SAMSUNG RESEARCH AMERICA

Implementing and Managing Open Source Compliance Programs

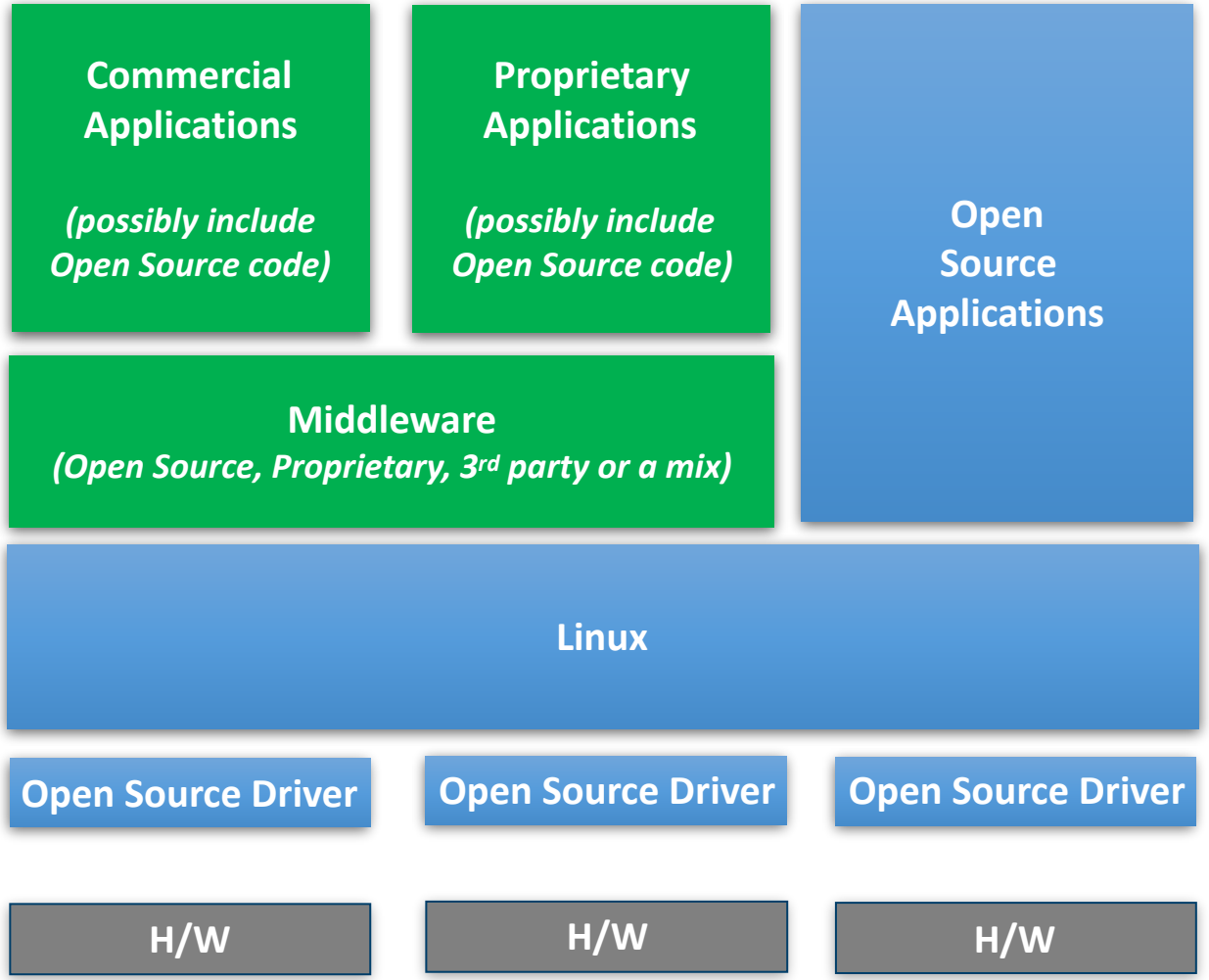
Ibrahim Haddad, Ph.D.
VP of R&D, Head of Open Source

Twitter: @IbrahimAtLinux
Web: IbrahimAtLinux.com

Open Compliance Summit
Yokohama, November 2017

Open source in modern software platforms

- 3rd Party Commercial
- Proprietary
- Open Source
- Proprietary + 3rd Party Commercial
- Proprietary + Open Source
- 3rd Party Commercial + Open Source
- Proprietary + Open Source + 3rd Party Commercial



Mitigation of risks through compliance practices

Protect your intellectual property and that of your customers and suppliers.

- Identify the origin and license of used software.
- Identify license obligations.
- Fulfill license obligations when products ship.

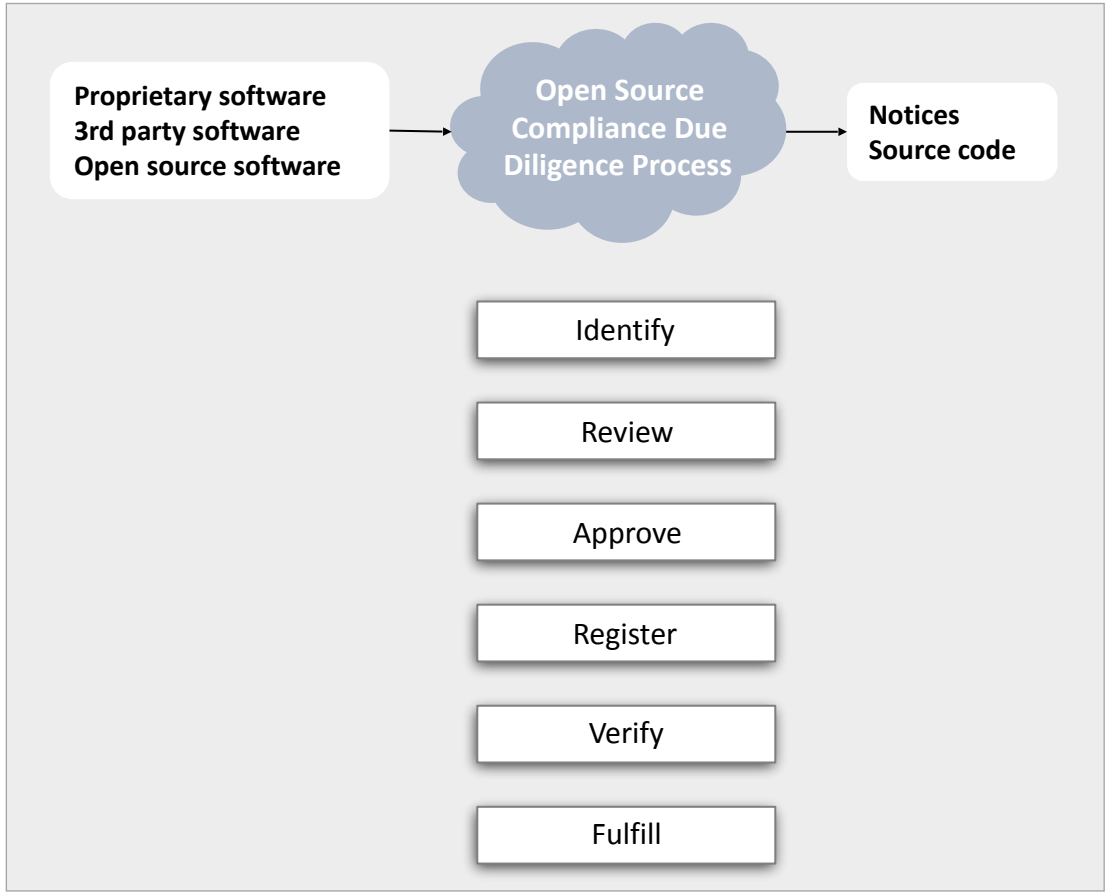
OPEN SOURCE CONSUMPTION AND COMPLIANCE PROGRAM ELEMENTS

Executive Sponsor + Financial Commitment

Strategy	Portals	Policy	Process	Development	Team	Education	Inventory	Communication	Tools	Initiatives
Compliance	Internal site (Educational)	Universal Policy	Universal Process	Integrate compliance checkpoints in the development and QA process	Compliance teams (core and support)	Training on company policy	Inventory management	Internal messaging	Source code scanning	OpenChain
Managing Inquiries	External site (Obligation fulfillment, source code distribution)	Distribution	Distribution	Integrate compliance tools with your build systems	Scoreboard and success metrics	Guidelines and best practices	Inventory of 3rd party code	External messaging	Linkage analysis	SPDX
Legal (Risk tolerance)		Auditing	Auditing			Training on open source licenses			Project management	Open Compliance Program
M&A, Corporate Development		Documentation	Documentation			New employee orientation			Bill of Material	TODO Group
Software Procurement		Notices	Notices			Checklist for product team			Automation for online forms and workflow	
		Usage	Usage			Checklist for developers				
		Company policy on licenses	Obligation Fulfillment			Checklist for SW procurement				
		Company policy on mixing code				Compliance mentorship				
						Professional formal training				
						Invited speakers				
						Brown bag seminars				

Compliance End-to-End Process

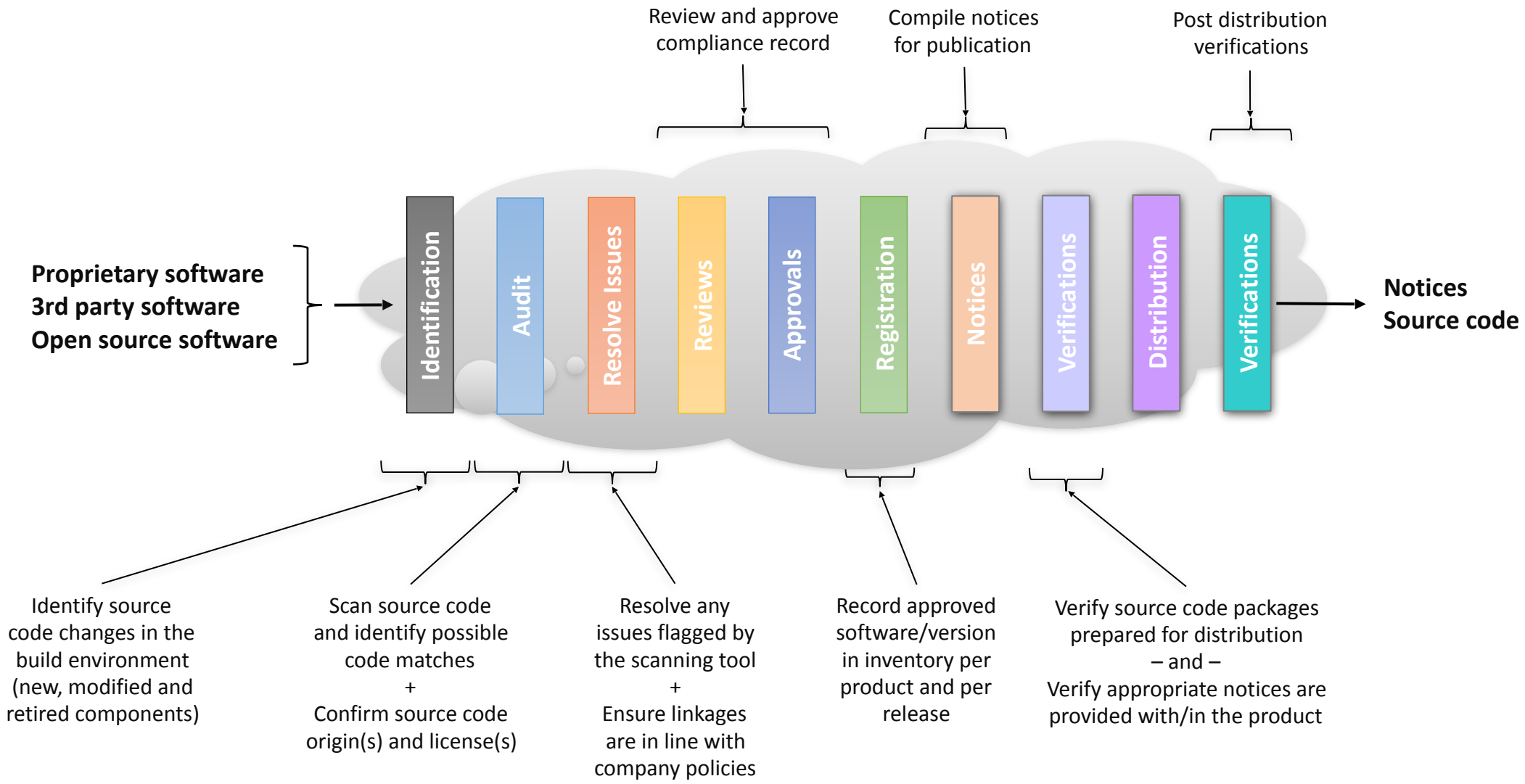
- Compliance due diligence involves the following:
  - OSS used in the product has been identified, reviewed and approved.
  - The product implementation includes only the approved OSS.
  - OSS used in the product have been registered in the OSS inventory system.
  - Obligations related to the use of licensed material have been identified.
  - Notices have been provided in the product documentation (written offer, attributions and copyright notices).
  - Source code including modifications (when applicable) are ready to be made available once the product ships.
  - Verifications of all the steps in the process.



Compliance end-to-end process – high level overview

Compliance End-to-End Process

Customize to your own environment



Metrics to evaluate source code scanning tools

There are a number of companies providing open source compliance tools and services. The question of what tool is best for a specific usage model and environment always comes up, especially around the time of license renewal. These few questions will help you create a comparison matrix for the tools you're comparing in an effort to make it as objective as possible to compare the tools and arrive to a decision with least subjectivity.

- Size of the knowledge base against which scanned code is being compared.
- Frequency of updates to the knowledge base.
- Support for different audit models / methods (traditional, blind, DIY).
- Speed of scans for same loads.
- Deployment models (local, cloud, hybrid).
- Ability to identify snippets (down to x lines).
- Ability to do auto-identification to avoid spending endless hours on manual labor.
- Support for discovery of vulnerabilities (there are two methods and only one is really meaningful when combined with the compliance context).

- Cost to deploy tool in terms of # of servers needed for your specific install.
- A simplified and intuitive UI that makes it easy and inviting to use the tool – minimizing learning curve.
- Support for APIs and a CLI that you can connect to for ease of integration with existing development and build systems.
- Ability to use the tool for M&A transactions.
- Programming languages agnostic.
- Licensing cost and cost for private customizations.

When a Vendor Discloses OSS, What do They Need to Tell You?

- Package Name
- Version
- Original download URL
- License
- Copyright notices
- Attribution notices
- Source code including modifications
- Included dependencies

- Development team's point of contact
- Inclusion of technology subject to export control
- <add more disclosures as needed>

What Should be Verified About the Disclosure?

Completeness, consistency, and accuracy.

- Use scanning and identification tools whenever the source code is available.
- Does the declared licensees match what's in the code files?
- Do version numbers match?
- Do the licenses truly permit the proposed use of the software?

Recommended open source compliance practices w.r.t. software suppliers

General

- Reform software acquisition agreements.
- Verify disclosures via efficient tooling and automation processes.

OpenChain-related

- Get suppliers to certify with OpenChain to understand their compliance adherence.
- Set targets similar to – any SW supplier need to meet Level 1 by 12/2017, Level 2 by 6/2018, Level 3 by 6/2019. etc.
- Mandate use of compliance tools.

Compliance Failures and How to Avoid Them

Unapproved “Copy/Paste”

Inclusion of open source code into proprietary or 3rd-party code (or vice versa)

Description	Discovery	Avoidance
This type of failure occurs during the development process when engineers add open source code into proprietary or 3rd party source code via copy and paste into proprietary or 3rd party software (or vice versa).	This type of failure can be discovered by scanning the source code to identify all open source code, their origin and license.	• Offer training. • Conduct regular source code scanning for the complete source code base.

Unapproved Linkages

Linking OSS into Proprietary Source Code (or vice versa)

Description	Discovery	Avoidance
This failure occurs as a result of linking software (OSS, 3rd party, proprietary) that have conflicting or incompatible licenses.	This failure can be discovered using a linking discovery tool that allows you to discover links between different software components	• Offer training to engineering staff to avoid linking software components with conflicting licenses. • Continuously run dependency tracking tools over build environment.

Source Code Related Compliance Failures

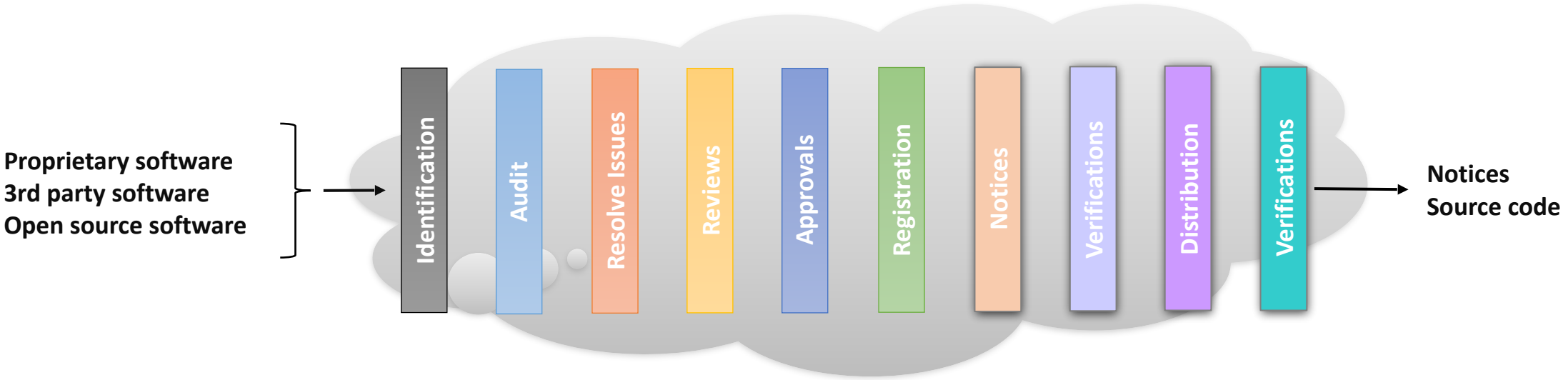
Description	Avoidance
Failure to publish source code	Milestone - part of product shipment checklist
Source code versioning failures	Add a verification step into the development / compliance process.
Failure to Publish Source Code Modifications	Add a verification step into the development / compliance process. 1. Use a bill of material difference tool that allows you to identify what software components have changed between different releases.
Failure to mark OSS source code modifications	1. Offer training to engineering staff. 2. Add a verification step into the development / compliance process. 3. Conduct source code inspections before releasing the source code.

Compliance Process Failures

Description	Prevention
Failure to submit a request to use open source	1. Conduct periodic full scans. 2. Training. 3. Include compliance in employee performance reviews.
Failure to take open source training	Mandatory – attached to performance metrics.
Failure to audit source code	1. Provide proper staffing. 2. Training. 3. Enforce periodic audits.
Failure to resolve audit findings	Implement a policy in the compliance management system that doesn't allow it to close a compliance ticket if it has open sub-tasks or issues.
Failure to submit OSRB form on time	1. Training. 2. Require form as soon as component or code is being evaluated.

Recommended Practices

Compliance End-to-End Process



Identification

- Identify **all the components and snippets** included in the product and their origin.
- Print out and retain the license information at the time you download the software.
- Double check that the license terms in the source distribution match the ones described on the project web site.
- If you cannot identify a license, ask legal to identify it for you.
- Document all changes to open source code.

Source Code Auditing

Scan all source code.

- Scan early and often. This practices allows you to:
  - Keep the delta with the previous scan to a minimum.
  - Discover compliance problems as they occur.
  - Provide solutions to discovered problem within acceptable delays.
  - Perform incremental scan in a very efficient way.
- Scan newer versions of previously approved packages:
  - Each time engineers modify a previously approved component or plan to use a previously approved component in a different product, the source code of the modified component is re-scanned and the component has to go through the approval process again.

Resolving Issues Identified by the Audit

- When in doubt, discuss with engineering and in some cases you may need to discuss with tool vendors if you suspect an unusual tool behavior.
  - The zlib test
- Inspect and resolve each file or snippet flagged by the scanning tool
- Identify if your engineers made any code modifications.
  - Don’t rely exclusively on engineers to remember if they made code changes.
  - Use tools to identify code changes, who made them and when.
- Re-scan the code after engineering has resolved a given issue to get a confirmation and a clean BoM.
- Provide legal with all information you discovered on the licensing of the specific component (COPYING, README, or LICENSE files).

If You Can’t Comply

4 possible scenarios to consider.

Remove and replace

Can you live without this code? Is there an alternative project with same function under a different license?

Re-engineer

Can you create a work around?

Version tracking

Is there a newer (or older) version of this code under a different license?

Re-license

Can you contact the author(s) and ask for an exception / different license?

Notices

- Companies using OSS in their products need to provide appropriate notices
  - Full License Text
  - Copyright Notices
  - Attribution Notices
  - Written offer / Information on Obtaining the Source Code

- Be clear, direct in the language of the written offer and inclusive of all OSS included in your product.

Presentation of the Notices

- Provide the written offer and notices in the product manual, on the web site, and inside the product.
- In some instances, depending on the product in question, the product may have a graphical user interface or a command line administrative interface; in this instance, you can also provide the option to display the attributions on the product (such as a mobile phone).
- For product updates, such as over-the-air (OTA) update for mobile phones, the notices must also be updated as part of the product updates when the update includes new or updated OSS components.

Verifications

- Due to the large number of verifications steps, we consider it a best practice to develop checklists that cover all the verification steps and the compliance team follows to ensure consistency and to ensure that no verification steps is overlooked.

Engineering guidelines

Good programming practices are also legal best practices.

General Guidelines to Engineers 1/3

- Fill out an request form for each open source software you are using in product, service or SDK.
- Save the web page from which you downloaded the package.
- Save a mint copy of the original package.
- Consult with compliance team when you upgrade your open source software version.
  - License changes can occur between versions.
- Do not check un-approved source code into any source tree without approval.
- Document your modification to open source packages following the change log practice of the project.
- Do not re-naming open source modules.

General Guidelines to Engineers 2/3

- Do not send modifications to any public source tree without getting proper approval(s).
  - Follow company policy/process.
- Do not discuss coding or compliance practices with persons outside the company.
- Document the interfaces between any code you write and other components.

General Guidelines to Engineers 3/3

- Copy/Paste
  - Do not copy/paste OSS code into proprietary or third party source code or vice versa without proper approval.
  - Follow your company's policy.

- Mixing Source Code with Different Licenses
  - Mixing of code coming under different OSS licenses must be avoided when licenses are incompatible with each other.
  - When in doubt, always refer to the FSF resource page on license compatibility available at http://www.fsf.org/licensing/licenses/index_html.
  - The OSRB must review all cases where more than one type of OSS license is used and provide approval on a case-by-case basis.

General considerations – recommended practices

General Considerations 1/2

- Compliance Verification Golden Rule
  - Compliance is verified on a product-by-product basis: Just because a OSS package is approved for use in one product does not necessarily mean it will be approved for use in a second product.

- Source Code Comments
  - Do not leave any inappropriate comments in the source code that includes private comments, product code names, mention of competitors, etc.

- Existing Licensing Information
  - Do not remove or in any way disturb existing OSS licensing copyrights or other licensing information from any OSS components that you use. All copyright and licensing information is to remain intact in all OSS components.

General Considerations 2/2

- Inbound Clean Bill of Material
  - Ensure that any in-bound software is not contaminated with OSS.
  - Always audit source code you received from your software providers or alternatively make it a company policy that software providers must deliver you a source code audit report for any source code you receive.

- Open Source in M&As
  - Understand the OSS implications of any software of an entity to be acquired as part of the due diligence performed prior to approval the corporate transaction.

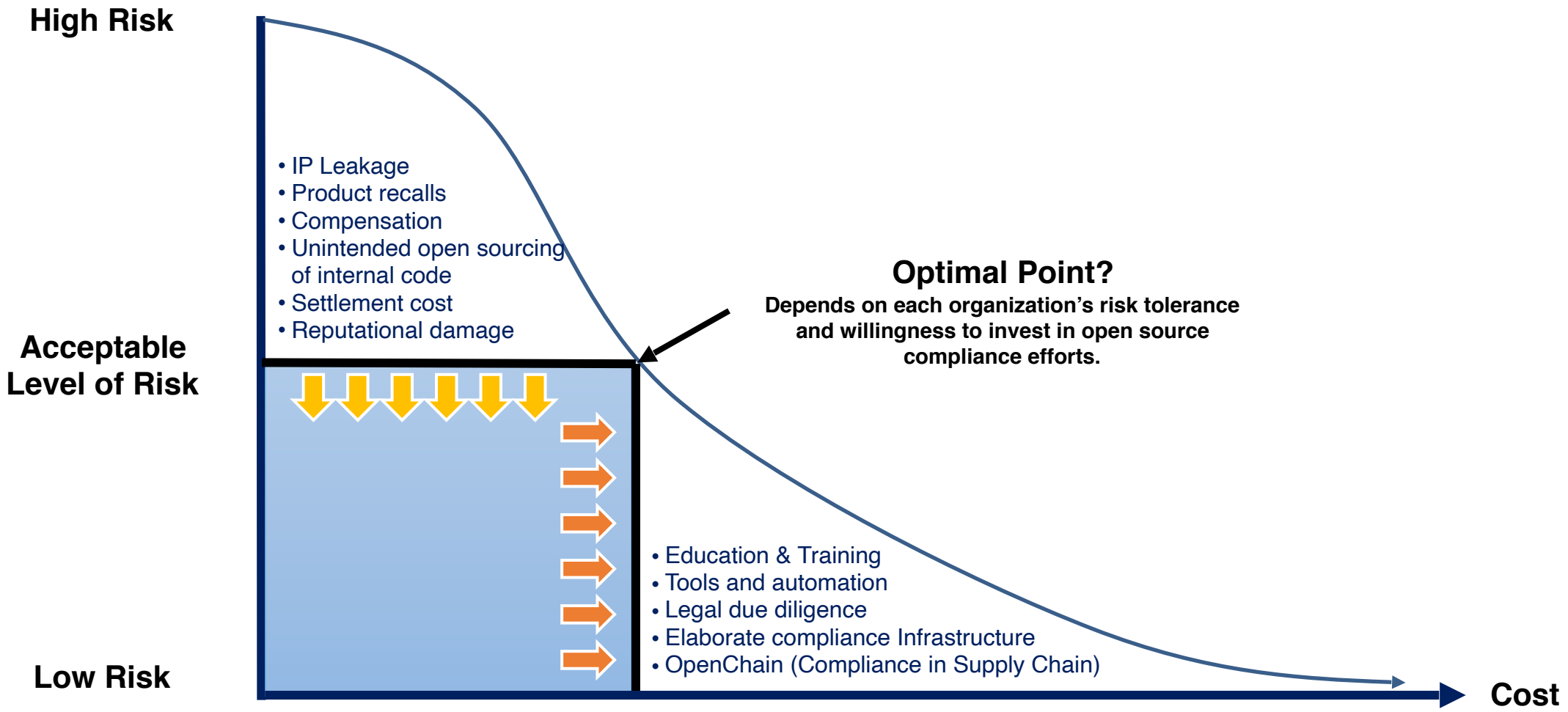
Distribution Considerations

- Ensure that source code subject to OSS distribution obligations is ready prior to distribution and ship acceptance.

- All modified GPL and LGPL files and any associated files that are required to build GPL and LGPL components must be available for distribution.
  - If a file is required in order to build a GPL or an LGPL component, it becomes a dependency for that component. Therefore, it must be part of the source code release and will be bound under the GPL or LGPL.

Compliance: A Balancing Act

Open Source Compliance: How Good is Good Enough?



Final Thoughts

- We’ve come a long way in open source compliance.
- We learned a lot.
- Much of open source compliance challenges today are related to scalability and cost.

Next Frontier

- How can we minimize the costs associated with being in compliance?
  - Costs = resources, tools, IT, support staff.
  - Minimize to eliminate common errors.
  - Adopt automation and tooling.
  - Increase education. Mandate training and internal certification to be eligible for promotions.
  - Improve practices with vendors in your software supply chain.
- How can we provide a consistent, repeatable approach that helps companies achieve proper compliance?
- How can we have a common method to assess compliance programs?
  - OpenChain.

1 Suggestion

- Collect recommended practices per area and publish them as part of an OpenChain educational package.
- Create consistency between companies on DO’s and DON’T’s of open source compliance.
- I volunteer to provide draft 0.1 with my content.

SAMSUNG RESEARCH AMERICA

Implementing and Managing Open Source Compliance Programs

Ibrahim Haddad, Ph.D.
VP of R&D, Head of Open Source

Twitter: @IbrahimAtLinux
Web: IbrahimAtLinux.com

Open Source Compliance Summit
Yokohama, November 2017