Suggestions to Scale your Open Source Legal Support

Ibrahim Haddad, Ph.D. – Head of Open Source Group
Samsung Research America (Silicon Valley)
Ibrahim.H@Samsung.com / @IbrahimAtLinux
For a detailed discussion about the compliance process, please refer to the Linux Foundation compliance publications available from http://compliance.linuxfoundation.org.
People Involved in the Compliance Process

- Developers / Software Architect
- Open Source Compliance Staff
- Software Development Managers
- Legal Counsel
- Compliance Officer

For a detailed discussion about the roles of teams and individuals responsible for ensuring open source compliance, please refer to the Linux Foundation compliance publications available from http://compliance.linuxfoundation.org.
Role of Legal Counsel in the Compliance Process

Core responsibilities include:

1. Advise on open source licensing
2. Provide approval around the use of open source in products
3. Contribute to establishing and running the compliance program
4. Provide training around open source licenses, policies and guidelines
How can we scale open source legal support?
Practical Legal Advice at Your Fingertips

- License playbooks
- License compatibility information
- License classification information
- Approved software interaction methods
- Checklists
1. License Playbooks

- An easy to read and understand summary of licenses intended for software developers.
- For each commonly used license provide a playbook that includes:
  - Name / Version / URL
  - Executive Summary
  - Grant
  - Limitations
  - Warranty
  - Obligations
  - Patent Notes
  - Etc.
License Playbook – Example from tldrlegal.com

This example is provided for illustration purposes only. This is not an endorsement.
License Playbook – Example from tldrlegal.com

This example is provided for illustration purposes only. This is not an endorsement.
License compatibility issues arise when developers combine source code incoming from different sources into a single work.

Incoming Licenses = A + B + C
Outgoing License(s) = ?
A license compatibility matrix is an easy visual method to identify if License-A is compatible with License-B.

A license compatibility matrix is prepared by Legal Counsels for the 10-15 most used licenses.
# License Compatibility Matrix – Simple View

<table>
<thead>
<tr>
<th>Is Compatible With:</th>
<th>License-A</th>
<th>License-B</th>
<th>License-C</th>
<th>License-D</th>
<th>License-E</th>
<th>License-F</th>
<th>License-G</th>
</tr>
</thead>
<tbody>
<tr>
<td>License-A</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>License-B</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>License-C</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>License-D</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>License-E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>License-F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>License-G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
# License Compatibility Matrix: Elaborate Example

<table>
<thead>
<tr>
<th>I want to release a project under:</th>
<th>GPLv2 only</th>
<th>GPLv2 or later</th>
<th>GPLv3 or later</th>
<th>LGPLv2.1 only</th>
<th>LGPLv2.1 or later</th>
<th>LGPLv3 or later</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GPLv2 only</strong></td>
<td>OK</td>
<td>OK [2]</td>
<td>NO</td>
<td>OK: Convey project under GPLv2 only [7]</td>
<td>OK: Convey project under GPLv2 only [7][2]</td>
<td>NO</td>
</tr>
<tr>
<td><strong>GPLv2 or later</strong></td>
<td>OK [1]</td>
<td>OK</td>
<td>OK</td>
<td>OK: Convey project under GPLv2 or later [7]</td>
<td>OK: Convey project under GPLv2 or later [7]</td>
<td>OK: Convey project under GPLv3 [8]</td>
</tr>
<tr>
<td><strong>GPLv2 only</strong></td>
<td>OK</td>
<td>OK [2]</td>
<td>NO</td>
<td>OK: Convey project under GPLv2 only [7]</td>
<td>OK: Convey project under GPLv2 only [7][2]</td>
<td>NO</td>
</tr>
<tr>
<td><strong>GPLv2 or later</strong></td>
<td>OK [1]</td>
<td>OK</td>
<td>OK</td>
<td>OK: Convey project under GPLv2 or later [7]</td>
<td>OK: Convey project under GPLv2 or later [7]</td>
<td>OK: Convey project under GPLv3 [8]</td>
</tr>
<tr>
<td><strong>LGPLv2.1 only</strong></td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td><strong>LGPLv2.1 or later</strong></td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td><strong>LGPLv3</strong></td>
<td>NO</td>
<td>OK: Convey project under GPLv3 [9]</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
</tr>
</tbody>
</table>
License Compatibility Matrix: Look at the Sources

- GNU.org
- Apache.org
- CreativeCommons.org
- Etc.
3. Classification

- An easy way to understand the approval process for different licenses and the course of action needed when using these licenses.
License Classification – Example 1

Example of classification system is to rank licenses from 0 to 5 where:

- 5 Pre-approved [Licenses: A, B, E, K]
- 4 High chance of approval [Licenses: C, G, J]
- 3 Medium chance of approval [etc.]
- 2 Low chance of approval [etc.]
- 1 Not approved – against policy [Licenses: F, L]
Another example of classification system:

<table>
<thead>
<tr>
<th>Permissive</th>
<th>Modifications to be released</th>
<th>Patent Clause</th>
<th>Not Allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>License-A</td>
<td>License-E</td>
<td>License-H</td>
<td>License-L</td>
</tr>
<tr>
<td>License-B</td>
<td>License-F</td>
<td>License-I</td>
<td>License-M</td>
</tr>
<tr>
<td>License-C</td>
<td>License-G</td>
<td>License-K</td>
<td></td>
</tr>
<tr>
<td>License-D</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
Source code licensed under these licenses is pre-approved and can be combined with proprietary software.

**Notes:**
Modifications made to source code licensed under these license must be released back

**Notes:**
Due to patent clause, you must discuss with legal counsel about your planned usage.

**Notes:**
Company policy prohibits use of source code under these licenses.

- Pre-approved
- Requires approval of engineering manager
- Requires Legal Counsel approval
- Not approved
4. Approved Software (License) Interactions

- The goal is to understand how that specific software component interacts with other software components and the method of interaction:
  - Components that are Open Source (used “as is” or modified)
  - Components that are proprietary
  - Components that are originating from third party software providers
  - Components dependencies
  - Communication protocols
  - Linkage method Dynamic versus static linking
  - Components that live in kernel space versus user space
  - Use of shared header files
  - Etc.
## Software Interactions

### Can Dynamically Link To

<table>
<thead>
<tr>
<th>Can Dynamically Link To</th>
<th>License-A</th>
<th>License-B</th>
<th>License-C</th>
<th>License-D</th>
</tr>
</thead>
<tbody>
<tr>
<td>License-A</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>License-B</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>License-C</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>License-D</td>
<td>X</td>
<td>[Requires approval]</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

### Can Statically Link To

<table>
<thead>
<tr>
<th>Can Statically Link To</th>
<th>License-A</th>
<th>License-B</th>
<th>License-C</th>
<th>License-D</th>
</tr>
</thead>
<tbody>
<tr>
<td>License-A</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>License-B</td>
<td></td>
<td>X</td>
<td>[Requires approval]</td>
<td></td>
</tr>
<tr>
<td>License-C</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>License-D</td>
<td>[Requires approval]</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
5. Checklists

- Establish a checklist for most milestones:
  - A checklist before approving integrating incoming code into your product’s source code repository
  - A checklist to ensure you fulfilled the obligations
  - A checklist for developers
  - A checklist for engineer managers
  - A checklist for compliance staff
  - Etc.

- After regular use, checklists become a default behavior.
Checklists – Example

Checklist for use before posting code on the web site (license obligation fulfillment):

- All source code components have a corresponding compliance ticket
- All compliance tickets have been approved by engineering and legal
- All compliance tickets are clear from any sub-tasks attached to them
- Notices for all of the software components have been sent to Documentation team and included in product documentation (including written offer)
- Legal has approved the written offer notice and overall compliance documentation
- Source code packages have been prepared and tested to compile on a standard development machine
- Source code provided is complete and corresponds to the binaries in the product
Benefits
Benefits to Providing Practical Legal Advice

- Easy access to commonly asked questions / use cases / scenarios

- Increase bandwidth of Legal Counsel supporting open source

- Fewer legal bottlenecks in enabling open source adoption and usage

- Documented open source legal practical guidelines, Do's and Don'ts

- Minimize engineering frustration surrounding open source legal stuff
Thank you.

Ibrahim Haddad, Ph.D. – Head of Open Source Group
Samsung Research America (Silicon Valley)
Ibrahim.H@Samsung.com | @IbrahimAtLinux