Towards a Fully Automated FOSS Compliance Tool Chain

This post summarizes the lecture ""*Towards a Fully Automated FOSS Compliance Tool Chain*" held Karsten Reincke and Daniel Eder at the "Standardization Offsite" 2022: We will describe and demo the challenge and offer an already usable solution (which is not the last word of wisdom!) before we sketch a *perfect* solution.

(1) Challenge

Products based on **F**ree and **O**pen **S**ource **S**oftware normally use complex software stacks with many different FOSS (sub) components.
Regardless of the size of this software stack, the one who wants to distribute his product compliantly, has to bundle the respective compliant artifacts with the product. To be able to do so, he must

- know which open source software components are (recursively) embedded into his product,
- know under which FOSS license the components and its (sub)parts are licensed
- know what the respective licenses require to bundle with the product
- create the respective *compliance artifacts*
- bundle them with his product.

We will demo the complexity of this constraint on the base of the FOSS Compliance TestCase No 11 released by Deutsche Telekom

(2) Semi Automated solution

The FOSS community offers already a set of supporting tools: http://oss-compliance-tooling.org/Tooling-Landscape/OSS-Based-License-Compliance-Tools/. These tools focus on scanning FOSS software for gathering information, necessary to solve the challenge. Or they solve important subtasks. But none of them tries to create the required compliance artifacts automatically on the base of the gathered information. It is clear why: In order to be able to do that, they would have to have implemented the legal knowledge in the form of an expert system.

We will demo the tool `sag.oscf` = *Semi Automated Generation of Open Source Compliance Files* https://github.com/Open-Source-Compliance/sag.oscf - released by Deutsche Telekom to close this gap (at least partially).

The advantage of this tool is, that one

- can use it already
- need no longer the respective legal advice.

The disadvantage of this tool is, that

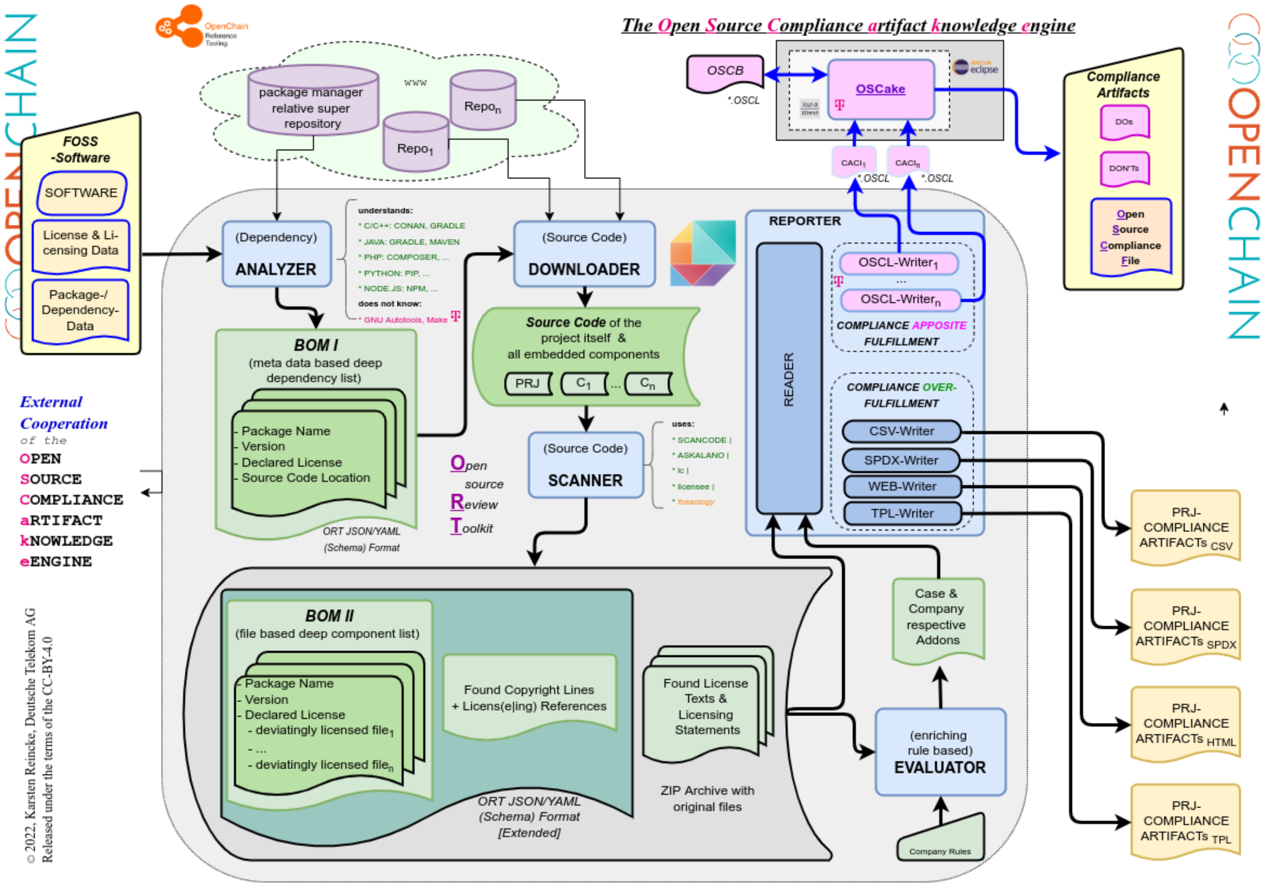
- one still must gather the information more or less manually
- the expert system is only able to deal with some FOSS licenses

(3) Towards A Fully Automated FOSS Compliance Tool Chain: ORT

Despite their limitations, the existing FOSS compliance tools offer one great benefit: doing the same work manually, would establish a big cost trap. In our times, the work of the compliance tools must become repeatable because they have to be integrated in a CI/CD chain.

We describe ORT, the Open Source Review Toolkit and the changes that DT has suggested and implemented to help close the gab.

To get a summary take a look at this image:

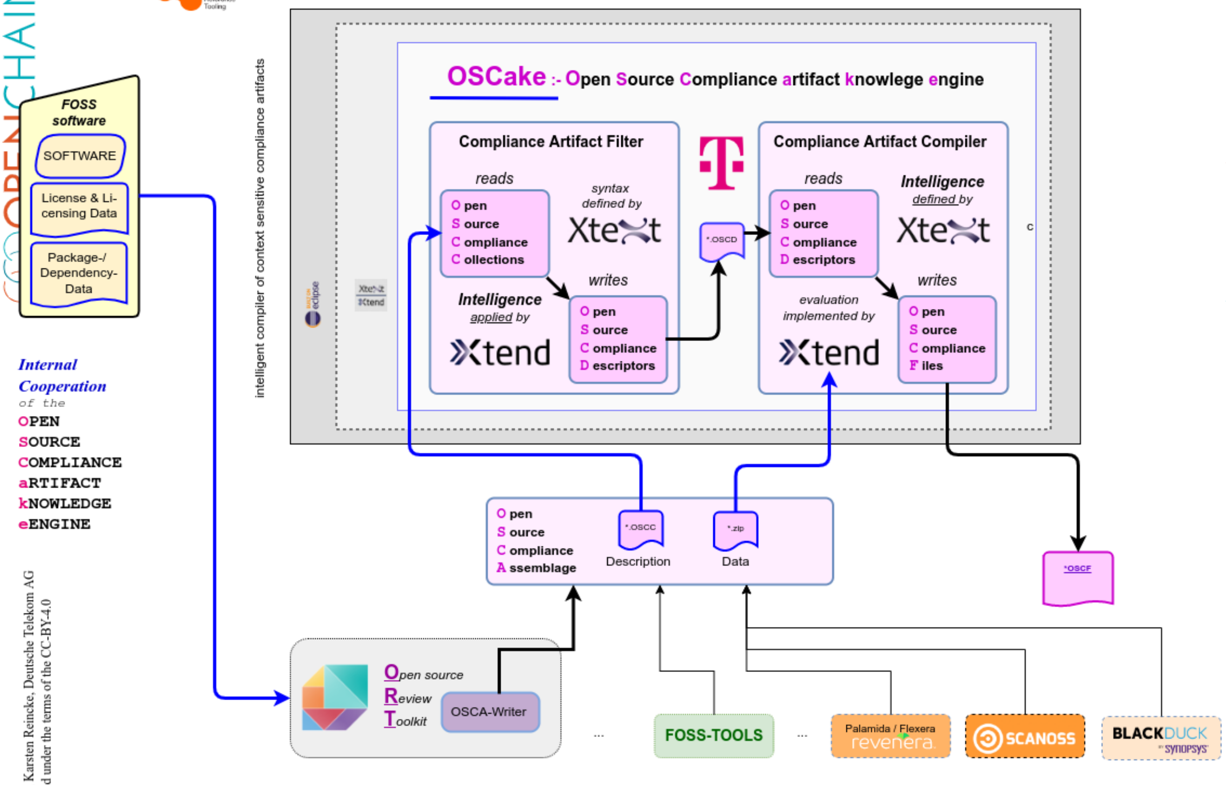

OPENCHAIN
OpenChain Reference Tooling
The Open Source Compliance artifact knowledge engine
FOSS -Software
SOFTWARE
License & Licensing Data
Package-/ Dependency-Data
package manager relative super repository
www
Repo$_n$
Repo$_1$
(Dependency) ANALYZER
understands:
* C/C++: CONAN, GRADLE
* JAVA: GRADLE, MAVEN
* PHP: COMPOSER, ...
* PYTHON: PIP, ...
* NODE.JS: NPM, ...
does not know:
* GNU Autotools, Make
(Source Code) DOWNLOADER
BOM I
(meta data based deep dependency list)
- Package Name
- Version
- Declared License
- Source Code Location
ORT JSON/YAML (Schema) Format
Source Code of the project itself & all embedded components
PRJ
C$_1$
...
C$_n$
(Source Code) SCANNER
uses:
* SCANCODE |
* ASKALANO |
* lc |
* licensee |
* fossology
Open source Review Toolkit
External Cooperation of the OPEN SOURCE COMPLIANCE aRTIFACT kNOWLEDGE eENGINE
OSCB
*.OSCL
OSCake
eclipse
CACI$_1$
*.OSCL
CACI$_n$
*.OSCL
Compliance Artifacts
DOs
DON'Ts
Open Source Compliance File
REPORTER
READER
OSCL-Writer$_1$
...
OSCL-Writer$_n$
COMPLIANCE APPOSITE FULFILLMENT
COMPLIANCE OVER-FULFILLMENT
CSV-Writer
SPDX-Writer
WEB-Writer
TPL-Writer
PRJ-COMPLIANCE ARTIFACTs CSV
PRJ-COMPLIANCE ARTIFACTs SPDX
PRJ-COMPLIANCE ARTIFACTs HTML
PRJ-COMPLIANCE ARTIFACTs TPL
BOM II
(file based deep component list)
- Package Name
- Version
- Declared License
- deviatingly licensed file$_1$
- ...
- deviatingly licensed file$_n$
Found Copyright Lines + Licens(e|ing) References
ORT JSON/YAML (Schema) Format [Extended]
Found License Texts & Licensing Statements
ZIP Archive with original files
Case & Company respective Addons
(enriching rule based) EVALUATOR
Company Rules
© 2022, Karsten Reincke, Deutsche Telekom AG
Released under the terms of the CC-BY-4.0

(4) Towards A Fully Automated FOSS Compliance Tool Chain: OSCake

Nevertheless, ORT itself is not an expert system, offering and applying the legal knowledge to a specific case. Thus, ORT alone is not able to close the gap. But if we had a system which takes and evaluates the results gathered by ORT and which - as an expert system like *sag.oscf* - applied its embedded legal knoweldge, we would have closed the described gap.

We will discuss the state of OSCake, the Open Source Compliance artifact knowledge engine developed on behalf of and under the leadship of Deutsche Telekom.

To get a summary take a look at this image:


FOSS software
SOFTWARE
License & Licensing Data
Package-/ Dependency-Data

intelligent compiler of context sensitive compliance artifacts

OSCake :- Open Source Compliance artifact knowlege engine

Compliance Artifact Filter
reads
Open Source Compliance Collections
syntax defined by
Xtext
writes
Intelligence applied by
Xtend
Open Source Compliance Descriptors
*.OSCD
Compliance Artifact Compiler
reads
Open Source Compliance Descriptors
Intelligence defined by
Xtext
writes
evaluation implemented by
Xtend
Open Source Compliance Files
c
Open Source Compliance Assemblage
*.OSCC
Description
*.zip
Data
*OSCF
Open source Review Toolkit
OSCA-Writer
...
FOSS-TOOLS
...
Palamida / Flexera
revenera
SCANOSS
BLACKDUCK
by synopsys

Internal
Cooperation
of the
OPEN
SOURCE
COMPLIANCE
aRTIFACT
kNOWLEDGE
eENGINE

© 2022, Karsten Reincke, Deutsche Telekom AG
Released under the terms of the CC-BY-4.0