Open Source Advising at Scale

Justin C. Colannino
Senior Attorney, Microsoft

FINOS
June 8, 2020
The Internet

The opinions in this presentation are those of the presenter, not Microsoft or its affiliates.
Agenda

Open Source Refresher
Definition & License Archetypes

Open Source Law
Making calls without (much) caselaw.

Counseling @ Scale
How to advise 60,000 developers for millions of use cases
The Open Source Stack For Lawyers

- Economic
- Political
- Social
- Legal
- Technical
## A Counseling Framework

<table>
<thead>
<tr>
<th>Economic</th>
<th>Political</th>
<th>Social</th>
<th>Legal</th>
<th>Technical</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Commodity</th>
<th>Custom</th>
</tr>
</thead>
</table>
What Is A License?
Permission
Permission
(usually subject to conditions or obligations)
Open Source: Permissions & Conditions or Obligations

Right to Use, Copy, Modify, and Distribute (FSF – Four Freedoms)

Must Meet Conditions or Obligations

Typical: provide notice and/or provide source
License Archetypes

- Ultra Permissive
- Permissive
- Weak Copyleft
- Copyleft
- Network Copyleft

Increasing Obligations
### Open Source License Archetypes

<table>
<thead>
<tr>
<th>Archetype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ultra Permissive</strong></td>
<td>- Goal: Maximum Rights, <strong>NO</strong> Obligations</td>
</tr>
<tr>
<td>(WTFPL, Unlicense, CC0)</td>
<td></td>
</tr>
<tr>
<td><strong>Permissive</strong></td>
<td>- Goal: Maximum Rights, Minimal Obligations</td>
</tr>
<tr>
<td>(MIT, BSD, Apache 2.0)</td>
<td>- Distribution Triggers Attribution Obligation</td>
</tr>
<tr>
<td><strong>Weak Copyleft</strong></td>
<td>- Goal: Preserve Freedom In A “Core”</td>
</tr>
<tr>
<td>(EPL?, LGPL, MPL)</td>
<td>- Distribution Triggers Attribution Obligation &amp; Source Code Obligation</td>
</tr>
<tr>
<td><strong>Copyleft</strong></td>
<td>- Goal: Preserve Downstream Rights</td>
</tr>
<tr>
<td>(GPL)</td>
<td>- Distribution Triggers Attribution Obligation &amp; Source Code Obligation</td>
</tr>
<tr>
<td><strong>Network Copyleft</strong></td>
<td>- Goal: Extend Copyleft to Network Services</td>
</tr>
<tr>
<td>(AGPL, EUPL, CPAL)</td>
<td>- Network Interaction Triggers Attribution Obligation &amp; Source Code Obligation</td>
</tr>
</tbody>
</table>
Notices: Facile Dictu, Difficile Factu
Agenda

Open Source Refresher
Definition & License Archetypes

Open Source Law
Making calls without (much) caselaw.

Counseling @ Scale
How to advise 60,000 developers for millions of use cases
The Open Source Stack For Lawyers

- Economic
- Political
- Social
- Legal
- Technical
Technical Aside: What is Linking?

PROGRAM

LIBRARY
Technical Aside: Static Linking

PROGRAM

LIBRARY
Technical Aside: Dynamic Linking
The Extent of Copyleft: Derivative Works

“A ‘derivative work’ is a work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which a work may be recast, transformed, or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications which, as a whole, represent an original work of authorship, is a ‘derivative work’.”

17 U.S.C. 101
The Extent of Copyleft: The Game Genie Case

Lewis Galoob Toys, Inc. v. Nintendo of Am., Inc., 964 F.2d 965 (9th Cir. 1992)
The Extent of Copyleft: Duke Nukem

*Micro Star v. Formgen, Inc.*, 154 F.3d 1107 (9th Cir. 1998)
Copyleft extends to changes / additions to the Program, but does not extend to “additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.”

The Eclipse Public License v1, Sec. 1
26. Some free software communities say that linking to their code automatically means that your program is a derivative work. Is this the position of the Eclipse Foundation?

No, the Eclipse Foundation interprets the term "derivative work" in a way that is consistent with the definition in the U.S. Copyright Act, as applicable to computer software. Therefore, linking to Eclipse project code might or might not create a derivative work, depending on all of the other facts and circumstances.
Quick Summary

This license, made and used by the Eclipse Foundation, is similar to GPL but allows you to link code under the license to proprietary applications. You may also license binaries under a proprietary license, as long as the source code is available under EPL.
Agenda

What Is An Open Source License?
Definition & License Archetypes

Open Source Law
Making calls without (much) caselaw

Counseling @ Scale
How to advise 60,000 developers for 2.5 million use cases
A Counseling Framework

Economic
Political
Social
Legal
Technical

Commodity
Custom
OSS Counselling @scale

Discover

Automate: Tooling

Automate: Human Process

Investigate
OSS Counselling @scale: Use

• Get into developer tools.
• Keep it simple.
• Iterate: turn bespoke into commodity.
Use: What do we care about?
Use: sharing creates value

- Feedback to Project
- Harvest
- Community Agreement
- Run Tools
Notices: Facile Dictu, Facile Factu
Example Automation Implementation Using Only Open Source Tools

- **CI / CD Infrastructure**
  - Build Tools
    - Dependency resolver
    - Container content resolver
    - Binary analyzer
    - Source package downloader
  - Artifact Repository
  - Continuous Integration
    - Component & application metadata repository
  - License metadata repository

- **Inbound software**
  - Contributions
  - Public compliance artifact repos
  - License & Copyright Scanner
  - License Classifier
  - ScanCode
  - Forensic Code Analysis Service

- **Outbound software & Compliance artifacts**
  - Issue Tracker
  - Source Code Repo
    - Policy checker
    - Obligation fulfillment
    - FOSS Compliance Bundle generator

- **Integration layer (API/Data)**
  - SW360
  - OSS Review Toolkit
  - TERN
  - SW360 fossology
OSS Counselling @scale: Contribute / Release

• What are you defending against?
• Small code exceptions to promote engagement?
• Distinction between utilities/products?
• Cost of forking maintenance.
• Meet communities where they are (CLA/DCO, etc.).
Questions?