Open Source Software Licensing: What Every Technologist Needs to Know

Heather Meeker
Foundations of Open Source
What is Open Source Development?

Cathedral

http://www.catb.org/esr/writings/cathedral-bazaar/

Bazaar
RMS v. the Xerox 9700

- Richard M. Stallman, a staff software programmer at the MIT AI lab, and, had trouble with a printer.
- The printer was a prototype from Xerox with a tendency to jam. When it jammed, print jobs stacked up.
- Stallman wanted to insert a software command into the printer software that ordered his computer to check the printer periodically and report back with a message if the printer was jammed.
- But the software in the printer was only in object code format.
- Xerox would not share the source code.

And then Mr. Stallman wrote the GPL...
For the full story, see http://www.oreilly.com/openbook/freedom
Open Source Is Not New

- Development of UNIX at AT&T Bell Labs
- Open source release
- Privatization
- The search for a UNIX successor
What is Open Source Licensing?

• Terms of art
  ○ Copyleft
    ■ Not public domain
    ■ Not a “virus”
  ○ Open Source
  ○ Proprietary
What is Open Source Licensing?

Unrestricted licenses with conditions

You can look, but not eat!

You can eat, but you must share!
The Open Source Licenses
Open Source Initiative has approved over 80 licenses, but almost all OSS is under these:

- GPL
- LGPL
- BSD
- MIT
- Apache 2.0
- Eclipse Public License, CDDL, Mozilla Public License
Two Philosophies: Open Source and Free Software

- Permissive software = MIT, BSD, Apache
  - If you distribute, provide notice

- Free software = GPL
  - If you distribute, use the same terms
  - If you distribute, provide source code
• **GPL -- Strong copyleft**
  ○ If any code in a program is GPL, it must all be GPL (all or none)
  ○ This means no “linking” to proprietary code

• **LGPL -- Weak copyleft**
  ○ If any code in a library is LGPL, it must all be LGPL
  ○ But you can dynamically link to proprietary code
• “Derivative work”
  ○ Mostly a red herring
  ○ If you put GPL code in a program, the entire program (including the GPL portion) is a derivative work of the GPL portion
• **Static v. Dynamic Linking**
  - Only matters to LGPL
  - Many languages do not use the concept of linking
  - GPL/LGPL were written for C/C++ programming, i.e. the Linux kernel
Top Questions in Open Source Licensing
Top Questions in Open Source Licensing

• What is distribution?
• If I release my software under an open source license, what happens to my patents?
• How do I put notices on my product?
• What is a “derivative work”? 
What is Distribution?

• Distribution is transferring a copy from one legal person to another
• Most open source licenses impose no conditions absent distribution
• For most licenses, SAAS is not considered distribution
• Exceptions:
  ○ Affero GPL (AGPL)
  ○ Open Software License
  ○ Non-Profit Open Source License
  ○ Academic Free License
  ○ Artistic 2.0
  ○ Apple Public Source License
  ○ RealNetworks Public Source License
  ○ Reciprocal Public License
What about Patents?

• Some open source licenses include express patent licenses
  ○ Apache 2.0
  ○ GPL3

• Some open source licenses say nothing about patents
  ○ BSD
  ○ MIT
  ○ GPL2

Either way, releasing open source code can limit your patent rights.
How to do Notices?

• Use the rules of GPL -- they work for most licenses

• Notices must be delivered with the product

• Delivering source code upfront is easiest

• If not, create a notice file
What is a “derivative work”?

• That is not the right question
• The right question is: how do I comply with GPL?
• Short answer: All code in a single executable process (i.e. a “Program”) must be either all GPL or no GPL.
For your ample free time...

HEATHER MEEKER
Partner
O’Melveny & Myers
Technology Transactions Group

hmeeker@omm.com
510-463-1116
blog at heathermeeker.com

Available in paper or ebook form