SAMSUNG RESEARCH AMERICA

Implementing and Managing Open Source Compliance Programs – A Crash Course

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

Twitter: @IbrahimAtLinux
Web: IbrahimAtLinux.com

Open Source Strategy Forum
November 14, , 2018 – London

Disclaimers

- I am not a legal counsel.
- This presentation does not offer legal advice.
- This presentations expresses my own views, and do not necessarily reflect those of my current or any of my previous employers.
- This is a large deck. Please excuse any typos.

Sections

1. Introduction to open source compliance
2. Compliance failures and how to avoid them
3. Overview of an open source compliance program
4. End-to-end compliance process
5. Challenges and solutions
6. Compliance teams
7. Recommended practices
8. Scaling open source legal support
9. Dealing with compliance inquiries
10. Open source compliance in M&A transactions
11. OpenChain Initiative

Open Source Compliance in the Enterprise

A practical guide for organizations on how best to use open source code in products and services, and participate in open source communities, in a legal and responsible way.

Key takeaways:

- How to structure an open source management program
- Set an open source compliance strategy
- Create compliance policies, tools, and processes
- When and how to involve executives and legal experts
- How to train and incentivize developers
- Common compliance tools and processes
- Open source license management best practices
- Much more!

http://www.ibrahimatlinux.com/

Additional resources

Free and Open Source Software Compliance: The Basic You Must Know

Establishing Free and Open Source Software Programs: Challenges and Solutions

A Five-Step Compliance Process for FOSS Identification and Review

Achieving FOSS Compliance in the Enterprise

A Glimpse into Recommended Practices in FOSS Compliance Management Process

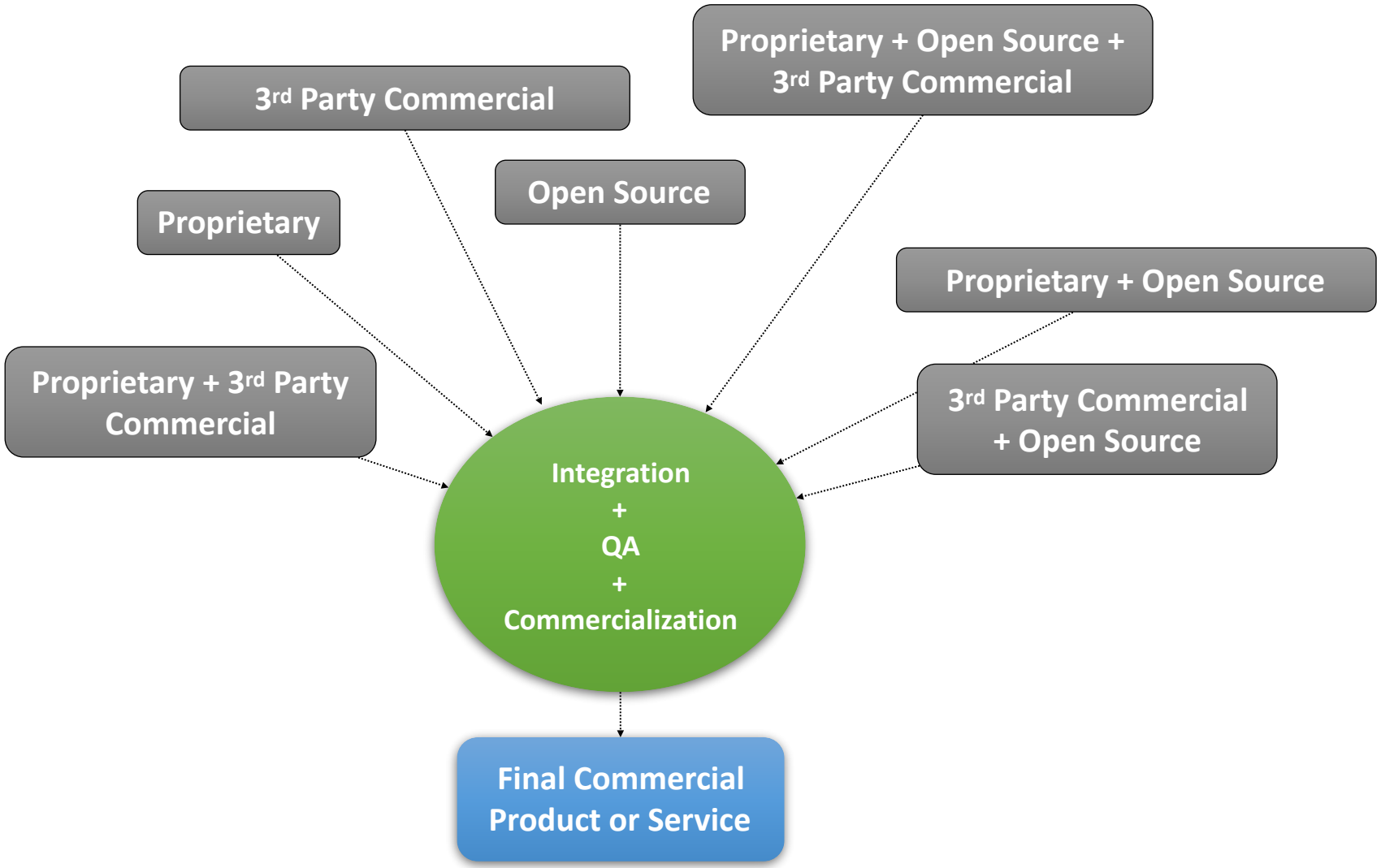
Free and Open Source Software Compliance: Who Does What

Publishing Source Code for FOSS Compliance: Lightweight Process and Checklists

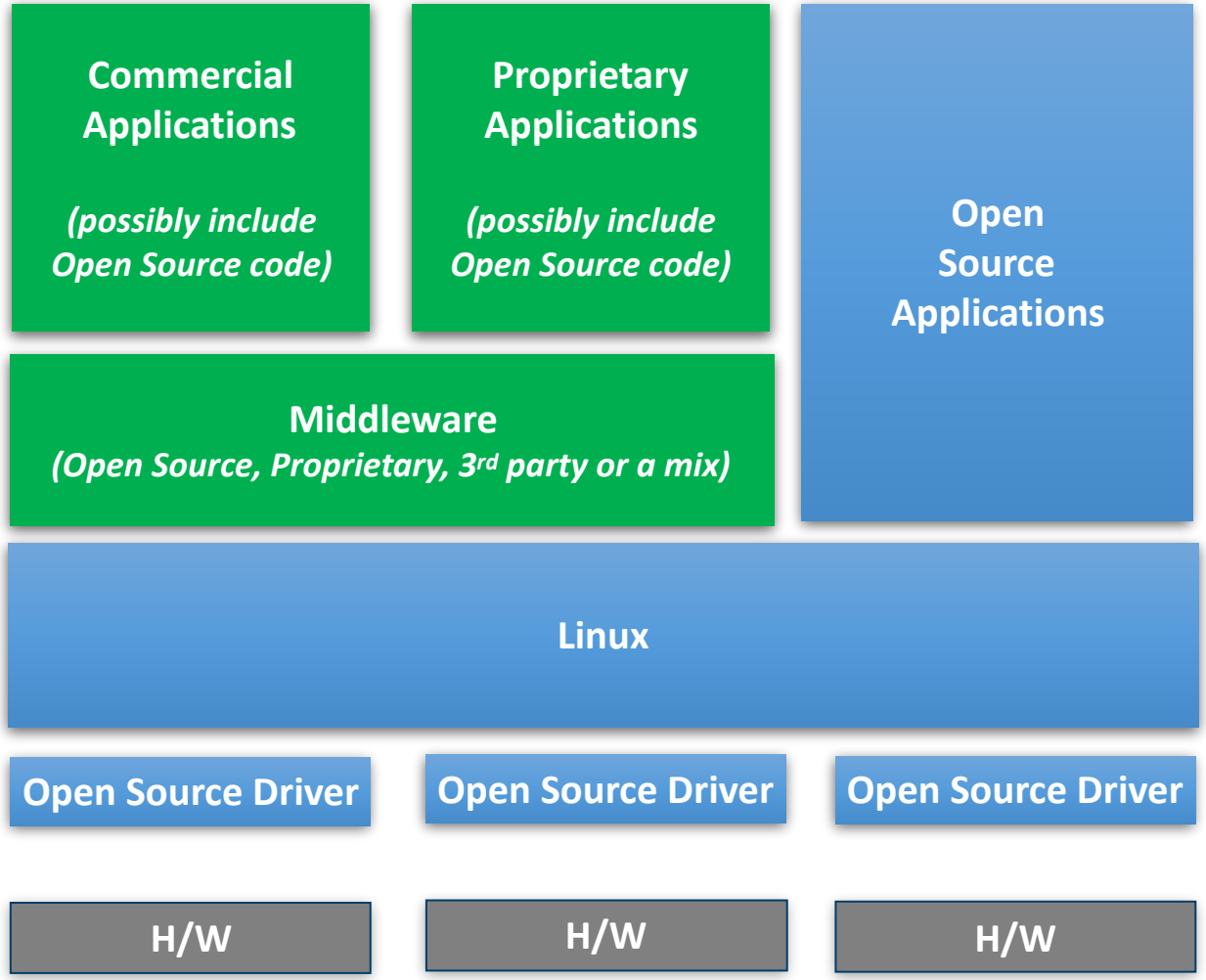
http://www.ibrahimatlinux.com/

Introduction

Multi-source development model



Use of open source in modern platforms



- Licenses are not negotiated.
- There are potentially tens or hundreds of licenses involved.
- The business environment is not as predictable as a commercial environment.
- There are potentially thousands of contributors to the various OSS used
- The origin of some components may not clear.
- Maintenance and support are variable and self-service.
- Security vulnerabilities are visible to everyone.
- Risks are mitigated through design, engineering practices and compliance

Mitigation of risks through compliance practices

- Identification of the origin and license of used software.
- Identification of license obligations.
- Fulfillment of license obligations when products ship.

What is OSS Compliance?

- Open source compliance refers to the aggregate of:
  - Policies
  - Processes
  - Training
  - Tools

- This enables an organization to effectively use open source software and contribute to open communities while
  - Protecting copyrights,
  - Complying with license obligations,
  - Comply with third party software supplier contractual obligations, and
  - Protecting the organization’s intellectual property and that of its customers and suppliers.

What are the Basic Compliance Obligations That Must Be Satisfied?

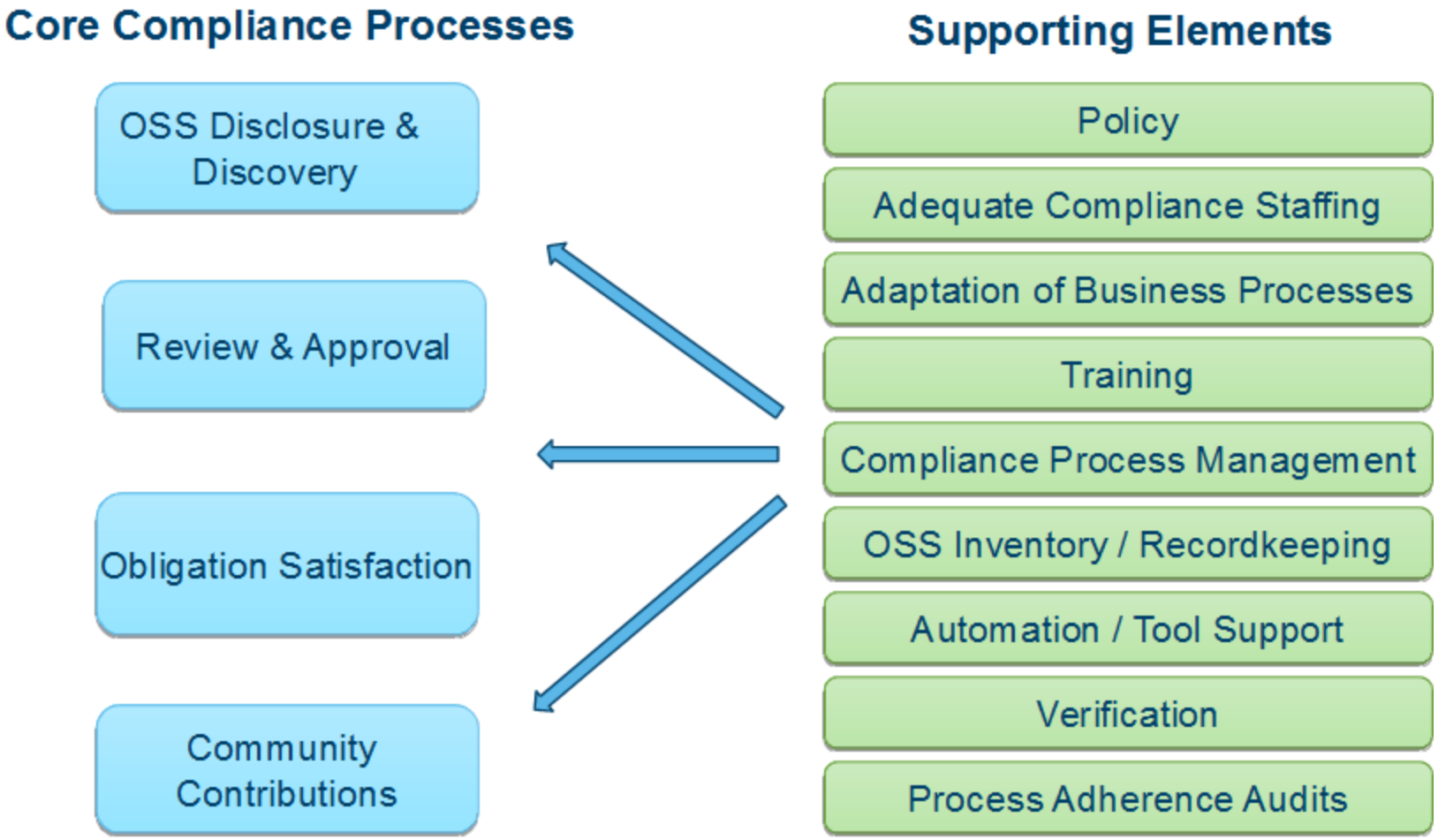
- Obligations differ per license.
- OSS license obligations generally are triggered when external distribution occurs.

  Discuss with your open source counsel.
- Source code scan and analysis is necessary to clarify obligations.
  - Need to know components, snippets, licenses and usage model.

Benefits to Achieving Compliance

- Increased understanding of the benefits of OSS and how it impacts your organization.
- Increased understanding of the costs and risks associated with using OSS.
- Build a relationship with the OSS community and organizations through involvement with OSS projects and participation in OSS events.
- Better preparation for possible acquisition, sale, and the launch of new products or services.
- Improved overall open source strategy.

The Compliance Approach in a Nutshell



What Source Code Must be Scanned and Audited?

- All software.

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

- Package Name
- Version
- Original download URL
- License and license documentation
- Description
- Modified code
- Included dependencies
- Intended use in product

- Development team's point of contact
- Availability of source code
- Where source code will be maintained
- Whether the package has previously been approved for use in another context
- License obligations
- Inclusion of technology subject to export control
- Etc.

What Else Should the Supplier Disclose?

- Other items that might be necessary for license obligations
  - Copyright notices and attributions
  - License text
  - Source code (including modifications the supplier made) for open source software that carries an obligation to offer source code to recipients.

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?

Ensure Compliance Prior to Product Shipment

- To avoid being challenged on OSS compliance, companies must make compliance a priority before products ship.

- Companies must establish and maintain consistent compliance policies and procedures, and ensure that OSS licenses, proprietary licenses, and 3rd party licenses can coexist before shipment.

Ensure Compliance Prior to Product Shipment

- To that effect, companies need to implement an end-to-end OSS management infrastructure that allows it to
  - Identify all OSS used in products,
  - Collect applicable OSS licenses for legal department review,
  - Institutionalize OSS and compliance training to ensure all employees are aware of company policies and the legal risks associated with using OSS,
  - Ensure that software vendors, suppliers, and subcontractors are adhering to OSS license requirements, and
  - Have an understanding of which OSS licenses are being used and also how they are being used.

Compliance Failures and How to Avoid Them

3 General Types of Compliance Failures

(my own classification not a legal advice)

- Intellectual property failures
- License failures
- Compliance process failures

Intellectual Property Failures

- These failures evolve around the contamination of proprietary, third party, or OSS code with source code that comes with incompatible or conflicting licenses.

- Such failures may result in companies losing their product differentiation through the release of the source code to the OSS community.

Examples of Intellectual Property Failures

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

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 (components, snippets), their origin and license.	• Offer training to engineering staff to bring awareness to compliance issues and to the different types and categories of OSS licenses and the implications of including OSS source code in proprietary source code. • Conduct regular source code scanning for all the source code used in your products or services.

Examples of Intellectual Property Failures

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.

Possible Results from Intellectual Property Failures

- An injunction that prevents the company from shipping a product.
- A requirement to publish a company's proprietary source code under an open source license.
- Significant re-engineering to eliminate compliance issue.
- Public embarrassment and tension in relationships with distributors, 3rd party software providers, and OSS communities.

License Compliance Failures

- Less damaging than intellectual property failures.

- License compliance failures may result in:
  - An injunction that prevents a company from shipping a product until source code is published.
  - Support or customer service headaches as a result of version mis-matches.
  - Embarrassment and/or bad publicity with customers and OSS suppliers.

License Compliance Failures

Description	Avoidance
Failure to publish source code	Avoided by making source code availability a checklist item in the product release cycle before the product becomes available in the marketplace.
Source code versioning failures	Avoided by adding a verification step into the compliance process to ensure the exact version of source code that corresponds to the distributed binary version is being published.
Failure to Publish Source Code Modifications	1. Use of a checklist. 2. Use a bill of material difference tool that allows you to identify what software components have changed between different releases. • Re-introduce revised software components into the compliance process. • Add the “compute diffs” of any modified OSS to the checklist item before releasing OSS used in the product.

License Compliance Failures

Description	Avoidance
Failure to mark OSS source code modifications	1. Offer training to engineering staff. 2. Add source code marking as a checklist item before releasing the source code. 3. Conduct source code inspections before releasing the source code. 4. Add milestones in the compliance process to verify that changed source code has been marked as such.

Compliance Process Failures

- When the compliance process fails to function correctly any one of the intellectual property or license compliance failures might occur and bring their respective consequences.

- Additionally, compliance process failures also tend to
  - Negatively impact product development and release schedules, and
  - Introduce bugs due to undocumented component version skew.

Compliance Process Failures

Description	Avoidance	Prevention
Failure to submit a request to use open source	Avoided by offering compliance training to engineering staff on the company's open source policies and processes. If an OSS component is found in the build system and does not have a corresponding compliance ticket, then a new ticket should be regenerated.	1. Conduct periodic full scans of the software platform to detect any undeclared open source code. 2. Offer training to engineering staff on the company's open source policies and processes. 3. Include compliance in employee performance reviews.
Failure to take open source training	Avoided by ensuring that the completion of compliance training is part of employees' professional development plans and is monitored for completion as part of performance reviews.	Mandate engineering staff to take open source compliance training by a specified date.

Compliance Process Failures

Description	Avoidance	Prevention
Failure to audit source code	1. Conduct periodic source code scans and audits. 2. Ensure that auditing is a milestone in the iterative development process.	1. Provide proper staffing as to no fall behind schedule. 2. Enforce periodic audits.
Failure to resolve audit findings	Avoided by not allowing compliance tickets to be resolved (i.e. closed) if the audit report is not finalized. A compliance ticket is closed only if there are no open sub-tasks or open issues attached.	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	Avoided by filing requests early even if engineering did not yet decide on the adoption of the OSS source code.	Prevented through better education and training.

Overview of the Enterprise Compliance Program

OPEN SOURCE IN THE ENTERPRISE

CONSUMPTION AND COMPLIANCE PROGRAM ELEMENTS

Executive Sponsor + Financial Commitment

Strategy	Portals	Policy	Process	Development	Team	Education	Inventory	Communication	Tools	Organizations
Compliance	Internal site (Educational)	Universal usage and compliance Policy	Universal usage and compliance 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	The Linux Foundation
Managing Inquiries	External site (Obligation fulfillment, source code distribution)	Distribution	Distribution	Integrate compliance tools with build systems	Scoreboard and success metrics	Guidelines and best practices	Inventory of 3rd party code	External messaging	Linkage analysis	OpenChain
Legal (Risk tolerance)		Auditing	Auditing			Training on open source licenses			Project management	SPDX
M&A, Corporate Development		Documentation	Documentation			New employee orientation			Bill of Material	Open Compliance Program
Software Procurement		Notices	Notices			Checklist for product team			Automation for online forms and workflow	TODO Group
		Usage	Usage			Checklist for developers			IP evaluation tool	Software Freedom Law Center
		Company policy on open source licenses	Obligation Fulfillment			Checklist for SW procurement				Open Source Initiative
		Company policy on mixing code				Compliance mentorship				Free Software Foundation
						Professional formal training				Software Freedom Conservancy
						Invited speakers				
						Brown bag seminars				

End-to-End Compliance Management

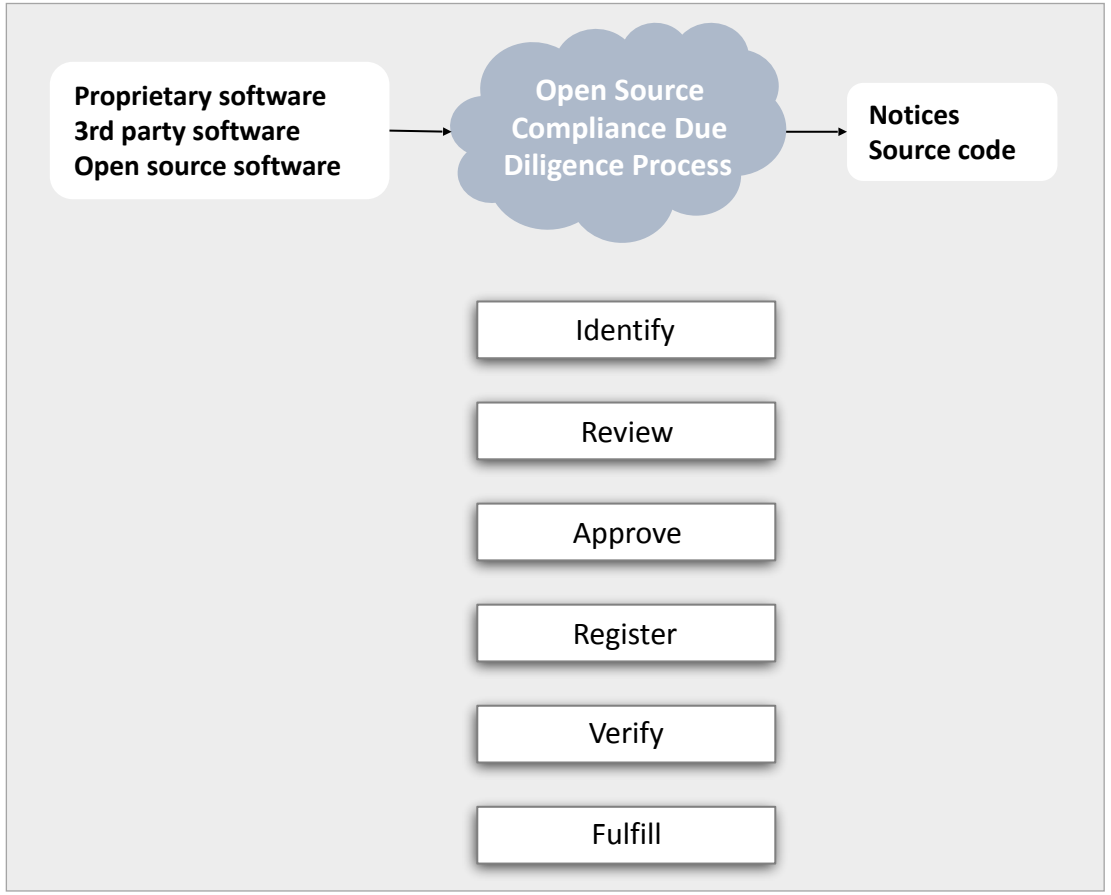
Introduction

- Compliance management consists of a set of actions that control the intake and distribution of OSS used in commercial products.

- The result of compliance due diligence is an identification of all OSS used in the product and a plan to meet the OSS license obligations.

Compliance End-to-End Process

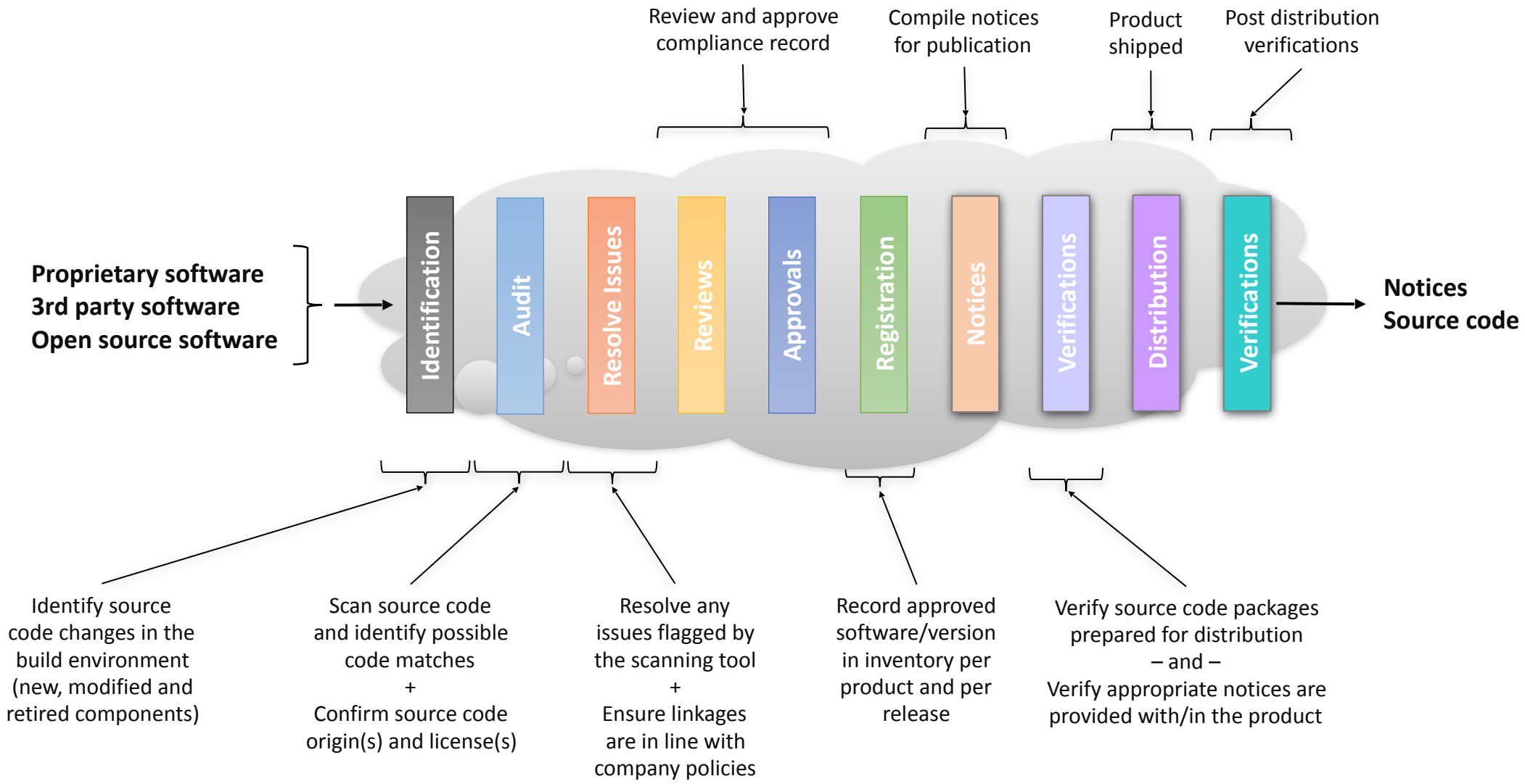
- Compliance due diligence involves the following:
  - OSS used in the product has been identified, reviewed and approved.
  - The product implementation includes only the approved OSS.
  - OSS used in the product have been registered in the OSS inventory system.
  - Obligations related to the use of 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.



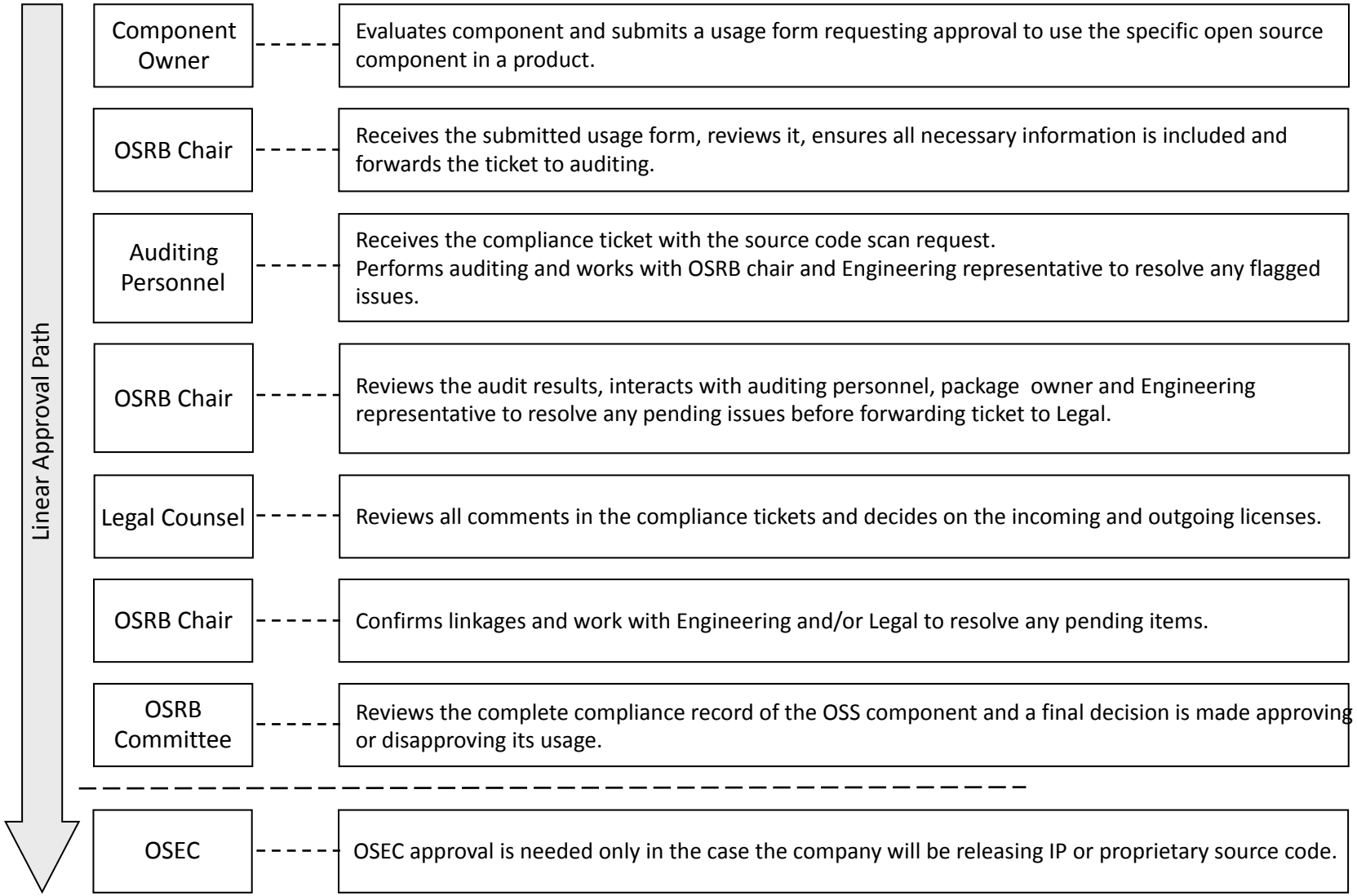
Compliance end-to-end process – high level overview

Open Source Compliance End-to-End Process

Customize to your own environment



Compliance Ticket Reviewers

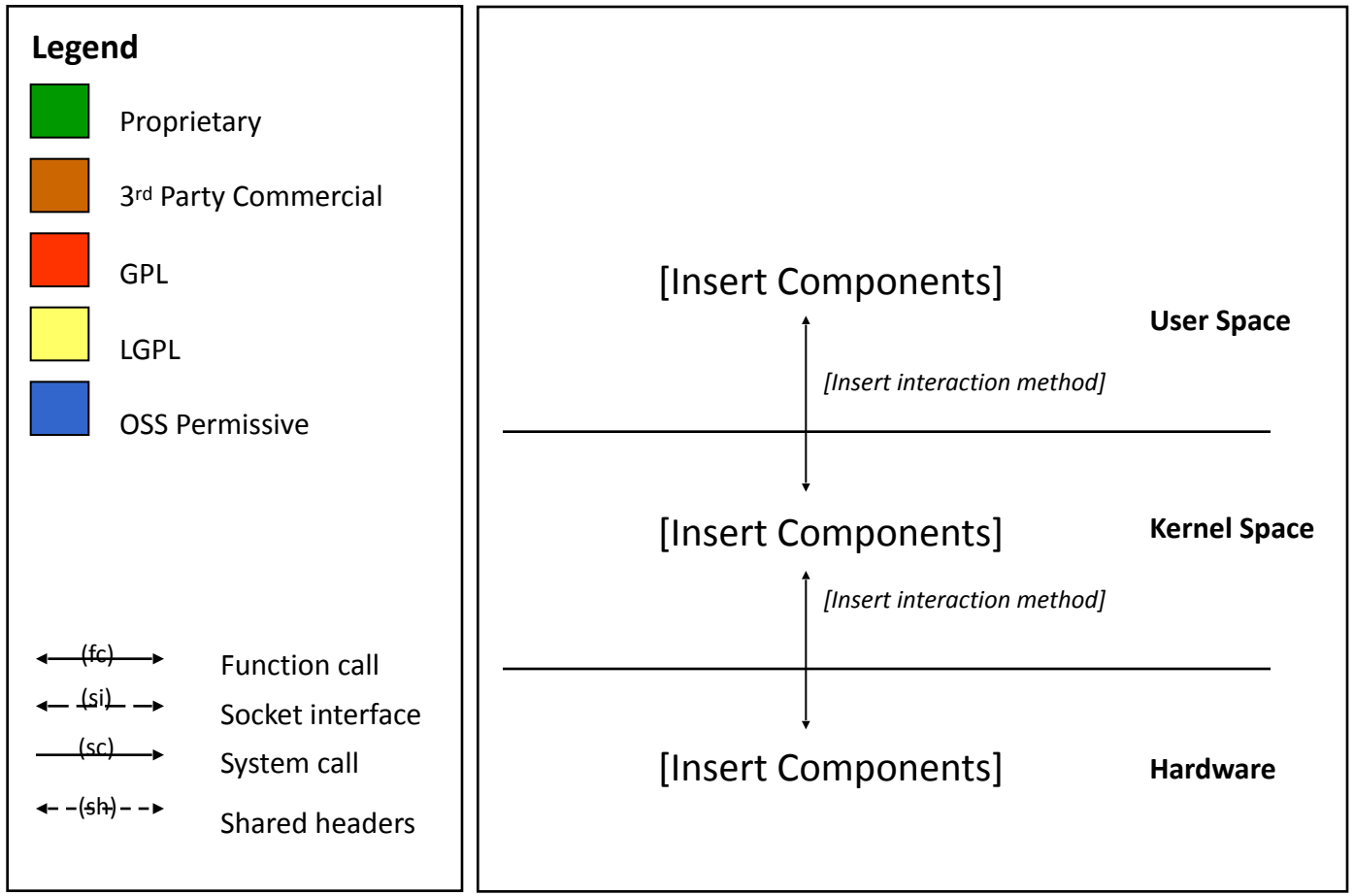


Architecture Review

- The goal with the architecture review is to analyze the interactions between the OSS code and third party and proprietary code.
- The result of the architecture review is an analysis of the licensing obligations that may extend from the OSS components to the proprietary components.
- The internal package owner, the OSRB engineering representative, and the OSS expert usually perform the architecture review. If they identify a dependency resulting in a licensing conflict, the OSRB Chair will issue ticket to engineering to resolve it.

Architecture Review - Template

Customize to your own preferences



Compliance Programs: Challenges and Solutions

Common Challenges

1. Create a compliance program that achieves the right balance between processes and product shipment deadlines
2. Think long term, while executing short term
3. Communicate on compliance
4. Establish a clean software baseline
5. Maintain compliance for evolving products
6. Institutionalize and sustain compliance efforts

Creating a Compliance Program

Description

- Create a compliance infrastructure while achieving the right balance between processes and product shipment deadlines.
  - Often, compliance activities are viewed as a burden to the development process.
  - Compliance activities that are well integrated into software management processes typically improve development efficiency

How to tackle it?

- Executive-level commitment and support.
- Simple and clear policy and process, communicated across the company.
- Incorporate compliance as part of development processes.
- Mandate the usage form for any planned use of OSS.
- Mandate architecture reviews.
- Mandate code inspections/reviews.
- Enforce due diligence on software received from 3rd party .

Thinking Long Term, While Executing Short Term

Description

- The priority of all companies is to ship products on time, at the same time they should also build and expand their internal open source compliance infrastructure.

How to tackle it?

- Plan a complete compliance infrastructure to meet long term goals, and implement pieces that are needed for short term execution.
- Expect to build your compliance infrastructure as you go while doing it the right way and keeping its scalability for future activities and products in mind.
  - Light-weight processes
  - Incorporate compliance as part of the development process
  - Pick the right source code scanning tool that ties into your code repos and build systems

Communicating Compliance

Description

- Ensuring that all company employees are aware of the risks associated with the inclusion of OSS in commercial product and ensuring they are well educated about the company's compliance policies, processes and guidelines.

- Ensuring that the OSS community is aware that you are undertaking serious efforts to ensure you meet the license obligations of the various OSS you are using in a commercial product.

How to tackle it?

- Internal Communication
  - All-hands meetings.
  - Formal training mandated to all employees.
  - Brown-bag OSS and compliance seminars.
  - OSS company newsletter.
  - Internal portal hosting company policies, procedures and a discussion forum related to OSS and compliance.
- External Communication
  - Web site for distribution of OSS packages with contact form.
  - Reaching out and supporting OSS organizations involved in enforcement.
  - Participation in OSS events and conferences.

Establishing a Clean Software Baseline

Description

- Figuring out how OSS software is licensed and where it's being used in your products or platform is a huge challenge.
- This challenge evolves around establishing a clean software baseline for your product or software platform.
- This is an intensive activity over a period of time that can extend for months depending on how soon you started the compliance activities in parallel to development activities.

How to tackle it?

- Tier 1 actions:
  - Full platform source code scan to establish baseline
- Tier 2 actions - supporting:
  - Simple and enforced policies
  - Light-weight processes
  - Tools and automation
  - Dedicated compliance team (or individual)
  - Embed compliance checkpoints in the software development process

Maintaining Compliance for Evolving Products

Description

- There are several challenges in maintaining open source compliance; they are similar to those faced when establishing baseline compliance.
  - In fact, many of the steps are the same, just on a smaller scale.

- Maintaining compliance is a continuous effort, comparatively small, and incremental effort that depends on discipline and commitment to build compliance activities into existing engineering and business processes.

How to tackle it?

- Tools and automation
- Continuous audits
- Enforcement of OSRB usage form (when applicable)
- Continuous due diligence on 3rd party software providers

Institutionalizing and Sustaining Compliance Efforts

Description

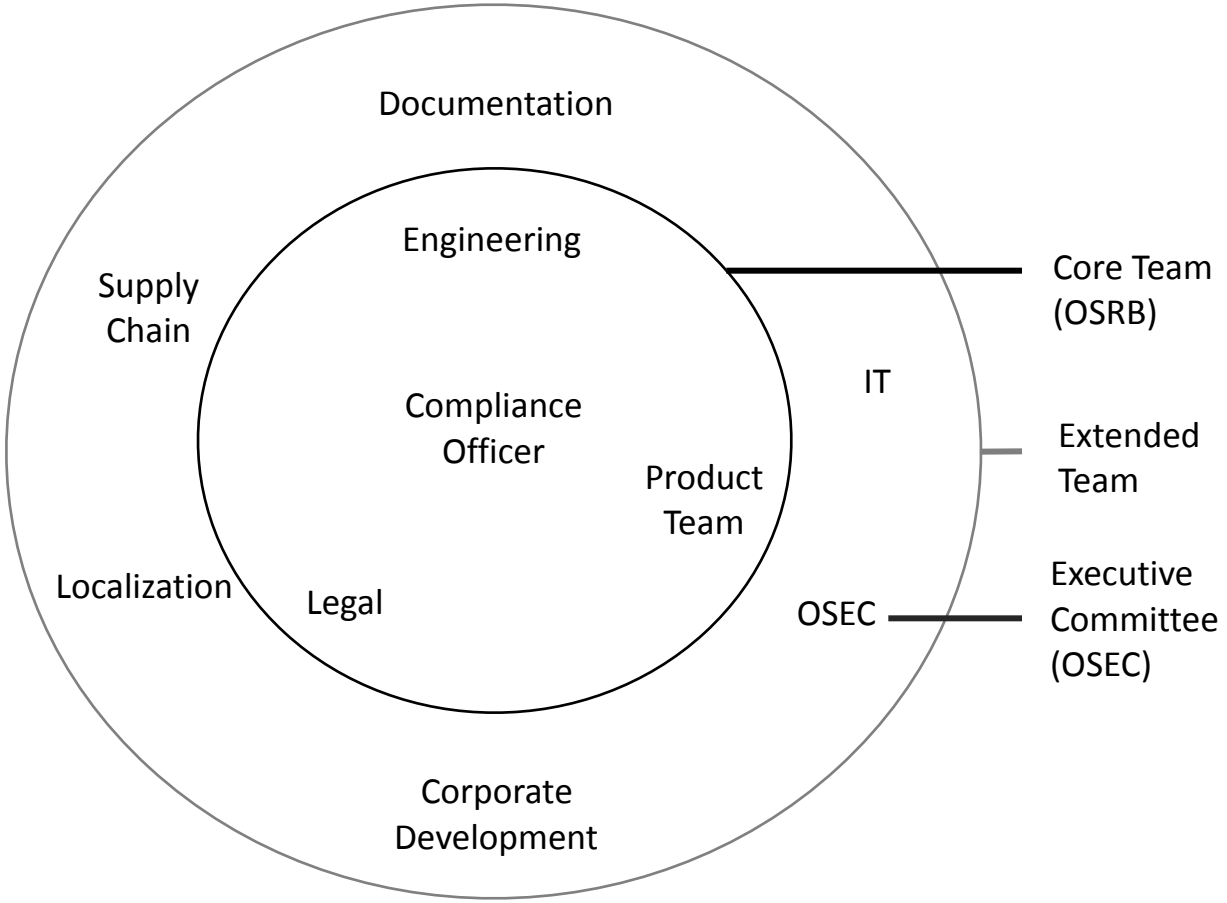
- One challenge faced during OSS compliance is keeping it going and establishing it as an integral part of the software development process.

How to tackle it?

- Executive-level commitment
- Streamlining compliance processes
- Training
- Staffing
- Measurement and analysis
- Achieving consistency across the company
- Enforcement mechanism
- Continuous improvements to the compliance program
- Communicate the productivity advantages that accrue from each program element when educating about the compliance program

Compliance Team

Teams Involved in Ensuring Compliance



Jump to p73

Core Team (OSRB)

Mission of the OSRB

1. Ensure mutual compliance with third party software and OSS licensing obligations by ensuring that all OSS has been identified, assessed, and formally approved and that it's licenses are compatible with the company's intended and actual use

2. Facilitate effective usage of OSS in commercial products within the company through the implementation of clear and light-weight OSS compliance policies, processes and procedures

3. Protect product differentiation while complying with OSS licensing obligations by ensuring that OSS license obligations do not propagate to proprietary software or third party software

Detailed Responsibilities of the OSRB

- Establish the policy
- Establish the compliance end-to-end process: including usage, audit, development, engagement, assurance, and compliance management.
- Create and maintain compliance policies, processes, guidelines, templates and forms used in the compliance program.
- Review requests for use, modification and distribution of OSS: The OSRB reviews incoming requests from engineering and product teams for using OSS and determines approval.
- Perform software audits: The OSRB performs audits on all software included in the product.
- Perform architectural reviews to analyze the interaction between the OSS source code, proprietary code and third party source code. The goal of this review is to ensure that architectural guidelines are respected and that the interactions between OSS, proprietary and third party software are within the acceptable legal guidelines.
- Perform linkage analysis to determine if any OSS license obligations propagates to proprietary or third party software through the linkage method.

Detailed Responsibilities of the OSRB

- Provide guidance on OSS questions incoming from company staff and engineers
- Perform code audits as part of the pre-distribution verification to ensure that OSS license, copyright notices have been intact, and that engineers have updated the change logs to reflect the changes introduced to the source code.
- Compile and maintain list of license obligations for OSS that is approved for use and pass it to appropriate departments for fulfillment
- Handle compliance inquiries sent to the company in relation to OSS compliance
- Review end-user documentation to ensure appropriate notices are given to consumers about OSS included in the product along with a written offer on how to obtain the source code when applicable.
- Recommends new tools to be used as part of the compliance infrastructure that will contribute to making the compliance work more efficient and highly automated.
- Work with the OSEC to determine whether the company needs to disclose or give away intellectual property, or issue a patent non-assert, as part of using OSS

Detailed Responsibilities of the OSRB

- Sign off on all product releases that contains any OSS
- Maintain records of compliance
- Develop and offer OSS and compliance training
- Host the OSS internal and external portals

OSRB Legal Representative Specific Duties

- The Legal representative is a member of the OSRB and provides legal advice and guidance to engineering teams on OSS issues.
- The Legal representative approves or disapprove the usage of a OSS after reviewing the compliance ticket and evaluating the risk factors based on the feedback provided by engineering and the Compliance Officer.

OSRB Legal Representative Specific Duties

- Advise on licensing:
  - Interpret OSS licenses and their obligations
  - Provide OSS license notes to engineering and product teams. In most cases, engineers do not have time to read lengthy licensing text and need a quick summary of most used OSS licenses that highlights the key points.
- Advise on licensing conflicts in relation to incompatible or conflicting licenses
- Resolve IP issues associated with the use of OSS: With OSEC to review and approve the release of IP or the release of proprietary source code as OSS.
- Approve updates to product documentation with respect to OSS notices:
  - Update corporate end user license agreement to include mentions of OSS.

Four Practical Tips for Legal Counsel

1. License Playbooks: An easy to read and digest summary of OSS licenses intended for software developers.

2. License compatibility matrix: An easy method to learn if License-A is compatible with License-B to be used by software developers as they merge code incoming from different projects under different licenses.

3. License classification: An easy way to understand the different licenses in play and the course of action needed when using these licenses.

4. Software interaction methods: A guide to understand how software components interact and if the method of interaction is allowed per company compliance policies.

Engineering Representative Specific Duties

- Engineering and product teams may have one or more representatives that participate in the OSRB to track down all compliance related tasks (sometimes called tickets) assigned to engineering.

- Engineering and products team have several responsibilities with respect to OSS compliance.

Engineering Representative Specific Duties

- Submit requests to use OSS: The engineering and product teams decide what external software to bring into the product baseline, including third party and OSS. Their primary responsibility from a compliance perspective is to submit OSRB usage forms for any OSS that is planned for inclusion in a product.
- Maintain a change log for each modified OSS: As part of meeting the OSS license obligations and depending on the OSS license in questions, some licenses impose the obligation of providing a change log that describes the changes that were introduced to the OSS package
- Follow technical compliance guidelines to architect, design, and implement source code as set by the OSRB.
- Conduct design reviews to discover and remedy any compliance issues in a timely manner.
  - The Compliance Officer drives the design reviews and invites different engineering participants depending on the software component in question.
- Cooperate with OSRB, respond promptly to questions asked by the OSRB, and be prompt in closing internal compliance tickets.

Engineering Representative Specific Duties

- Take OSS training: All engineers must take the available OSS training.
- Monitor the OSS projects to determine whether any bug fixes or security patches have become available and take responsibility for updating the OSS component used in the product.
- Prepare source code packages for distribution as part of meeting the OSS license obligations, in addition to any build scripts or utilities needed to build the OSS that corresponds to the binaries in the software release.
- Integrate compliance milestones as part of the development process: This exercise take places in collaboration with the OSRB and the Compliance Officer.

Compliance Officer

- The Compliance Officer chairs the OSRB and manages the compliance program.

- The compliance officer must possess the following expertise:
  - Understanding of OSS licenses and obligations to discuss with legal counsels
  - Knowledge of industry practices
  - Knowledge and experience in establishing corporate policies and processes
  - Technical knowledge to discuss with developers directly
  - Historical perspective on OSS
  - Knowledge of community consensus and practices
  - Contacts in the OSS community
  - Contacts in the OSS organizations that could be called upon for clarification

Compliance Officer Duties

- Drives the compliance due diligence end-to-end process and acts as the compliance program manager ensuring all compliance related tasks are resolved and there are no compliance issues blocking product shipment
- Coordinates source code scans and drive all auditing issues to closure
- Participates in engineering design reviews, code inspections and distribution readiness assessment to assure that the engineering and product teams follow all compliance processes and policies and conforms to the approved OSRB usage form
- Coordinates with engineering and product team on source code distribution of OSS packages, including preparing and verifying distribution checklist for each OSS package
- Acts as liaison
  - Between OSEC and OSRB
  - Between the engineering and product team and the OSRB and OSEC in regard to usage plan approval processes
- Escalates compliance issues to OSEC
- Reports on compliance activities to the OSEC including flagging any issues that stand in the way of shipping a product

Extended Teams

Open Source Executive Committee (OSEC)

- The Open Source Executive Committee (OSEC) consists of engineering, legal and product marketing executives in addition to the Compliance Officer.

- The OSEC is responsible for:
  - Setting the OSS strategy
  - Reviewing and approving OSS Policy (as developed by the OSRB)
  - Reviewing and approving release of IP
  - Providing approvals to release proprietary source code to the OSS community under a specific OSS license

Documentation Team

- The documentation team is responsible for ensuring that all documentation requirements for OSS used are met:
  - Appropriate notices (copyrights and attributions) are included in the product documentation
  - A written offer to provide source code for included OSS is included where required

Compliance Officer	Prepares the draft notices document based on the final software bill of material in addition to a proposal on where and how these notices should be presented.
Legal Counsel	Reviews, edits and approves proposal from the Compliance Officer.
Documentation Team	Update product documentation as approved by Legal.

Localization Team

- Responsible for translating OSS notices into the supported languages (depending on the countries in which the product will be available).
- Industry practice:
  - Keep OSS licenses in their native language
  - Supply notices in language of localized releases
- Free Software Foundation Position:
  - Does not provide or approve any translations of the GNU GPL, LGPL, AGPL, and FDL into other languages.
    - Verifying the translation is a difficult and expensive task and often involves the help of bilingual lawyers in other countries
    - If a translation error occurs, the results could be catastrophic.
- Recommendation:
  - If you want to provide translations of OSS license, label your translations as unofficial.

Localization Team

- The FSF gives permission to publish translations of the GNU GPL, LGPL, AGPL, and FDL into other languages on two conditions:
  - The translation must be labeled as unofficial to inform people that they do not count legally as substitutes for the authentic version
  - You agree to install changes at FSF's request, if they identify that changes are necessary to make the translation clearer
- According to the FSF, to label translations as unofficial, you need to add the following text at the beginning, both in English and in the language of the translation.

This is an unofficial translation of the GNU General Public License into language. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU GPL—only the original English text of the GNU GPL does that. However, we hope that this translation will help language speakers understand the GNU GPL better.

—Free Software Foundation (http://www.gnu.org/licenses/translations.html)

Replace language with the name of that language, and "GNU General Public License" and "GPL" with the name and abbreviation of the license you are translating, if it is not the GPL.

Supply Chain

- You must know what goes into all of the product's software, including software provided by outside suppliers.
- Supply Chain personnel are usually involved in moving software from the suppliers to your company.
- Supply chain procedures must be updated to address the acquisition and use of OSS.
  - Examine software supplied to you by 3rd party software providers.
- Supply Chain can support OSS compliance activities by:
  - Mandating 3rd party software providers to disclose the OSS being delivered to your company
  - Assisting with licensing-in 3rd party software that is bundled with OSS packages

Supply Chain

- Mandate third party software providers to disclose OSS used in their offering along with a statement on how they plan to meet the applicable OSS license obligations.

- It is not sufficient to point at the supplier and inform the OSS community that meeting OSS license obligations is the responsibility of the supplier.

IT

- IT provides support and maintenance for the tools and automation infrastructure used by the compliance program.

- IT support includes:
  - Servers hosting the various tools
  - Tools
  - Mailing lists
  - Web portals

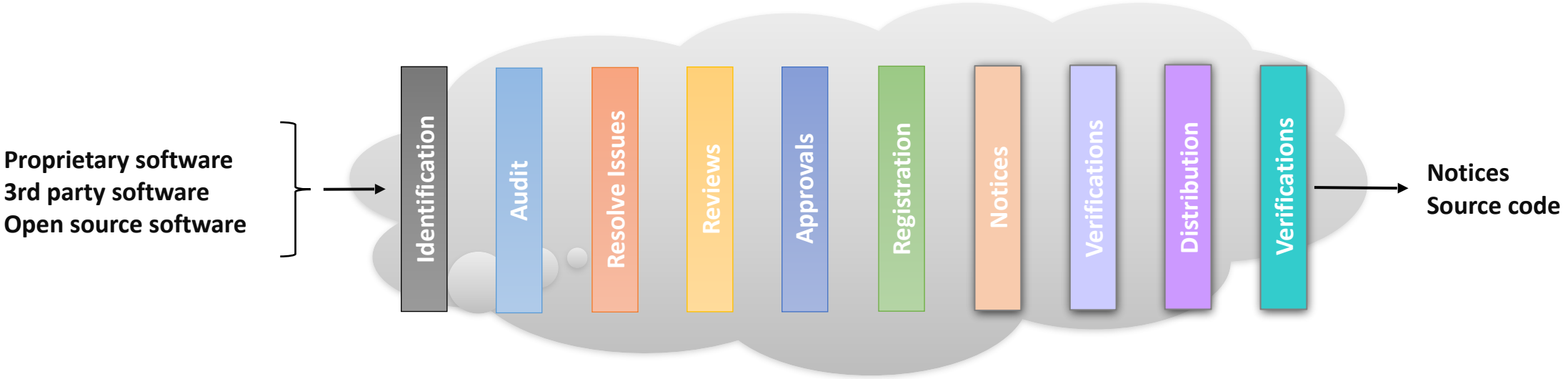
- In addition, IT may get requests from the OSRB to develop tools that will be used to improve effectiveness the compliance activities.

Corporate Development

- Corporate development in involved with OSS compliance in the following two major scenarios:
  - Mergers and acquisitions transactions
  - Outsourced development

Recommended Practices

Compliance End-to-End Process



Identification

- Identify **all the components and snippets** included in the product and their origin.

- There are three main sources for incoming source code:
  - Proprietary:
    - Developed by your own engineers (may include snippets of OSS or in many cases depends on or links to OSS)
  - Third Party Commercial:
    - Developed by third party software providers or consultants and offered to you under a commercial or OSS license (may include snippets of OSS or in many cases depends on or links to OSS)
  - OSS:
    - Developed by members of the OSS community

Identification – Basic Housekeeping

- Print out and retain the license information at the time you download the software.
  - Backtracking doesn't always work.
- 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. It allows you to:
  - Discover compliance problems as they occur
  - Provide solutions to discovered problem within acceptable delays
  - Keep the delta with the previous scan to a minimum
  - 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 with the scan results discuss with engineering and in some cases you may need to discuss with tool vendors if you suspect an unusual tool behavior
- 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.
- If the scanning tool identifies GPL-licensed source code (for instance) integrated in a proprietary component, you should report to engineering and request correction.
  - Re-scan the code after engineering has resolved the issue to get a solid confirmation that engineering has removed the code and replaced it with proprietary source code.
- In preparation of the legal review, provide legal with all information you discovered on the licensing of the specific component.
  - For OSS components, this includes COPYING, README, or LICENSE files.

If You Can't Comply

- If there are conflicts or compliance is not possible:
  - **Remove / 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 a written offer (in the case of GPL and LGPL for instance) informing end users how to contact the company to request a copy of the source code
  - Provide appropriate license, copyright and attribution notices for all OSS
- Be clear and direct in the language of the written offer and inclusive of all OSS included in your product.
- Make the written office available in the product manual, on the web site, and inside the product.

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.

General Guidelines to Engineers 1/2

- Fill out an OSRB form for each open source software you are using in product or in SDK.
- Save the web site from which you downloaded the open source package and save a mint copy of the package you downloaded.
- Consult with OSRB team when you upgrade your open source software version. License changes can occur between versions Don’t change or eliminate existing comments in headers
- Document compliance information in your build instructions.
- Do not check un-approved code into any source tree without authorization.
- For each file you modify in a GPL/LGPL licensed open source software, include a header comment that says: “Modified on mm-dd-yyyy”.
- Avoid re-naming open source modules.

General Guidelines to Engineers 2/2

- Do not send modifications to any public source tree without getting proper internal approval(s).
- Making even small contributions without your company's permission can compromise your company's IP (due to implicit or explicit patent licenses).
- Do not discuss coding or compliance practices with persons outside the company.
- Good programming practices are also legal best practices.
- Document the interfaces between any code you write.

Source Code Modification Considerations

- Source code modifications that you wish to remain proprietary must not be made to an OSS package that has derivative work obligations:
  - Proprietary source code must not link (statically or dynamically) to an OSS package that has a derivative work obligation (e.g., GPL).
  - This typically means those libraries under GPL or other OSS licenses that have derivative work obligations will not be approved for use.
- Ensure that any modifications to source code are documented in compliance with the OSS License prior to distribution
- All modifications to open source code modules shall be captured in the revision history of the module.

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.

Software Design Considerations

- Protect proprietary source code from being infected by GPL/LGPL code by developing proprietary code and OSS code to run in separate processes and use system calls to interact with each other.

Usage Considerations 1/3

- 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.

- OSRB Form for Each OSS
  - Fill out an OSRB usage request form for each OSS you are using in product.
  - Avoid using any OSS unless you have OSRB approval.

- Understand the Risks
  - 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.

Usage Considerations 2/3

- Retired OSS Packages
  - If an approved OSS package is not is use anymore, engineers must inform the OSRB to update the OSS inventory; alternatively, the OSRB will discover that the package is not used anymore when they run the BOM diff tool.

- Major Source Code Changes
  - If an approved package went through any major change, inform the OSRB to re-scan the source code; alternatively, the OSRB will discover that the package has been modified when they run the BOM diff tool.

- References, Original Download Source
  - Save the web site from which you downloaded the OSS package in addition to a mint copy of the package.

Usage Considerations 3/3

- Upgrading to Newer Versions of OSS
  - Ensure that each new version of the same OSS component is reviewed and approved. When you upgrade the version of an OSS package, make sure that the license of the new version is the same as the license of the older used version. License changes can occur between version upgrades. If the license changed, contact the OSRB to ensure that compliance records are updated and that the new license does not create a conflict.

- 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 Mixing Considerations

- Copy/Paste
  - Do not copy/paste OSS code into proprietary or third party source code or vice versa without OSRB approval.
  - Approvals are given on a case-by-case basis.

- Mixing Source Code with Different Licenses
  - Mixing of code coming under different OSS licenses must be avoided.
  - Many OSS licenses are incompatible with each other, especially when mixing licenses with the GPL.
  - 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

- 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.

Notice Types

- OSS attribution requirements differ from license to license but we can generally group them into four categories:
  - Full License Text
  - Copyright Notices
  - Acknowledgment Notices
  - Information on Obtaining the Source Code

Presentation of the Notices

- For each product containing OSS, the notices must be included in published user documentation (such as the product manual) that is distributed in printed or electronic form (on a CD or downloadable via a web site).
- 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.

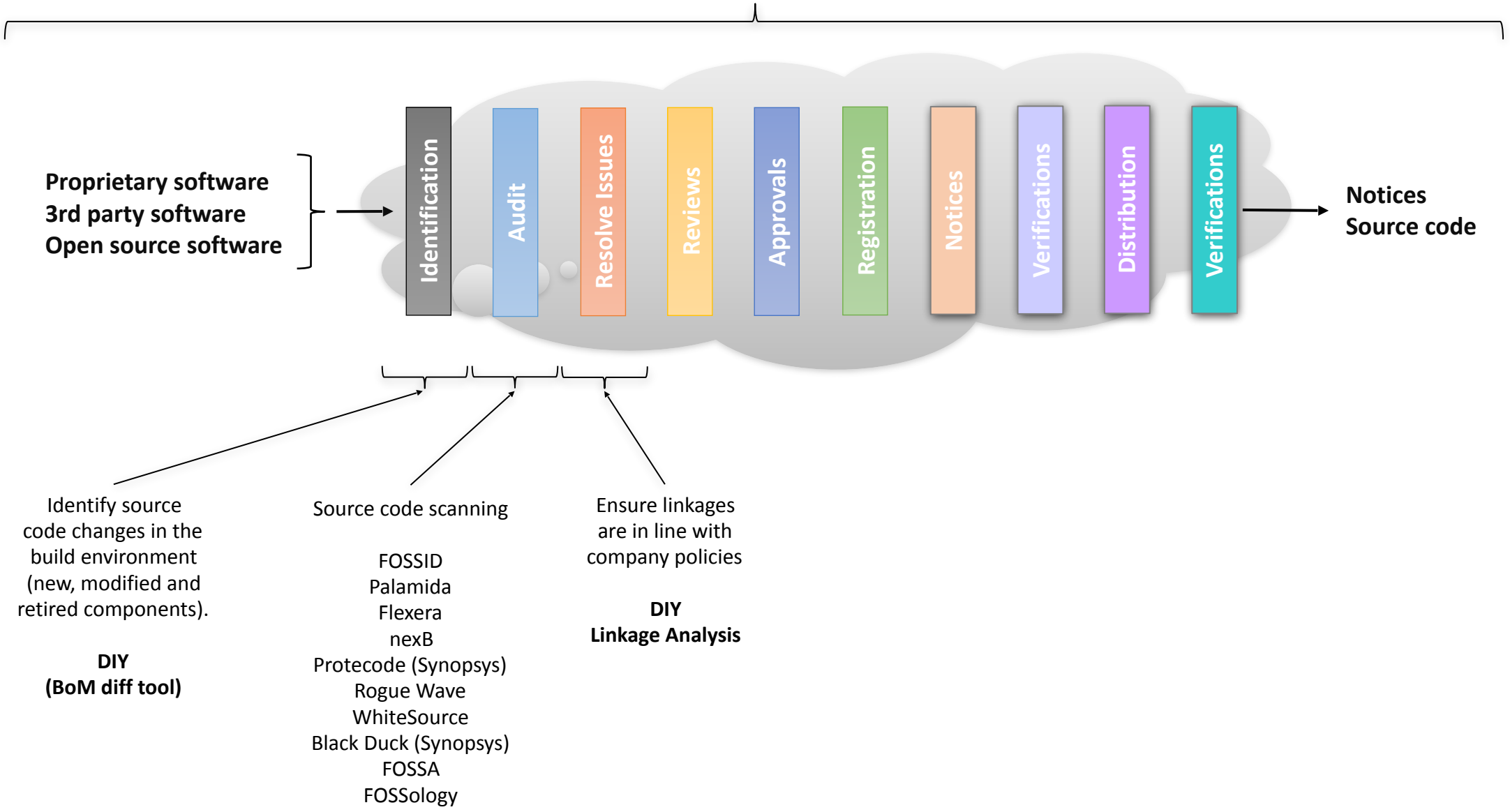
Information on Obtaining the Source Code

- Most licenses with a source code redistribution obligation require that the product is either accompanied with the source code or accompanied with a written offer of how to obtain the source code.

- The GPL and LGPL are examples of licenses that fall in this category.

Tools and Automation

Process Management
JIRA / Bugzilla / or similar



Metrics to evaluate source code scanning tools 1/2

Metric	Specifics
Knowledge Base	Size
	Frequency of update
Detection Capabilities	Whole components
	Partial snippets
	Ability to auto-identify code with proper origin and license
Ease of use	Intuitive, requires minimal amount of training
Operational Capabilities	Speed of scans
	Ability to use for M&A scans
	Support for different audit models
	Programming language agnostic
Integration Capabilities	Integration with build systems via APIs and a CLI
	Integration of company compliance policies within the tool
Security Vulnerabilities Database	Size of database
	Frequency of update
	Sources of vulnerabilities
	Support for advanced vulnerability discovery (i.e. identifying a vulnerability when vulnerable code was copied into a new component)

Metrics to evaluate source code scanning tools 2/2

Metric	Specifics
Cost	Infrastructure cost
	Operational cost
	Licensing cost
	Integration cost
	Lock-in cost
	Cost of engineering customization
Deployment models	On-site, Cloud, Hybrid
Other	Modular installation
	Generation of required notices
	Reporting abilities

Recommendations – Scaling Legal Support

Practical Legal Advice at Your Fingertips

1. License playbooks
2. License compatibility information
3. License classification information
4. Approved software interaction methods
5. Checklists

1. License Playbooks

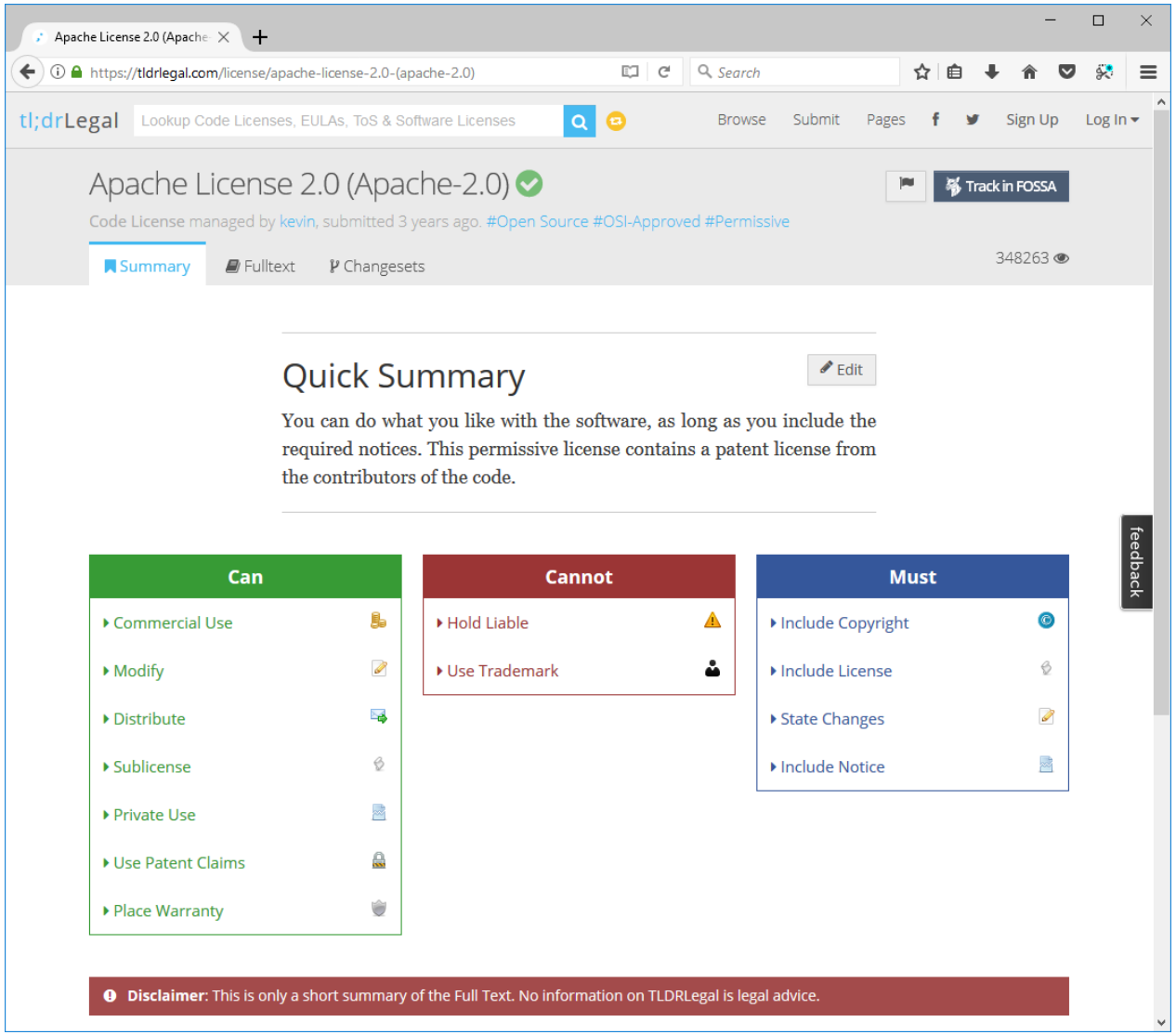
An easy to read and understand summary of licenses intended for software developers.

For each commonly used license provide a playbook that includes:

- **Name / Version / URL**
- **Executive Summary**
- **Grant**
- **Limitations**
- **Warranty**
- **Obligations**
- **Patent Notes**
- **Etc.**

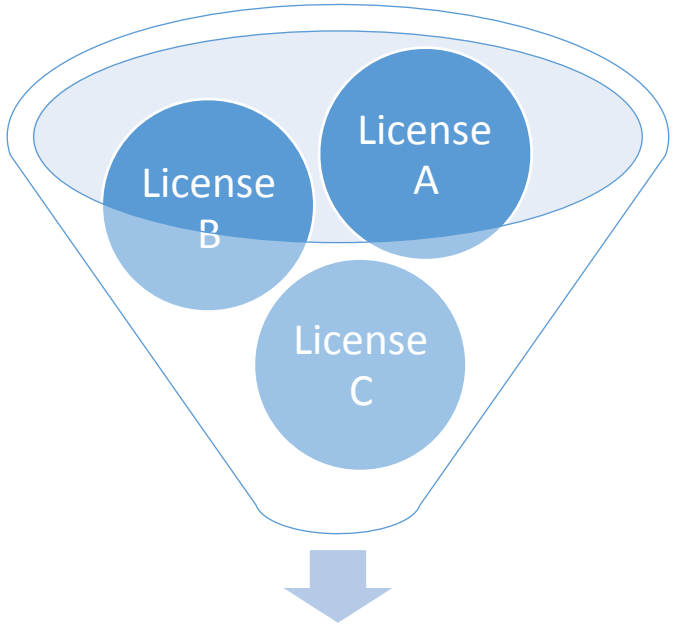
License Playbook – Example from `tldrlegal.com`

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



2. Compatibility Matrix

License compatibility issues arises when developers combine source code incoming from different sources, under different licenses, into a single work.



License(s) ?

Incoming Licenses = A + B + C
Outgoing License(s) = ?

License Compatibility Matrix

A license compatibility matrix is an easy visual method to identify if a given license is compatible with another license.

A license compatibility matrix is prepared by Legal Counsels for the 10-15 most used licenses.

License Compatibility Matrix – Simple View

Is Compatible With:	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

(example classification)

An easy way to understand the approval process for licenses and the course of action needed when using them.

Permissive	**Modifications to be released**	**Patent Clause**	**Not Allowed**
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.

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	**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

Can Statically Link To	**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.

- It is expected that after regular and consistent 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

Dealing with Compliance Inquiries

Introduction

- Several companies received negative publicity and/or got sued because they either:
  - Ignored requests to provide Open Source compliance information
  - Did not know how to handle compliance inquires
  - Lacked or had a poor compliance program
  - Simply refused to cooperate thinking it is not enforceable.

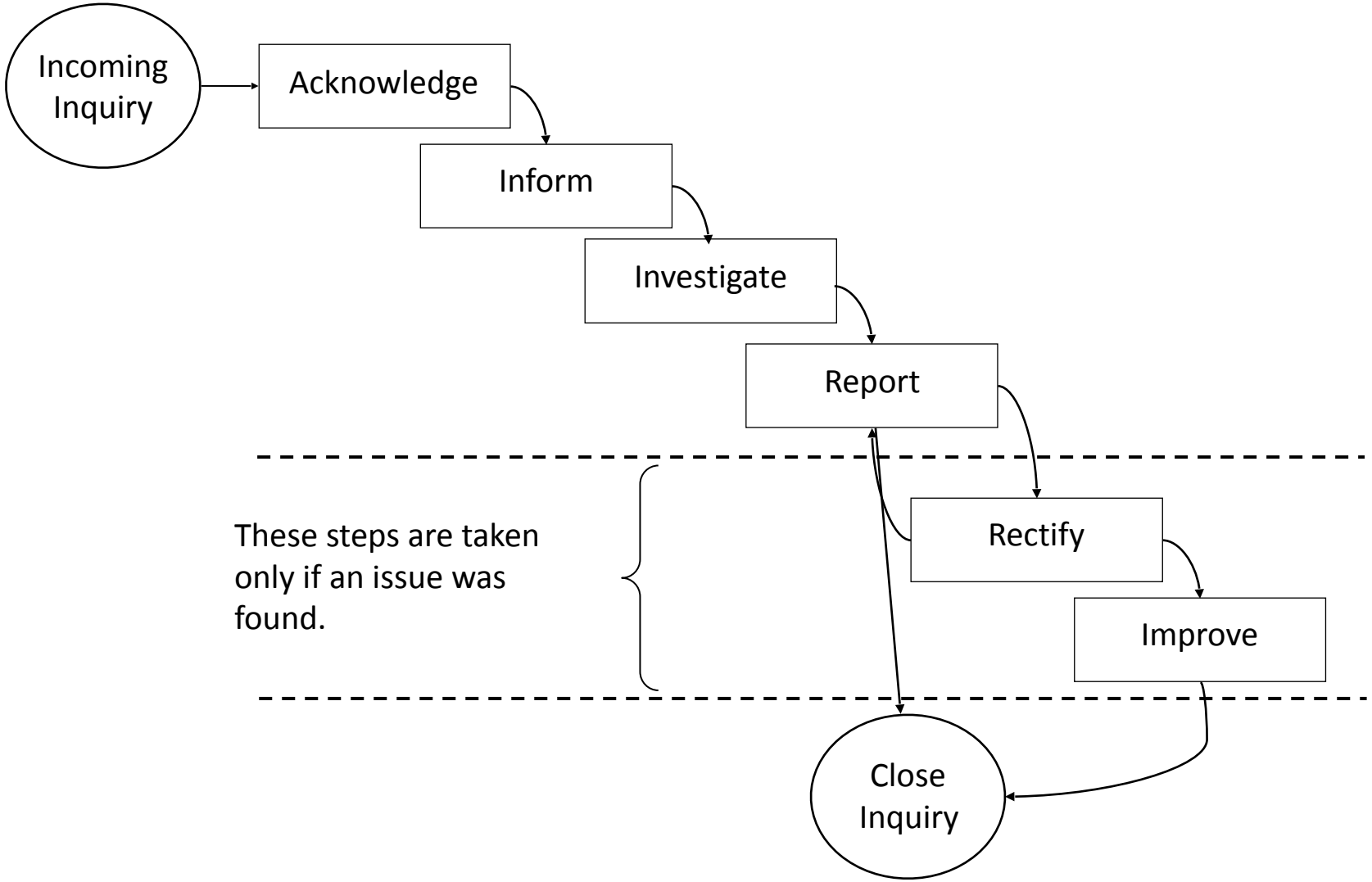
- We know that none of these approaches is fruitful and beneficial to any of the parties involved.

General Rule

- Companies should not ignore compliance inquiries.

- Companies should:
  - Acknowledge the receipt of the inquiry
  - Inform the reported that they will be looking into it
  - Provide a certain date on when to expect a follow-up
  - Work with inquirer to resolve the issue at hand (if there is one)

Responding to Compliance Inquiries



Not All Inquiries Come Through Your Compliance Email Address

- You also need to watch social networks, discussion board, blogs of enforcement community, etc.
- You will need to have a team ready to engage (Legal + PR + Open Source Expert)

Some Inquiries (Trolling) Are Financially Motivated

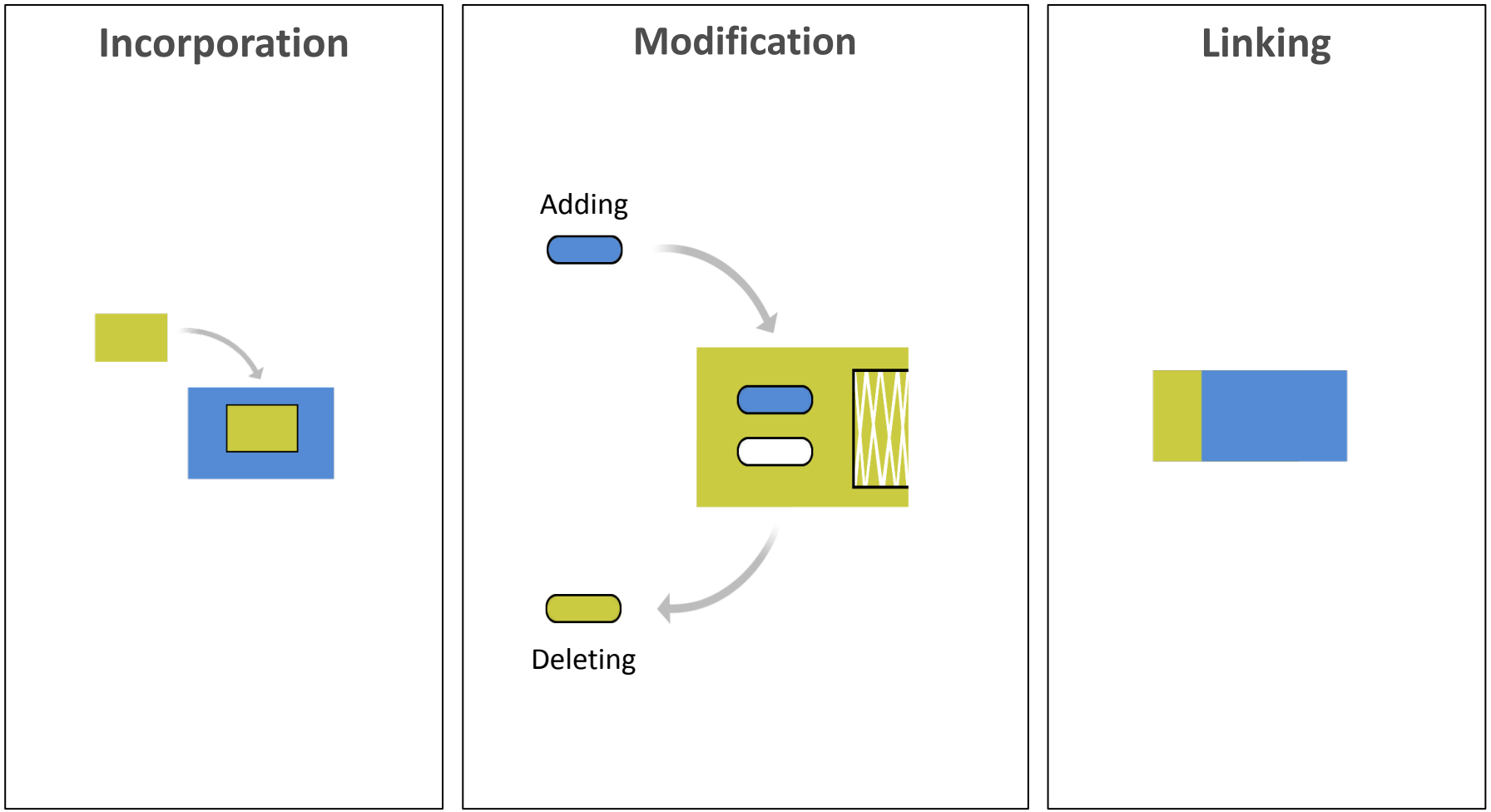
A great FAQ on the topic by Heather Meeker.
https://opensource.com/article/17/8/patrick-mchardy-and-copyright-profiteering

General Considerations

- Assume that any information you disclose can become public
- Treat all inquiries as formal inquires
- Consider how your existing OSS compliance efforts would measure up in an enforcement action

Open source compliance in M&A transactions

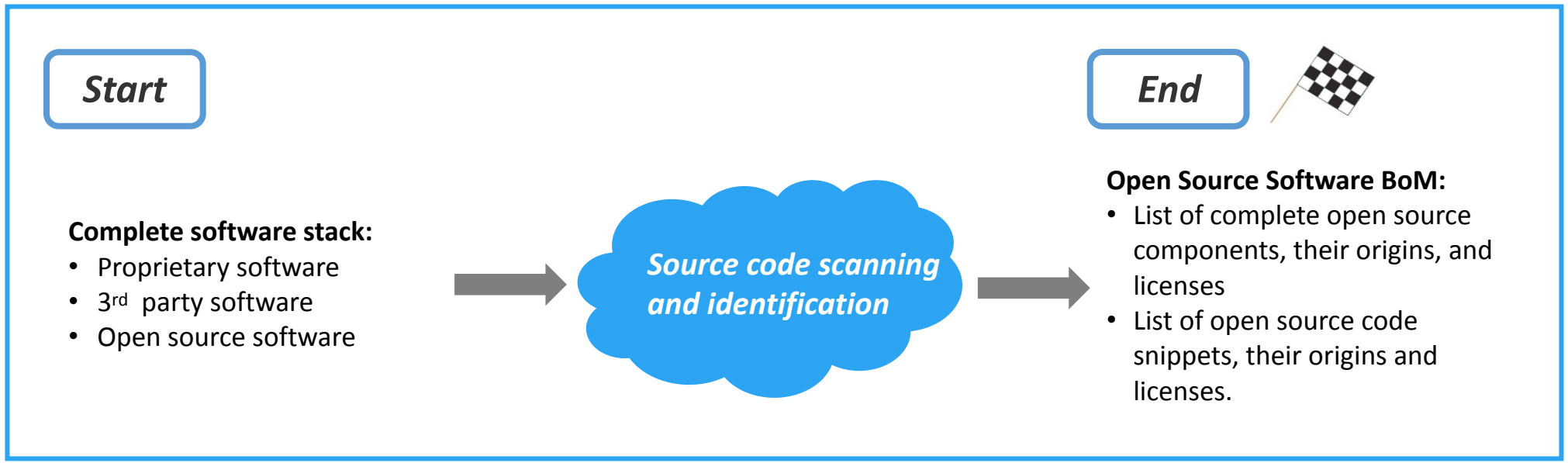
Common Open Source Usage Scenarios



Every deal is different.

Open source is a constant.

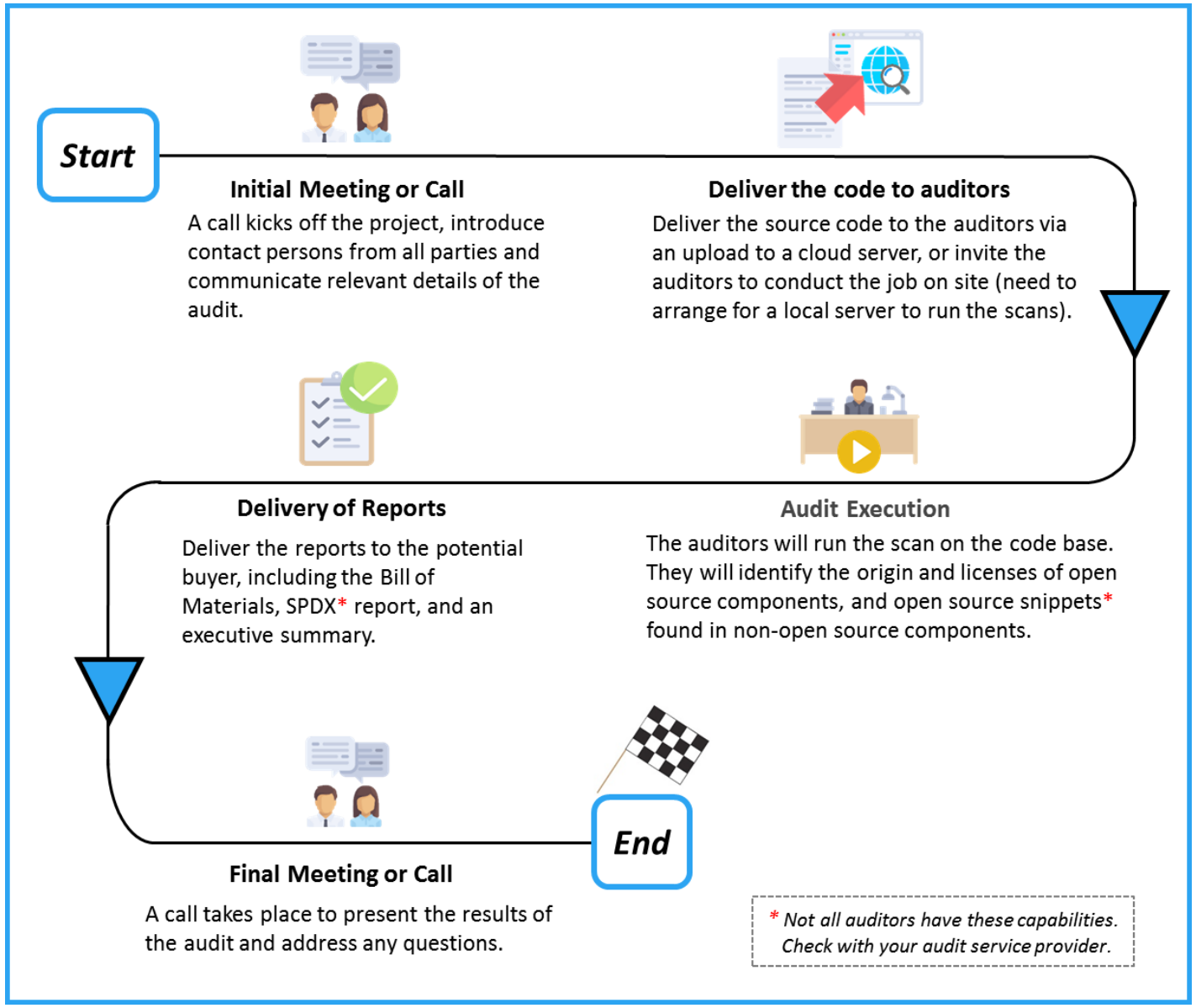
What specific due diligence open source software is required in M&A transactions?


Start
Complete software stack:
• Proprietary software
• 3rd party software
• Open source software
Source code scanning and identification
End
Open Source Software BoM:
• List of complete open source components, their origins, and licenses
• List of open source code snippets, their origins and licenses.

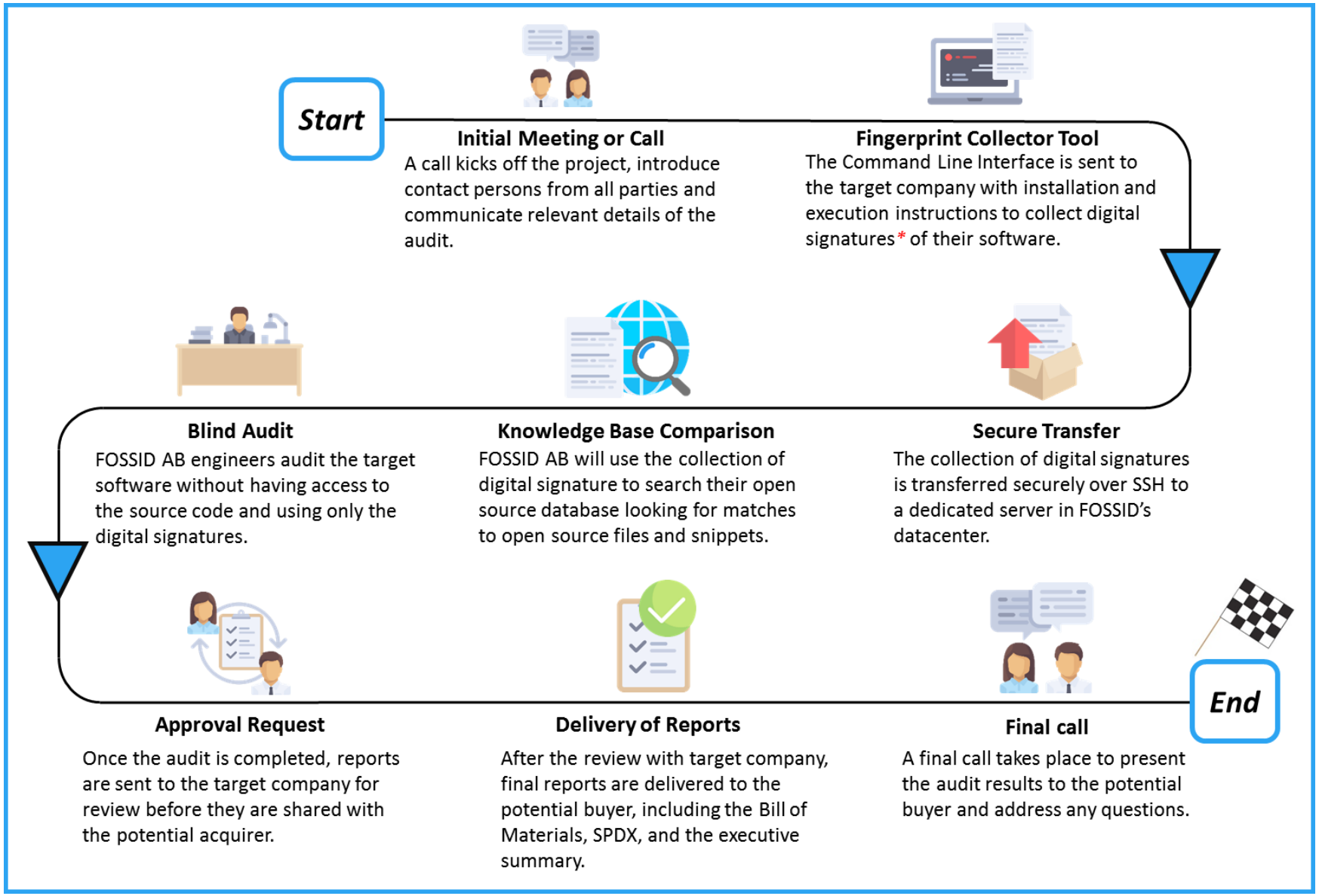
Audit methods

- Traditional
- Blind
- DIY

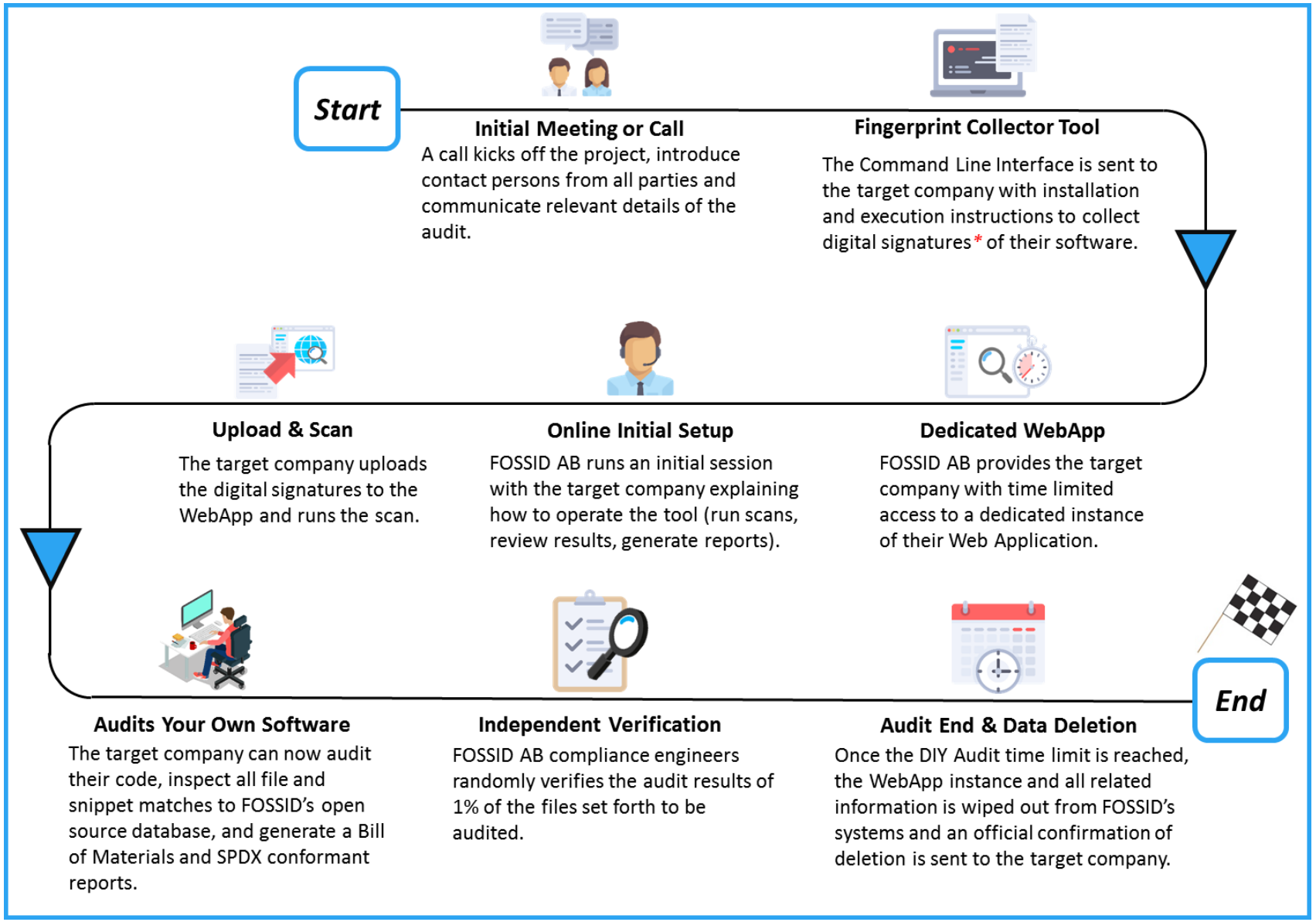
Traditional



Blind


Start
Initial Meeting or Call
A call kicks off the project, introduce contact persons from all parties and communicate relevant details of the audit.
Fingerprint Collector Tool
The Command Line Interface is sent to the target company with installation and execution instructions to collect digital signatures* of their software.
Secure Transfer
The collection of digital signatures is transferred securely over SSH to a dedicated server in FOSSID's datacenter.
Knowledge Base Comparison
FOSSID AB will use the collection of digital signature to search their open source database looking for matches to open source files and snippets.
Blind Audit
FOSSID AB engineers audit the target software without having access to the source code and using only the digital signatures.
Approval Request
Once the audit is completed, reports are sent to the target company for review before they are shared with the potential acquirer.
Delivery of Reports
After the review with target company, final reports are delivered to the potential buyer, including the Bill of Materials, SPDX, and the executive summary.
Final call
A final call takes place to present the audit results to the potential buyer and address any questions.
End

DIY


Start
Initial Meeting or Call
A call kicks off the project, introduce contact persons from all parties and communicate relevant details of the audit.
Fingerprint Collector Tool
The Command Line Interface is sent to the target company with installation and execution instructions to collect digital signatures* of their software.
Upload & Scan
The target company uploads the digital signatures to the WebApp and runs the scan.
Online Initial Setup
FOSSID AB runs an initial session with the target company explaining how to operate the tool (run scans, review results, generate reports).
Dedicated WebApp
FOSSID AB provides the target company with time limited access to a dedicated instance of their Web Application.
Audits Your Own Software
The target company can now audit their code, inspect all file and snippet matches to FOSSID's open source database, and generate a Bill of Materials and SPDX conformant reports.
Independent Verification
FOSSID AB compliance engineers randomly verifies the audit results of 1% of the files set forth to be audited.
Audit End & Data Deletion
Once the DIY Audit time limit is reached, the WebApp instance and all related information is wiped out from FOSSID's systems and an official confirmation of deletion is sent to the target company.
End

What Insights Can You Gain From Such Diligence?

Engineering Insights

Good programming practices are also legal best practices.

- High correlation between good compliance practices and good engineering practices.
- Modularity of software components.
- Clean integration of various components or modules.
- Transparent APIs.
- Good documentation.
- Minimization of data sharing.

Legal and Compliance Insights

- Policies and processes setup to handle open source compliance.
- Adequate mechanisms to satisfy open source license obligations by offering access to the source code and other notices as required by the licenses.
- Practices that conflict with the acquiring company's open source policies, to what extent, and a way to compare the target company's record of fulfilling of open source license obligations for current commercial offerings.
- Proprietary software assets are at risk due to misuse of open source software with strong copyleft license.
- The risk portfolio of the open source licenses the target uses and if it is aligned with the comfort zone of the acquiring company.

Business Insights

- A better understanding of whether the bulk of the target's valuation is a result of the integration of open source or in proprietary added value.

- A confirmation whether the target company has identified all open source software contained in distributed products and services and whether or not they've satisfied all obligations resulting from mixing the open source code with code under a proprietary or alternative open source license.

Preparation as a Target

- Process and policy
- Staff
- Training
- Tooling
- Use latest releases for security purposes
- Measure up your compliance efforts
- Educate

Avoid Common Pitfalls

Type	Avoidance
Unplanned inclusion of OSS into proprietary or 3rd party code (or vice versa)	Training. Regularly scheduled scans.
Unplanned linking of OSS into proprietary source code (or vice versa).	Training. Dependency tracking tool.
Failure to provide accompanying source code.	Checklist. Post shipping to-do.
Providing the incorrect version of accompanying source code.	Update process to ensure that the accompanying source code for the binary version is being published.
Failure to provide accompanying source code for OSS component modifications.	Update process to ensure that source code for modifications are published.
Failure to mark OSS source code modifications.	Training. Verification before posting source code.
Failure by developers to seek approval to use OSS.	Conduct periodic full scan to detect undeclared OSS. Training. Accountability (including compliance in performance metrics).
Failure to audit the source code.	Provide proper staffing. Enforce periodic audits.
Failure to resolve the audit findings.	Time limit before escalation kicks off automatically.
Failure to seek review of OSS in a timely manner.	Training.

Preparations as Acquirer

- Choose the right audit model and right auditor for your needs.
- Know what you care about.
- Ask the right questions.
- Identify items to be resolved before executing the transaction.
- Create a compliance improvement plan for post-acquisition.

Recommendations for Target

- Identify the origin and license of all internal and external software.
- Track open source software within the development process (components and snippets).
- Perform source code reviews for new or updated code entering the build.
- Fulfill license obligations when a product ships or when software is updated.
- Offer open source compliance training to employees.

Recommendations for Acquirer

- Decide with the target company on the appropriate audit method to use, and which 3rd party to engage for the audit
  - Audit method, inputs and outputs
  - Primary contact
  - Timeline and logistics especially if it involves an on-site visit
  - Confidentiality parameters
  - Code vulnerabilities and version control

OpenChain

Building trust with your company’s compliance practices will add trust to the software supply chain.

Scaling Across Ecosystem Partners.

Compliance Hiccups Fall Under 6 Buckets

1.	Policy failure	Employee did not follow policy / internal guidelines
2.	Process failure	Process oversight, corner cases, human error
3.	Tooling failure	Tooling imperfection, human error
4.	Intellectual Property Failure	Copy / Paste syndrome
5.	SW Procurement failure	Non-compliance via 3rd party
6.	Miscellaneous failure	Notice error, code versioning error, web site access error, incomplete code, etc.

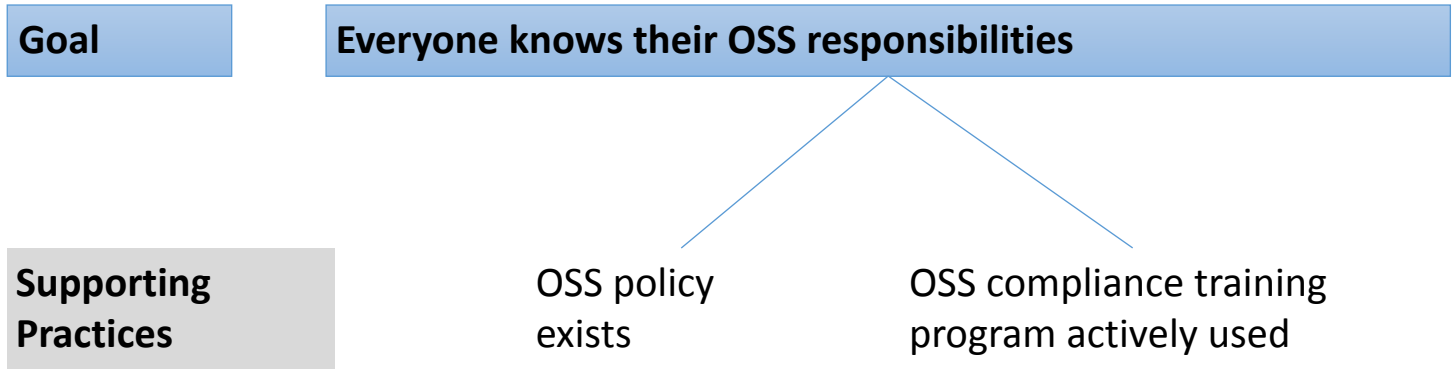
Learning From Our Experiences

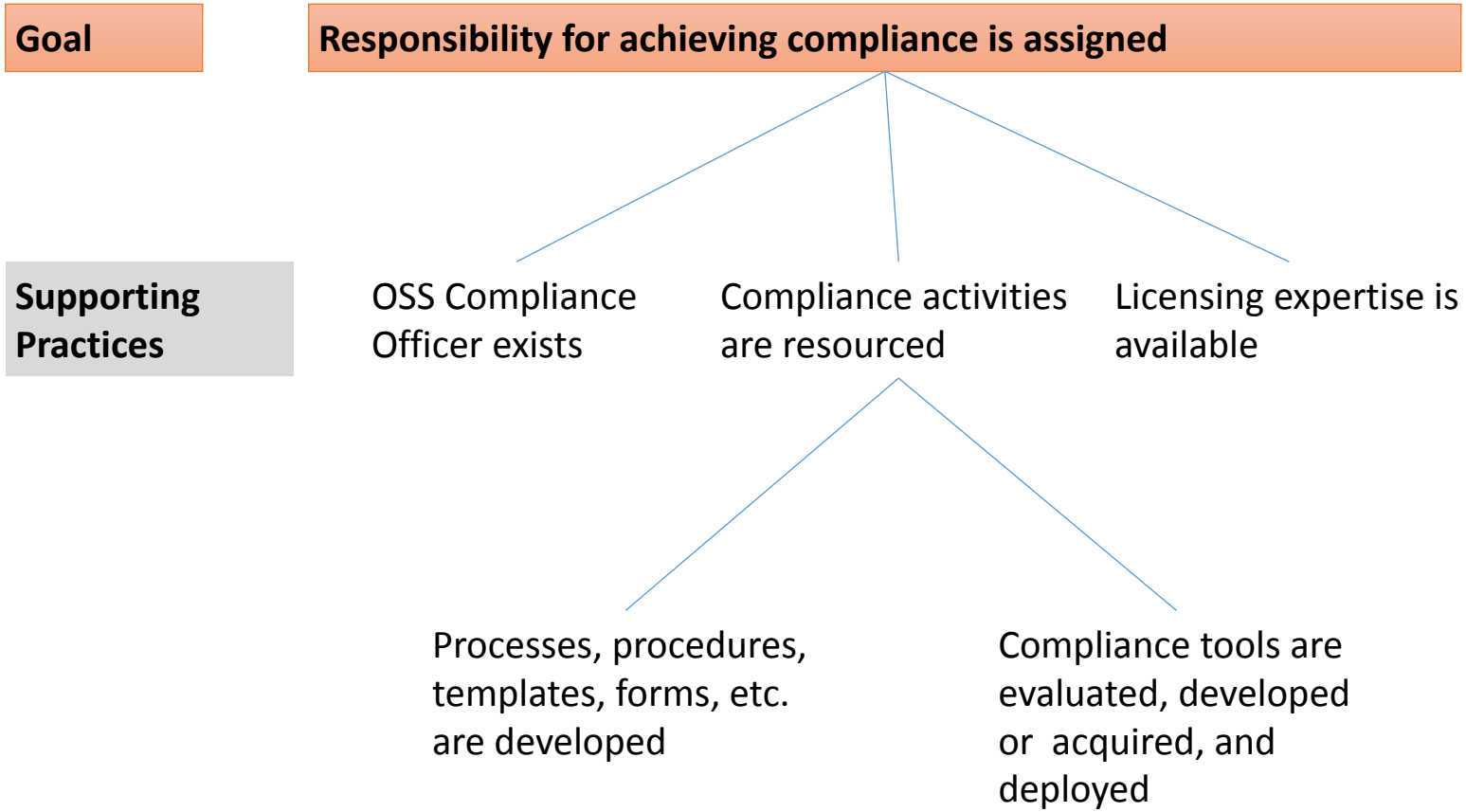
1.	Policy and process	Training + ongoing seminars + central policy & process
2.	Tooling	Training + additional tooling (including in-house)
3.	SW Procurement	Training + reform agreements + templates
4.	IP Failure	Require approval for code re-use
5.	Misc.	Update process to include verification steps

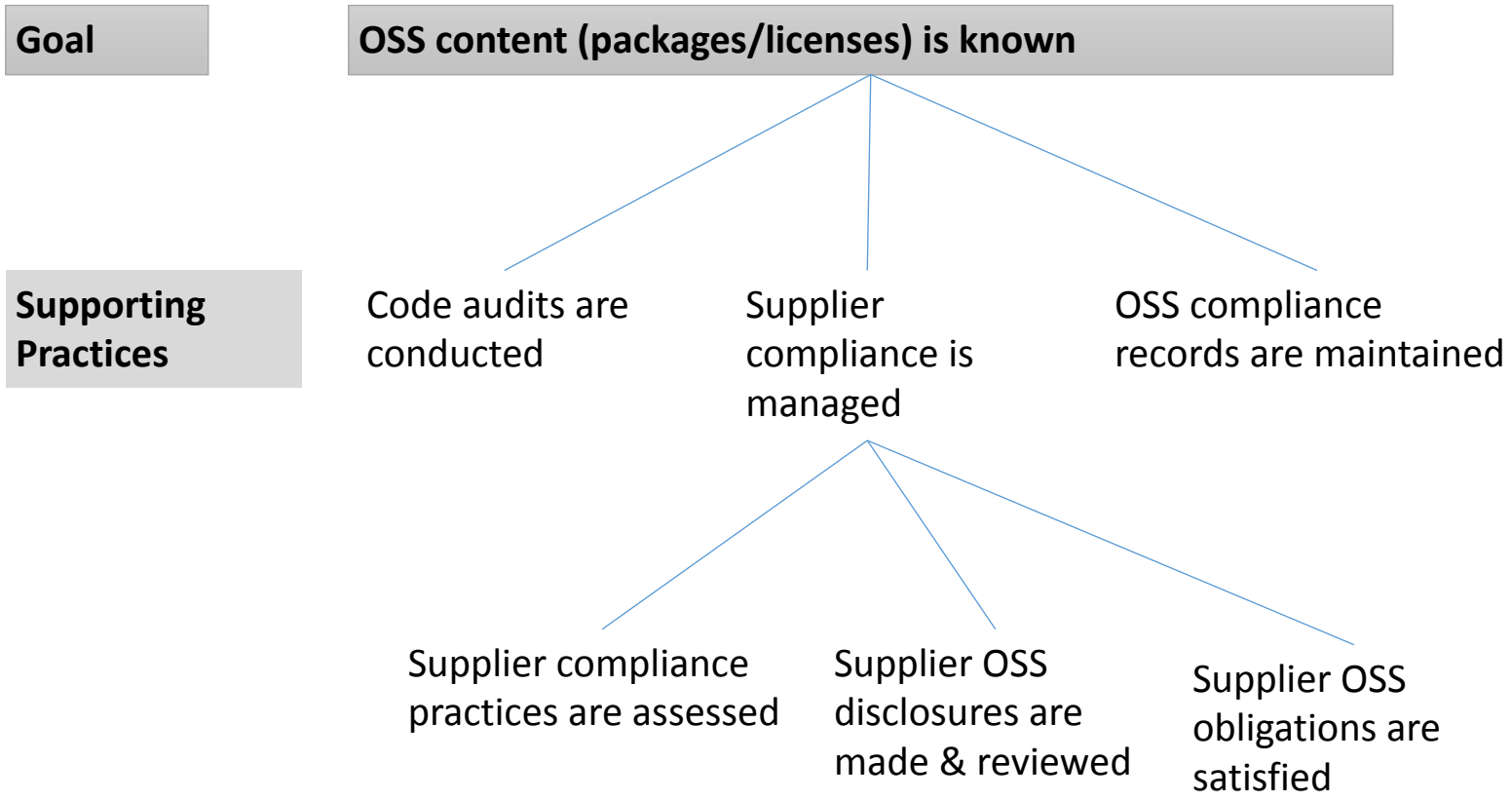
What Does This Mean?

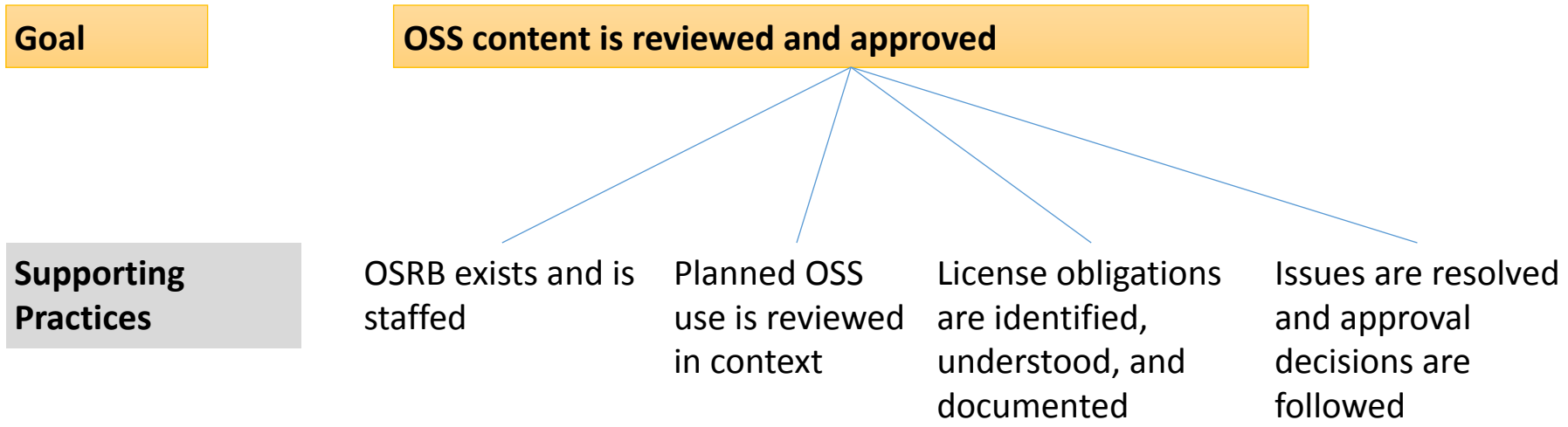
- Across the companies we work with as suppliers and partners:

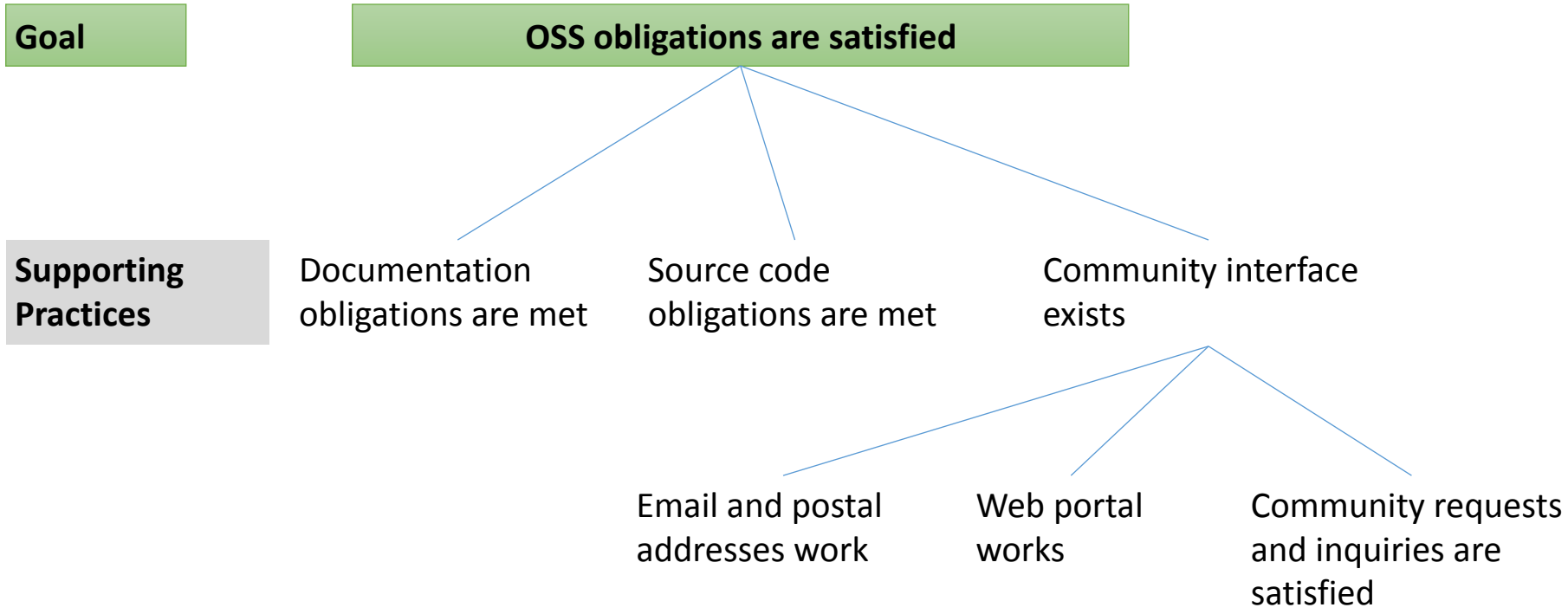
Everyone knows their OSS responsibilities	Policy + Process + Education
Responsibility for achieving compliance is assigned	Staffing + Education
OSS content (packages/licenses) is known	Process + Tools
OSS content is reviewed and approved	Process + Policy + Staffing
OSS obligations are satisfied	Process + Operation/Execution


Goal
Everyone knows their OSS responsibilities
Supporting Practices
OSS policy exists
OSS compliance training program actively used


Goal
Responsibility for achieving compliance is assigned
Supporting Practices
OSS Compliance Officer exists
Compliance activities are resourced
Licensing expertise is available
Processes, procedures, templates, forms, etc. are developed
Compliance tools are evaluated, developed or acquired, and deployed


Goal
OSS content (packages/licenses) is known
Supporting Practices
Code audits are conducted
Supplier compliance is managed
OSS compliance records are maintained
Supplier compliance practices are assessed
Supplier OSS disclosures are made & reviewed
Supplier OSS obligations are satisfied


Goal
OSS content is reviewed and approved
Supporting Practices
OSRB exists and is staffed
Planned OSS use is reviewed in context
License obligations are identified, understood, and documented
Issues are resolved and approval decisions are followed


Goal
OSS obligations are satisfied
Supporting Practices
Documentation obligations are met
Source code obligations are met
Community interface exists
Email and postal addresses work
Web portal works
Community requests and inquiries are satisfied

Building trust within the SW supply chain is doable

Goal 1	**Everyone knows their OSS responsibilities**
Goal 2	**Responsibility for achieving compliance is assigned**
Goal 3	**OSS content (packages/licenses) is known**
Goal 4	**OSS content is reviewed and approved**
Goal 5	**OSS obligations are satisfied**

Imagine a world where all companies
you exchange software with have met these 5 basic goals:
Policy, Process, Tool, Staffing, Education.

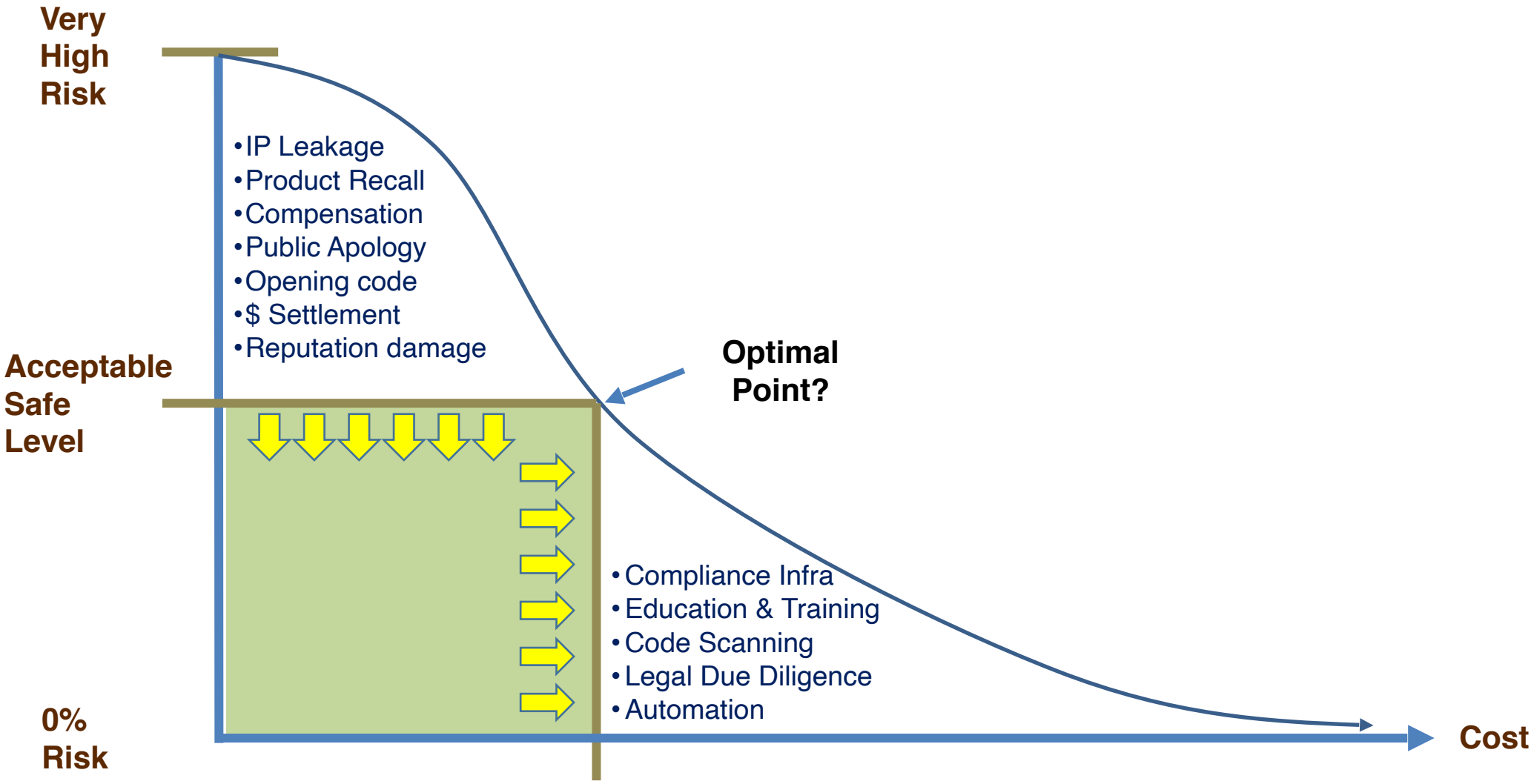
Compliance: A Balancing Act

Balancing What?

- Internal & external legal counsel opinions
- Business (incl. engineering) needs
- Open source community views on license compliance
- Enforcers views on license compliance

Sweet Spot

How Good is Good Enough?



Final Thoughts

- We've come a long way in open source compliance.
- We learned a lot.
- Compliance today is now more of a scalability and a cost issue, not as much of a license interpretation debate.

<u>The Next Frontier:</u>

- How can we take cost out of compliance and provide a consistent, repeatable approach that helps companies avoid compliance hiccups?

Open source compliance is an ongoing process, not a destination.

Maintaining good open source compliance practices enables companies to be prepared for any scenario where software changes hands, from a possible acquisition, a sale, or product or service release.

SAMSUNG RESEARCH AMERICA

Implementing and Managing Open Source Compliance Programs – A Crash Course

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

Twitter: @IbrahimAtLinux
Web: IbrahimAtLinux.com

Open Source Strategy Forum
November 14, , 2018 – London