



Improving the Open Source Legal System

Richard E. Fontana

Red Hat, Inc.

SCALE 8x

Sunday, 21 February 2010

Introduction, license, disclaimers

- Me
 - Open Source Licensing and Patent Counsel at Red Hat since 2008
 - Formerly Counsel at Software Freedom Law Center
- Slides are © 2010 Red Hat, Inc.
 - License: CC-BY-SA 3.0 (er, CC-BY-ND 3.0 with Shadowman image)
- IAAL, IANYL, TINLA, and

the views expressed in this presentation are solely my own and do not represent the views of any current or former employer or client.



Summary

- As a system of rights, FLOSS should make legal sense
- Local law doesn't satisfactorily explain FLOSS community behavior
- Helpful to see FLOSS as a transnational customary legal system, but this exposes problems
- By improving those problems we make FLOSS more rational, predictable, legitimate



Making legal sense out of open source

- Open source is about rights to use code
- A system of rights can only work well if it is:
 - comprehensible
 - legitimate
 - certain



Law, software and control

- Since ~1980, use of software is controlled legally through:
 - Monopoly rights: ©, trade secrets, patents (~1990)
 - Contracts
 - Aided by object code-only distribution (technical control)
- This is local law
- Public domain: what's uncontrolled



Proprietary software is boring

- Users have no rights
- Users agree they have even fewer rights
- Technical barriers to exercising freedoms
- Entrenched commercial interests keep the system under control



Free software is interesting – in tension with law

- State of nature (antedates use of law to control code)
- Licenses originate as acts of legal rebellion
 - Using machinery of control to give up control
- Because there's freedom, there are complex issues
- Is idea of code ownership legitimate?



Two problems

- Open source doesn't make sufficient sense as part of underlying legal system
- Open source doesn't make sufficient sense if you ignore underlying legal system



The conventional view

- Open source is just exotic variant of proprietary software licensing, **well-grounded in underlying law**
 - Trade secret license that gives up all trade secrets
 - Copyright license with few restrictions, enforceable through © infringement action (*Jacobsen v. Katzer*)
 - Anything that doesn't make sense is also present in proprietary software (e.g. issues of contract assent)
 - Compliance is equated with minimizing risk (esp. litigation)



Inside vs. outside community

- “As I find violator after violator, I look forward to welcoming them to our community in a friendly way, and ask them to respect the commons that gave them so much, and give their code back to the community that got them started.”
– Bradley M. Kuhn, *The Anatomy of a Modern GPL Violation* (2009)

Conventional view provides good way of understanding behavior ‘outside the community’



Doesn't fully explain intra-community behavior

- License interpretation
 - Territoriality is ignored; no one knows or cares what law governs – license should have universal interpretation
 - What about GPLv3 internationalization?
 - Even if we think we know what law governs, we don't use copyright/contract law to understand licenses
 - If you try to map to local law, you get the wrong answers
- Licenses go unenforced: fear of litigation does not shape behavior



Disturbing implications

- Does open source licensing make any sense?
- One possibility: none of this matters because FLOSS licenses are purely aspirational
- Another possibility: it's okay that mapping to local law gives bad answers, because risk of enforcement is so low



Consider FLOSS as its own legal system

- Transnational – cf. international commercial law
- Based on *sui generis* code authorship rights masquerading as copyright
- Legal rules/norms based on community customs
- Disputes resolved through community institutions, without relying on governments
- What software IP law would look like if hackers had developed it



Sounds good, but . . .

- Suggests the system is more rational than it actually seems to be
- How do you know what these customs are?
- If you can figure out the rules, how do you know if they reflect consensus?
- There really aren't any dispute resolution institutions
- So this may be closer to truth but is partly wishful thinking



Any real problem?

- Open source seems to work very well, legally
 - Few serious disputes arise
 - No perception in community of enforcement crisis
 - But is this because developers don't care enough?
- Long-term damage if system lacks legitimacy and rationality
- If we believe in free software ideals, we should strive to make FLOSS law meaningful and rational



Making FLOSS a more rational legal system

- Understanding definition of FLOSS
- Understanding outbound licensing and inbound licensing
- What are the sources of law?
- How do we resolve legal disputes?



Customary understanding of FLOSS definition

- There should be broad agreement on free/nonfree boundaries
- FSF, Debian, OSI, Fedora try to develop this doctrine (a good example of customary law)
- Numerous cases of nominally open source projects
 - Distributing nonfree code
 - Putting their own code under nonfree licenses
 - Applying antifree interpretations to free licenses
- Distro projects can help police and explain rationale



Customary understanding of outbound licensing

- 'License stewards' of widely used licenses should provide more public guidance on interpretation
- Projects should document their interpretation of licenses
- License conflicts
 - Often impossible to make any coherent sense of how a project's code release is licensed
 - Distro projects can help police and explain rationale



Customary understanding of inbound licensing

- FLOSS is held to higher standard because of transparency
- Often impossible to determine who wrote code, and what license covered it
- Projects can help by documenting policies on inbound contributions
- Remember, your code may be reused out of original context



Sources of law

- Mainstream law: constitutions, statutes, treaties, regulations, court decisions, custom ...
- FLOSS: FLOSS definitions, licenses, custom
 - How is custom documented?
 - FAQs, mailing lists, actual development practices ...
 - By and large it *isn't* documented, at least accessibly
 - We need to start producing and assembling this material



Dispute resolution

- FOSS licensing disputes traditionally resolved through community dialogue/pressure
- Consider building community dispute resolution institutions to resolve intracommunity FLOSS licensing conflicts
 - Inspiration: SFLC's handling of ath5k dispute



Thank you.

rfontana@redhat.com

