FREE SOFTWARE FOUNDATION

Bulletin

Issue 30
Spring/Summer 2017

Contents

Copyleft's decline: A claim without evidence

By John Sullivan
Executive Director

We know that one tactic for convincing people to stop doing something you don't want them to do is to tell them nobody else is doing it anymore. Peer pressure is a powerful force, and in the world of technology, there's a particularly strong desire to be seen as current. That could be why we've been seeing reports in the tech press that the use of copyleft licenses, like the GNU General Public License (GPL), is declining in comparison to the use of lax permissive licenses like the Apache or Expat (commonly but unfortunately called MIT) licenses.

All of the articles I've seen making this claim cite the same few corporate "studies" as their primary sources. The evidence they present is not evidence at all, because neither the specific data set nor the methodology used are published. No field of science accepts experimental conclusions that can't be reproduced by others. We shouldn't accept such conclusions in the area of counting license use either.

Counting the licenses used by free software projects may seem straightforward. By definition, all of their code is published in publicly-available repositories, and should carry easy-to-read notices indicating the applicable licenses. But doing it turns out to involve a minefield of potential errors and biases.

Sumana Harihareswara, keynote speaker at LibrePlanet 2017

I reviewed some of the data likely used by companies counting licenses,

and found obvious mistakes. As of May 2017, `openhub.net`, operated by the company Black Duck and used in its license-counting data set, lists GNU Bash as GPLv2-or-later. Bash has been GPLv3-or-later for several years. While it's now been corrected, the site also listed GNU Emacs as GPLv2-only, a license the project has never had. I found these errors on the first two projects I spot-checked. How many more would we find if the full data set were identified?

Even if the inputs were perfect, writing software to count licenses is extremely difficult and requires making many normative choices. These choices need to be disclosed if we're to draw any accurate conclusions. The problems start with deciding what qualifies as a project to count. Do you care whether the code actually works, or whether it's had contributions from more than one person? Projects often change code hosting sites without removing their old home. If you are crawling multiple hosts, is your code smart enough to tell when two programs are the same? Does a forked or slightly modified version count as a separate program? Versions of the same program for different operating systems can conceivably each be under a different license. Do you count them separately?

After you've determined which projects qualify, you have to parse their license information. License notices are not yet predominantly in structured, machine readable formats. They are written by and for humans, with typos and inconsistent formatting that confound automated parsers. When licenses are recognized, there may be several of them. A GPL-covered project can contain files carrying lax permissive license notices, because it is allowable—and common—to redistribute such files as part of a copyleft work. Does that add one just to the GPL column, or do you also increment the noncopyleft license columns?

Once you've decided a project qualfies, and have figured out how to represent its license(s), you then have to decide how much weight to give it. Do you care about the size of the codebase? If you don't, then you will count a large package like GNU Emacs as equal to a small node.js library. If you do care, then you have to create categories to better compare apples to apples, and those criteria need to be shared for others to properly understand the results. Do you care about the size of the user base? If you don't, you will count a GitHub repo containing someone's personal configuration files, kindly shared under a free license but really intended only for their personal use, the same as GCC, used as the foundation for billions of dollars in economic value. If you do care, then you need to share how you determined the user base and how that was incorporated.

Counting licenses used across the entire universe of free software is not an easy job. Whether any given article claiming that copyleft is declining is part of an intentional anti-copyleft effort or not, it risks creating a self-fulfilling prophecy by increasing peer pressure against choosing copyleft licenses. As an individual advocate for user freedom, you can make a difference by questioning these claims when you see them.

Ask two questions: First, is the

methodology, including the code used to do the counting, published? Second, is the data set published? If the answer to either one of the above is no, then the claim should be ignored entirely. It's no better than an assertion, and the interpretation of the "data" will be like reading tea leaves—just the author's own confirmation bias from within their particular bubble.

You can avoid the self-fulfilling prophecy by choosing copyleft for your own projects. Individual license choices have a large impact, because they influence the decisions made by future projects based on yours, or that integrate with yours. From my bubble, I see plenty of people continuing to choose copyleft. We interview some of them every month in a blog series.[1] Recently, the Department of Defense chose the Affero GNU GPL as the license for a new project, and plans to use the GPL as the default for its future projects.[2]

You can also help efforts to scientifically collect information about software license usage. Our Free Software Directory is growing into a useful resource for this, and welcomes volunteer contributions. The Software Heritage Project will be extremely useful in this area as well, and there are packages like FOSSology which aim to do the work of license counting with free, auditable software.[3]

In the end, we need to remember that numbers about who chooses which free license may not be that useful or interesting. All of this is part of the same pie as proprietary software, and so increases in noncopyleft use may be trading off with proprietary licenses, not copyleft, and noncopyleft licenses are still free software licenses. If every proprietary license were replaced with a noncopyleft free license tomorrow, that would be an amazing victory for our movement.

Licenses are a means to the end of user freedom. Copyleft remains the best tool we have for achieving and securing that freedom in the context of our current global regimes on copyright, patents, and contracts. We need it now more than ever. Software under noncopyleft licenses is free, but contingent—future improvements to it can be made proprietary, essentially pulling the rug out from under us. Only copyleft builds a solid foundation for freedom. If we want to measure something, let's focus on metrics of how more or less free we are in our daily, increasingly digital, lives.

[1] `https://fsf.org/blogs/licensing`

[2] `https://code.mil`

[3] Software Heritage: `https://softwareheritage.org`, FOSSology: `https://www.fossology.org/`

Respects Your Freedom certification program continues to grow

By Donald Robertson
Licensing and Compliance Manager

The Free Software Foundation's Respects Your Freedom (RYF) certification program is growing rapidly. In March of this year, we certified three devices from Vikings GmbH, bringing the total number of certified devices to twenty-two.[4] Certifying multiple devices at once quickly expands the pool

[4] The Vikings devices most recently certified include: Vikings USB Stereo Sound Adapter, Vikings D16 Mainboard, and Vikings X200 libre-friendly laptop.

of hardware that users can trust, but can also work as a stress test of our certification system.

When the RYF program first launched, no one knew how many companies would be up to the task of ensuring that their hardware only came with freedom inside. We are certainly pleased with the response. Increasingly, however, companies are looking to follow in Vikings' footsteps with ever larger launches. It makes sense: once you know what it takes to meet the RYF criteria, it becomes easier to see all of your hardware as potential candidates.

As the number of applicants and devices has risen, so has the need to refine the certification process to better handle increased interest in RYF. We plan on publishing more information about the process, so applicants can better know what to expect. Right now, the criteria are published, but the actual process from initial contact and application form, to rounds of review, to certification and announcement aren't publicly documented in full.[5] We are also working with potential partner organizations to help set up something like a mentoring program to help first time applicants through the process.

Another big item is that we are working with current applicants to rethink how we handle reviewing the physical devices themselves. Currently, we ask for two samples of each device be sent to the FSF. That certainly isn't too onerous when dealing with a single device, but that changes with the prospect of potentially dozens of pieces of hardware. Particularly, we are looking at how we handle what are essentially variations of the same device, such as a laptop with different pre-installed distributions. The same base device can be sold with many different potential configurations of components. Each configuration can represent an issue regarding what software might be hiding inside, or what free software is compatible with that component. This part of the process is not easy to improve, however. The RYF program certifies a particular piece of hardware as it is sold to a user. It is not a general recommendation of a particular retailer, so we need to check all devices that are up for certification. We want to streamline the process while still maintaining a robust standard of review, and we are working with current applicants to figure out the proper balance between those two goals. With these upcoming changes, we hope to continue to help the program expand while maintaining its rigorous standards.

Historically, RYF devices have leaned heavily on refurbishing existing hardware with a fully free stack of software. But more and more we are hearing from companies looking to build RYF devices from the ground up. Controlling the design of their own hardware means they can avoid problems from the start, rather than having to reverse engineer solutions on existing devices. Because hardware manufacturers are increasingly locking down machines, being able to

[5] https://u.fsf.org/ryfcriteria

create works designed with freedom in mind is necessary for the future of the RYF program. This is an exciting development, and one that is coming much sooner than anticipated. In addition, we are also receiving applications for many types of devices that we haven't previously certified, bringing us closer to one day having a pool of certified devices that could meet all of a user's needs.

There is an incredible number and variety of devices currently working their way through our certification program, so keep an eye out for upcoming announcements.

There is more information about RYF, including a list of certified devices, on `fsf.org/ryf`.

Images are from *Wires for Empathy*. A film inspired by the epic *Gilgamesh*, it is directed by Bassam Kurdali and made using Blender: `wiresforempathy.org`.

Join the federation

By Georgia Young
Program Manager

Since 2004, online interaction between friends and people with shared interests has slowly become dominated by a few giant social networking sites. You probably know of them: Facebook, Twitter, and Instagram are among those with the most users, and focus on sharing images, links, video, and chat.

These sites want to control your computing. They use their own servers, which you can't access. They force you to run their preferred systems for accessing these sites. They promote use of nonfree software by serving users nonfree JavaScript (for example, Facebook does this for some features) that runs on users' local machines, and by promoting and distributing proprietary mobile apps to use their services.[6] They control what technologies you can use to access their servers, what you are allowed to do on their site, and the data you generate.

Federated and free: why it's good

Luckily, there's an option that allows more people to have direct control over their social network activity: free software federation.[7] There are

[6] Facebook, Twitter, and Instagram all use and contribute individual components that are available under licenses included in the GNU Project's list of free software licenses: `https://u.fsf.org/lb`

[7] `https://u.fsf.org/284`

many reasons why free software federation is great, including:

- Interoperability: On a federated (also known as distributed) network, people whose accounts are located on different instances can communicate with each other.[8]

- Freedom: In theory, federated networks can include nodes pushing proprietary software, but in practice, it's an approach favored more by free software. You can examine the code yourself in order to understand what it does and determine whether it can be trusted. You can modify it, too. The microblogging software Mastodon is an alternative implementation of GNU social–and that's possible because GNU social is free software.[9] So the ideal social web isn't just federated, it's a federation of free software nodes.

- Resilience: Because a federated network is made up of multiple instances, each used by a different set of people, rather than being operated by one company with all users relying on the same server, the failure of one instance doesn't affect all users.

- Privacy: Using a federated network means it is harder for a large company to spy on you. And if you run your own node in the network, you can inspect the code to make certain that the privacy of data associated with your account is being respected.

- Preserving your data: When you place photos or other documents that are important to you in the hands of centralized, corporate-controlled social media software, you could unexpectedly lose those things if the company that controls the servers decides to discontinue the program or block access to your account. If you host your own instance, you decide.

Federation sounds weird. How does it work?

In practice, someone using a federated social network is likely to detect only a slight difference. Take the experience of GNU social versus Twitter, for example. If you want to use Twitter, you need to create an account on `twitter.com`, nowhere else. You can only reply to another person if they also have an account on twitter.com, and each handle looks like this: `@fsf`. (Yes, the FSF does use Twitter, in a way that avoids using any proprietary software.[10])

But if you use a federated social network, like GNU social, you might create your account on a GNU social instance, like `https://quitter.se/`, but that site is only one of many options.[11] If the FSF (`fsf@status.fsf.org`) wants to talk to 2016 Free Software Award winner Alexandre Oliva, we would tag his full handle: `@lxoliva@social.libreplanetbr.org`.

The protocols underlying free decentralized social media are continuing to advance. Three years ago, the World Wide Web Consortium (W3C) created a Social Web Working Group whose goals include creating a Web

[8] `https://u.fsf.org/286`

[9] Mastodon: `https://u.fsf.org/285/`, GNU social: `https://gnu.io/social/`

[10] `https://www.fsf.org/twitter`

[11] `https://gnu.io/social/try/`

protocol for "federating social information such as status updates," explicitly to "facilitate access to social communication on the Web." The working group's charter includes in its use cases user control over personal data and cross-organization adhoc federation.[12] Unlike other efforts by the W3C, it's nice to see the Social Web Working Group engaged in some freedom-respecting goals.[13]

Now that you understand a bit more about how federated social networks behave, and why they're good for your freedom, why not try one? Visit a diaspora pod for a general social network, a GNU social or Mastodon instance for microblogging, or a GNU MediaGoblin instance for sharing media like video and images.[14]

What not to miss from LibrePlanet

By Various LibrePlanet Attendees

The 2017 LibrePlanet team put a lot of effort into creating a diverse, interesting, balanced schedule full of talks we really wanted to see. During the event itself, however, we found ourselves lacking the time necessary to sit through a whole session. We reached out to a few attendees for recommendations.

Understanding the complexity of copyleft defense by Bradley Kuhn[15]

I liked Bradley Kuhn's talk. I thought he made some really important points about the place the legal system has in the fight for software freedom. He made a convincing argument I now subscribe to.

—Carol Smith

The set of programmers: How math restricts us by Carol Smith[16]

I found this to be a really thoughtful and engaging overview of a topic I hadn't thought very much about. I wish every technical recruiter, hiring lead, and/or admissions committee would watch it.

—Shauna Gordon-McKeon

Freedom and loathing on the campaign trail '16 by Remy DeCausemaker[17]

This is a super interesting look into the technology of a presidential political campaign, and the opportunities for free software and open community values to fit into that.

—Shauna Gordon-McKeon

The Lisp machine and GNU by Christopher Webber[18]

I really liked Christopher Webber's talk about Lisp machines! I thought it was a really informative history lesson about this sort of alternate reality of what our desktops could have become—with a lot of good research and guest star Gerald Sussman! Plus he gave it entirely in Emacs.

—Noah Swartz

Rock and roll bands and free software: A comparative analysis by Pamela Chestek[19]

Pam Chestek's talk had stories, music and legal drama! So great! Bands and free software projects aren't so different. We have so much to learn

[12] Social Web Working Group charter: `https://u.fsf.org/287`

[13] `https://u.fsf.org/252`

[14] Diaspora: `https://podupti.me/`, MediaGoblin: `https://u.fsf.org/288`

[15] `https://u.fsf.org/27w`

[16] `https://u.fsf.org/27x`

[17] `https://u.fsf.org/27y`

[18] `https://u.fsf.org/27z`

[19] `https://u.fsf.org/27-`

from other people who started out doing something for love but one day found themselves doing it (at least partially) for money. Plan for success and register your trademarks!

—Deborah Nicholson

Meet them where they are: Free software and social justice today by Brett Smith[20]

I loved Brett's talk on what we're really asking users to do when we recommend free software. Software supply chains are hard but important. Security and software freedom should be synonymous, but when they aren't? Our community has work to do.

—Deborah Nicholson

A fully-free cell phone experience, no baseband required by Denver Gingerich[21]

I like Denver's talk about making the entire cell phone experience as free as possible. The whole project is actually more accessible than I thought. It's still at a "hackers only" stage, for sure, but it's easier to get started than I realized, and maybe even more importantly, he showed a lot of incremental steps you can take to get more free software on your cell phone without completely writing off today's networks.

—Brett Smith

Running a TV channel with free software by Zeeshan Hasan[22]

TV is not dead as some of us would like to believe. It is alive and aggregated to a point of absurdity. The monopolistic entities in control now must be challenged and thwarted by independent sources for news and information. Free TV projects are necessary for people to have control over freedom of information and autonomy as we are ruled by the information we receive. Zeeshan shows one important way that we can turn that around.

—Micky Metts

Free software & the law: A lighthearted trip down memory lane by Robinson Tryon[23]

Robinson has put into words the obvious elephant in the room. How could we spend hours, years and decades writing code and never paying heed to the laws that bind us? The free software community is fortunate to have such forward looking people focused on the laws surrounding software use and licensing. Most people never read the license of a product or service they use. This session should inspire more people to become lawyers for good.

—Micky Metts

We would like to thank Brett, Carol, Deb, Micky, Noah, and Shauna for sharing their LibrePlanet recommendations with us. There are dozens of other great presentations, including keynotes by Kade Crockford, Richard M. Stallman, Cory Doctorow, and Sumana Harihareswara, all of which are available at `media.libreplanet.org`.

Edits made for grammar and clarity.

FSF infrastructure upgrades

By Andrew Engelbrecht
Web Developer

Over the last eight months, the FSF tech team has been upgrading our physical infrastructure and

[20] `https://u.fsf.org/280`
[21] `https://u.fsf.org/281`
[22] `https://u.fsf.org/282`
[23] `https://u.fsf.org/283`

software to more effectively serve the free software community and help others do so as well.

Thanks to a generous $40,000 donation, we are migrating to Librebooted KGPE-D16 motherboards with 32 cores per board and are adding 4-channel, 10 GBit Ethernet to speed up our new Ceph (distributed storage) systems.[24] We are also migrating to libvirt, which offers an awesome interface to the KVM features in the kernel Linux. Senior systems administrator Ruben Rodriguez made a cool hack that lets us boot MBR-free filesystems with a custom reusable GRUB image, a method inspired by the Xen paravirtualization system.

Our new infrastructure will propel development within and beyond GNU by increasing the much-needed storage space of the GNU Savannah software collaboration system, which hosts over three thousand GNU and non-GNU projects; and by massively boosting Savannah's core count, which will improve the performance of interactions with high-demand source code repositories. We will also migrate many of our virtual machines—including `libreplanet.org`, the Free Software Directory, and our CiviCRM + SQL instance—to this new infrastructure for improved performance, fault-tolerant, high speed data storage, and the ability to perform live migrations of virtual machines.

We're also in the process of upgrading very old servers to Trisquel 7 and 8, which I hope will be officially released quite soon.

[24] An RYF certified version of these motherboards is available from Vikings `https://vikings.net`

We updated our staff and member ejabberd (XMPP) servers, which facilitate decentralized instant messaging, hardened SSL configurations, and optional end-to-end encryption. We migrated our FSF staff StatusNet server to GNU social, a decentralized short message system, which interoperates with other GNU social servers and the popular Mastodon platform. We updated MediaWiki, the software that powers Wikipedia, on our `libreplanet.org` and `directory.fsf.org` sites. We also upgraded our internal instance of Request Tracker, used by FSF staff and many volunteers.

The FSF doesn't work alone; we receive much help from volunteers who maintain the servers that comprise Savannah and other systems, such as `gnu.org`. We also benefit greatly from the programmers, documentation writers, packagers, and artists who work on GNU and non-GNU, and whose generous efforts everyone is free to make use of.

An important part of the FSF's role in the world is to demonstrate to other nonprofits our ability to run exclusively free software on Trisquel-based, Librebooted, self-hosted systems. Although our technical team is small, we are able to deploy and maintain a large array of services that we happily use on a regular basis. We do this for greater autonomy, full control over our systems, and to make great use of the awesomeness that is free software. I hope that we inspire you to do so as well. :-)

Mastodon interview

By John Hsieh
Deputy Director

This past April, the federated social network Mastodon exploded, rapidly gaining hundreds of thousands of users across more than a thousand instances of the platform. These instances are being run independently across personal and public servers—a benefit of a federated social network.

Recently, the FSF had an opportunity to interview Eugen Rochko over e-mail. To learn more about Mastodon, you can visit `mastodon.social`.

Can you tell us a bit about yourself?

I am a recent graduate from Friedrich-Schiller-Universität Jena, where I studied computer science. I am German, of Jewish/Russian origin.

What inspired you to create Mastodon?

I was disappointed with Twitter, and have a love for free software.

Can you tell us a bit about the technical side of Mastodon?

It's made with Ruby and JavaScript. It uses Ruby on Rails as a framework, and React.js as well.

Who contributes and how are they organized?

Officially, the Mastodon team is just me (main developer, founder) and `@maloki@mastodon.social` (project manager). Everyone else is on a volunteer basis—according to GitHub there are 323 different contributors as of 5/13/2017.[25] There are only about a dozen regular contributors; most of them have been given write access to the repository, which allows them to authoritatively review pull requests. But only I and one other person can merge into the master branch.[26]

How/why did you choose the GNU Affero General Public License version 3 for Mastodon?

Originally I started with the GPL, because I was familiar with it from other projects like Discourse, a free online discussion platform that can be used as a mailing list, online forums, or chat rooms. It was suggested that I change to the AGPLv3 to prevent the XMPP/gTalk/WhatsApp situation, and I found that point compelling. To preserve federation, AGPLv3 was chosen.[27]

How does Mastodon relate to GNU social?

Mastodon is an OStatus application, just like GNU social.[28] They are both part of the same network ("fediverse") based on this protocol.

What kinds of technical and/or social challenges did you experience during development?

Technical challenges have included a rush for large-scale optimization during the activity explosion and pinpointing bugs in a distributed networking environment.[29] We have also been adjusting to people's expectations of how things should work.

[25] Mastodon has 3,031 commits by 335 contributors on GitHub (5/24/2017). The Patreon is supported by 727 individuals.

[26] Write access grants contributors a number of permissions, including the creation of repositories, the ability to review pull requests, and manage various reported issues, project boards, and team repos.

[27] To read more on licensing and federation: `https://u.fsf.org/27u`

[28] OStatus is a standard for distributed status updates and includes a number of protocols. Microblogging applications using the same protocol are able to talk with one another across instances and even specific software.

[29] Following the early-April publication of several articles on Mastodon, user numbers went from 20,000 to 42,000 over two days.

How many Mastodon users are there today?

Today Mastodon has over 620,000 users on over 1,200 instances. These numbers are available on `instances.mastodon.xyz/list`.
I do not track any other specific stats, but any time I look there are about 6,000 users accessing `mastodon.social` at the same time (this includes websockets connections of online users).

We'd like to thank Eugen for taking the time to do the interview, as well as thank the entire Mastodon team for their efforts. If you have suggestions for future interview candidates, email `campaigns@fsf.org`.

Responses edited for content and clarity.

On the road with RMS

By Jeanne Rasata
Assistant to the president

Richard M. Stallman, president of the Free Software Foundation, is known around the world as RMS, the founder of the free software movement. He continues to travel, speaking to diverse audiences. Here is a little bit about what he's been up to between November 2016 - May 2017.

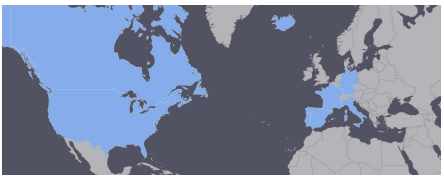
In the past six months, Richard has...

...visited 29 cities.

...given 32 talks.

...participated in 1 panel.

Where has RMS been?



Richard has been to schools, conferences, and organizations like UNESCO across Canada, France, Germany, Iceland, Italy, Portugal, Spain, and the United States.

Talks given include:

- El software libre en la administración
- El software libre y tu libertad
- Free Software, Your Freedom, Your Privacy
- A Free Digital Society
- Le logiciel Libre, la conception libre du matériel
- Gouvernance et regulation de la securité numérique: Quel role pour chacun?
- Free Software, Your Freedom and Medicine

Please write to `rms-assist@fsf.org` with any photographs you would like us to share on RMS's blog, at `fsf.org/blogs/rms`; with recordings of his speeches for our audio-video archive `audio-video.gnu.org`; or to extend a speaking invitation to RMS. See `u.fsf./org/zi` for a list of his confirmed engagements.

Donate to the FSF with Bitcoin.

Published twice yearly by the Free Software Foundation, 51 Franklin Street, 5th Floor, Boston, MA 02110-1301, (617) 542-5942.
info@fsf.org, fsf.org

How to Contribute

Associate Membership: Become an associate member of the FSF. Members will receive a bootable USB card, e-mail forwarding, and an account on the FSF's Jabber/XMPP server. To sign up or get more information, visit member.fsf.org or write to membership@fsf.org.

Online: Use your credit card or PayPal account to make a donation at donate.fsf.org or contact donate@fsf.org for more information on supporting the FSF.

Jobs: List your job offers on our jobs page: fsf.org/jobs.

Free Software Directory: Browse and download from thousands of different free software projects: directory.fsf.org.

Volunteer: To learn more, visit fsf.org/volunteer.

LibrePlanet: Find local groups in your area or start your own at libreplanet.org!

Free Software Supporter: Receive our monthly email newsletter: fsf.org/fss.