Contribution Policies for Open Source Projects

Richard Fontana

redhat.

LinuxCon 2010

Me

- IAAL, IANYL, TINLA

- Open Source Licensing and Patent Counsel at Red Hat, 2008-Present

- Counsel, Software Freedom Law Center, 2005-2008

Agenda

- Background

- Types of contribution policies

- Controversy over copyright assignment

- Red Hat's experience

- Conclusions

Copyright

- *"subsists ... in original works of authorship fixed in any tangible medium of expression"* (17 USC § 102)

- Right to exclude others from copying, preparing derivative works, distribution (etc.)

- Arises automagically - no formalities

- Lasts a really long time

- Holder can transfer (in writing) or grant licenses

- Enforcement: © infringement, breach of contract

Contribution Policies

Rules governing acceptance/use of inbound contributions

- Open source projects vary in collaborativeness

- Outbound license: terms presented by project to users

- Inbound: © permissions inhering in contributions
  - Limits range of possible outbound licenses

Types

- Inbound = outbound FOSS license

- Copyright assignment

- Formal contributor license agreement (CLA)

- Miscellaneous

Inbound = Outbound

- Majority rule
  - Contributor uses global outbound FOSS license
  - Project passes through contribution FOSS license

- Variance in formality
  - Undocumented, manifest in project custom
  - Informally documented
  - Formal agreement (uncommon)

Copyright Assignment

- Structured as formal agreements

- Prominent examples: FSF, FSFE FLA, Sun CA
  - SCA: joint ownership but no duty to account

- Most notably used by companies with dual-licensing/open core business model

- Uncommon for noncorporate projects

Copyright Assignment

Some common features:

- Grant-back of maximally broad copyright license

- Fallback maximally broad copyright license

- How assignee will use contribution/ derivative work

- Patent license grant

- Reps/warranties: originality/third-party claims

- Attempts to deal with work-for-hire issue

CLAs

- Typically resemble assignment agreements but contributee just gets maximally broad license grant

- Best-known: Apache CLAs
    - ASF commits not to use "contrary to the public benefit or inconsistent with its nonprofit status"
    - Reused by community and corporate projects (e.g. Django, Google, Zend Framework)

- Projects with permissive as well as copyleft outbound

Miscellaneous

- Limit inbound FOSS license choice (non-outbound)
  - CiviCRM; PSF CA

- Lightweight approach to small contributions

- Contributor selects from multiple policies
  - KDE: choose from acceptable licenses or FLA
  - MariaDB: joint © or 3-clause BSD

The Case Against © Assignment

- Perennially controversial

- "My property"; dangers of single-entity control; red tape; evils of dual-licensing/open-core

- Harmful effects of discouraging contributions (community, adoption, forking, re-implementation)

- Some critics distinguish between trusted nonprofits and for-profits

The Case For © Assignment

- Three main claims:
    - Concentrating © ownership facilitates enforcement
    - Facilitates project relicensing
    - Protects project/assignee from third-party claims

- Additional arguments (for-profit assignee)
    - Facilitates dual-licensing → attracts investment
    - Fair for mere patch authors to have less power
    - Residual doubt over scope of broadest © licenses
    - Convince cautious companies to "open source"

Enforcement Argument

- Distributed © ownership impedes enforcement

- Mere nonexclusive license not enough to enforce

Joinder of Necessary Parties

- In infringement suit, *"court may require the joinder ... of any person having or claiming an interest in the ©"* (17 USC § 501(b))

- Rationale: anyone who could grant a license to infringer is necessary party
    - Then assignor with broad grantback license may be necessary party

Standing

- Only © holders can sue for © infringement

- But nonexclusive licensee can hold © on a derivative or collective work

- Fair to expect a company to have contributed something creative in order to enforce a license

What About CLAs?

- © assignment vs. Apache-style CLAs

- Apache-style CLAs vs. ordinary permissive FOSS

- Confusion around Apache-style CLAs
    - Often described as copyright assignments
    - Some assume CLAs are inherently nonproblematic
    - CLA-using corporations exploiting confusion?

Red Hat's Experience

- Contribution asymmetry
  - Substantial contributor to many upstream projects
  - Difficulty attracting contributors to "our" projects

- By 2008, patchwork of contribution policies
  - Fedora: Apache-style CLA
  - JBoss.org: (L)GPL CLA
  - Cygwin: copyright assignment
  - Others: inbound=outbound

Fedora ICLA

- Huge cultural/technical mismatch
  - Fedora community tends to be pro-copyleft
  - Fedora mainly packages other projects' code

- Led to confusion and mistrust
  - Assumed by many to supersede upstream licenses
  - Fear of proprietization by Red Hat

- Discouraged some contributions

- Docs relicensing (2009): 'nuclear option'?

FPCA (2010)

- Fedora developed clever reinterpretation

- Codified in new agreement written from scratch
  - Minimalist design; no scary legalese
  - Revised with community input

- Covers only original material with no explicit license
  - Default free license: MIT, CC-BY-SA
  - Contributors can opt out by explicitly licensing

JBoss

- Inherited committer CLA: use only under (L)GPL

- Supplemented by undocumented customary practices

- Complicates project relicensing (LGPL → Apache)

- Replaced with Apache-based CLA in 2009 as stopgap

- Future: FPCA? Apache 2.0/LGPLv2.1 dual-license?

Other Projects

- Briefly considered universal imposition of modified Fedora CLA
    - FreeIPA, Spacewalk

- Michael DeHaan (2008):*"Cobbler itself has not required copyright assignment, so files are © their original author and major contributors, Linux kernel-style."*

Red Hat as Contributor

- Most upstreams do not require special agreements

- Exceptions
  - Trusted project nonprofits (FSF/GNU, ASF, PSF, Mozilla)
  - Established relationships with Sun projects (OOo, OpenJDK)
  - For-profits: protracted negotiations over badly drafted, overreaching agreements

Conclusions

- Formal contribution agreements are Bad:
    - No real legal advantages
    - Signals basic lack of legal confidence in FOSS
    - Negates social/technical advantages of FOSS
    - Ethical concerns

- More justifiable if assignee is nonprofit fiduciary

- Otherwise, best policies are informal, pure FOSS, documented

The End

Thank you!

rfontana@redhat.com

This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 United States License.