

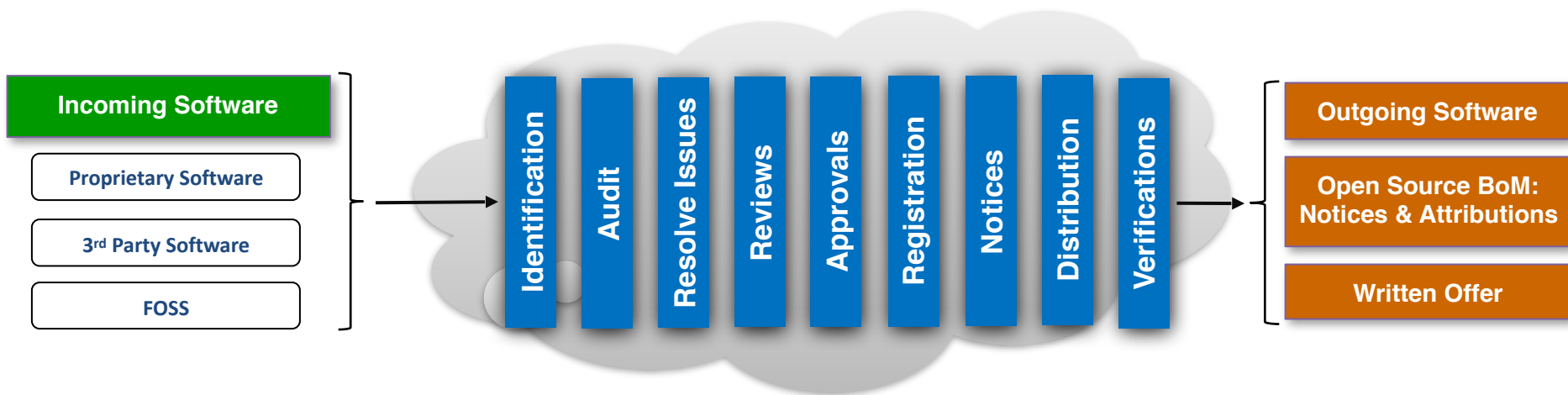
Suggestions to Scale your Open Source Legal Support

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

Disclaimers



Generic Usage / Compliance Process



- 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.

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">Commercial UseModifyDistributeSublicenseWarranty	<ul style="list-style-type: none">Use TrademarkHold Liable	<ul style="list-style-type: none">Include Copyright NoticeInclude LicenseState Changes

None of the information on TLDRLegal 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.


Browse Software Licenses & Summaries

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


Filters:

Commercial Use Modify Distribute Sublicense Include Copyright Notice Use Trademark Include Original Include License Private Use Rename


State Changes Warranty Non-Static Linkage Hold Liable Disclose Source Incomplete

 MIT License


Commercial Use Modify Distribute Sublicense Include Copyright Notice Hold Liable

 Microsoft Public License (Ms-PL)


Commercial Use Modify Distribute Sublicense Include Copyright Notice Use Trademark Hold Liable

 GNU General Public License v2 (GPL-2)


Commercial Use Modify Distribute Sublicense Include Original Warranty Hold Liable Disclose Source

 GNU General Public License v3 (GPL-3)


Commercial Use Modify Distribute Sublicense Include Original State Changes Warranty Hold Liable 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 Liable

 Mozilla Public License 2.0 (MPL-2)

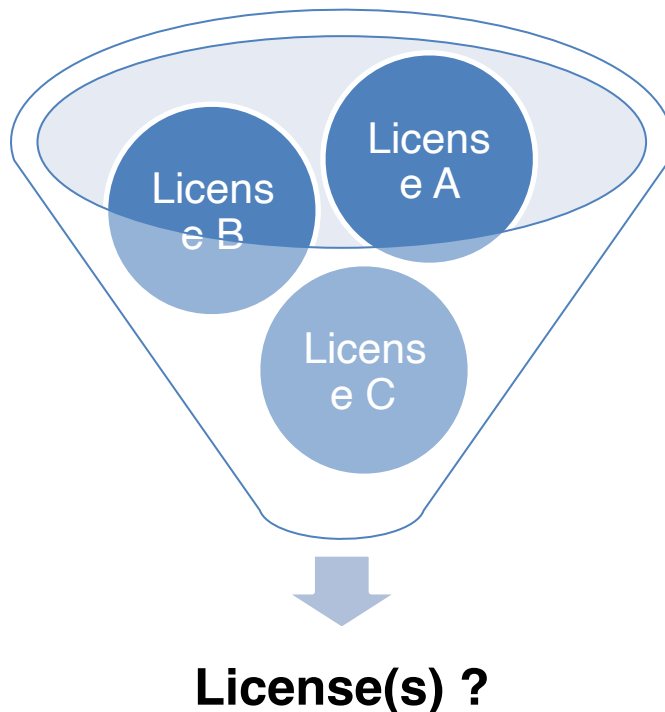
Commercial Use Modify Distribute Sublicense Include Copyright Notice Use Trademark Include License Warranty Hold Liable Disclose Source

 GNU Lesser General Public License v2.1 (LGPL-2.1)

Commercial Use Modify Distribute Sublicense Include Original Non-Static Linkage Hold Liable

2. Compatibility Matrix

- 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) = ?

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>Is Compatible 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

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 [2]	NO	OK: Convey project under GPLv2 only [7]	OK: Convey project under GPLv2 only [7][2]	NO
	GPLv2 or later	OK [1]	OK	OK	OK: Convey project under GPLv2 or later [7]	OK: Convey project under GPLv2 or later [7]	OK: Convey project under GPLv3 [8]
	GPLv3	NO	OK: Convey project under GPLv3 [3]	OK	OK: Convey project under GPLv3 [7]	OK: Convey project under GPLv3 [7]	OK: Convey project under GPLv3 [8]
	LGPLv2.1 only	OK: Convey code under GPLv2 [7]	OK: Convey code under GPLv2 or later [7]	OK: Convey code under GPLv3 [7]	OK	OK [6]	OK: Convey code under GPLv3 [7][8]
	LGPLv2.1 or later	OK: Convey code under GPLv2 [7][1]	OK: Convey code under GPLv2 or later [7]	OK: Convey code under GPLv3 [7]	OK [5]	OK	OK
	LGPLv3	NO	OK: Convey project and code under GPLv3 [8][3]	OK: Convey code under GPLv3 [8]	OK: Convey project and code under GPLv3 [7][8]	OK: Convey project under LGPLv3 [4]	OK
I want to use a library under:	GPLv2 only	OK	OK [2]	NO	OK: Convey project under GPLv2 only [7]	OK: Convey project under GPLv2 only [7][2]	NO
	GPLv2 or later	OK [1]	OK	OK	OK: Convey project under GPLv2 or later [7]	OK: Convey project under GPLv2 or later [7]	OK: Convey project under GPLv3 [8]
	GPLv3	NO	OK: Convey project under GPLv3 [3]	OK	OK: Convey project under GPLv3 [7]	OK: Convey project under GPLv3 [7]	OK: Convey project under GPLv3 [8]
	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 [9]	OK	OK	OK	OK

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] |

License Classification – Example 2

● Another example of classification system:

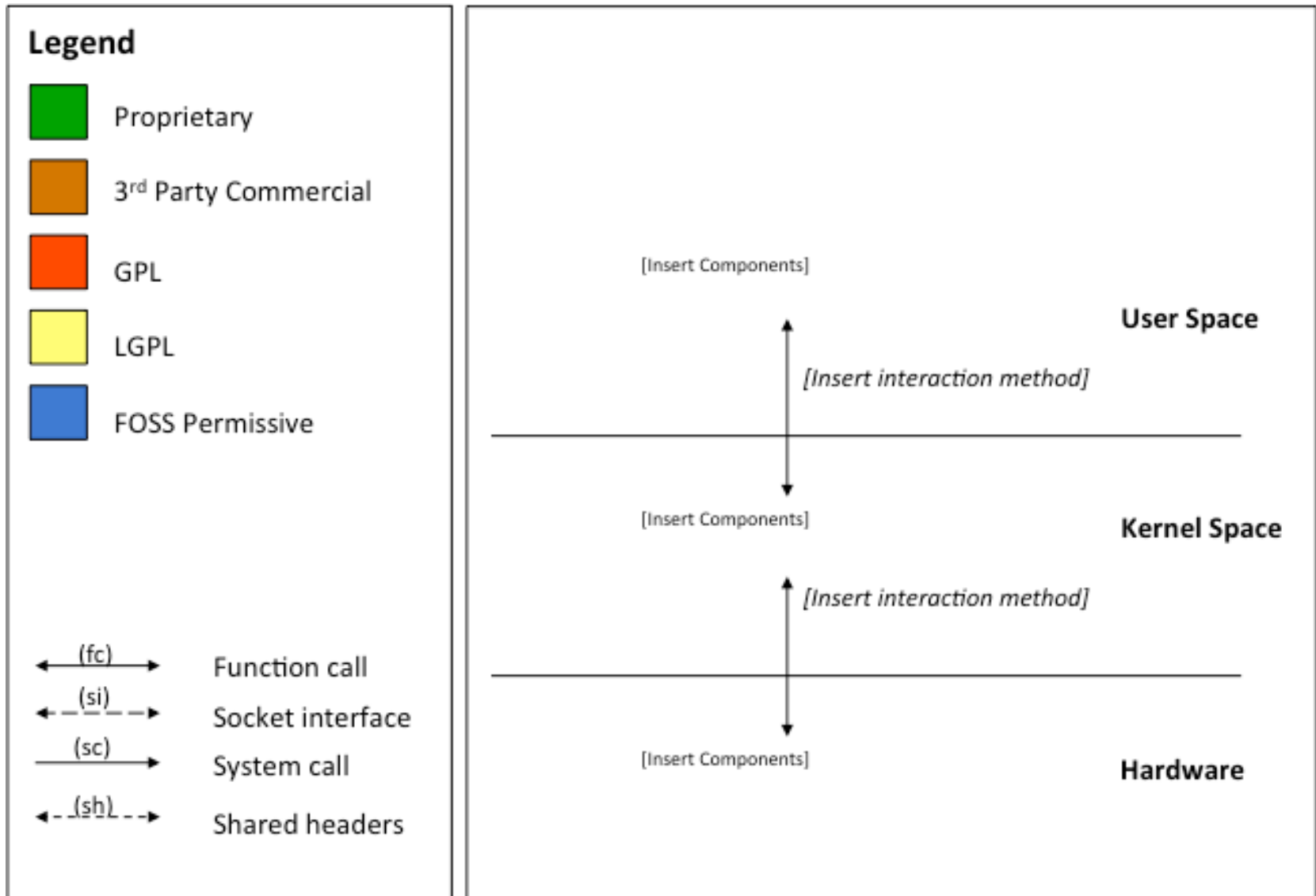
<u>Permissive</u>	<u>Modifications to be released</u>	<u>Patent Clause</u>	<u>Not Allowed</u>
License-A License-B License-C License-D	License-E License-F License-G	License-H License-I License-K	License-L License-M
Notes: Source code licensed under these licenses is pre-approved and can be combined with 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



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

- **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