



Improving Legal Certainty in FOSS

Richard Fontana
Red Hat, Inc.

OSCON 2009

Me:

- IAAL, IANYL, TINLA
- Open Source Licensing and Patent Counsel at Red Hat
- Before that: Counsel at the Software Freedom Law Center
- Before that: just another “intellectual property” lawyer

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



Agenda

- Relationship between FOSS and IP law
- FOSS as an independent legal system
- If FOSS is/were a legal system, how could we make it better?



A Brief History of FOSS & the Law

- Prehistoric era (Time Immemorial™-1983)
 - Freedom from IP law
- Classical era (1983-1998)
 - Subverting IP law
- Modern era (1998-present)
 - Normalization within IP law



Two Prevailing Views of FOSS & the Law

- Conflict
 - Free software forced to adapt to and manipulate IP regime
 - IP regime basically evil
- Harmony
 - FOSS licensing is about ownership & control, consistent with mainstream IP legal norms
 - IP regime basically good (maybe modulo software patents)



Is FOSS Really a Creature of IP Law?

- Maybe, but that can't be the whole story
 - We don't think about territoriality
 - We don't use the IP/contract legal regime to understand these licenses
 - Our actions are not really motivated by avoiding litigation
- (Who's "we"?)



Another View

- FOSS can be seen as its own legal system
 - Supra-territorial
 - Sui generis code authorship/distribution rights masquerading as ©
 - Legal norms are based on custom/tradition
 - Community dispute resolution
- Normative (what reform of national software IP law might look like)



Historical Analogue: *Lex Mercatoria*

- Medieval traders developed private system of international commercial law to fill regulatory gaps in local law
- Based on customary practices, well-documented
- Came to influence national commercial law
- ‘New’ lex mercatoria (1990s-)
 - Arbitration system enforced via national contract law
 - Going outside ‘real’ legal system raises issues of legitimacy



Judging the FOSS Legal System

- If FOSS is a legal system, how good is it?
- Well-functioning legal systems feature certainty/predictability:
 - Known guiding principles reflecting community consensus
 - Definite (non-vague) rules about rights/responsibilities
 - Well-understood, legitimate dispute resolution process



Given some code, can we say with reasonable certainty:

- This is (or isn't) FOSS, and if it is,
- We know who wrote it (rights holder)
- We can figure out what expectations govern use of the code (rights holder's license)
- But if conflicts do arise over such use, there's a process for resolving them

- (Still assuming – we're not truly inside national IP regime)



How are we doing?

- Foundational norms
- Theory of licensing
- Code provenance
- Sources of law
- Risk of third-party claims
- Dispute resolution



Foundational Norms

- Trusted institutions have done a decent job of articulating and applying strict definitions of FOSS
- Nevertheless, we too commonly see nominally open source projects:
 - using 3rd-party non-FOSS code
 - choosing non-FOSS licenses
 - applying anti-FOSS interpretations to FOSS licenses
- Community distros can play key role in enforcing norms
 - We need more public rationale for these decisions



Theory of (Outbound) Licensing

- Two issues:
 - What do particular licenses mean
 - Widely-used licenses are well understood – projects should prefer
 - Projects should provide explicit guidance regarding license interpretation
 - How do multiple licenses interact (~compatibility)
 - Projects should take license conflicts seriously; often easy to resolve by contacting licensors
 - Often difficult to derive coherent understanding of how project is licensed



Code Provenance (and Inbound/Outbound Licensing)

- Several issues
 - 1) Who wrote this code?
 - 2) Who is supposed to have rights to license out this code? (Pretend that © matters)
 - 3) What license governs the inbound contribution?
 - 4) What license governs the outbound code?
- Projects should do a better job at transparency here
- FOSS is held to a higher standard than proprietary software



Code Provenance (and Inbound/Outbound Licensing)

- Inbound licensing
 - Projects should have documented licensing policies governing patches
 - Reasonable effort to ensure contributor has right to make the contribution
 - Problem is more apparent than real



Code Provenance (and Inbound/Outbound Licensing)

- Problems arise with cross-project code reuse, especially legacy code
- It would be ideal for projects to document relevant legal information in the individual source file (contributor, applicable outbound license, perhaps existence of inbound contribution license)
 - BTW, please do not have source file merely refer users to legal files in your source distribution!



Sources of Law

- Mainstream law: constitutions, statutes, treaties, regulations, court decisions, custom ...
- FOSS: FOSS 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



What About Risks of Third-Party Claims?

- On ©/trade secret/TM sides, this rarely comes up
- Perhaps mainly an issue involving software patents:
 - Unlike with ©, can't really assimilate into FOSS legal system (despite license attempts)
 - Advice:
 - Avoid well-known patent thickets
 - Otherwise, ignore patents as much as possible
 - Some projects will need to formulate and publicize coherent patent policy



Dispute Resolution

- FOSS licensing disputes traditionally resolved through community dialogue/pressure
 - License compliance is suboptimal?
- Recent increase in GPL enforcement litigation is separate phenomenon (use conventional legal system against clear-cut material license violations)
- Consider building community dispute resolution institutions to resolve other kinds of FOSS licensing conflicts



Conclusions

- Useful, at least as a thought experiment, to evaluate FOSS as a separate legal system
 - May better capture what's going on
 - May point to law reform possibilities
- Suggests ways in which projects can improve FOSS legal infrastructure and practices, regardless of what legal framework is applicable



Questions?

rfontana@redhat.com

