

# The Role of Legal Counsels in Focusing Open Source Compliance on Scaling and Execution

**Ibrahim Haddad, Ph.D.**  
**Open Source Group**  
**Samsung Research America**  
**@IbrahimAtLinux**

# Abstract

---

**Open Source initiatives and projects provide companies with a vehicle to accelerate innovation through collaboration with the global community of open source developers.**

**However, accompanying the benefits of teaming with the open source community are important responsibilities: Companies must ensure compliance with applicable open source license obligations.**

**In this talk, we look closely at the role of the Legal Counsel in ensuring open source compliance and discuss a number of practical advice that a Legal Counsel can provide to the software development team. Such practical advice will enable software developers to make daily decisions related to open source licenses without having to go back to the Legal Counsel for every single question.**

# Disclaimers

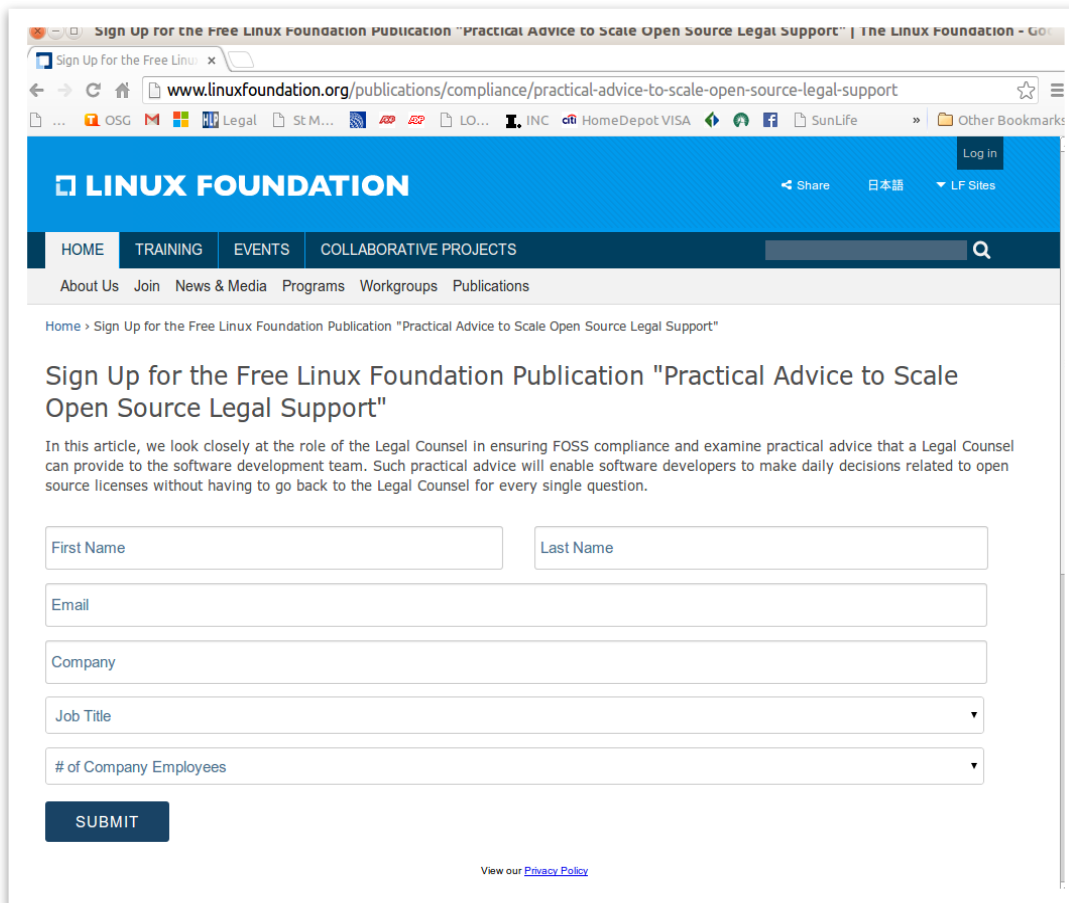
**I am not a lawyer.**

**This presentation is not a legal advice**

**I advise the Samsung compliance team.**



# Download the full paper



Sign Up for the Free Linux Foundation Publication "Practical Advice to Scale Open Source Legal Support" | The Linux Foundation - Go

Sign Up for the Free Linux: x

www.linuxfoundation.org/publications/compliance/practical-advice-to-scale-open-source-legal-support

Linux FOUNDATION

Log in

Share 日本語 LF Sites

HOME TRAINING EVENTS COLLABORATIVE PROJECTS

About Us Join News & Media Programs Workgroups Publications

Home > Sign Up for the Free Linux Foundation Publication "Practical Advice to Scale Open Source Legal Support"

### Sign Up for the Free Linux Foundation Publication "Practical Advice to Scale Open Source Legal Support"

In this article, we look closely at the role of the Legal Counsel in ensuring FOSS compliance and examine practical advice that a Legal Counsel can provide to the software development team. Such practical advice will enable software developers to make daily decisions related to open source licenses without having to go back to the Legal Counsel for every single question.

First Name

Last Name

Email

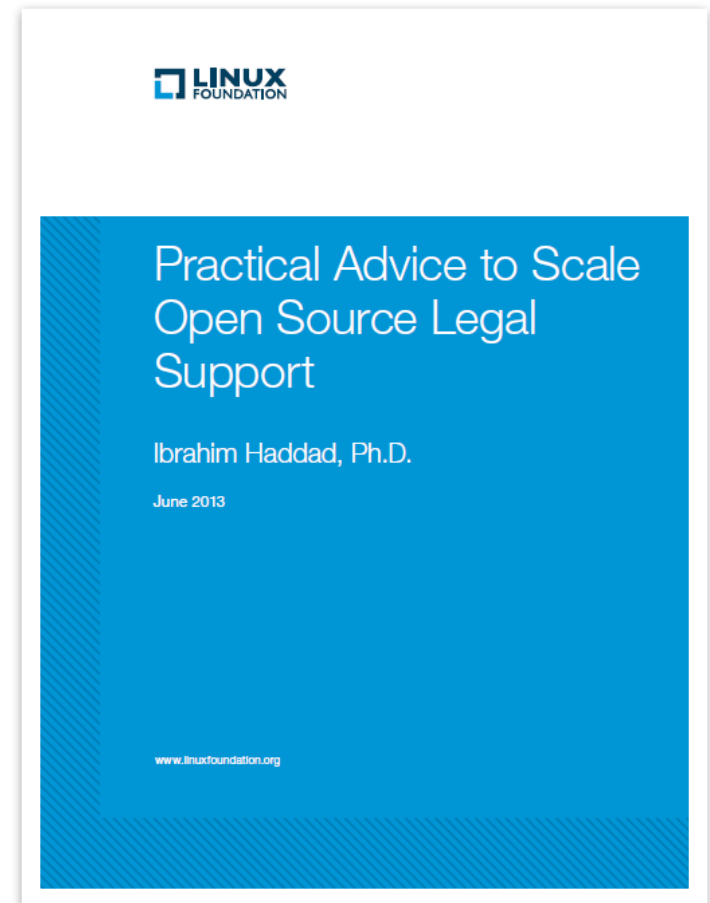
Company

Job Title

# of Company Employees

SUBMIT

View our [Privacy Policy](#)



<http://www.linuxfoundation.org/publications/compliance>

# Smart Companies Have an Open Source Strategy

(and the infrastructure to support it)



# Clear is the new Smart.

Clear internal open source governance

Clear policies.

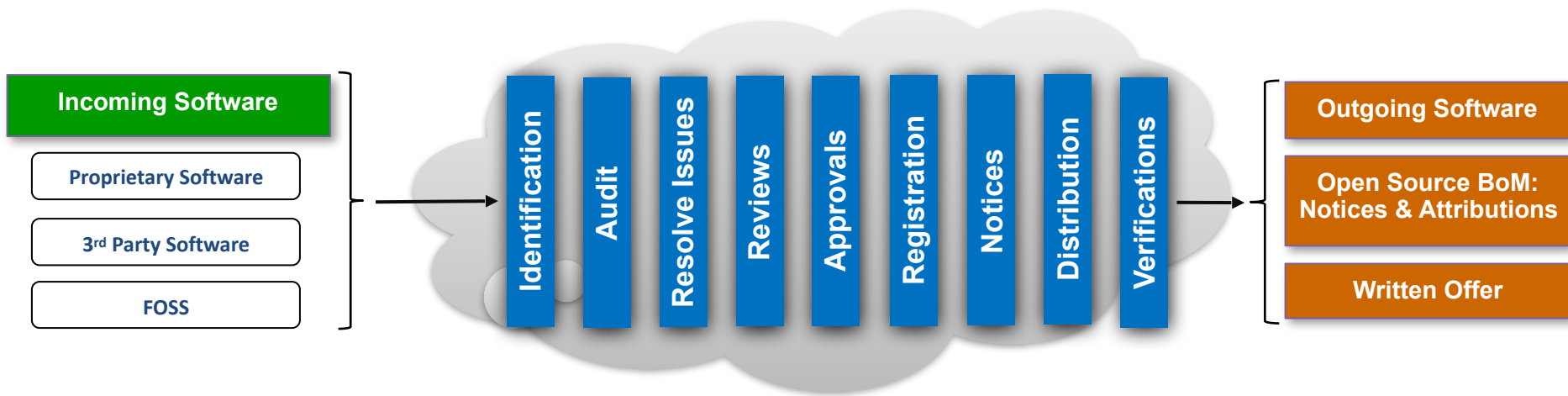
Clear guidelines.

Clear process.



# Example of a Usage / Compliance Process

(Used to approve the inclusion of open source code in a commercial product)



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 Architects**      **I write code**
- **Software Development Managers**
- **Open Source Compliance Staff**      **I approve technical merit for oss usage**
- **Legal Counsel**
- **Compliance Officer (aka Director or Manager of Open Source )**      **I scan code and report results**  
      **I review scan results and advise**  
  
      **I manage and execute compliance**



# Role of Legal Counsel in the Compliance Process

---

- **Establish**                      **Contribute to establishing the compliance program** *[one time effort for a period of time]*
- **Train**                         **Provide training around open source licenses, policies and guidelines** *[occasional]*
- **Approve**                      **Usage and contribution requests – that includes advising on open source licensing**  
*[almost daily depending on your company's adoption rate]*

# How can the Legal Counsel scale support for open source in their org?



# 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.

Apache License 2.0 (Apache-2.0) Explained  
(Open Source)

**Quick Summary:**

A popular license that allows you much freedom with the software, including an explicit right to a patent. "State changes" means that you have to include a notice in each file you modified.

[Tiny Embed Link](#) | [View Full Text](#) | [Use With Other](#)

Can	Cannot	Must
<ul style="list-style-type: none"><li>Commercial Use</li><li>Modify</li><li>Distribute</li><li>Sublicense</li><li>Warranty</li></ul>	<ul style="list-style-type: none"><li>Use Trademark</li><li>Hold Liable</li></ul>	<ul style="list-style-type: none"><li>Include Copyright Notice</li><li>Include License</li><li>State Changes</li></ul>

None of the information on TDRLegal should be considered legal advice. Please remember to consult an attorney.

Other licenses:  
[Beerware License](#) | [Sleepycat License \(Sleepycat\)](#) | [Academic Free License 3.0 \(AFL\)](#) |

**Full Text:**

Apache License, Version 2.0 Apache License Version 2.0, January 2004 <http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

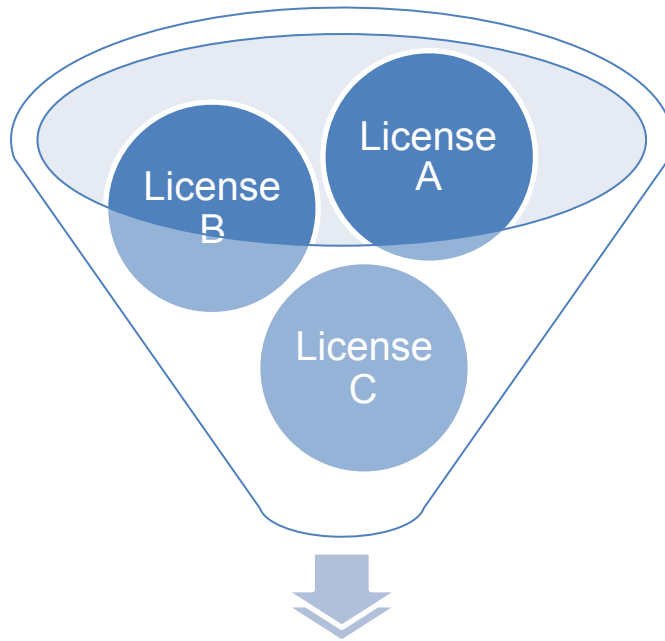
# License Playbook – Example from tldrlegal.com

This example is provided for illustration purposes only.  
This is not an endorsement.

The screenshot shows the TLDRLegal website interface. At the top, there's a navigation bar with the site name, social media links, and a 'Get Updates!' button. Below this is a search bar and a 'Browse' button. The main heading is 'Browse Software Licenses & Summaries'. Underneath, there's a filter section with buttons for 'ALL', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'. Below the filter section, there's a 'Filters:' section with buttons for 'Commercial Use', 'Modify', 'Distribute', 'Sublicense', 'Include Copyright Notice', 'Use Trademark', 'Include Original', 'Include License', 'Private Use', 'Rename', 'State Changes', 'Warranty', 'Non-Static Linkage', 'Hold Liabe', 'Disclose Source', and 'Incomplete'. The main content area displays a list of licenses with their respective options: MIT License (Commercial Use, Modify, Distribute, Sublicense, Include Copyright Notice, Hold Liabe), Microsoft Public License (Ms-PL) (Commercial Use, Modify, Distribute, Sublicense, Include Copyright Notice, Use Trademark, Hold Liabe), GNU General Public License v2 (GPL-2) (Commercial Use, Modify, Distribute, Sublicense, Include Original, Warranty, Hold Liabe, Disclose Source), GNU General Public License v3 (GPL-3) (Commercial Use, Modify, Distribute, Sublicense, Include Original, State Changes, Warranty, Hold Liabe, Disclose Source), Apache License 2.0 (Apache-2.0) (Commercial Use, Modify, Distribute, Sublicense, Include Copyright Notice, Use Trademark, Include License, State Changes, Warranty, Hold Liabe), Mozilla Public License 2.0 (MPL-2) (Commercial Use, Modify, Distribute, Sublicense, Include Copyright Notice, Use Trademark, Include License, Warranty, Hold Liabe, Disclose Source), and GNU Lesser General Public License v2.1 (LGPL-2.1) (Commercial Use, Modify, Distribute, Sublicense, Include Original, Non-Static Linkage, Hold Liabe). A 'feedback' button is visible on the right side of the page.

## 2. License Compatibility Matrix

- License compatibility issues arises when developers combine code from different sources into a single work.



*Incoming Licenses = A + B + C*

*Outgoing License(s) = ?*

**License(s) ?**

# License Compatibility Matrix

---

- **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

---

<u>Compatible</u> <u>With:</u>	License-A	License-B	License-C	License-D	License-E	License-F	License-G
License-A	X				X	X	
License-B		X					
License-C			X				
License-D		X		X			X
License-E					X		
License-F			X			X	
License-G	X						X

***Only Top 10-15 Used Licenses.***

# License Compatibility Matrix: Elaborate Example

		I want to release a project under:					
		GPLv2 only	GPLv2 or later	GPLv3 or later	LGPLv2.1 only	LGPLv2.1 or later	LGPLv3 or later
I want to copy code under:	GPLv2 only	OK	OK <a href="#">[2]</a>	NO	OK: Convey project under GPLv2 only <a href="#">[7]</a>	OK: Convey project under GPLv2 only <a href="#">[7][2]</a>	NO
	GPLv2 or later	OK <a href="#">[1]</a>	OK	OK	OK: Convey project under GPLv2 or later <a href="#">[7]</a>	OK: Convey project under GPLv2 or later <a href="#">[7]</a>	OK: Convey project under GPLv3 <a href="#">[8]</a>
	GPLv3	NO	OK: Convey project under GPLv3 <a href="#">[3]</a>	OK	OK: Convey project under GPLv3 <a href="#">[7]</a>	OK: Convey project under GPLv3 <a href="#">[7]</a>	OK: Convey project under GPLv3 <a href="#">[8]</a>
	LGPLv2.1 only	OK: Convey code under GPLv2 <a href="#">[7]</a>	OK: Convey code under GPLv2 or later <a href="#">[7]</a>	OK: Convey code under GPLv3 <a href="#">[7]</a>	OK	OK <a href="#">[6]</a>	OK: Convey code under GPLv3 <a href="#">[7][8]</a>
	LGPLv2.1 or later	OK: Convey code under GPLv2 <a href="#">[7][1]</a>	OK: Convey code under GPLv2 or later <a href="#">[7]</a>	OK: Convey code under GPLv3 <a href="#">[7]</a>	OK <a href="#">[5]</a>	OK	OK
	LGPLv3	NO	OK: Convey project and code under GPLv3 <a href="#">[8][3]</a>	OK: Convey code under GPLv3 <a href="#">[8]</a>	OK: Convey project and code under GPLv3 <a href="#">[7][8]</a>	OK: Convey project under LGPLv3 <a href="#">[4]</a>	OK
I want to use a library under:	GPLv2 only	OK	OK <a href="#">[2]</a>	NO	OK: Convey project under GPLv2 only <a href="#">[7]</a>	OK: Convey project under GPLv2 only <a href="#">[7][2]</a>	NO
	GPLv2 or later	OK <a href="#">[1]</a>	OK	OK	OK: Convey project under GPLv2 or later <a href="#">[7]</a>	OK: Convey project under GPLv2 or later <a href="#">[7]</a>	OK: Convey project under GPLv3 <a href="#">[8]</a>
	GPLv3	NO	OK: Convey project under GPLv3 <a href="#">[3]</a>	OK	OK: Convey project under GPLv3 <a href="#">[7]</a>	OK: Convey project under GPLv3 <a href="#">[7]</a>	OK: Convey project under GPLv3 <a href="#">[8]</a>
	LGPLv2.1 only	OK	OK	OK	OK	OK	OK
	LGPLv2.1 or later	OK	OK	OK	OK	OK	OK
	LGPLv3	NO	OK: Convey project under GPLv3 <a href="#">[9]</a>	OK	OK	OK	OK

# License Compatibility Matrix: Look at the Sources

---

- **GNU.org**
  - **Apache.org**
  - **CreativeCommons.org**
  - **Etc.**
- 
- **If you can't find an answer, email them directly.**

### 3. License 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]

# License Classification – Example 2

Another example of classification system:

<u>Permissive</u>	<u>Modifications to be released</u>	<u>Patent Clause</u>	<u>Not Allowed</u>
License-A License-B License-C License-D	License-E License-F License-G	License-H License-I License-K	License-L License-M
<b>Notes:</b> Source code licensed under these licenses is pre-approved and can be combined with proprietary software.	<b>Notes:</b> Modifications made to source code licensed under these license must be released back	<b>Notes:</b> Due to patent clause, you must discuss with legal counsel about your planned usage.	<b>Notes:</b> 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 a 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 originating from third party software providers
- Component 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

## Legend



Proprietary



3<sup>rd</sup> Party Commercial



GPL



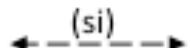
LGPL



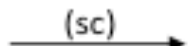
FOSS Permissive



Function call



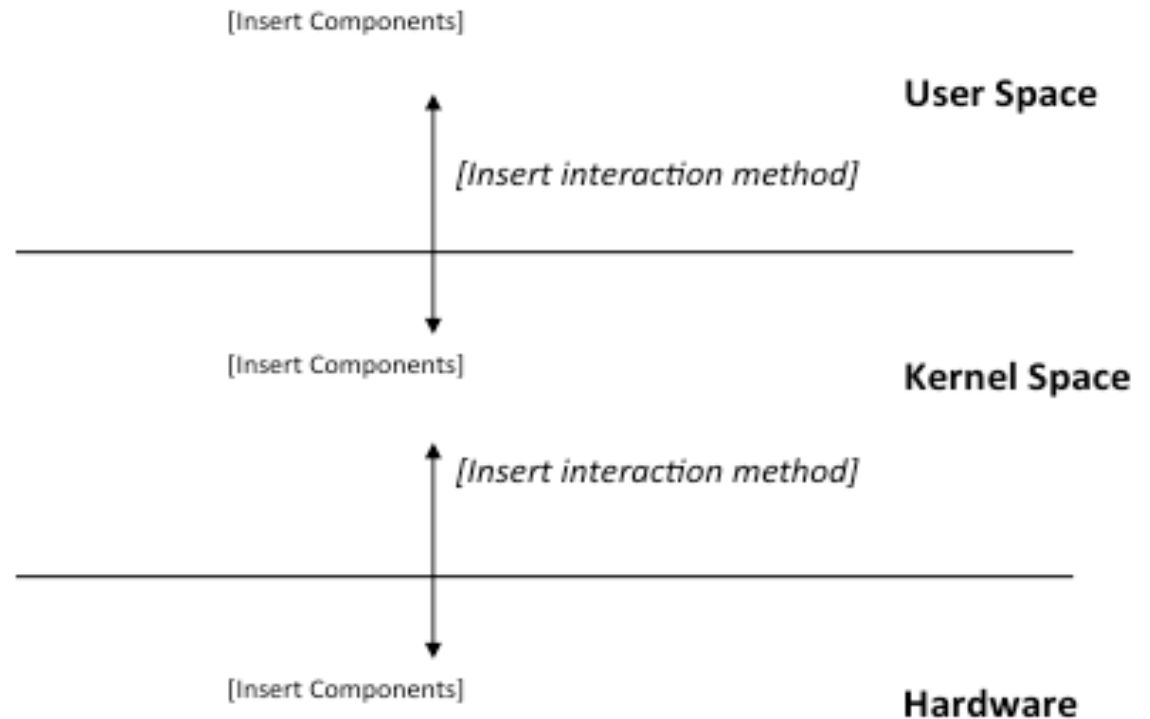
Socket interface



System call



Shared headers





# Software Interactions

<u>Can Dynamically Link To</u>	License-A	License-B	License-C	License-D
License-A	X	X	X	X
License-B		X		X
License-C	X		X	
License-D		X	[Requires approval]	X

<u>Can Statically Link To</u>	License-A	License-B	License-C	License-D
License-A	X		X	
License-B		X	[Requires approval]	
License-C	X		X	
License-D	[Requires approval]			X

# 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

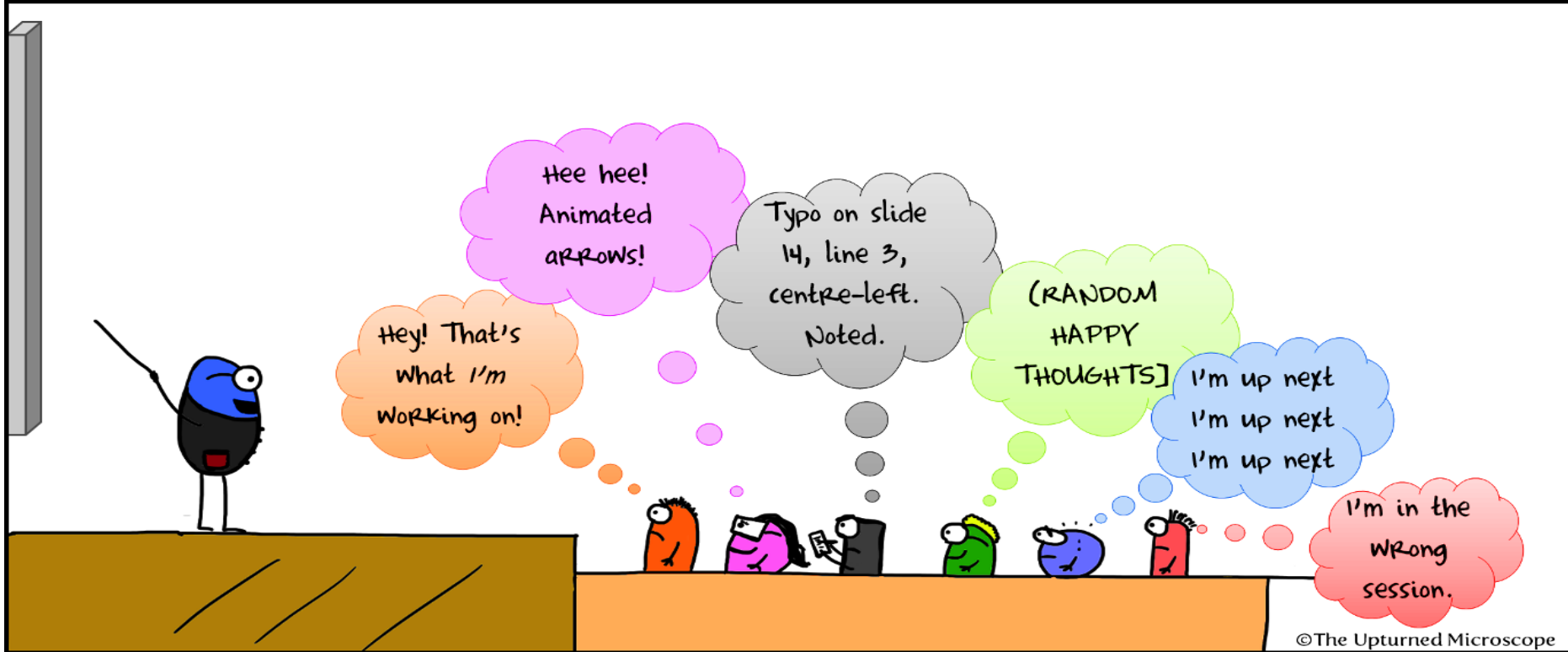


# Benefits to Providing Practical Legal Advice

---

- **Engineers**
  - Easy access to commonly asked questions / use cases / scenarios
  - Minimize frustration surrounding open source *legal stuff*
- **Legal Counsels**
  - Increase bandwidth of Legal Counsel supporting open source
  - Act as enablers to the adoption and use of open source software
- **Company**
  - Documented open source legal practical guidelines, Do's and Don'ts
  - Fewer legal bottlenecks in enabling open source adoption and usage
  - Increased focus on practical open source legal advice

## What people think about during your **conference talk**



# Thank you.

Ibrahim Haddad, Ph.D.  
Open Source Group  
Samsung Research America  
@IbrahimAtLinux