensing, or specific business models)
- user choice: different, alternative (OR-ed) licenses apply to the same copy of the software; the user choose the license
  - degenerate case: version N or above clauses. The user is free to choose *which version* of the license (within limits) apply to them

Motivations:

- License compatibility (e.g., Perl, Firefox)
- Business models based on market segregation (MySQL, OCaml)
- Future-proof license-based strategies

Proliferation of licenses

- Vanity licenses: known problem in the community for a few years.
- A growing number of licenses increases exponentially the possible combinations and interactions.
- That, in turn, makes difficult to merge code from diverse sources, both for incompatibility issues and unacceptable clauses.
- It introduces juridical uncertainty requiring lawyers, that it is what *public* licenses were trying to avoid in the first place (i.e., the EUPL license and EULAs).
- It favors FUD-mongers (Fear, Uncertainly, Doubt).

Outline

1. Free Software licensing

2. License bestiary
   - Academic licenses
   - Public Domain & Co.
   - Other permissive licenses
   - Weak copyleft licenses
   - Strong copyleft licenses

3. License popularity

Popular and noteworthy licenses

- Permissive
  - BSD 3-Clause “New” or “Revised” license
  - BSD 2-Clause “Simplified” or “FreeBSD” license
  - Apache License 2.0
  - MIT license
  - ISC License
- Strong copyleft
  - GNU General Public License (GPL), versions 2 and 3
  - GNU Affero General Public License (AGPL), version 3
- Weak copyleft
  - GNU Lesser General Public License (LGPL), versions 2.1 and 3
  - Mozilla Public License (MPL), version 2.0
  - Common Development and Distribution License (CDDL)

Outline

1. Free Software licensing

2. License bestiary
   - Academic licenses
   - Public Domain & Co.
   - Other permissive licenses
   - Weak copyleft licenses
   - Strong copyleft licenses

3. License popularity

BSD License — origins

- BSD (Berkeley Software Distribution) is a Unix flavor developed by University of Berkeley (CA).
- BSD Unix was released under the terms of a “minimalistic” license, which permits both source and binary redistribution; with or without modifications, without any other restriction.
- Historical origin of the most liberal tradition in Free Software, opposing the use of copyleft as a strategy to liberate more software
    - intuition: favor “developer’s freedoms” over “user’s freedoms”
- Several revisions of the license exists
- Each revision is in fact a template, where copyright notices should be properly instantiated

Modern BSD Licenses

- Based on the original BSD license.
- Very popular (BSD userland, PF, TCP/IP, OpenSSH, TCL/Tk. . .)
- You may redistribute the work, in any form (source or binary), as long as you preserve copyright notices
- Includes “as is” and “no warranty” clauses
- “Liberal (= libertarian) license”: no control over software evolution

BSD License — advantages

- BSD license places minimal restrictions on developers and future evolutions.
- This allows BSD code to remain Free Software or become integrated into closed solutions.
- No legal complexity (unlike *GPL family of licenses)
- It allows developers and companies to spend their time creating and promoting good code rather than worrying about license violations.

Prior BSD License (1988)

```
Copyright (c) <year> <copyright holder>.
All rights reserved.

Redistribution and use in source and binary forms are permitted
provided that the above copyright notice and this paragraph are
duplicated in all such forms and that any documentation,
advertising materials, and other materials related to such
distribution and use acknowledge that the software was developed
by the <organization>. The name of the
<organization> may not be used to endorse or promote products derived
from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
```

- *de facto* obsolete
- approved by: FSF, Debian
- not approved by: OSI
- GPL compatible

4-clause / Original BSD license (1990)

```
Copyright (c) <year>, <copyright holder>
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software
   must display the following acknowledgement:
   This product includes software developed by the <organization>.
4. Neither the name of the <organization> nor the
   names of its contributors may be used to endorse or promote products
   derived from this software without specific prior written permission.

[ as-is + no warranty disclaimer ]
```

4-clause / Original BSD license (cont.)

As-is + no warranty disclaimer:

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

- shared with other (modern) variants of the BSD license

4-clause / Original BSD license (cont.)

- infamous advertisement clause
  - advertisements escalation, up to 70 in NetBSD
  - further restriction
- *de facto* obsolete
- approved by: FSF, Debian
- not approved by: OSI
- GPL incompatible

3-clause / Revised / New BSD License (1999)

```
Copyright (c) <year>, <copyright holder>
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
    * Redistributions of source code must retain the above copyright
      notice, this list of conditions and the following disclaimer.
    * Redistributions in binary form must reproduce the above copyright
      notice, this list of conditions and the following disclaimer in the
      documentation and/or other materials provided with the distribution.
    * Neither the name of the <organization> nor the
     names of its contributors may be used to endorse or promote products
      derived from this software without specific prior written permission.

[ as-is + no warranty disclaimer ]
```

3-clause / Revised / New BSD License (cont.)

- intuition: 4-clause without the infamous advertisement clause
- popular permissive license
- approved by: FSF, OSI, Debian
- GPL compatible

2-clause / FreeBSD / Simplified BSD License

```
Copyright (c) <YEAR>, <OWNER>
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

[ as-is + no warranty disclaimer ]

The views and conclusions contained in the software and documentation
are those of the authors and should not be interpreted as representing
official policies, either expressed or implied, of the FreeBSD Project.
```

2-clause / FreeBSD / Simplified BSD License (cont.)

- intuition: 3-clause without the non-endorsement clause
- used by the FreeBSD project
- approved by: FSF, OSI, Debian
- GPL compatible

Popular BSD-like licenses

- ISC
- MIT

ISC: the shortest Free Software license

```
Copyright (c) 4−digit year, Company or Person's Name

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
```

- Functionally similar to the 2-clause BSD.
- Language made “unnecessary” by Berne removed.[1]
- BIND, DHCP, and preferred license by the OpenBSD project.
- approved by: FSF, OSI, Debian
- GPL compatible

[1] according to `http://www.openbsd.org/policy.html`

The MIT License

```
The MIT License (MIT)
Copyright (c) <year> <copyright holders>

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.
[ as−is + no warranty disclaimer ]
```

- Functionally similar to the 2-clause BSD.
- Used by X11/X.org, Symfony, RoR, Lua, Putty, Mono, CakePHP, ...
- It doesn't contain an explicit notice prohibiting the use of the name of the copyright holder in promotion.
- It states more explicitly the rights given to the end-user
- approved by: FSF, OSI, Debian
- GPL compatible

Other BSD-like licenses

- Zope Public License 2.0
  - Used by the Zope application server and related products.
  - Similar to the BSD license, but with explicit clauses that prohibits the use of Zope Corporation trademarks and require documentation of changes
  - approved by: FSF, OSI, Debian
  - GPL compatible

`http://opensource.org/licenses/ZPL-2.0`

Outline

1 Free Software licensing

2 License bestiary
- Academic licenses
- Public Domain & Co.
- Other permissive licenses
- Weak copyleft licenses
- Strong copyleft licenses

3 License popularity

Extreme liberal licensing

How far can we go with liberal licensing?
i.e., maximizing freedom guarantees and minimizing constraints

WTFPL License

```
            DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
                    Version 2, December 2004

Copyright (C) 2004 Sam Hocevar <sam@hocevar.net>

Everyone is permitted to copy and distribute verbatim or modified
copies of this license document, and changing it is allowed as long
as the name is changed.

            DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
  TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

 0. You just DO WHAT THE FUCK YOU WANT TO.
```

- Irreverent text, on purpose
- Licensees are encouraged to do what the @*&%!#* they want to
- Not very popular, not necessarily a good choice
- Approved by: FSF, Debian
- Not approved by: OSI

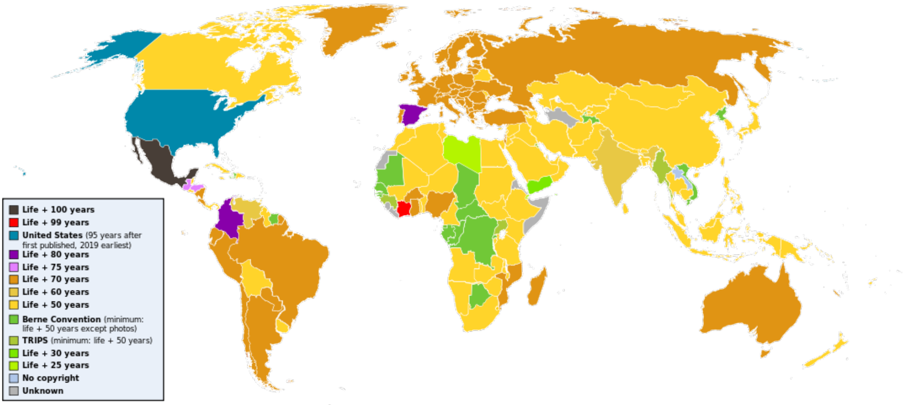
Public Domain

Works in the public domain are those works for which copyright rights have expired, forfeited, or not applicable.

There are thus various ways works enter the public domain:

1. copyright is not applicable to this kind of works (facts, theories, short phrases, . . .)
2. the copyright has expired
   - copyright expired *tout court*
   - the copyright owner failed to follow copyright renewal rules, where/when applicable (e.g., the novel *Anthem* by Ayn Rand)
3. the copyright owner *deliberately* places it in the public domain

World copyright terms



https://commons.wikimedia.org/wiki/File:World_copyright_terms.svg

Public domain dedication I

To deliberately put a work in the public domain, authors should use a public domain dedication, e.g.:

> *The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.*
>
> *A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.*
>
> *Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law,*

Public domain dedication II

> *whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.*
>
> *Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.*

http://creativecommons.org/licenses/publicdomain/

Public Domain Issues

- Is it actually possible *before* copyright expiration?

Public Domain Issues

- Is it actually possible *before* copyright expiration?
- Many legal systems effectively prohibit any attempt by the owners to surrender copyright rights automatically conferred by law.
    - Particularly moral rights (unwaiverable, inalienable).
- A solution: the CC0 license by Creative Commons — waive all copyright rights to the fullest extent allowed by law

CC0 I

(excerpt.)

[...]

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

CC0 II

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

CC0 III

[. . .]

`http://creativecommons.org/publicdomain/zero/1.0/`

Exercise: ipfilter license (2000)

```
/*
 * Copyright (C) 1993–2000 by Darren Reed.
 *
 * The author accepts no responsibility for the use of this software
 * and provides it on an ‘‘as is’’ basis without express or implied
 * warranty.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that this notice is preserved and due credit is given
 * to the original author and the contributors.
 *
 * This program 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.
 *
 * I hate legalese, don’t you?
 */
```

ipfilter license “clarification” (2001)

```
/*
 * Copyright (C) 1993-2000 by Darren Reed.
 *
 * The author accepts no responsibility for the use of this software
 * and provides it on an ‘‘as is’’ basis without express or implied
 * warranty.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that this notice is preserved and due credit is given
 * to the original author and the contributors.
 *
 * Yes, this means that derivitive or modified works are not
 * permitted without the author's prior consent.
 *
 * This program 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.
 *
 * I hate legalese, don't you ?
 */
```

Theo de Raadt announces ipfilter replacement

```
Date: Tue, 29 May 2001 19:13:11 -0600
From: Theo de Raadt <deraadt@cvs.openbsd.org>
Subject: ipf

sometime in the next 20 hours, i will be removing ipf from the
source tree since it does not meet our freedom requirements,
as have been outlined in policy.html and goals.html since the
start of our project.

we will have to work on an alternative.
```

https://groups.google.com/d/msg/fa.openbsd.tech/q3b--naHTF0/iERRvuKkTFEJ

ipfilter case

- The real problem is that code with a non-free license was incorporated into the core of a free operating system.
- Carelessness with licenses invites trouble.

Outline

1 Free Software licensing

2 License bestiary
- Academic licenses
- Public Domain & Co.
- Other permissive licenses
- Weak copyleft licenses
- Strong copyleft licenses

3 License popularity

Apache License

- Old versions: 1.0 (original) and 1.1 (ASF, 2000).
- An extension of the 3-clause BSD license.
- Permits to be integrated into closed source projects.
- Apache License 2.0 (January 2004): permissive license.
  - Make the license easier for non-ASF projects to use.
  - Explicitly grants patent rights where necessary to operate, modify and distribute the software (sect. 3).
  - Patent retaliation (terminating the license upon the initiation of a lawsuit)

Apache License 2.0

- popular license
  - ≈150 projects hosted by the Apache Software Foundation (2015)
  - over 8'000 non-ASF projects located at SourceForge are available under Apache License (2012)
  - 25% of Google Code projects, including Android user space (2008)
- approved by: FSF, OSI, Debian
- compatible with GPLv3
- incompatible with GPLv2

Apache License: Grant of Patent

3. Grant of Patent License

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable, patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work , where such license applies [...].

Apache License: Patent retaliation

3. Grant of Patent License

If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

Outline

1. Free Software licensing

2. License bestiary
   - Academic licenses
   - Public Domain & Co.
   - Other permissive licenses
   - Weak copyleft licenses
   - Strong copyleft licenses

3. License popularity

Mozilla Public License (MPL)

1998 version 1.0, as a successor of the NPL (Netscape Public License)

1999 version 1.1 by Mozilla Organization

- public feedback/review process on how to improve version 1.0
- allow for multiple licensing

2012 version 2.0

- public process again
- GPL-compatible

Mozilla Public License (MPL) 2.0

- weak (or partial) copyleft license, with file-based boundaries on the reach of copyleft requirements
- separation between covered software and larger work
  - Covered Software

    means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof

  - Larger Work

    means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software

- see `http://opensource.org/licenses/MPL-2.0`

Mozilla Public License (MPL) 2.0 (cont.)

- covered software
  1. copyleft-like clause

     > *All distribution of Covered Software in Source Code Form, including any Modifications that You create or which You contribute, must be under the terms of thi License*

  2. source code requirement

     > *If You distribute Covered Software in Executable Forn then [. . .] such Covered Software must also be made available in Source Code Form*

- larger work: its own license

  > *If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, [. . .], this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).*

Mozilla Public License (MPL) 2.0 (cont.)

- Explicitly grants patent rights where necessary to operate the software.
- approved by: FSF, OSI, Debian
- version 2.0 of the license is compatible with the GPL
- version 1.1 is *in*compatible with the GPL
  - A module covered by the GPL and a module covered by the MPL version 1.1 cannot be linked together.
  - For this reason, Firefox has been relicensed under multiple licenses (MPL, GPL, LGPL).
  - MPL 1.1 can be specifically amended to allow combining with GPL and others (sect. 13, “Multiple-licensed code”).

CDDL License

- The Common Development and Distribution License (CDDL) is based on the MPL, version 1.1.
- Produced by Sun Microsystems for the OpenSolaris projects (kernel, userland, ZFS, DTrace, NetBeans, GlassFish, . . .)
- approved by: FSF, OSI, Debian
- It tries to amend GPL-incompatibility issues in the MPL 1.1, without succeeding (at least according to the FSF and Debian)
- Such as some non-compliance with European law system in the MPL that have been corrected in the CDDL.
- approved by: FSF, OSI, Debian.
- GPL-incompatible.

GNU LGPL

1991 GNU *Library* General Public License, version 2 (for uniformity with GPL version)

1999 GNU *Lesser* General Public License, version 2.1

- name change to emphasize that it is inferior (from a copyleft POV) to the GPL, rather then the recommended variant of the GPL for software libraries

2007 GNU LGPL, version 3

- reimplemented as GPLv3 + additional permissions

- very popular license for libraries (and more)
- approved by: FSF, OSI, Debian
- GPL compatible

GNU LGPL 2.1

- *4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, [. . .]*

GNU LGPL 2.1

- *4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, [. . .]*
- *5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.*

Note the lack of explicit file boundaries (contrary to, e.g., MPL)

GNU LGPL 2.1

- *4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, [. . .]*
- *5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.*

Note the lack of explicit file boundaries (contrary to, e.g., MPL)

- *3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library [. . .] This option is useful when you wish to copy part of the code of the Library into a program that is not a library.*

GNU LGPL 3

- *A “Combined Work” is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the “Linked Version”.*
- *The “Minimal Corresponding Source” for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version*
- *The object code form of an Application may incorporate material from a header file that is part of the Library.*

GNU LGPL 3(cont.)

- *You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering [. . .]*
- *[provided that] you do one of the following:*
  1. *Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version [. . .]*
  2. *Use a suitable shared library mechanism for linking with the Library. [. . .]*

things start to get quite technical for a legal document. . .

LGPL vs GPL — rationale

Why you shouldn't use the Lesser GPL for your next library
`https://www.gnu.org/licenses/why-not-lgpl.html`

> *The GNU Project has two principal licenses to use for libraries. One is the GNU Lesser GPL; the other is the ordinary GNU GPL. [...] using the Lesser GPL permits use of the library in proprietary programs; using the ordinary GPL for a library makes it available only for free programs. [...]*
>
> *Which license is best for a given library is a matter of strategy [...]. [...] Free software developers need to make advantages for each other. Using the ordinary GPL for a library gives free software developers an advantage over proprietary developers: a library that they can use, while proprietary developers cannot use it. [...]*
>
> *There are reasons that can make it better to use the Lesser GPL in certain cases. The most common case is when a free library's features are readily available for proprietary software through other alternative libraries. In that case, the library cannot give free software any particular advantage, so it is better to use the Lesser GPL for that library.*

Outline

1. Free Software licensing

2. License bestiary
   - Academic licenses
   - Public Domain & Co.
   - Other permissive licenses
   - Weak copyleft licenses
   - Strong copyleft licenses

3. License popularity

GNU General Public License (GPL)

- considered to be the most popular Free Software license
- approved by: FSF, Debian, OSI

1989 version 1 (by RMS), as a generalization (hence the name) of licenses already used by the GNU project for: Emacs, GDB, GCC

1991 version 2 (by RMS)

- "liberty or death"; early ex. of defense against patents and similar threats to user freedoms

2007 version 3 (by RMS with counsel from E. Moglen/SFLC)

- public review process
- software patents clauses
- DRM clauses (anti "tivoization")
- license compatibility provision
- internationalization
- self-defense against further restrictions

GPL — relevance

What makes the GPL so special?

- It was the first license to outline the copyleft principle.
- Highly influential on all subsequent copyleft/share-alike licenses, including the Wikipedia license.
- Without the GPL, copyleft would have been just an abstract idea.
- Designed to prevent proprietary relicensing of Free Software code.

GPL — characteristics

- This license guarantees the four FLOSS freedoms.
- "Copyleft" clause: all derivative works should be licensed also under the same license.
- Since in USA software patents are admissible, GPL includes a clause for avoiding GPL licensing of patented software or algorithms.
- GPL code can not be mixed with other code under "GPL-incompatible" license.

GPLv2 — source code requirement

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) *Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or, [. . .]*

[2] the so called system library exception

GPLv2 — source code requirement

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) *Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or, [. . .]*

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable.
However, as a special exception,[2] *the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.*

[2] the so called system library exception

GPLv2 — copyleft

> *2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:*
>
> a) *You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.*
>
> b) *You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.*

As a consequence of the fact that derived works fall under the terms of the GPL themselves, *their* source code must be distributed as well.

GPLv2 — no EULA

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

GPLv2 — no EULA

> *5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.*

How about permission to *use* the Program?

GPLv2 — no EULA

> *5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.*

How about permission to *use* the Program?

> *0. [. . .] Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, [. . .]*

GPLv2 — “or later”

Recommended—by the license itself—way to apply the GPL(v2) license to source code:

> *This program 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 2 of the License, or (at your option) any later version.*

- part of the copyright/license notices, not of the license itself
- individual software authors *can* leave the “or later” clause out
- other licenses include implicit “or later” requirements in the license text itself (e.g., MPL)

GPLv2 — looking back

- Written by Richard Stallman and the FSF, published in 1991.
- The most popular Free Software license: estimated to cover 50-70 % of all Free Software projects (at the time)
- It’s more than a software license: it is a social contract, imposing that all players have the same rights and obligations

Why update in?

GPLv2 — looking back

- Written by Richard Stallman and the FSF, published in 1991.
- The most popular Free Software license: estimated to cover 50-70 % of all Free Software projects (at the time)
- It's more than a software license: it is a social contract, imposing that all players have the same rights and obligations

Why update in?
After 15 years, needed updating in order to remain effective against new threats to user freedoms.

Intuition: the GPL is a mean to an end. It is an implementation that might have bugs (or grow them over time), which need to be fixed in further releases of the license.

GPLv3 — public consultation

Public consultation process:

- very relevant and the social responsible thing to do: given the abundance of “or later” software, the effects of the release of GPLv3 might be huge
- It lasted eighteen months: from January 16, 2006 (first draft) to June 29, 2007 (final version).
- Selected/invited participants from high-profile Free Software projects.
- 4 drafts.
- 5 International Conferences (Boston, Porto Alegre, Barcelona, Tokyo and Brussels)

GPLv3 — DRM

> *3. Protecting Users' Legal Rights From Anti-Circumvention Law.*
>
> *[. . .]*
>
> *When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.*

- does not *forbid* to implement DRM & co. in software
- but allows to write interoperable software and bypass restrictions
- neutralize laws that get in the way of user freedoms (e.g., DMCA, EUCD)

GPLv3 — DRM (cont.)

- Together with

> *6. Conveying Non-Source Forms.*
>
> *[. . .]*
>
> *"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source.*

it also neutralizes "tivoization", i.e., the circumvention of the GPL by using cryptography to disallow the installation/execution of modified versions of a GPL'd program

GPLv2 — patents

Protection against patent threats is implemented by GPLv2 only in the "Liberty or Death" clause:

> *7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all.*
>
> *[. . .]*
>
> *It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. [. . .]*

GPLv3 — patents

GPLv3 *adds* (i.e., “liberty or death” remains) stronger protection against patent threats through legal-engineering:

> *11. Patents*
>
> *[. . .] Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor’s essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.*

> *10. Automatic Licensing of Downstream Recipients.*
>
> *[. . .] you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.*

GPLv3 — self-defense

7. Additional Terms.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. [require copyright ownership]

[. . .]

All other non-permissive additional terms are considered "further restrictions" [. . .] If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term.

GPLv3 — variants and compatibility

- *7. Additional Terms.*
  [. . .] for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms: [. . .]
  e) *Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or*

  (and other similar permissions for adding warranties/“as is” disclaimers, limiting the use for publicity purposes, etc.)

- *Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work.*

The Application Service Provider (ASP) loophole

- the ASP loophole:
  1. obtain a copy of some GPL'd program
  2. modify it
  3. offer remote access to your modified version over the Net (e.g., web app, remote API, etc.)

  Does the GPL force you to redistribute the code of your modified version?

The Application Service Provider (ASP) loophole

- the ASP loophole:
  1. obtain a copy of some GPL'd program
  2. modify it
  3. offer remote access to your modified version over the Net (e.g., web app, remote API, etc.)

  Does the GPL force you to redistribute the code of your modified version? **No.**
- GPL (both v2 and v3) copyleft clauses trigger upon "distribution"/"convey" of the modified copy, in either source or non-source form
- if you do not do any of that, copyleft *de facto* does not apply
- from copyleft POV, this is very problematic for web apps

  > *"GPL is the BSD of Web applications"* — *Bradley Kuhn*

- but in an increasingly more connected world, the problem is more general

GNU Affero GPL (AGPL)

- Based on the GPL
- Published by the Free Software Foundation (version 3: 2007).
- It contains the extra Affero clause that requires distribution of modified source code of applications to users interacting remotely over the network with the program
- The clause has initially been considered for inclusion in GPLv3, but then relegated into a separate license
- approved by: FSF, Debian, OSI
- GPL compatible (explicitly so)

AGPL — Affero clause

13. Remote Network Interaction; Use with the GNU General Public License.

[. . .] if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software.

Derivative works and the GPL

GPL copyleft propagation applies to (GPLv2 language):

> *a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language.*

Can you link a GPL program/library with a non-GPL program/library, without applying the GPL to the obtained binary?

Derivative works and the GPL

GPL copyleft propagation applies to (GPLv2 language):

> *a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language.*

Can you link a GPL program/library with a non-GPL program/library, without applying the GPL to the obtained binary?

- FSF/FSF's lawyers (and popular) answer: no; the GPL applies
- some corporate lawyers' answer: yes; the GPL doesn't apply
- court cases/tribunal answer: none (yet)

(according to some, the resulting answer is thus "we don't know")

Derivative or collective works?

[US law language]

- a derivative work is a "work based upon one or more preexisting works", which requires some transformation or adaption of the original
- a collective work is created when a person brings together "preexisting materials... in such a way that the resulting work as a whole constitutes an original work of authorship"
  - individual parts remain under their individual licenses
  - a separate license apply to the collection

Does linking create a derivative or a collective work (or both)?

Linking and the GPL — FSF position

License text (redux):

> *a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language.*

From the GPL FAQ[3]

> *Q: Does the GPL have different requirements for statically vs dynamically linked modules with a covered work?*
> *A: No. Linking a GPL covered work statically or dynamically with other modules is making a combined work based on the GPL covered work. Thus, the terms and conditions of the GNU General Public License cover the whole combination. [. . .]*
> *Q: Can I release a non-free program that's designed to load a GPL-covered plug-in?*
> *A: [. . .] Using shared memory to communicate with complex data structures is pretty much equivalent to dynamic linking*

[3] http://www.gnu.org/licenses/gpl-faq.html

Linking and the GPL — arguments

- dynamically linked executables contains “annotations and elaborations” on a base binary
  - does a Linux kernel module contains annotations and elaborations on the base expression of the kernel?
  - if yes, then it might be a derived work of the kernel, which is GPLv2

[4] https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/COPYING

Linking and the GPL — arguments

- dynamically linked executables contains “annotations and elaborations” on a base binary
  - does a Linux kernel module contains annotations and elaborations on the base expression of the kernel?
  - if yes, then it might be a derived work of the kernel, which is GPLv2
  - how about user programs that run on Linux?
  - according to Linus:[4]

    > *NOTE! This copyright does *not* cover user programs that use kernel services by normal system calls - this is merely considered normal use of the kernel, and does *not* fall under the heading of "derived work". Also note that the GPL below is copyrighted by the Free Software Foundation, but the instance of code that it refers to (the Linux kernel) is copyrighted by me and others who actually wrote it.*

    … but is he right?
- the legal principle of usage of trade might play a role too

[4] https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/COPYING

Linking and the GPL — arguments (cont.)

- arguments to the contrary (often by corporate lawyers) claim that linking only creates collective works—not subject to the GPL as a whole—because there is no substantial difference between two executables on disk and two in memory
- header files might also play a role
  - during compilation (before linking) you might use header files to prepare your executable for dynamic linking
  - if the headers used at compile time are GPL'd, then your dynamically linked executable might be a derived work of the headers

Linking and the GPL — arguments (cont.)

- there are also other types of “linking”: RPC, RMI, REST API, etc. When do they constitute “linking” in a sense that would trigger strong copyleft requirements?
    - no consensus yet
    - folklore suggests that:
        - loosely coupled and/or popular and/or standardized APIs with several alternative implementations should not trigger the GPL
        - tightly coupled and/or ad-hoc and/or single-implementation APIs should trigger the GPL
- on the other hand, it seems consensual that static linking will produce a derivative work of the GPL part

Linking and the GPL — arguments (cont.)

- there are also other types of “linking”: RPC, RMI, REST API, etc. When do they constitute “linking” in a sense that would trigger strong copyleft requirements?
  - no consensus yet
  - folklore suggests that:
    - loosely coupled and/or popular and/or standardized APIs with several alternative implementations should not trigger the GPL
    - tightly coupled and/or ad-hoc and/or single-implementation APIs should trigger the GPL
- on the other hand, it seems consensual that static linking will produce a derivative work of the GPL part

Ultimately, this GPL linking dilemma seems to be problematic only for those who want to somehow circumvent the main principle of the GPL which, *per se*, is very clear.

Outline

1. Free Software licensing

2. License bestiary
   - Academic licenses
   - Public Domain & Co.
   - Other permissive licenses
   - Weak copyleft licenses
   - Strong copyleft licenses

3. License popularity

Warning

Due to the free circulation of Free Software, it is very difficult to get hard number about software—and therefore license—popularity.

Nonetheless, many actors of the FOSS ecosystem publish statistics about those facts. Unfortunately, most of them do so in rather unscientific ways: without disclosing the details about the dataset they are using, and without liberating the software they use to compile their statistics.

Use caution in interpreting the data.

Blackduck: “Top 20 Open Source Licenses”

GPL 2.0	25%
MIT	19%
Apache License 2.0	16%
GPL 3.0	10%
BSD 3-clause	7%
Artistic (Perl)	5%
LGPL 2.1	5%
LGPL 3.0	2%
MS-PL	2%
EPL	2%
Code Project Open License	1%
MPL 1.1, BSD 2-clause, CDDL 1.0, AGPL, Microsoft Reciprocal License, Sun GPL w/ classpath exc., CDDL 1.1, zlib/libpng, CPL	<1%[5]

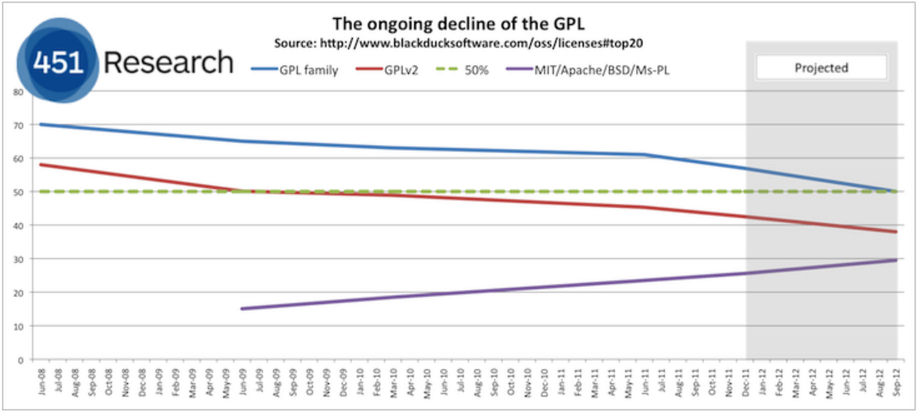
https://www.blackducksoftware.com/resources/data/top-20-open-source-licenses

February 2015

[5] each license

Blackduck: the (alleged) decline of the GPL

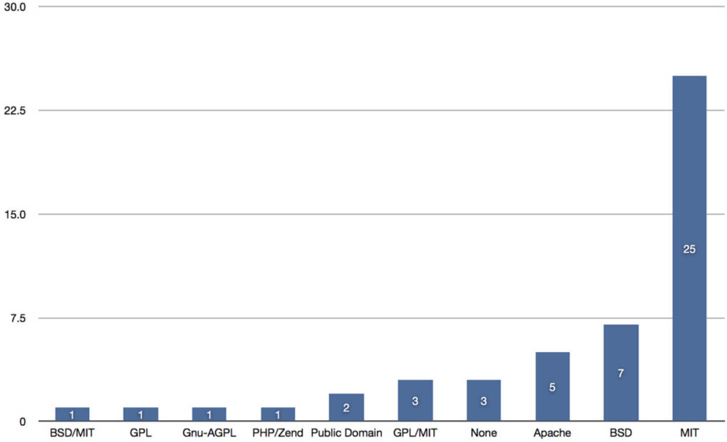
According to Blackduck, in recent years the GPL is losing ground to permissive licenses:



The ongoing decline of the GPL

Source: http://www.blackducksoftware.com/oss/licenses#top20

GitHub: top licenses



http://ostatic.com/blog/the-top-licenses-on-github

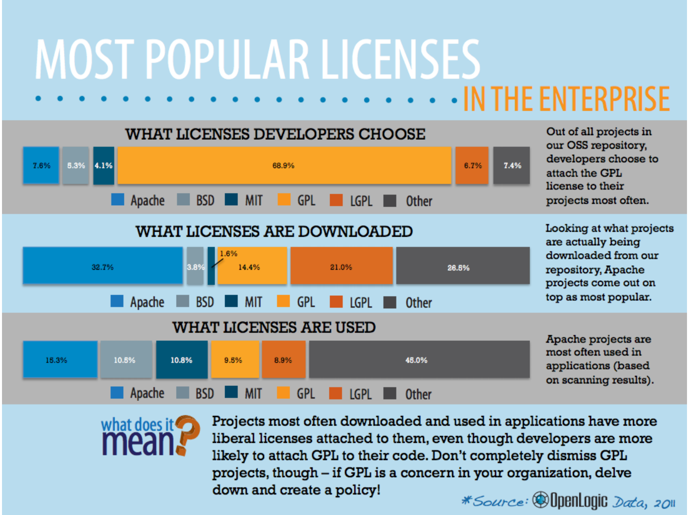
February 2012

GitHub — a bit more complex than that

- 2013 analysis of GitHub by Aaron Williamson “Licensing of Software on Github: A Quantitative Analysis”[6]
- only 14.9% have a top-level license file
- only 3.6% mention a license in README
- for the remaining projects, the breakdown is largely confirmed
- POSS (“Post Open Source Software” debate)

[6] http://www.softwarefreedom.org/resources/2013/lcs-slides-aaronw/

OpenLogic: most popular licenses in the enterpises



Out of all projects in our OSS repository, developers choose to attach the GPL license to their projects most often.

Looking at what projects are actually being downloaded from our repository, Apache projects come out on top as most popular.

Apache projects are most often used in applications (based on scanning results).

Projects most often downloaded and used in applications have more liberal licenses attached to them, even though developers are more likely to attach GPL to their code. Don't completely dismiss GPL projects, though – if GPL is a concern in your organization, delve down and create a policy!

*Source: OpenLogic Data, 2011

http://www.openlogic.com/blog/bid/197148/open-source-software-101-understanding-compliance

July 2012

Discussion: GPL vs BSD

- "BSD code is free, but GPL code stays free"
- copyleft: (non) business friendly?