Trail map

- Introduction to intellectual property law as related to open source
- Introduction to open source (licenses)
- Overview of how organizations use or engage with open source
  - Key considerations
Some basics relevant to open source software

**Intro to Intellectual Property**
What is “intellectual property”? 
Trademarks

- Protects marks (word, logos, slogans, color, etc.) that identify the source of the product
  - Consumer and brand protection; avoid consumer confusion and brand dilution
- Mark must be source identifying and distinct
- Must be first to use mark in commerce
- Trademark owner must enforce the mark
- Trademarks continue in force until abandoned by owner or become un-protectable
- Can be registered or unregistered
Patents

- Protects useful inventions (ideas) – requirements:
  - New (novel) as compared to prior art
  - Non-obvious (inventive) to a person having ordinary skill in the art
  - Must be useful (capable of industrial application)
  - Patentable subject matter

- 20 year term

- Rights are assignable (transfer ownership/all rights)
Patents: What do you get?

- A patent does not give you any rights: only the right to stop others
  - Limited term monopoly in return for making invention available to public
- Patent owner has right to stop others from:
  - Makes, uses, sells, offers for sale, or imports the invention described in the claims of the patent
Copyright

- Protects original works of authorship fixed in a tangible medium
  - Low threshold of creativity
- Protection attaches as soon as work is fixed/created
- Covers: Literary, dramatic, choreographic, pictorial/graphic/sculpture, motion picture/audiovisual, sound recordings, architecture
- Copyright owner = author
  - If “work made for hire” = employer is author
- Lasts for life of author + 70 years
  - 95 years in case of “entity authors”
Copyright: What do you get?

• The owner of copyright ... has the exclusive right to do and to authorize any of the following:
  • To reproduce the copyrighted work...
  • To prepare derivative works based upon the copyrighted work
  • To distribute copies ... to the public by sale or other transfer of ownership, or by rental, lease or lending

• Rights are assignable (transfer ownership/all rights)
Copyright: What it doesn’t cover

- Copyright protects the expression (not the idea)

- Merger doctrine = if the idea underlying the work can be expressed only in one way, the idea and expression “merge”

- de minimus copying

- Fair use (fair dealing) = permitted use for limited reasons
  - E.g., commentary, search engines, criticism, parody, news reporting, research, teaching, library archiving and scholarship

- Public domain = works for which © protection has expired or otherwise not protected
Licenses

- Gives permission to exercise a right otherwise reserved for owner
  - Types of use allowed (distribution, derivative works / made, have made, manufacture)
  - Conditions placed upon exercise of enumerated rights
- Exclusive or non-exclusive
- Geographical scope
- Perpetual or time limited

- May also include contractual terms re: warranties, indemnification, support, upgrade, maintenance
Licenses

- Open source licenses are copyright licenses
- May include explicit patent grant
  - Implied patent grant?
- May include contain clauses restricting use of trademarks or names
- Conditions placed upon exercise of enumerated rights
- Non-exclusive, perpetual
Open source software, licenses, and projects

Basics of open source software (licenses)
source code = human readable, editable format

object code = compiled code

binary, executable (non-source form)

compilation = process of turning human readable code into machine executable code
static linking = all code compiled into a single executable (or standalone binary file); done at compile time

dynamic linking = code or modules compiled into separate files (libraries); done at execution

scripts = human readable, no object form
embedded system = a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints

firmware = software that is embedded in a piece of hardware
A note on terminology

- open source software
- free software
- free and open source software
- free/libre open source software
- OSS
- FOSS
- FLOSS
What is open source (software)?

- Development model?
- Ideology?
- Legal construct?
What is open source (software)?

✓ Development model?
  - Access to source code and means anyone can tinker with, modify, improve
  - Project can benefit from collective intelligence of many developers

✓ Ideology?

✓ Legal construct?
What is open source (software)?

✓ Development model?
  - Access to source code and means anyone can tinker with, modify, improve
  - Project can benefit from collective intelligence of many developers

✓ Ideology?
  - Belief in freedom and desire to share

✓ Legal construct?
What is open source (software)?

✓ Development model?
  - Access to source code and means anyone can tinker with, modify, improve
  - Project can benefit from collective intelligence of many developers

✓ Ideology?
  - Belief in freedom and desire to share

✓ Legal construct?
  - Collaborative model and ideology “implemented” via (copyright) licenses
What makes open source open?

- Open source license gives you:
  - permission to use the program for any purpose; modify it (create derivative works); and redistribute the program to others (original or your modified version)
- May or may not also be no cost
- Just because you have the source code, doesn’t mean it’s open source

_free as in freedom, not free as in beer_
Defining open source

- Defines free software via the four freedoms
- Mission to promote computer user freedom
- Founded in 1985 by Richard Stallman
- Maintains the Open Source Definition (OSD)
  - Approves licenses that meet this definition
- Mission to educate and advocate for the benefits of open source
- Formed in 1998
Open source licenses are... licenses

Open source

- Unilateral permission
  - Anyone can use code
  - No affirmative acceptance needed
  - Direct license from licensor

- Grant of rights
  - Can copy, modify, distribute

- Licensor obligations
  - "as is"
  - No warranty, support, updates

Proprietary/commercial (closed source)

- Negotiated terms
  - Specific to parties

- Grant of rights
  - Restricted: # of copies, cannot modify, specific use case, may prohibit distribution

- Licensor obligations
  - Warranties, support, upgrades
  - IP indemnification
## What is NOT open source software?

<table>
<thead>
<tr>
<th>Freeware</th>
<th>Shareware</th>
<th>Public Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ Free (as in cost)</td>
<td>▪ Free (as in cost) for a set period</td>
<td>▪ Expired © or not protectable</td>
</tr>
<tr>
<td>▪ No access to source code</td>
<td>▪ A.K.A. trialware or demoware</td>
<td>▪ Some s/w is “released” into the public domain via a “dedication”</td>
</tr>
<tr>
<td>▪ License usually has various restrictions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>▪ License is specific to that program</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Because these are “free” (no cost), sometimes mistaken for free software or open source software.
“Types” of Open Source Licenses

Permissive
- Broad grant of rights
- License requirements are minimal
  - E.g. retain notices, include copy of license, notice of modifications

Copyleft
- Source code must be made available for binary distribution
- Original work, any modifications, any derivative work must remain under same license
  - Entire derivative work or only modified files
“Types” of Open Source Licenses

Permissive
- MIT
- BSD-2-Clause
- BSD-3-Clause
- Apache 1.1
- Apache 2.0

Copyleft
- Mozilla Public License
- Eclipse Public License
- Lesser General Public License
- General Public License
- Affero General Public License
NOT ALL OPEN SOURCE LICENSES ARE CREATED EQUAL!!
Structure, governance

Open source projects
Open source project governance
Open source project governance
Open source project governance

- Central or umbrella organization
  - Non-profit, foundation, corporation

- Formal governance
  - Governing board
  - Technical steering committee
  - Other committees

- Membership agreement and dues
Open source project agreements

Inbound agreement

<code contribution>

Three variations:
1) assignment
2) contributor license
3) same as outbound

Outbound license

E.g.,
- Apache-2.0
- GPL-2.0-or-later
- BSD-3-Clause
- MIT
- etc....
Pros: Ultimate control for project owner.
Cons: Highly frowned upon. High cost of entry.

Inbound: copyright assignment

- Transfer of all rights (ownership), unconditionally
  - May also include explicit patent license
- Contributor relinquishes copyright
- Includes a license back to contributor
Inbound: contributor license agreement (CLA)

- Copyright remains with each contributor
- Contributor grants broad rights to project maintainer with no conditions
  - May include explicit patent license
- Project maintainer can then change license

Pros: Provides more control for project maintainer.
Cons: Seen as unfair (unequal rights). Administrative burden.
Pros: Provides equality. More easily understood.
Cons: Lack of control for project maintainer. Proper authority?

Inbound: same as outbound license

- Copyright remains with each contributor
- Contributions to project are under same license as outbound license
  - Direct license to all recipients
Developer’s Certificate of Origin (DCO)

- Personal accountability via “sign-off”
- Attestation (“I certify that”) by the contributor:
  - that the contribution was created by that person or they have the rights to contribute it under the license indicated in the file
  - Record of contribution (name) will be maintained
Code of Conduct

- Set of rules or guidelines to establish social and behavioral norms and method for enforcement
- Becoming more common for open source projects
  - Response to failings of “pure meritocracy”, examples of egregious behavior, and severe lack of diversity in open source projects
Membership agreements / project charter

- States project mission and goals
- Sets out membership types and dues
- Establishes project governing board or other roles or committees and how they operate, voting rights, etc.
- IP section often names or refers to open source licensing model chosen for project
Using and engaging with open source

Challenges, considerations, tips
How do organizations use or engage with open source?
Strategy, guidance, governance

Contribute to existing projects
• As employee
• On own-time

Join a project

Create an open source project

Roles and responsibilities as to who manages processes and how

Internal use
Use in products (external distribution)
Consume: external distribution

Open source license compliance

- Have to know *what* open source software you are using
  - Have to know what *license* applies
    - Have to know *how* you are using
  - What license conditions are triggered? = open source license compliance
What are you using?

GitHub

SourceForge

Etc....

your product source code
What license applies?

Open Source Package
License

Main Open Source Package
Bundled Open Source Packages
Main License
Bundled License
Bundled License
Bundled License
license link broken #6

jlovejoy opened this issue on Aug 26, 2014 · 1 comment

jlovejoy commented on Aug 26, 2014

the license link listed in the LICENSE file is broken. Can you please provide a copy of the proper license with the code itself? thanks!

ar7886 commented on Sep 8, 2014

Thanks for contacting at&t. The code-kit is under the Apache license. We will publish the license text in upcoming release.
What license applies?

Multiple licenses

- Conjunctive = AND
  - Multiple licenses apply to code, must comply with all
- Disjunctive = OR
  - Choose one of set of licenses
- Dual = OR or AND?
  - (usually) Choice of open source license or commercial license

- Additional terms/exceptions = WITH
  - Modifiers to existing open source licenses
Open source license compliance analysis

- **IF I am...** distributing, modifying... (in binary or source form)...
  - Trigger or usage model - How am I using the software?
- **THEN I must...** provide a copy of the license, retain notices, provide the source code...
  - **WHAT** is the requirement?
  - **HOW** does that requirement need to be met?
    o Devil’s in the details...
How are you using the OSS?

- **Distribution**
  - Dissemination of material to a change of hands
  - May be broader than non-lawyers think
  - GPL-2.0: “Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, . . .”

- **Is access via a computer network (ever) a “distribution”?**
  - Applications download to the user’s machine (e.g. mobile)
  - Javascript, web client, or other code that is downloaded to the user’s machine
  - Some licenses define as such: Affero GPL, Apple Public Source License, RealNetworks, Reciprocal Public License
How are you using the OSS?

- **Modifications**
  - Changes to the existing program: additions, deletions, etc.
    - See definition in license
  - Usually need to provide notice of modification, so important to track this
What license conditions are triggered?

- Retain copyright (and other) notices
- Provide a copy of the license
- Provide attribution
- Provide notice of modifications
- Provide access to source code (whether you modified it or not)
- Maintain modified versions (derivative works) under same license
- Do not use the project or copyright holder name or trademark
- Do not restrict others of the rights granted
- Termination clauses (if you breach, you lose license)
Other considerations: derivative works

- Definition may vary by license or even license version
  - But what about the legal definition?
- No judicial opinion as to the derivative work question (re: GPL)
  - Fact-based, case-by-case analysis...
- Does static linking and dynamic linking create a derivative work?

A company’s position will be based on a legal interpretation, practical reality, and risk assessment.
Other considerations: license compatibility

- You create a derivative work combining or incorporating code under different licenses and the license terms conflict with each other or create obligations you don’t want to comply with
  - Combining open source with other open source
  - Combining open source with proprietary-licensed code

- What creates incompatibility?
  - Impossible to comply with both
  - Terms otherwise conflict with each other (“further restrictions”)
  - Other terms conflict or create obligations you don’t want
Consume: process

- System for request, approval, and tracking open source components
  - License identification, component, use-case
  - Maintaining information (license, source code, etc.)
  - Generation of license compliance artefacts delivered with products
- Training, identified roles, guidance
Contribute, collaborate

- Contribute to existing projects
  - As employee on behalf of organization
  - On own-time

- Join a project (e.g., membership agreement, dues)
Contribute, collaborate

- Why are you contributing?
  - Support technology you rely upon
  - Enablement of other technology
  - Upstream modifications / reduce technical debt
  - A good thing to do
    - Not part of your business or IP strategy
Contribute, collaborate

- What are the legal requirements?
  - What is the inbound license?
    - What IP are you giving a license to?
  - What are the expectations for membership?
Contribute, collaborate

- Who is contributing?
  - Coordinating efforts across your organization

- Public participation and organization reputation
Contribute, collaborate: process

- Request, approval process, scope of approval
  - Are certain areas for contributing “auto-approved”?
- Tracking of contributions and participation
  - Not just about managing risk...
Creating open source projects

- Strategy
- Licensing model
- Code review
- Marketing
- Infrastructure

Strategy, guidance, governance
Consume
Contribute
Create
Roles and responsibilities

(\textit{the open source program office})

- Who is responsible for what?
  - Is that role properly resourced?
- What expertise is needed?
  - Open source is not one-dimensional area
- Cross-functional
  - Engineering, legal, strategic, marketing
Collaboration beyond code

- SPDX (Software Package Data Exchange) = Standard format for communicating information about software components
  - https://spdx.org/

- OpenChain = standard defining requirements for effective open source management (license compliance)
  - https://www.openchainproject.org/

- TODO Group = guidance related to open source community engagement
  - https://todogroup.org/
KEEP CALM AND USE OPEN SOURCE
Tips for developers/engineers

- Look for the license (and be skeptical)
  - In the source code (don’t trust the website)
  - Consider there may be other OSS with different licenses included in the project you want to use

- If you don’t have / can’t find a license, then you don’t have one / can’t use it
- Record what you find and where (and how)
- Not all “free” licenses are open source
- Keep track of what you are using, how you are using, and if modifying it (and how)

- Identify the inbound and outbound license(s) for the OSS project
- Understand why contributing to or creating the OSS project provides value (why are we doing this?)
Extra slides
SPDX Specification

- Instructions for collecting, storing and communicating data
- Goal to be human and machine readable
- Focus on facts (not interpretations)
- Associated tooling
SPDX License List

- Matching guidelines to determine license matches (and templates)
- License expression syntax
- AND, OR, WITH, +
- Tools to programmatically access the SPDX License List
- Actively maintained by SPDX legal team

[Link to SPDX License List](https://spdx.org/licenses/)
SPDX in action

- Generating or consuming SPDX documents
- Use of SPDX License List
- Use of SPDX license identifiers in source code
- Use of SPDX specification fields in internal systems

  - Can you talk about how you are using SPDX?