Open Source- Legal Strategy – Avoiding Legal Pitfalls

BIJU K NAIR

Advocate

Do you know any of these companies.

- Panasonic
- Cisco
- VMWare
- Samsung
- Verizon

Example of Violations Litigation

- CoKinetic Systems Pursues $100 Million GPL License Violation Case Against Panasonic Avionics
- Cisco sued for Linksys GPL violation.
- Linux developer sues VMware over GPL violations.
- Best Buy, Samsung, Westinghouse, And Eleven Other Brands Named In SFLC Lawsuit. Evidence of GPL Violations and Copyright Infringement Found in TVs, DVD Players, and Dozens of other Electronic Devices

Case & License Violated

- Cisco
- Verizon
- Samsung
- Vmware

- GNU General Public License" version 2 ("GPL"), and the "GNU Lesser General Public License" (also versions 2 and 2.1 ("LGPL")
- GPL v2
- GPLv2
- GPLv2.

Open Source: Internal Supply Chain/ TEAM WORK

Developers and engineers

In addition to software developers, hardware engineers are deeply involved in developing device driver software, board support packages (BSP) and software development kits (SDKs) for their hardware.

Procurement personnel

OSS may be included in deliverables from the supply chain, such as software, hardware modules, SoCs, semiconductor products, and products designed and developed by ODM/OEM manufacturers.

Sales personnel

Sales personnel are required to understand the reasons that customers need the OSS-related information, including copyright and license information.

Open Source: Internal Supply Chain/ TEAM WORK

Quality assurance personnel

OSS that is included in a product may affect its quality or introduce bugs. QA personnel need to be aware of such issues.

Legal/Intellectual Property personnel

Legal and intellectual property personnel are required to know the laws, legal precedents, and legal remedies that relate to OSS license interpretation and adherence

Executives and managers

Executives and managers develop strategyaround using, contributing to, and distributing Open Source; build teams to promote OSS usage; and oversee OSS processes, and investment in required software tools.

Open Source License

Open Source licenses by definition make source code available under terms that allow for modification and redistribution

Open Source licenses may have conditions related to providing attributions, copyright statement preservation, or a written offer to make the source code available

One popular set of licenses are those approved by the Open Source Initiative (OSI) based on their Open Source Definition (OSD). A complete list of OSI-approved licenses is available at http://www.opensource.org/licenses/

Permissive Open Source Licenses

Permissive Open Source license: a term used often to describe minimally restrictive Open Source licenses

Example: BSD-3-Clause

The BSD license is an example of a permissive license that allows unlimited redistribution for any purpose in source or object code form as long as its copyright notices and the license's disclaimers of warranty are maintained

The license contains a clause restricting use of the names of contributors for endorsement of a derived work without specific permission

Other examples: MIT, Apache-2.0

License Reciprocity & Copyleft Licenses

- Some licenses require that if derivative works (or software in the same file, same program or other boundary) are distributed, the distribution is under the same terms as the original work
- This is referred to as a “copyleft” or “reciprocal” effect
- Example of license reciprocity from the GPL version 2.0:
- *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 […] under the terms of this License.*
- Licenses that include reciprocity or Copyleft clauses include all versions of the GPL, LGPL, AGPL, MPL and CDDL

Proprietary License or Closed Source

- A proprietary software license (or commercial license or EULA) has restrictions on the usage, modification and/or distribution of the software
- Proprietary licenses are unique to each vendor – there are as many variations of proprietary licenses as there are vendors and each must be evaluated individually
- Open Source developers often use the term “proprietary” to describe a commercial non-Open Source license, even though both Open Source and proprietary licenses are based on intellectual property and provide a license grant to that property

Other Non-Open Source Licensing Situations

- Freeware – software distributed under a proprietary license at no or very low cost
  - The source code may or may not be available, and creation of derivative works is usually restricted
  - Freeware software is usually fully functional (no locked features) and available for unlimited use (no locking on days of usage)
  - Freeware software licenses usually impose restrictions in relation to copying, distributing, and making derivative works of the software, as well as restrictions on the type of usage (personal, commercial, academic, etc.)
- Shareware – proprietary software provided to users on a trial basis, for a limited time, free of charge and with limited functionalities or features
  - The goal of shareware is to give potential buyers the opportunity to use the program and judge its usefulness before purchasing a license for the full version of the software
  - Most companies are very leery of Shareware, because Shareware vendors often approach companies for large license payments after the software has freely propagated within their organizations.

Other Non-Open Source Licensing Situations

- “Non-commercial” – some licenses have most of the characteristics of a Open Source license, but are limited to non-commercial use (e.g. CC-BY-NC).
  - Open Source by definition cannot limit the field of use of the software
  - Commercial use is a field of use so any restriction prevents the license from being Open Source

(Source: https://www.openchainproject.org/resources)

License Compatibility

- License compatibility is the process of ensuring that license terms do not conflict.
- If one license requires you to do something and another prohibits doing that, the licenses conflict and are not compatible if the combination of the two software modules trigger the obligations under a license.
  - GPL-2.0 and EPL-1.0 each extend their obligations to “derivative works” which are distributed.
  - If a GPL-2.0 module is combined with an EPL-1.0 module and the merged module is distributed, that module must
    - (according to GPL-2.0) be distributed under GPL-2.0 only, and
    - (according to EPL-1.0) under EPL-1.0 only.
  - The distributor cannot satisfy both conditions at once so the module may not be distributed.
  - This is an example of *license incompatibility.*
- The definition of “derivative work” is subject to different views in the Open Source community andits interpretation in law is likely to vary from jurisdiction to jurisdiction.

Notices

- Notices, such as text in comments in file headers, often provide authorship and licensing information. Open Source licenses may also require the placement of notices in or alongside source code or documentation to give credit to the author (an attribution) or to make it clear the software includes modifications.
- **Copyright notice** – an identifier placed on copies of the work to inform the world of copyright ownership. Example: Copyright © A. Person (2016)
- **License notice** – a notice that specifies and acknowledges the license terms and conditions of the Open Source included in the product.
- **Attribution notice** – a notice included in the product release that acknowledges the identity of the original authors and / or sponsors of the Open Source included in the product.
- **Modification notice** – a notice that you have made modifications to the source code of a file, such as adding your copyright notice to the top of the file.

What Compliance Obligations Must Be Satisfied?

- Depending on the Open Source license(s) involved, your compliance obligations may consist of:
- **Attribution and Notices.** You may need to provide or retain copyright and license text in the source code and/or product documentation or user interface, so that downstream users know the origin of the software and their rights under the licenses. You may also need to provide notices regarding modifications, or full copies of the license.
- **Source code availability.** You may need to provide source code for the Open Source software, for modifications you make, for combined or linked software, and scripts that control the build process.
- **Reciprocity.** You may need to maintain modified versions or derivative works under the same license that governs the Open Source component.
- **Other terms.** The Open Source license may restrict use of the copyright holder name or trademark, may require modified versions to use a different name to avoid confusion, or may terminate upon any breach.

Intellectual Property Pitfalls

TYPE & DESCRIPTION	DISCOVERY	AVOIDANCE
Unplanned inclusion of copyleft Open Source into proprietary or 3rd party code: This type of failure occurs during the development process when engineers add Open Source code into source code that is intended to be proprietary in conflict with the Open Source policy.	This type of failure can be discovered by scanning or auditing the source code for possible matches with: • Open Source source code • Copyright notices Automated source code scanning tools may be used for this purpose	This type of failure can be avoided by: • Offering training to engineering staff about compliance issues, the different types of Open Source licenses and the implications of including Open Source in proprietary source code • Conducting regular source code scans or audits for all the source code in the build environment.

Intellectual Property Pitfalls

Type & Description	Discovery	Avoidance
Unplanned linking of copyleft Open Source and proprietary source code: This type of failure occurs as a result of linking software with conflicting or incompatible licenses. The legal effect of linking is subject to debate in the Open Source community.	This type of failure can be discovered using a dependency tracking tool that shows any linking between different software components.	This type of failure can be avoided by: 1. Offering training to engineering staff to avoid linking software components with licenses that conflict with you Open Source policies which will take a position on these legal risks 2. Continuously running the dependency tracking tool over your build environment
Inclusion of proprietary code into copyleft Open Source through source code modifications	This type of failure can be discovered using the audits or scans to identify and analyze the source code you introduced to the Open Source component.	This type of failures can be avoided by: 1. Offering training to engineering staff 2. Conducting regular code audits

License Compliance Pitfalls

Type & Description	Avoidance
Failure to Provide Accompanying Source Code/appropriate license, attribution or notice information	This type of failure can be avoided by making source code capture and publishing a checklist item in the product release cycle before the product becomes available in the market place.
Providing the Incorrect Version of Accompanying Source Code	This type of failure can be avoided by adding a verification step into the compliance process to ensure that the accompanying source code for the binary version is being published.
Failure to Provide Accompanying Source Code for Open Source Component Modifications	This type of failure can be avoided by adding a verification step into the compliance process to ensure that source code for modifications are published, rather than only the original source code for the Open Source component

License Compliance Pitfalls

Type & Description	Avoidance
Failure to mark Open Source Source Code Modifications: Failure to mark Open Source source code that has been changed as required by the Open Source license (or providing information about modifications which has an insufficient level of detail or clarity to satisfy the license)	This type of failure can be avoided by: 1. Adding source code modification marking as a verification step before releasing the source code 2. Offering training to engineering staff to ensure they update copyright markings or license information of all Open Source or proprietary software that is going to be released to the public

Compliance Process Failures

Description	Avoidance	Prevention
Failure by developers to seek approval to use Open Source	This type of failure can be avoided by offering training to Engineering staff on the company's Open Source policies and processes.	This type of failure can be prevented by: 1. Conducting periodic full scan for the software platform to detect any "undeclared" Open Source usage 2. Offering training to engineering staff on the company's Open Source policies and processes 3. Including compliance in the employees performance review
Failure to take the Open Source training	This type of failure can be avoided by ensuring that the completion of the Open Source training is part of the employee's professional development plan and it is monitored for completion as part of the performance review	This type of failure can be prevented by mandating engineering staff to take the Open Source training by a specific date

Compliance Process Failures

Description	Avoidance	Prevention
Failure to audit the source code	This type of failure can be avoided by: 1. Conducting periodic source code scans/audits 2. Ensuring that auditing is a milestone in the iterative development process	This type of failure can be prevented by: 1. Providing proper staffing as to not fall behind in schedule 2. Enforcing periodic audits
Failure to resolve the audit findings (analyzing the "hits" reported by a scan tool or audit)	This type of failure can be avoided by not allowing a compliance ticket to be resolved (i.e. closed) if the audit report is not finalized.	This type of failure can be prevented by implementing blocks in approvals in the Open Source compliance process
Failure to seek review of Open Source in a timely manner	This type of failure can be avoided by initiating Open Source Review requests early even if engineering did not yet decide on the adoption of the Open Source source code	This type of failure can be prevented through education

Solutions: Open Chain :Vision & Mission

Vision

A software supply chain where free/open source software (FOSS) is delivered with trusted and consistent compliance information.

Mission

Establish requirements to achieve effective management of free/open source software (FOSS) for software supply chain participants, such that the requirements and associated collateral are developed collaboratively and openly by representatives from the software supply chain, open source community, and academia;

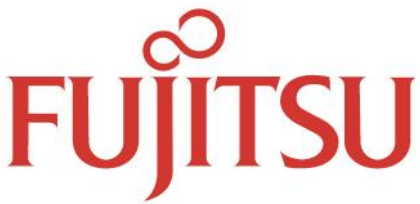
ii) Establish a governance model that provides long-term support for the current work of the OpenChain work group and enables OpenChain to grow into an effective and reliable software supply chain standard; and

iii) Develop and implement specific projects (including, for example, training and educational programs) to further the goals of the Project.

What is the OpenChain Curriculum?

- The OpenChain Project helps to identify and share the core components of a Free and Open Source Software (Open Source) compliance program.
- The core of the OpenChain Project is the **Specification**. This identifies and publishes the core requirements a Open Source compliance program should satisfy.
- The OpenChain **Curriculum** supports the Specification by providingfreely available training material.
- These slides help companies satisfy the requirements of the Specification Section 1.2. They can also be used for general compliance training.
- Learn more at: https://www.openchainproject.org

The Companies behind OpenChain




AB EHR Digital
A to B
Electronic Health Records


B2M
SOLUTIONS
ANALYTICS. INSIGHT. RESULTS.


Cognizant


credativ


>_ ENDOCODE


GE Digital


Google


HARMAN


Hella Aglaia


HITACHI
Inspire the Next


Infosys


Interneuron


LYRA
Infosystems


Nextcloud


NODEWEAVER


PELAGICORE
Experience Change


Qualcomm
Qualcomm
Technologies, Inc.


SCANIA


SIEMENS


SOURCE
Auditor
"Keeping Your Source Code Yours"


Source Code Control
Open Source Risk Management Specialists


SUSE


CRD
the Center for Research and
Development Hong Kong


Tjaldur Software Governance Solutions


TogánLabs


ura


WIND

GPL Cooperation Commitment

Leading companies, developers, and other leaders in the open source community who have all committed to provide GPLv2 and LGPLv2.x licensees a fair chance to correct violations before their licenses are terminated

Goal:

reduce opportunities for abusive enforcement tactics and, more broadly, to promote greater predictability in the enforcement of GPLv2 and LGPLv2.x licenses.

increase participation in the use and development of open source software by helping to ensure that enforcement, when it takes place, is fair and predictable

GPL Cooperation Commitment

The “automatic termination” feature of GPLv2 and LGPLv2.x does not provide an express “cure” period in the event of a violation. This means that a single act of inadvertent non-compliance could give rise to an infringement claim, with no obligation to provide notice prior to taking legal action. When GPLv3 was introduced in 2007, one of the key improvements was the inclusion of a cure period.

Imbalance in GPLv2 and LGPLv2.x license enforcement, Red Hat, IBM, Google, and Facebook announced in November 2017 a commitment to apply the GPLv3 cure provisions for their GPLv2 and LGPLv2.x licensed software. 40 Companies are present participants.

Open Source & Patent Tension

OPEN INVENTION NETWORK

Overview

MEMBERS

~3300

COMMUNITY MEMBERS FROM **START-UPS** TO **LARGE CORPORATIONS**

CROSS LICENSE POOL

~2.0 million

TOTAL PATENTS AND **APPS OWNED BY OIN LICENSEES**

OIN PATENT PORTFOLIO

~1300

GLOBAL PATENTS AND **APPS WITH BROAD SCOPE**

~$100 million

SPENT ACQUIRING DEFENSIVE PATENTS

LINUX SYSTEM

>2885

OPEN SOURCE **CORE LINUX TECHNOLOGY PACKAGES** COVERED

The Linux System definition comprises core software packages from various open source projects which, taken together, provide the scope of the cross-license obligation to which each community member commits.

A complete and current listing of Linux System packages is available on our website under the Joining OIN tab.

openinventionnetwork®

Enabling Open Source

Who are we?

- We are the world's largest patent non-aggression community, and its largest free defensive patent pool.
- We protect open source through patent collaboration.
- We remove patent friction in core open source technologies, which drives higher levels of innovation.
- By enabling open source software,
  - We advance the collective intelligence of a global community.
  - We promote greater diversity of thought, perspective and talent.

openinventionnetwork®

THE PATENT NON-AGGRESSION COMMUNITY

The Linux Foundation & Project-based Innovation

THE PATENT NON-AGGRESSION COMMUNITY

Member Representation Sampling

NETWORK/ TELECOM

INTERNET

SOFTWARE

NETWORKING

AUTOMOTIVE

THE PATENT NON-AGGRESSION COMMUNITY

Member Representation Sampling

HARDWARE

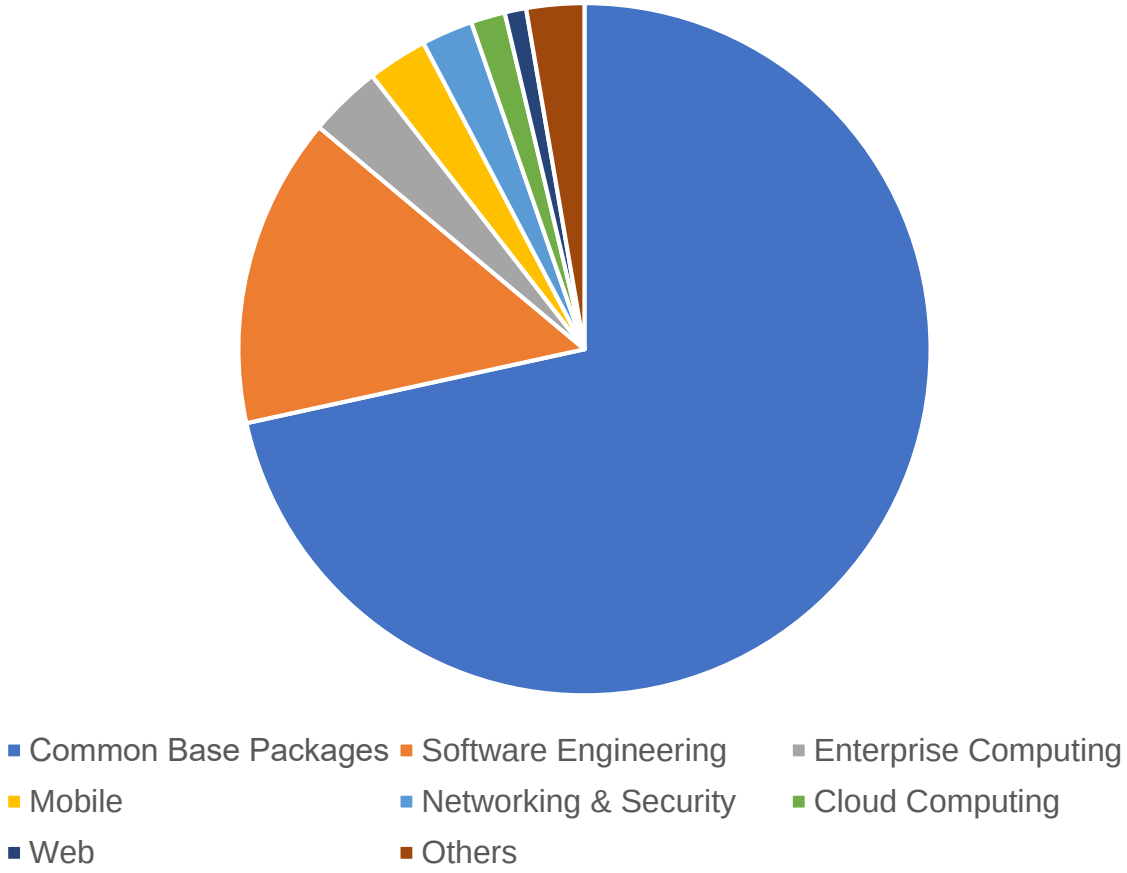
SEMI-CONDUCTOR

FINTECH

INDUSTRIALS

ENERGY

Linux System Overview



Common Base Packages (71.5%) - Linux kernel, system programs, common libraries

Software Engineering (14.5%) - Perl, Python, Go, Lua, etc.

Enterprise Computing (3.4%) - JBOSS, Jakarta, OpenShift (high-functionality packages)

Mobile (2.8%) - Android, Web OS

Networking & Security (2.4%) - OpenSSL, OpenSSH, OpenVPN

Cloud Computing (1.6%) - OpenStack, Qemu, libvirt

Web (1.0%) - Apache web server, NGINX

Others (2.7%) - Container Technologies, Configuration Management, Automotive, Software Development

IoT, ONAP/OPNFV & Hyperledger - Future roadmap of Linux and LF Projects will translate into near term incorporation but given the reuse of code all of these projects are to some degree already represented in the Linux System

Joining Open Invention Network

How can you participate?

- Everyone can join; Everyone signs the same terms
- No barrier to entry; No requirement to hold patents
- **Zero** cost for membership
- No ongoing time commitment
- Everyone makes the same cross-license commitment in the Linux System, as defined on our website
- Members and licensees are listed on our website
- Join now through our electronic signature process online

openinventionnetwork®

Our Role Beyond the License

How Do We Help Our Community Members Against Patent Aggressors?

- We **collect and share prior art** to help community members defend themselves.
- We **sell patents** to OIN community members for defensive purposes.
- We **acquire patents from patent antagonists** asserting Linux-centric patents when this clears patent threats broadly for the OIN community.
- We routinely utilize the AIA's pre-issuance submission program to **limit claim scope of overly broad patent claims** in key technology areas.

Sample Community Benefits

- **Threat Clearing:** We acquired 22 patents that were being marketed for sale as reading on "Open Source Software", thereby clearing this patent threat for its community.

- **Patent Sales:** A community member was asserted against by a Fortune 50 company, acquired patents from us and used our patents to negotiate a far better deal than originally demanded.

- **Membership Alone as a Deterrent:** Facing a major lawsuit from a Fortune 50 company, another entity decided to join our community. After its membership became known, a settlement was reached within 72 hours on terms that were extremely favorable for them.

Sample Community Benefits

- **Prior Art Assistance:** In over 45 cases where one of our members was the subject of claims, we had identified and shared prior art for use against the NPE, or corporate patent aggressors.

- **Patent Acquisitions:** We have expended in excess of $15M to purchase Linux-centric patents owned by NPE's that were being used in active assertions or litigation - 15% of ours circa $100M total patent acquisition investment to date.

- **Claim Scope Reduction:** We are among the leading users of the AIA's pre-issuance submission program, having helped to get more than 25 overly broad patent applications rejected and significantly reducing claim scope in another 40 plus patent applications.

Thank you

bnair@openinventionnetwork.com

91-9711956777