



# Open Source: Better for Companies, Communities and Developers

Ric Wheeler  
[ricwheeler@gmail.com](mailto:ricwheeler@gmail.com)

January 2019

# Overview

- Some definitions of “open”
- Early days in open source
- Why is open source good for business
- Why is open source good for communities
- Why is open source better for you as a developer
- Questions



# What is Open Source?

# What Many People think about Open Source



# What Does Open Mean?

- Open is a very popular term
  - Used to describe standards
  - Used to name formats
  - Used to make companies sound progressive and trendy!
- Open standards provide for interoperability between different implementations
  - Not always a promise that the implementations are open source
  - Not always a promise to share IP like patents
  - Examples include W3C (<https://www.w3.org/>) or IETF (<http://ietf.org/>)
- This talk is about the software that uses open source licenses

# What is Open Source?

- This talk is focused on “open source” software licenses
  - Open source is built around a legal license to use code
  - Different licenses give different rights and have various requirements
  - <https://opensource.org/licenses> has a list of 83 common licenses
  - copyright is critical: it expresses ownership which gives the right to license code
- Open source provides users and developers the freedom to see, modify and share the code and executables
  - Permissive licenses like BSD let you embed code in non-open source work
  - GPL style licenses require sharing of the code if you ship product
  - Examples at: (<https://opensource.com/resources/what-open-source>)

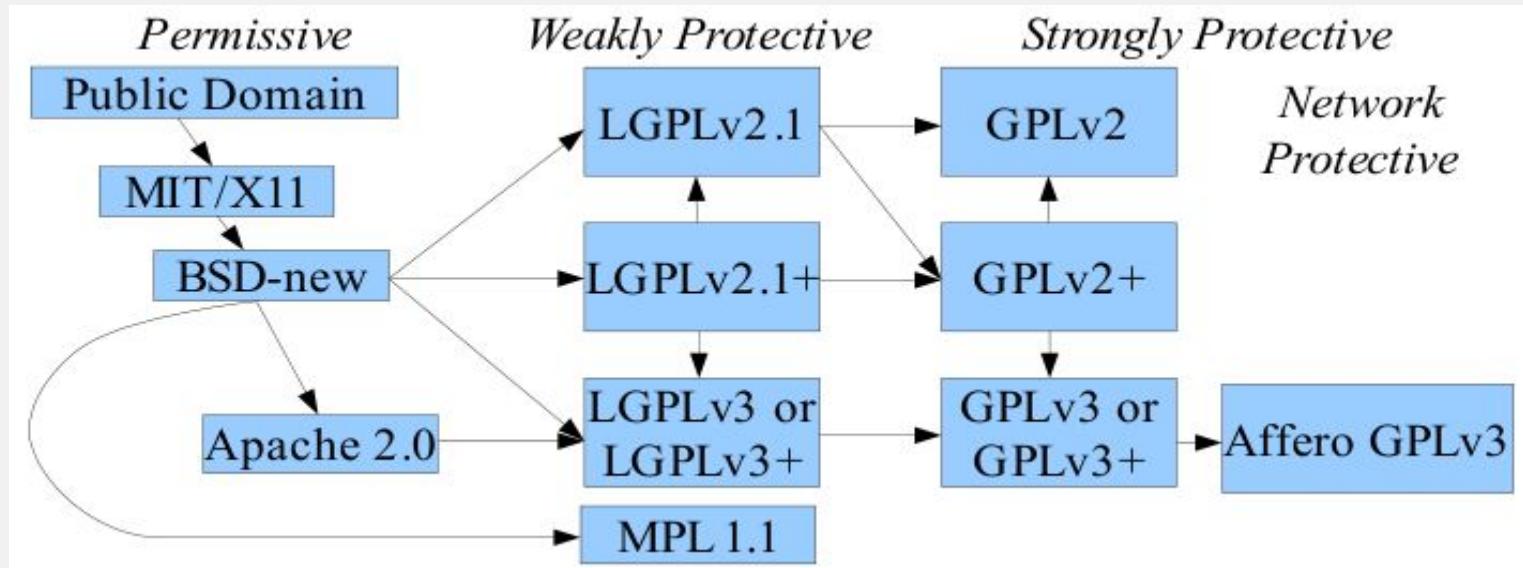
# Examples of Open Source Licenses

- Ugandan Orphans License - used by VIM
  - <https://www.gnu.org/licenses/vim-license.txt>
  - *Vim is Charityware. You can use and copy it as much as you like, but you are encouraged to make a donation for needy children in Uganda.*
- GPLv2
  - <https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>
  - Requires distribution of the code if you ship the binary, Linux kernel license
- GPLv3
  - <https://www.gnu.org/licenses/gpl.html>
  - Used by samba and other projects
  - Attempts to close loopholes in GPLv2

# Examples of Permissive Open Source Licenses

- BSD or MIT style license
  - <https://opensource.org/licenses/BSD-3-Clause>
  - Pretty free to use in any way you want
  - Don't blame the authors - use at your own risk
  - Can be bundled in non-open source projects
- Apache Public License
  - <https://www.apache.org/licenses/LICENSE-2.0>
  - Similar to BSD, but adds some implied grant for patents

# Open Source License Chart



<https://www.dwheeler.com/essays/floss-license-slide.html>

# License Compatibility

- All open source licenses are not equivalent
  - Important to understand the license when you build a project
  - Common licenses may not be combined
- GPLv2 is not compatible with GPLv3 for example!
  - Protective licenses tend to restrict combinations more than permissive licenses
  - Trade off is between you preserve end user rights versus ease of combination
- Talk to a lawyer!

# How Do You Choose a License for a New Project?

- Talk to a lawyer
  - Do not rely on a google search or suggestions from friends (or me!)
- Talk to your target community of developers
  - Developers can be passionate about licenses
  - Your license choice can encourage or discourage others
- Think about what you need to combine your project with
  - Is it a library or service or will it be embedded in another bit of code?
- Think about tracking contributions
  - You cannot change a license without permission from your developers
  - Contributor agreements can be a shortcut here, but can depress contributions

# How to Cause Hackers to Argue

- License choice:
  - Debate the merits of BSD (permissive) versus GPL
  - Right answer: GPL!
- Best editor for hacking
  - Debate the merits of VI versus emacs
  - Right answer: emacs!
- Choice of programming language
  - Debate C versus C++ versus Golang
  - Right answer: C!



# Early Days in Open Source

# Jump Back to the 1980's



# Ancient Days of Early UNIX



- Lots of access to UNIX from ATT and UC Berkeley
  - Needed a signed license from ATT
  - Non-commercial use only
  - Popular at universities
- I worked on UNIX first in 1981
  - VAX 11/780 ran at 1 MIP
  - Had 4MB of DRAM
  - 2 disks 300 MB each (cabinet sized)
  - 80 students ran on each machine

# UNIX in the 1980's

- AT&T UNIX
  - Source code was well studied
  - No one outside of AT&T could write about UNIX algorithms or share changes
  - BSD code changes could be shared
- Some commercial companies shipped UNIX variations
  - SUN, HP, Digital, SGI, IBM, Prime, etc
  - Code was not portable between versions
  - Mostly ran on different architectures
- No free version of UNIX for home users or enthusiasts

# Software Sharing Predates the GPL



- Richard Stallman was still just hacking at MIT
  - GNU Manifesto 1985 -  
[https://en.wikipedia.org/wiki/GNU\\_Manifesto](https://en.wikipedia.org/wiki/GNU_Manifesto)
  - GNU project focused on user space
  - Image from  
<https://stallman.org/saintignucius.jpg>
- Would take until 1989 for the first version of the GPL to be released
  - <https://www.gnu.org/copyleft/copying-1.0.html>

# Some Truth in those “hippie” Worries

- Some of us did have too much hair
- Most UNIX server companies bundled operating systems with hardware
  - Was a cost of doing business
- Open Software Foundation founded in 1988 tried to make a common UNIX for companies



# Conferences & Books: First Step to Freeing Code

- USENIX association conferences and events
  - Started in 1975 - <https://www.usenix.org/about/history>
  - Home of historic innovation in UNIX and UNIX-like systems
  - Free exchange of deep technical papers and talk
- The Design of the UNIX Operating System by Maurice Bach, 1986
  - Based on AT&T internal courses, described in technical depth the algorithms
  - Paid two young graduate students to write the teacher's "answer" set
- Other researchers wrote books on operating systems internals
  - Operating Systems: Design and Implementation by Andrew Tanenbaum, 1987
  - The Design and Implementation of the 4.3 BSD UNIX Operating System, 1989
  - The MOSIX Distributed Operating System, 1993

# Sharing UNIX Code

- Berkeley Computer System Research Group
  - Home of BSD distributions
  - Worked in the late 80's to remove AT&T source code from BSD
  - Release Net/1 in 1989, mostly full version released (Net/2) in 1991
  - Lawsuit with AT&T stalled the BSD evolution until settled in 1994
- Growth of Linux
  - Kicked off by Linus Torvalds in 1991
  - Shared ideas and code early on with MINIX and the GNU projects
  - Did not use BSD code base and largely avoided the legal worries
  - Started as a fun toy like experiment, 27 years on runs everywhere!
  - <https://arstechnica.com/information-technology/2015/08/how-linux-was-born-as-told-by-linus-torvalds-himself/>

# Early 1990's Restrictions on Open Source

- Major companies banned the use of open source code and tools
  - No use for gcc for example
  - Certainly no open source licensed code
- Non-open source UNIX vendors dominated serious computing
  - Saw Linux and \*BSD as toys
  - Fought each other more than open source or Microsoft
- Open Software Foundation's OSF1 shipped
  - Standard version of UNIX
  - Hated by SUN and AT&T
  - Home of the Mach microkernel (still around in OSX for Apple users)!
  - Proprietary license common



# Roll Forward to Today...





# Why is Open Source Good for Business?

# Open Source Business Models

- Companies like Red Hat or SUSE favor pure open source models
  - Customers can get all of the source code
  - Competitors can use your code in their own products
  - Lack of “lock in” to one implementation
- “Open Core” model
  - The heart of the software is built around open source components
  - One critical component is kept as proprietary
- Traditional Proprietary Tech Firms
  - Use Open Source as a base for a proprietary system or application

# Open Source Software used in Enterprises Today

- Linux platforms and open source applications are everywhere
  - Rare to see a firm that does not use open source for mission critical applications
  - Open source provides quality, enterprise strength platforms and tools
- Businesses as customers have choice
  - Can hire technical staff to deploy and manage open source directly
  - Can purchase open source from one or more vendors
  - Can hire consultants or services
- Drives down costs of software for end users and customers

# Even Wall Street Loves Open Source!



# Open Source Components in Proprietary Systems

- Open source tools are commonly used to develop software
  - Compilers, debuggers, etc
  - Open source platforms like Linux are common development platforms
- Many “proprietary” enterprise storage systems are built on open source platforms
  - Big companies like Dell EMC have moved a lot their software onto Linux
  - Some non-open source, key components of course exist like drivers, applications and management applications
  - Critical business need to make sure that their systems inter-operate well with Linux clients
- Great way to reduce the cost of engineering and product production
  - Focus your engineering investment strategically on your special technology

# Pure Play Open Source Companies

- A relatively few brave companies have only open source code in their products
  - Red Hat for example re-licenses newly code from acquired companies
  - Gives away technology they paid money for to everyone - including competitors and potential customers
- Why do customers pay for open source products?
  - Most customers want to buy a product that has quality and features
  - No desire to build, patch and test software themselves - they make money doing something else!
  - Open source does allow flexibility to go it on your own or switch vendors without “lock in”
  - Access to the lead developers and architects of the product

# Open Source Companies Efficiencies

- Many open source projects have contributions that span the globe
  - You can focus your efforts on what is most critical to you and your market
  - Leverage less strategic code from the project and others who might have a different focus
  - Referred to occasionally as “Scratch your own itch!”
- Customers can try easily before purchasing
  - Very little barrier to entry for potential customers
  - Can provide options for revenue through consulting and contracting
  - Possibly can be migrated to a firm’s commercially supported product

# Open Source Encourages Innovation

- Start up companies can build on a huge base of sophisticated, powerful code
  - Enables rapid innovation to explore a specific niche or additional feature
  - No need to start from scratch and rebuild the world before starting to leverage your idea!
- Open source is common in universities and education
  - Students and researchers get to use real, production systems
  - Researchers can modify those systems without fee
  - Freedom to share results and code
  - Freedom to just write about systems, design and performance results



# Why is Open Source Good for Communities?

# Communities are About Togetherness



# Communities

- We use the word “community” to describe
  - Can be a group of people in the same region
  - “Hippies” from the 1960’s and 1970’s who lived in the same place
  - A group with a shared passion for a sports team, hiking or a project
- Technical communities historically mapped to vendor specific events
  - Microsoft customers or developers
  - Customers of large companies like IBM or EMC
  - Enthusiasts for trains, radios, etc



# Open Source Communities

- Open source has created very fertile ground for a myriad of communities!
- No cost to belong - all code is open and shared
  - Anyone can evaluate or study it
  - No “walls” around key infrastructure technology
  - Easy to lurk at the edges by reading upstream mailing lists
  - Easy to make initial contributions around testing & suggestions
  - Easy to contribute directly and publically to help the community evolve
- This is a unique kind of technical community
  - The members of the community control and guide it
  - Not beholden to the large corporation or politicians

# Building Open Source Communities

- Every open source community has a unique feel and way of doing things
  - When we do this well, the result is a lot of diversity and choice
  - When we do it poorly, we can limit the appeal of our community
- It does take a lot of work to build communities properly
  - Encourage new members of the community with kindness and advice
  - Focus on technical feedback rather than personal attacks
  - Keep reviewing and improving your community rules and guidelines
  - Learn from our sister communities when they do something well or poorly!

# LSF/MM Kernel Community





# Why is Open Source Good for Developers?

# How Can Open Source Help Me?



# Open Source Gives Real Software to Students

- Students can use and contribute to real, enterprise class projects
  - Can start at an early age as users
  - Serious contributions can start at an early age, even before university
- Can empower people who do not follow traditional education paths
  - Some leaders in open source are self taught
  - University is great but not the only path to learning!
- Open source enables adult learners to explore and expand their career options
  - No one on the upstream lists knows or cares what your academic history is!

# Open Source Gives Working Engineers Control

- Access to enterprise, world class software without walls
  - Let's anyone start using and contributing
  - Can start without any experience and with very little resources
- Empowers you as an individual to advance your career
  - Build your reputation by contributing to a community in your own way
  - You can point any potential employer at your public contributions
  - Reduces worries about violating NDA's or employee agreements with open source
- Employers can find you by searching the upstream mailing lists and commit logs
  - Your public record can work to find you a job before you even start looking!

# Individuals Can Launch Projects and Businesses

- When you identify a need or interesting new area
  - Open source allows you to plant the seeds of your idea out in public
  - It might well start & end as a hobby, but just might grow into something awesome
- Individuals can take community projects and build up a business
  - Common to start with consulting or support for commercial users
  - Can spin out a new idea into a business around the technology
  - Need to follow the rules of the license and play well with the community!
- Can form the basis for critical infrastructure for empowering social change
  - Open source certainly makes enterprise software more affordable
  - Some social change can be for good or for not so good!

# Open Source Can Bring You To Exotic Places!

