THE LINUX FOUNDATION

» The Open Compliance Program

Achieving FOSS Compliance in the Enterprise

A Close Look at a Sample End-to-End Compliance Process

By Ibrahim Haddad (Ph.D.), The Linux Foundation

A White Paper By The Linux Foundation
http://www.linuxfoundation.org

This white paper is fifth in a series that focuses on the various practical aspects of ensuring free and open source software (FOSS) compliance in the enterprise. This paper examines a sample end-to-end compliance process.

Introduction

Implementation of open source compliance processes can vary from company to company based on a number of factors: the underlying product development process into which compliance must fit, the size and nature of the code base, number of products turned out, the amount of externally supplied code, size and organizational structure of the company, and so on. But the core elements of compliance usually remain the same: identifying the open source in the code base; reviewing and approving its use; and satisfying obligations. This paper focuses on the core elements of a compliance process, examining the inputs, actions, and outputs of each element. The result of compliance due diligence is an identification of all free and open source software (FOSS) used in a product intended for external distribution and a plan to meet the attendant license obligations.

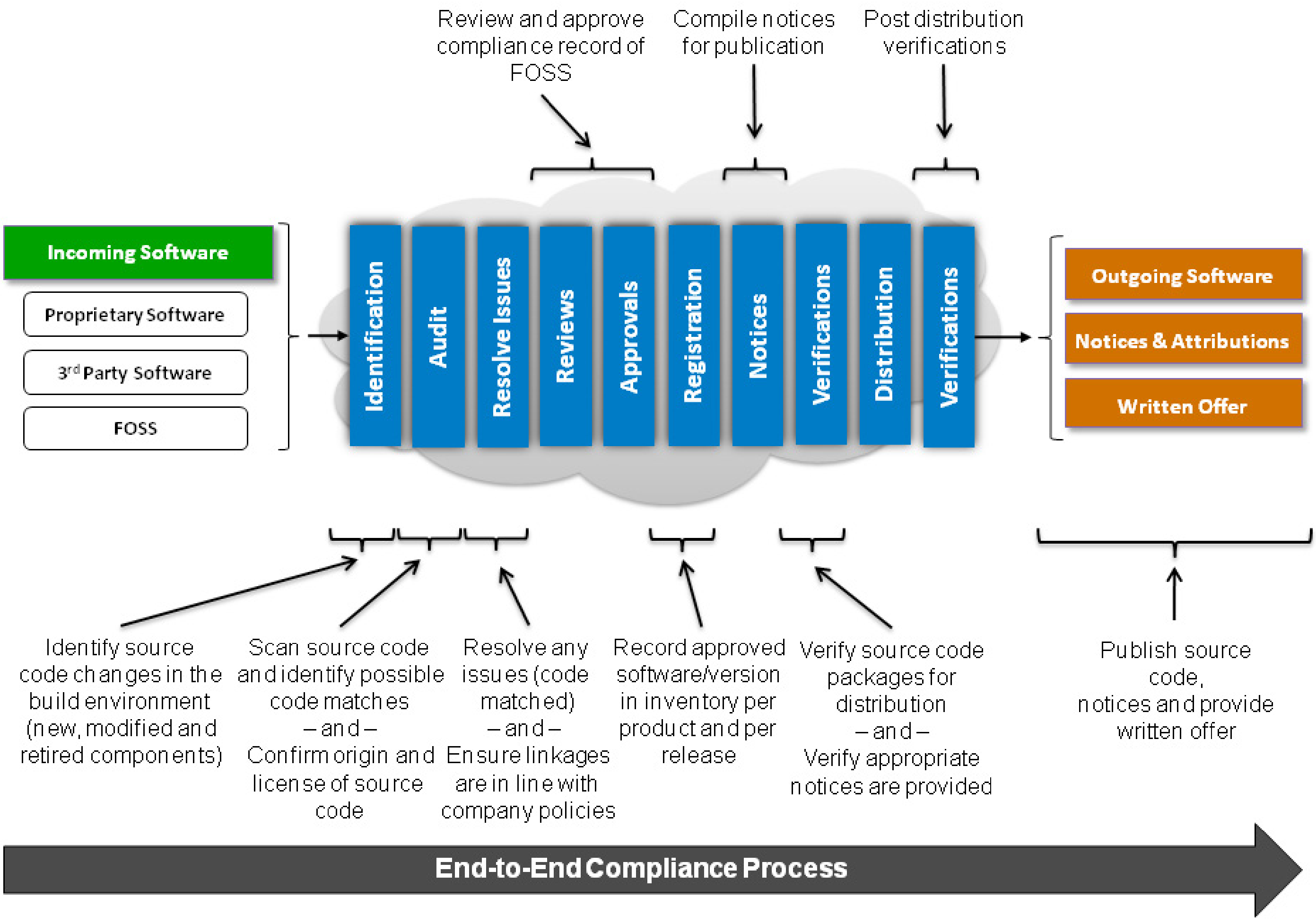

Review and approve compliance record of FOSS
Compile notices for publication
Post distribution verifications
Incoming Software
Proprietary Software
3rd Party Software
FOSS
Identification
Audit
Resolve Issues
Reviews
Approvals
Registration
Notices
Verifications
Distribution
Verifications
Outgoing Software
Notices & Attributions
Written Offer
Identify source code changes in the build environment (new, modified and retired components)
Scan source code and identify possible code matches – and – Confirm origin and license of source code
Resolve any issues (code matched) – and – Ensure linkages are in line with company policies
Record approved software/version in inventory per product and per release
Verify source code packages for distribution – and – Verify appropriate notices are provided
Publish source code, notices and provide written offer
End-to-End Compliance Process

Figure 1. Simple view of the compliance end-to-end process

Figure 1 offers a high-level overview of a sample end-to-end compliance process and illustrates the various compliance steps or phases that components containing free and open source software go through before they get approved for use in a product intended for external distribution (Other ways of organizing the compliance process may well accomplish the same goals of achieving compliance.) Throughout the paper, we will be looking at what happens in these various phases, what are the inputs and outputs of each phase and how to keep tight control on software usage via the compliance process.

Effective Compliance

I use the term "due diligence" to refer to a number of concepts involving either the performance of source inspection or source surveillance, or the performance of quality duties or system audits. In the case of FOSS compliance, due diligence is required to ensure the following:

- FOSS used in the product has been identified, reviewed and approved
- The product implementation includes only the approved FOSS
- FOSS used in the product has been registered in the FOSS inventory system
- All obligations related to the use of licensed material have been identified
- Appropriate notices have been provided in the product documentation, including attributions and copyright notices and perhaps, depending on the FOSS involved and the compliance approach, a written offer to provide source code upon request
- Source code, including modifications (when applicable), has been prepared and is available at the point the product ships
- Verifications of all the steps in the process

There are great benefits to having an end-to-end compliance process that is simple, communicated and well understood within the organization

- It enables organizations to benefit from FOSS while complying with obligations
- It moves FOSS use inside the organization from ad-hoc to a standardized process
- It helps manage acquisition of FOSS components
- It helps employees understand how to work with FOSS
- It improves the relationship with the FOSS community because the community knows the organization has a plan to comply with FOSS licenses
- It accelerates exchange of information and ideas with the FOSS community through sharing of source code modifications
- It speeds innovation and adoption of new products and services since the organization is able to safely adopt FOSS components and use them as enablers for services and products

Elements of an End-to-End Compliance Process

An effective compliance process will include several steps (Figure 1):

- Identification of FOSS
- Auditing source code
- Resolving any issues uncovered by the audit

- Completing appropriate reviews
- Receiving approval to use FOSS
- Registering FOSS in the software inventory
- Updating end user documentation to reflect FOSS usage in product
- Performing verification of all previous steps prior to distribution
- Distributing FOSS including modifications (if any) when applicable
- Performing final verifications in relation to distribution

Identification of FOSS

The identification of FOSS (Figure 2) is the first and initial step in the compliance due diligence process. The goal of this step is to monitor the intake and incorporation of FOSS in the product either as a standalone package or embedded within third party or company-developed proprietary software.

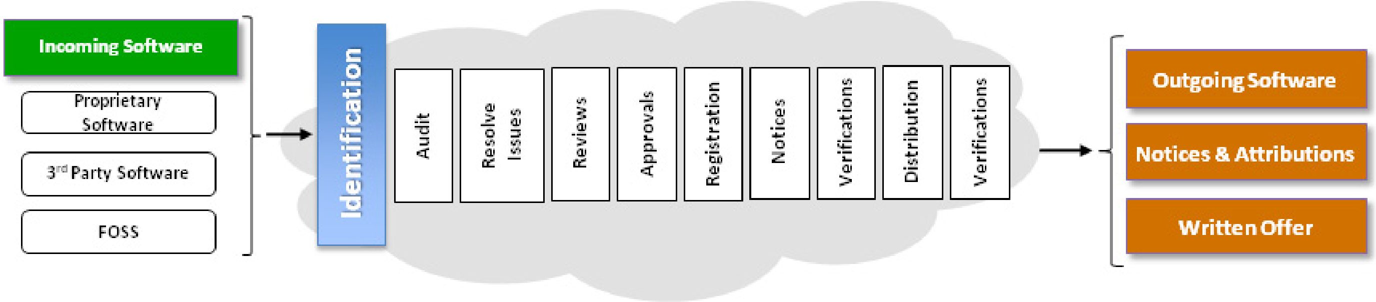


Figure 2. Identification of FOSS step as part of the compliance end-to-end process

There are several methods to identify the use of FOSS in the product:

- Incoming request to use FOSS: This is the most common method for identifying the usage of FOSS in product. Engineering staff or Product Management is required to inform an Open Source Review Board (OSRB) or compliance team of the intent to use specific FOSS in a specific product or platform release. The submitter provides information regarding the intended use of the FOSS package for review and approval.
- Auditing the full platform or product code to establish a compliance baseline, then auditing code modules that are changed in subsequent releases.
- Third party software provider due diligence: Requiring a full disclosure of FOSS inclusion in products provided by third party suppliers, with an accompanying review of the disclosure by the FOSS compliance team. In some cases it will make sense to require that third party software vendors to provide an audit of the supplied code as an additional layer of diligence to ensure you are controlling the intake of FOSS by your product.
- Auditing proprietary (company-developed) software components: In some instances, engineers may decide to copy/paste source code from FOSS components and include it in proprietary software components. Therefore, it is important to also audit company-

developed software components since they may include FOSS code, which may lead to compliance failures if not discovered before product ship date.

- FOSS component added to the organization's source code repository that does not correspond to an incoming request to use FOSS: Relying on engineers to fill out forms announcing their intent to use FOSS is not always a reliable method. Therefore, as a backup, you should also consider setting up a source code control system with a separate folder for FOSS and a notification alert any time there is a check-in to this folder. Since it is always a recommended practice to separate FOSS, company-developed proprietary software and third party software in different folders in your build system, it becomes feasible to set up alerts when new code is being submitted. If a new component was submitted and it does not correspond to an existing usage (FOSS request) form, then this means it is a new component and a new form must be filled out.

Identification Phase	
Prerequisites	**Outcome**
• An incoming OSRB form requesting use of a specific FOSS package • A discovery of FOSS being used (without proper authorization) via a scan • A discovery of FOSS being used as part of third party software	• A compliance record is created (or updated) for the FOSS • An audit is requested to scan the source code

Auditing Source Code

Auditing source code is the second step in the compliance due diligence (Figure 3). It consists of scanning the source code using automated source code analysis tools to discover source code matching FOSS source code.

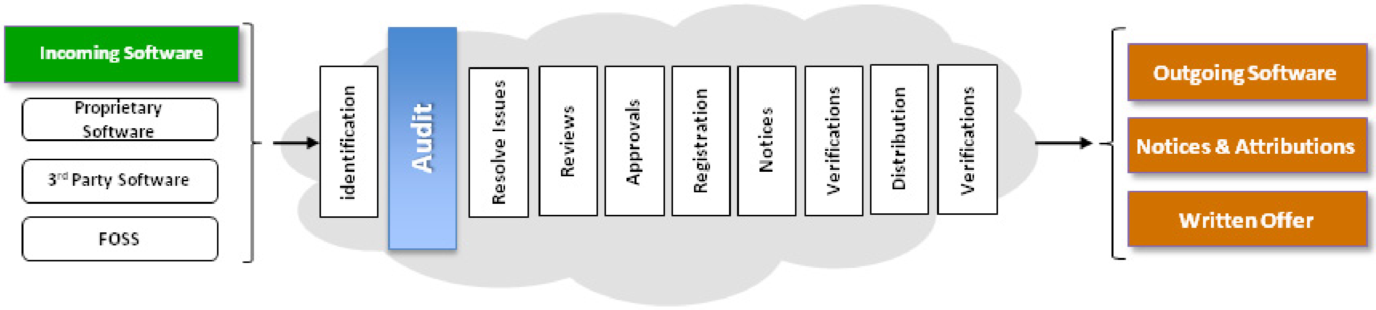


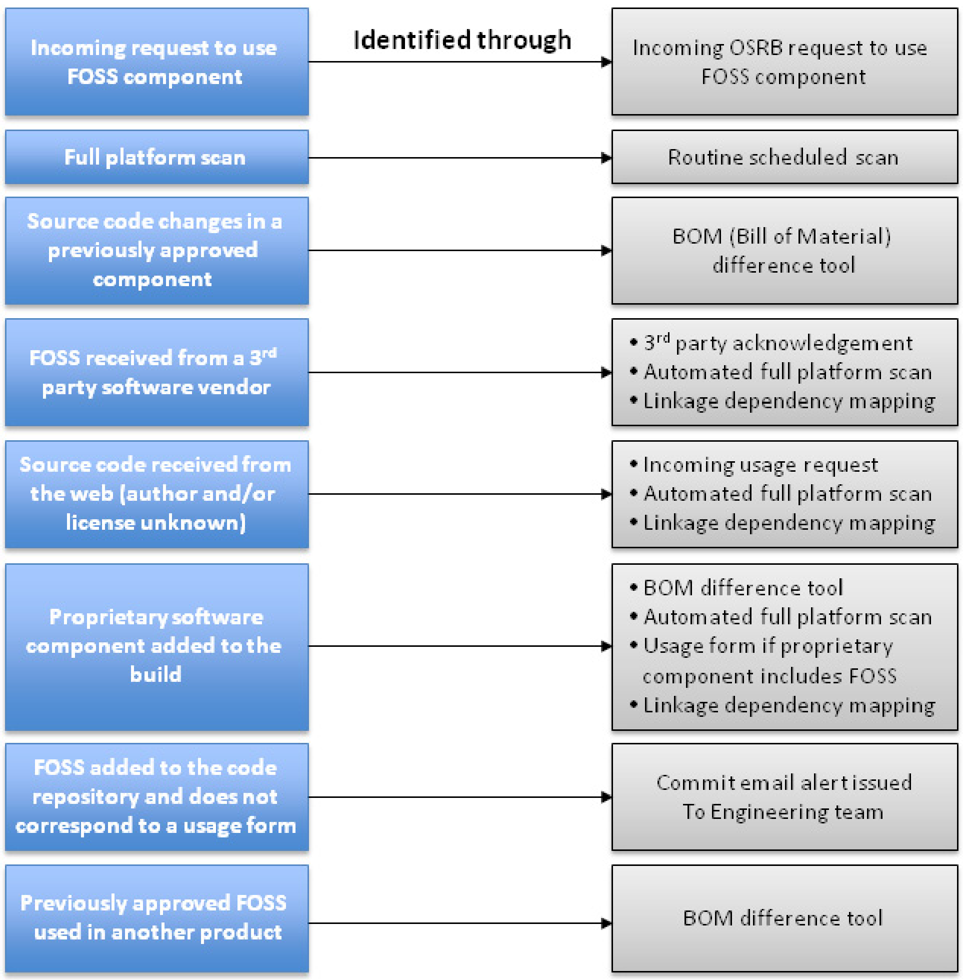
Figure 3. Auditing source code as part of the compliance end-to-end process

The auditing personnel perform a source code scan iteratively from one release to another release label, to build a chain of evidence that what is included in the release is compliant to the various applicable FOSS licenses.

The goals of the audit are to:

- Update the release's bill of materials to account for any FOSS added in releases since the last previous scan
- Confirm the origin(s) of the source code, including the provenance of any FOSS
- Flag dependencies, code matches and licensing conflicts

Auditing Phase	
Prerequisites	**Outcome**
• A proper compliance record (also called ticket) is created capturing all necessary information about the usage of that specific FOSS and providing the location of the source code within the internal build system. • In some cases, specifically when a full platform scan is done, a FOSS component may be scanned before having a proper compliance record. In this case, a record is created when the FOSS component is discovered.	• An audit report identifying the origins and licenses of the source code. • Change request tickets are filed against the appropriate engineering team for any issues identified during the audit that require resolution.



	Identified through
Incoming request to use FOSS component	Incoming OSRB request to use FOSS component
Full platform scan	Routine scheduled scan
Source code changes in a previously approved component	BOM (Bill of Material) difference tool
FOSS received from a 3rd party software vendor	• 3rd party acknowledgement • Automated full platform scan • Linkage dependency mapping
Source code received from the web (author and/or license unknown)	• Incoming usage request • Automated full platform scan • Linkage dependency mapping
Proprietary software component added to the build	• BOM difference tool • Automated full platform scan • Usage form if proprietary component includes FOSS • Linkage dependency mapping
FOSS added to the code repository and does not correspond to a usage form	Commit email alert issued To Engineering team
Previously approved FOSS used in another product	BOM difference tool

Figure 4. Methods to identify and audit incoming FOSS

Several actions can trigger a discovery and an audit for software component (Figure 4):

- Incoming request to use a FOSS component
- Full platform scan
- Source code changes in a previously approved component
- FOSS received from a third party vendor
- Source code downloaded from the web (unknown author or license)
- Proprietary software added to the software platform
- FOSS added to the code repository and does not correspond to a usage form
- Using previously approved FOSS in a different product

Resolving Issues

In this step of the compliance due diligence, all issues identified during the auditing step are resolved (Figure 5).The OSRB Chair will monitor closure of tickets assigned to engineers during the Audit step. Once the engineers have resolved the identified issues, the OSRB Chair will request a new audit to confirm that the resolved issues do not exist anymore.

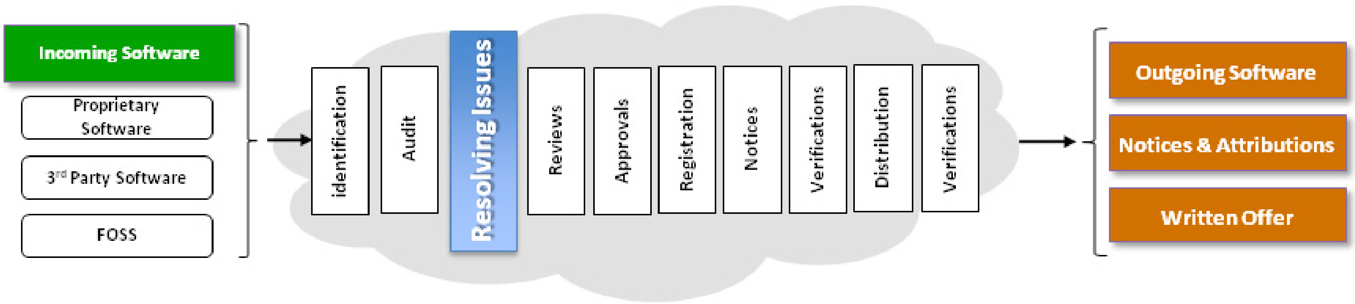


Figure 5. Resolving issues as part of the compliance end-to-end process

Resolving Issues Phase	
Prerequisites	**Outcome**
A source code scan has been completed and an audit report is generated identifying the origins and licenses of the source code and flagging source code files that were not identified and that need further investigation.	A resolution for each of the flagged files in the report and a resolution for any flagged license conflict.

Reviews

Once the auditing is complete and all issues identified earlier have been resolved, the compliance ticket for a specific software component moves to the review step (Figure 6). Various reviews are performed, as depicted in Figure 7, and all identified issues must be resolved. The reviewers need to understand the licenses that govern use, modification and distribution of the software and identify the obligations of the various licenses.

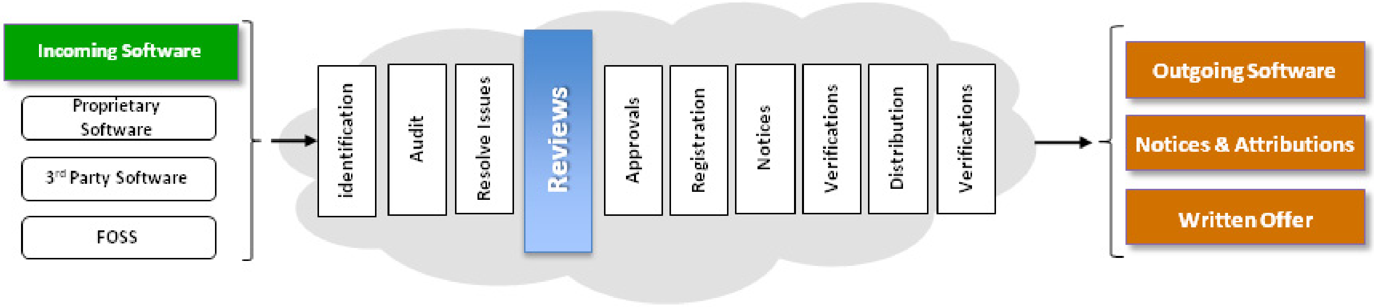


Figure 6. Reviews of FOSS usage as part of the compliance end-to-end process

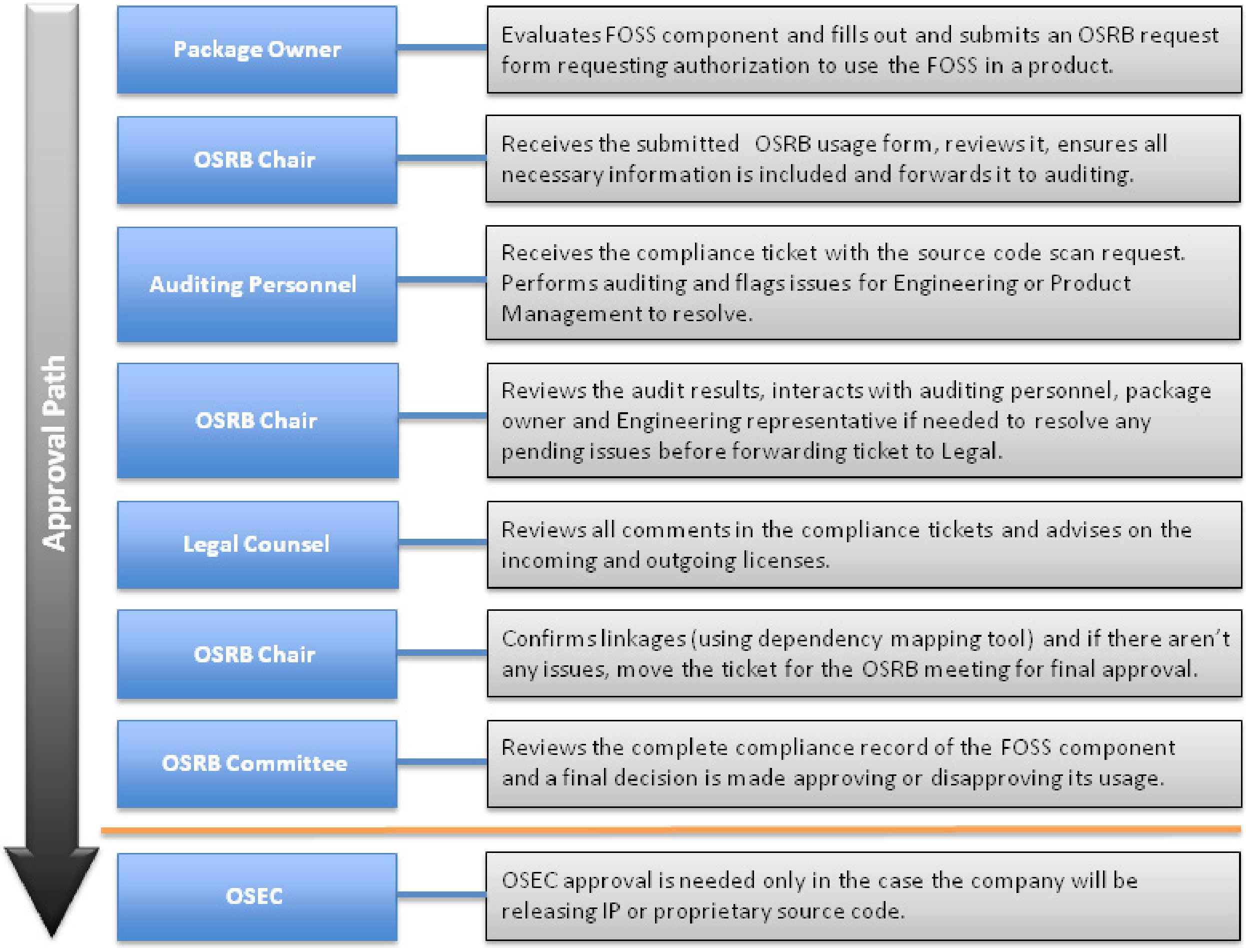
For any given software components, the reviewers of the compliance ticket are:

- Internal package owner
- Auditing personnel
- OSRB (Open Source Review Board) which includes the OSRB chair (Compliance Officer), the

Legal counsel and the OSRB engineering representative

- OSEC (Open Source Executive Committee)

Review Phase	
Prerequisites	**Outcome**
Source code has been audited and all issues have been resolved.	OSRB members perform an architecture review and a linkage analysis for the specific component and mark it as ready for the next step (i.e. Approval) if no issues were uncovered.



Package Owner: Evaluates FOSS component and fills out and submits an OSRB request form requesting authorization to use the FOSS in a product.

OSRB Chair: Receives the submitted OSRB usage form, reviews it, ensures all necessary information is included and forwards it to auditing.

Auditing Personnel: Receives the compliance ticket with the source code scan request. Performs auditing and flags issues for Engineering or Product Management to resolve.

OSRB Chair: Reviews the audit results, interacts with auditing personnel, package owner and Engineering representative if needed to resolve any pending issues before forwarding ticket to Legal.

Legal Counsel: Reviews all comments in the compliance tickets and advises on the incoming and outgoing licenses.

OSRB Chair: Confirms linkages (using dependency mapping tool) and if there aren't any issues, move the ticket for the OSRB meeting for final approval.

OSRB Committee: Reviews the complete compliance record of the FOSS component and a final decision is made approving or disapproving its usage.

OSEC: OSEC approval is needed only in the case the company will be releasing IP or proprietary source code.

Figure 7. Reviewers of the compliance ticket and their roles and responsibilities

As part of this step of compliance due diligence, there are two important reviews: architecture review and linkage analysis review.

Architecture Review

The goal of the architecture review is to analyze the interactions between the FOSS 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 FOSS components to the proprietary components. The internal package owner, the OSRB engineering representative and the FOSS 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 the dependency problem by reworking the source code.

Link Analysis Review

The goal of linkage analysis is to find potentially problematic code combinations at the static and dynamic link level, such as linking a GPL library to a proprietary source code component. The OSRB Chair performs this review using an automated tool. If the OSRB Chair identifies a linkage conflict, it is reported to Engineering to resolve it.

Approvals

In this step, the software component is either approved for usage in the product or not. The approval comes from the OSRB, which includes Legal Counsel, Engineering representative and FOSS expert. Once all reviews have been completed, the compliance ticket for a specific software component moves to the approval step (Figure 8).

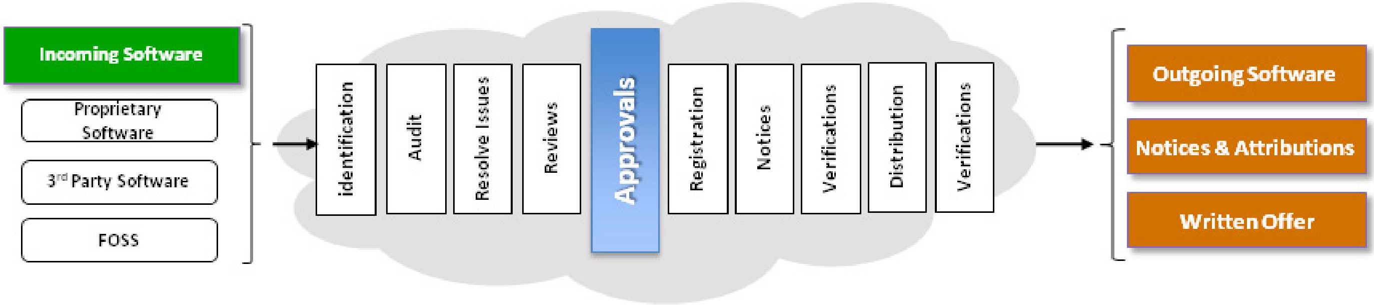


Figure 8. Approvals as part of the compliance end-to-end process

For most software components, the approval is granted by the OSRB given that the ticket has made it that far in the compliance process. In some cases, when the software component in question involves a patent non-assert, the approval is escalated to the OSEC (Open Source Executive Committee). Once the OSRB approves the usage of a FOSS component, the OSRB communicates to the product teams about the approval so they understand their responsibilities and start their preparations to fulfill the license obligations. If the OSRB rejects the use of the FOSS component, they communicate the reason for rejection to the requester and this information is recorded as part of the compliance ticket. As a result, the FOSS component cannot be used in the product, though the requester can consider submitting a rebuttal for reconsideration by the OSRB.

Approval Phase	
Prerequisites	Outcome
All OSRB members have reviewed the compliance ticket, and the OSRB has completed the architecture review and linkage analysis.	Approval or denial of usage of the specific component.

Registration

Once a software component has been approved for usage in a product, its compliance ticket will be updated to reflect the approval and it will be added to the software inventory that tracks FOSS used in products (Figure 9).

It is important to note that a FOSS software component is approved based on a specific version and usage in a specific product version. If a new version of this software component is available, engineering teams need to go through the process again to get approval for using the new version. We have witnessed on several occasions FOSS projects changing licenses in between versions, hence the importance of going through the process again. Furthermore, if engineering teams want to use the same software component in a different product, they need to issue a new request. Approvals are dependent on usage models and for instance, a GPL software component that is approved for inclusion in Product A may not be approved for inclusion in Product B based on a different usage model. The OSRB may decide to issue an approval of broader scope for FOSS components under certain licenses.

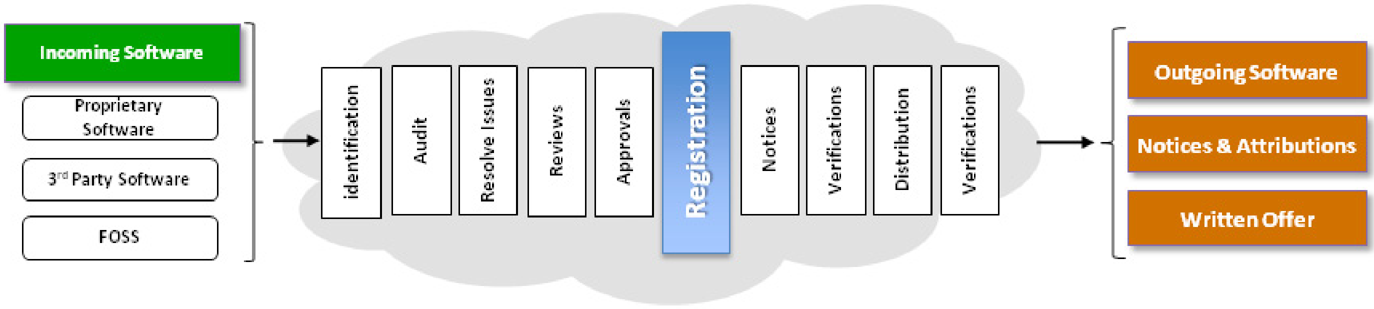


Figure 9. Registration of the approved FOSS in the software inventory

Registration Phase	
Prerequisites	**Outcome**
The OSRB has approved the component's usage in the product.	The component is registered in the software inventory marking the component name, version, internal owner and the details of the product where it's being used such as product name, version, release number, etc.

Notices

One of the key obligations of using FOSS is the documentation obligation, also referred to as the notice obligation. Companies using FOSS in an externally distributed product must:

- Inform the end user of their product how to obtain a copy of the FOSS source code (when applicable)
- Acknowledge the use of FOSS by providing full copyright and attribution notices
- Reproduce the entire text of the license agreements for the FOSS code included in the product.

If companies are non-compliant with FOSS license obligations, they are not licensed and thus can be sued by the copyright holder for copyright infringement and can potentially lose the right to use and distribute the software in question. In order to fulfill documentation obligations, appropriate notices must be included with the product. In this step of the compliance due diligence, the OSRB Chair prepares the notices and passes them to the appropriate departments for fulfillment (Figure 10).

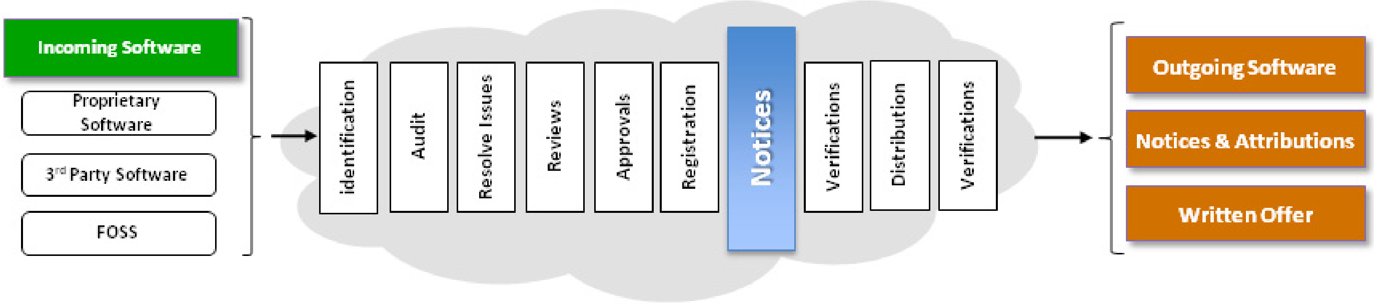


Figure 10. Updating notices as part of the compliance end-to-end process

Notices Phase	
Prerequisites	**Outcome**
The software component has been approved for usage and it has been registered in the software inventory.	The license, copyright and attribution notices for that specific component are prepared and passed along to the appropriate departments to be included in the product documentation.

Pre-Distribution Verifications

The next step in the compliance due diligence is to decide on the method and mode of distribution, type of packages to distribute and mechanism of distribution (Figure 11).

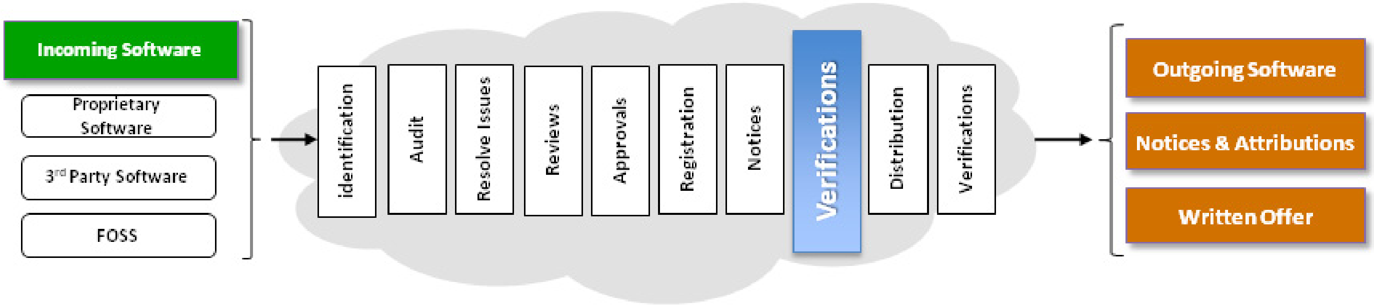


Figure 11. Pre-distribution verifications as part of the compliance end-to-end process

In addition, part of the pre-distribution verification is to ensure that:

- FOSS packages destined for distribution have been identified and approved
- The source code packages (including modifications) have been verified to match the binary equivalent shipping in the product
- All appropriate notices have been included in the product documentation to inform end-users of their right to request source code for identified FOSS
- All source code comments have been reviewed and any offending or inappropriate comment has been removed

Pre-Distribution Verifications Phase	
Prerequisites	**Outcome**
• Component has been approved for usage • Component has been registered in the software inventory • All notices have been captured and sent for fulfillment	• Decide on distribution method and mode • Ensure that all the pre-distribution verifications have been successfully completed.

Distribution

Once all pre-distribution verifications have been completed, the next step is to upload the FOSS packages to the distribution web site, identified with labels as to which product and version it corresponds to (Figure 12). Note that this action is helpful to those desiring code download but may not be sufficient by itself to satisfy license obligations.

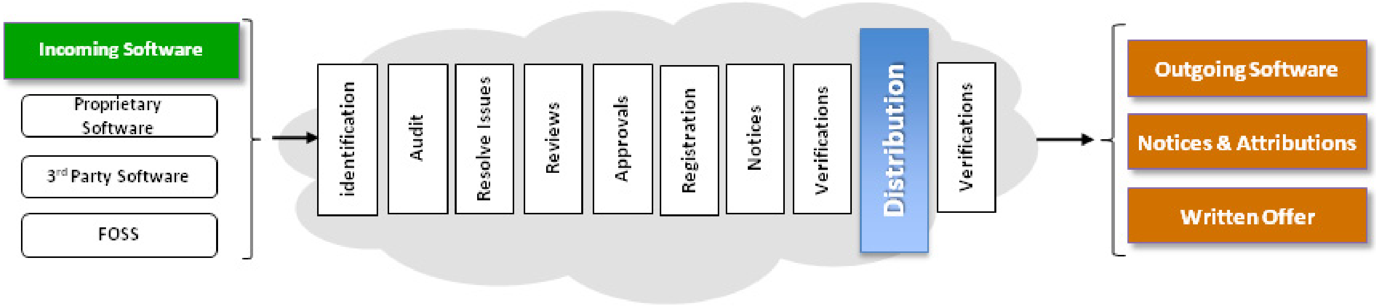


Figure 12. Distribution of source code as part of the compliance end-to-end process

Furthermore, a recommended practice is to provide email and snail mail contact information for any compliance or FOSS-related questions.

Distribution Phase	
Prerequisites	**Outcome**
All pre-distribution verification have been checked and no issue is discovered.	The source code of the component in question is uploaded to the web site for distribution (if that was the distribution method of choice).

Final Verifications

Once you upload the FOSS packages to the distribution web site, there are verification steps to validate that the package has been uploaded correctly, and can be downloaded and uncompressed on an external computer without errors (Figure 13).

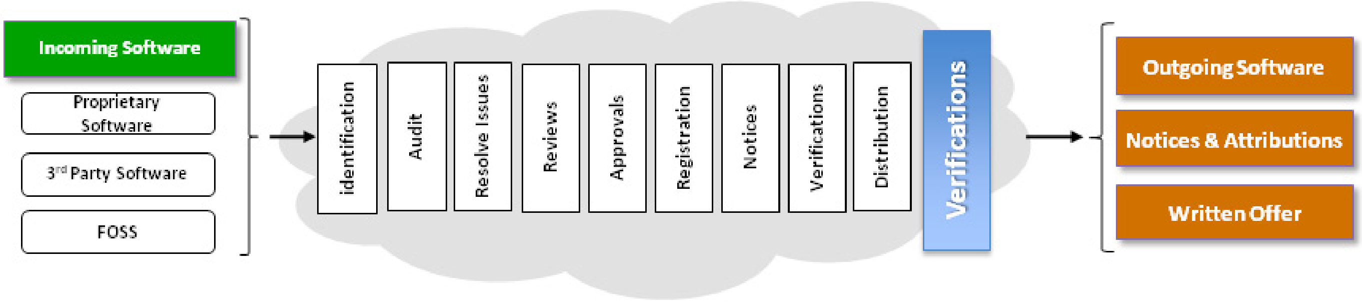


Figure 13. Conducting final post-distribution verifications

Final Verifications Phase	
Prerequisites	Outcome
The source code is published on the web site.	Verifications that the source code is: • Uploaded correctly • Corresponds to the same version that was approved • Accessible for download for the public

Conclusion

In this paper, we provided a detailed review of an end-to-end compliance process and illustrated the various compliance steps or phases that FOSS or software components containing FOSS go through before they get approved for use in a product intended for external distribution. Throughout the paper, we looked at what happens in these various phases, identified the inputs and outputs of each phase and how to keep tight control on software usage via the compliance process. How can you help your company be ahead of the curve on FOSS compliance? Check out the resources that the Linux Foundation is making available ranging from professional training to tools and everything in between. Details are in the resources section. Thanks for reading and stay tuned for more publications on the topic of FOSS compliance!

Linux Foundation Resources

- **Open Compliance Program:** http://www.linuxfoundation.org/programs/legal/compliance
- **Professional and Comprehensive Compliance Training:** The Linux Foundation offers hands-on training from compliance experts for individuals and companies responsible for achieving compliance with open source licenses and establishing an open source compliance program, as well as for those who simply want to learn more about compliance. Options available include live onsite training in addition to instructor-led live remote training. http://www.linuxfoundation.org/programs/legal/compliance/training-and-education
- **Compliance Publications:** http://www.linuxfoundation.org/publications
- **Open Compliance Directory and Rapid Alert System:** http://www.linuxfoundation.org/programs/legal/compliance/directory
- **Compliance Tools:** http://www.linuxfoundation.org/programs/legal/compliance/tools
- **The Software Package Data Exchange™:** The SPDX™ specification is a standard format for communicating the components, licenses and copyrights associated with a software package. http://www.spdx.org/
- **FOSSBazaar:** An open community of technology and industry leaders who are collaborating to accelerate adoption of free and open source software in the enterprise. http://fossbazaar.org/

Acknowledgments

The author would like to express his gratitude to Karen Copenhaver (Legal Director of the Linux Foundation and Partner in Choate, Hall & Stewart LLP 's Business & Technology practice) and to Philip Koltun (The Linux Foundation Director of Open Compliance Program) for their reviews and valuable input.

About the Author

Dr. Ibrahim Haddad manages The Linux Foundation's Mobile Linux initiatives and works with the community to facilitate a vendor-neutral environment for advancing the Linux platform for next-generation mobile computing devices.

About the Open Compliance Program

The Linux Foundation's Open Compliance Program is the industry's only neutral, comprehensive software compliance initiative. By marshaling the resources of its members and leaders in the compliance community, the Linux Foundation brings together the individuals, companies and legal entities needed to expand the use of open source software while decreasing legal costs and FUD. The Open Compliance Program offers comprehensive training and informational materials, open source tools, an online community (FOSSBazaar), a best practices checklist, a rapid alert directory of company's compliance officers and a standard to help companies uniformly tag and report software used in their products. The Open Compliance Program is led by experts in the compliance industry and backed by such organizations as the Adobe, AMD, ARM Limited, Cisco Systems, Google, HP, IBM, Intel, Motorola, NEC, Novell, Samsung, Software Freedom Law Center, Sony Electronics and many more. More information can be found at http://www.linuxfoundation.org/programs/legal/compliance.

OPEN COMPLIANCE PROGRAM

1796 18th Street, Suite C
San Francisco, CA 94107
+1 415 723 9709
http://www.linuxfoundation.org

THE LINUX FOUNDATION

The Linux Foundation promotes, protects and advances Linux by providing unified resources and services needed for open source to successfully compete with closed platforms.

To learn more about The Linux Foundation, the Open Compliance Program or our other initiatives please visit us at http://www.linuxfoundation.org/.

THE
LINUX
FOUNDATION