Open Source for Open Source License Compliance

Oliver Fendt, Michael C. Jaeger

Open Source for Open Source License Compliance

Oliver Fendt and Michael C. Jaeger

[1] Siemens AG, Corporate Technology,
Otto-Hahn-Ring 6, 81379 Munich, Germany
{oliver.fendt|michael.c.jaeger}@siemens.com

Abstract. Today, many software systems are of a level of complexity that no single company can implement modern solutions alone. Thus many companies engage in the open source software (OSS) ecosystem to keep the development costs manageable. But the usage of third-party components (both OSS and commercial) also mandates the need of a license compliance process supported by suitable tools. This paper is focused on using open source tools and relevant processes for open source license compliance. OSS license compliance is a very important topic, and requires appropriate processes, culture, and tools.
This work is based on extensive practical industrial experience and broad use at Siemens AG. We first describe the process and culture, then a set of tools. We complement this with related work in the community and future directions.

Keywords: license compliance, license scanning, component inventory, open source management

1 Elements for an OSS Management Process

The clearing of components and involved licenses is part of an OSS management process that covers the handling of 3rd party software. As described in [1] the following main goals have to be achieved by the OSS management process:

- Assurance that only suited components are approved for integration – after the involved licensing has been determined and understood.
- Assurance of license requirement fulfillment – determining the involved licenses and understanding involved terms has the purpose to actually implement those license requirements in order to provide a compliant product or delivery.
- Storing and tracking of OSS components. An organization takes advantage from keeping track of 3rd party software use: on one hand, it serves the purpose of documentation if questions or inquiries arise about OSS usage for example. On the other hand, an organization wants to prevent redundant clearing work and reuse clearing results for future uses of the same component.

The management process requires different elements in an organization for its implementation. On one side there are organizational aspects, which can be responsibilities, roles, contact persons and a decision board. These aspects are for example summarized by the OpenChain (https://www.openchainproject.org) standard which describes which basic organization elements should be present.

And of course, a culture that is "open" to the use of open source is a key element of any open source management. Mentioning our organization as an example, we have regular internal events on OSS and a broadly-used web-based training educating all employees not only in software development related roles, but also beyond e.g. procurement and product management. Since its introduction in 2015, several ten thousands have attended the Web-based training so far. As an example for a dedicated role with decision responsibility, "third-party software experts" of the various business units regularly meet to discuss and agree on common approaches, best practices and challenging cases of 3rdparty software usage.

In addition to company-internal activities described here, we also actively engage in world-wide activities of the OSS community. Such engagement is not only limited to re-using OSS components, and re-using OSS clearing results, but also other aspects such as agreeing on standard formats like SPDX (https://spdx.org/) and joining activities like the OpenChain projects that promote a high level of license clearing processes and enable sharing of cleared components

This paper focuses on the use of tools and services which are available as open source from the community for implementing license compliance. Despite commercial tools being available, our work shows that OSS tools can be adopted by organizations and provide an effective and open approach. Among many advantages, OSS software allows for modification and adaptation to own needs; can better use the latest innovative approaches from the OSS community; do not require the establishment of a commercial contractual relation; and it allows for using software without spending monetary resources, which is an advantage also for non-commercial organization.

An open source management process needs to count on different artifacts for a successful implementation. These include a tool for analyzing the licenses present in 3rd party software components, a database that holds license interpretation of used licensing, a catalogue application that keeps track of 3rd party software in products and services, an application that keeps track of the progress of clearing tasks as well as providing an overview of all involved 3rd party component w.r.t. clearing status, a source code and component and repository that stores used software for analyses and reference to clearing results, an application which generates the licensing documentation for distribution as well as for internal approval purpose, product OSS code collector, and a code verifier.

The management of software component takes place inside the software engineering process – there are software repositories, dependency management systems and packing tools which already create various software artifacts for distribution. Naturally, the open source management tools must be integrated with the already existing software engineering facilities, for example continuous integration tools.

Figure 1 shows a general setup and context of software management, including basic elements for a software development tool chain. On the left there are different sources for 3rd party software as well as information about them. All these elements are outside of the organization and are relevant sources for OSS management. In addition to inbound 3rd party software, public repositories and databases holding information about 3rd party software components exist which can hold relevant information for component and license clearing.

In the middle of this diagram, the integration of OSS management tools with existing software engineering tools is outlined: internal repositories and tools for building the

software are integrated with clearing tools. Organizational internal repositories for software exist that contain information used for component clearing, but more importantly, checks and analyses tool for a clearing are triggered by the build and software production automation.

In basic terms, the input for managing 3rd party components, originates from the software building infrastructure. Going further right, the software is prepared for distribution which involves the generation of distribution documentation as well as checks if the license terms are fulfilled. Around all this, organization internal repositories, such as internal git servers, but also artifact servers are not only relevant for software development, but also for OSS management. In additional, a central element is a component catalogue application which acts as a central inventory capturing component usage. It is fed by analysis information of 3rd party components as well as usage information in products and services.

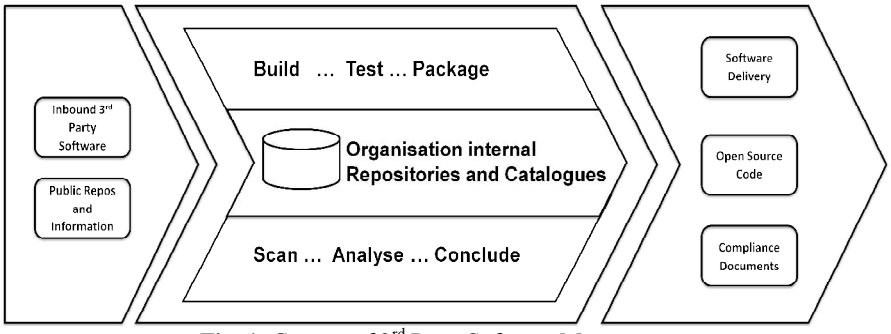

Fig. 1: Context of 3rd Party Software Management

2 OSS Software Projects and Tools

For the elements depicted in the figure 1, different OSS tools exist for the implementation of the given elements. We would like to introduce a minimal set for implementing a OSS management process with the goals given above. The first part on the right hand side refers, of course, to OSS published software (e.g. Sourceforge, or nowadays Github or git servers of OSS foundations), but also software package servers, such as those for Linux distributions as well as those for developing software for a particular languages (e.g. Nuget for C# components). In addition, software can be obtained from commercial vendors and as such, represents also inbound 3rd party software.

As a second element that an organization can use are the now growing offerings for information about 3rd party software. Some companies offer concluded licensing information, some analyses and metadata describing how sane the software project is. Moreover, public information relevant for an OSS management process are license (text) collections and interpretations about licenses. A license collection is very useful for identifying license texts as found in OSS. As for the interpretation of licenses, there are also libraries in the form of offering interpretations for licenses, as well as information pages about articles that discuss licenses cases from a legal point of view.

There are a number of open source tools for various aspects of open source license clearing and management. In our organization, the two main OSS tools we use for this are FOSSology (https://www.fossology.org/, https://github.com/fossology/fossology)

and SW360 (https://eclipse.org/sw360, https://github.com/eclipse/sw360). FOSSology is a Linux Foundation project which has a meanwhile over 10 year project history providing a solution for license scanning [2]. It has a very precise license scanning facility that allows us to identify licenses as well as copyright information well. SW360 is a project hosted by the Eclipse foundation. It provides both a web application and a repository to collect, organize and make information available about software components.

For the "Assurance of license requirement fulfillment" step in the license clearing process all of the OSS licenses of the approved component have to be thoroughly examined by FOSSology to clearly identify the requirements (license obligations) to fulfill as well as to define the ways of fulfilling the obligations. For the "Storage and tracking of OSS components" step we use SW360. With it, the usage of the approved components are typically registered and uniquely tracked for future reference and reuse. The goal is not only to have a database of already approved OSS components with all their associated information (e.g. suitability ranking, copyright holders, applicable licenses, set of requirements the products have to fulfill derived from the license situation, etc.). An additional goal is also to provide a means for internal (and external) knowledge sharing on how to use, integrate and analyze the component.

Figure 2 shows a high level overview of an integrated compliance tool chain, which represents also a more detailed diagram compared to Figure 1. On the left side, the incoming software is depicted. For successful use of OSS, also contributions to the project should be considered. The right side shows the deliverables, products, or the software which the organization conveys. In the middle part, a collection of elements is shown which play a role license compliance and OSS management. The connection of all the elements together, integrated with build infrastructure provides us with a compliance tool chain.

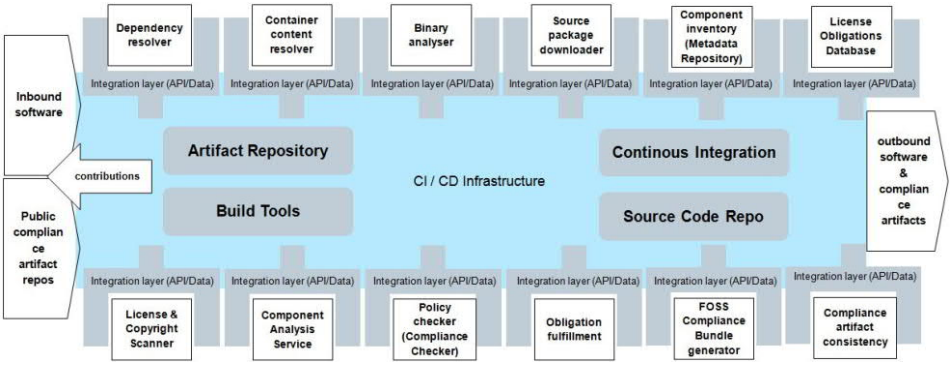

Fig. 2. Overview of an Integrated Compliance Tool Chain

3 Related Work

We are part of a vibrant and growing community in the area of open source tools of open source license compliance. Numerous members of this community presented at a forum organized by the German BITKOM organization [3] and the community strives to find ways to further foster the ecosystem of OSS-based tools for OSS license compliance and related topics. In addition to our own participation (FOSSology, SW360), other tools

include Quartermaster (https://qmstr.org, https://github.com/qmstr/qmstr) and OSS Review Toolkit (https://github.com/heremaps/oss-review-toolkit).

More relevant OSS tools and activities include: ScanCode, another OSS license clearing tool, Tern which analysis containers for their used OSS for compliance (https://github.com/vmware/tern), and ClearlyDefined (https://clearlydefined.io, https://github.com/clearlydefined). ClearlyDefined and its hosting foundation, the Open Source Initiative, are on a mission to help OSS projects by being, clearly defined in terms of compliance relevant information. According to ClearlyDefined, a lack of clarity around licenses and security vulnerabilities reduces engagement and this can lead to fewer user, contributors and a smaller community. Although not being a tool, the Software Heritage project (https://www.softwareheritage.org/) is an important part of the OSS ecosystem. Ideally, it could be a central source that the various tools could use as the central repository of the OSS source code which would be of significant benefit to the entire community.

4 Summary and Future Work

Modern software engineering seems to be increasingly use OSS including many of the aspects that have been a hallmark of OSS –transparency, improvement, sharing, and collaboration. As open source becomes more and more prevalent in our products, and also in the tools that help create those products, it is logical to also increasingly use open source tools and processes to do OSS license compliance. The open source community on this topic is active and includes well-established tools with a long history such as FOSSology, but also a set of numerous other tools as well. We encourage the community to continue to work together to further extend the scope and the increased use of these tools in practice.

We are firmly convinced that an open source approach is the best way to able to keep up with the fast and ever-faster changing software world. Some of the future directions and areas for future work for the international OSS community are:

- Reuse of (a subset of) clearing results across an external ecosystem,
- License compliance in the context of continuous delivery / DevOps ecosystem identification of dependencies, automatic download of the source code packages of the used packages (incl. dependencies), license analysis – and licenses determination, copyrights and automatic generation of license compliance artifacts. To realize a fully functional DevOps setup the license compliance as well as cyber-security processes need to be seamless integrated in the environment
- Enhancing automatic container analysis for license compliance with the goal to identify all applicable licenses.

References

1. Oliver Fendt, Michael C. Jaeger, Ricardo Jimenez Serrano: Industrial Experience with Open Source Software Process Management, IEEE COMPSAC 2016
2. Michael C. Jaeger, Oliver Fendt, Robert Gobeille, Maximilian Huber, Johannes Najjar, Kate Stewart, Steffen Weber, Andreas Würl: The FOSSology Project: 10 Years of License Scanning, International Free and Open Source Software Law Review, Vol. 9, Issue 1
3. Forum Open Source 2018 - BITKOM 2018, https://www.bitkom.org/-Themen/Technologien-Software/Open-Source/Forum-Open-Source-2018.html