The OpenChain Reference Tooling Work Group

The OpenChain reference tooling work group 2020

Situation today

The use of OSS in products and services is increasing exponentially.

The OSS compliance process needs to be as automated as possible and fully integrated in the CI/CD pipelines.

An integrated OSS compliance toolchain needs to be modular, flexible and adaptable

Objective and purpose

We are building an open source compliance toolchain ecosystem with open source tools as an open source project.

To accomplish this we:

- Use existing independent tooling projects

- Provide reference workflows to allow their adoption

- Provide the concepts and glue to ensure easy interoperability and integration in existing environments

- Provide reference turnkey toolchains that can be used without fees by anybody

Objective and purpose

Reducing resource costs and improving the quality of results around open source compliance activities.
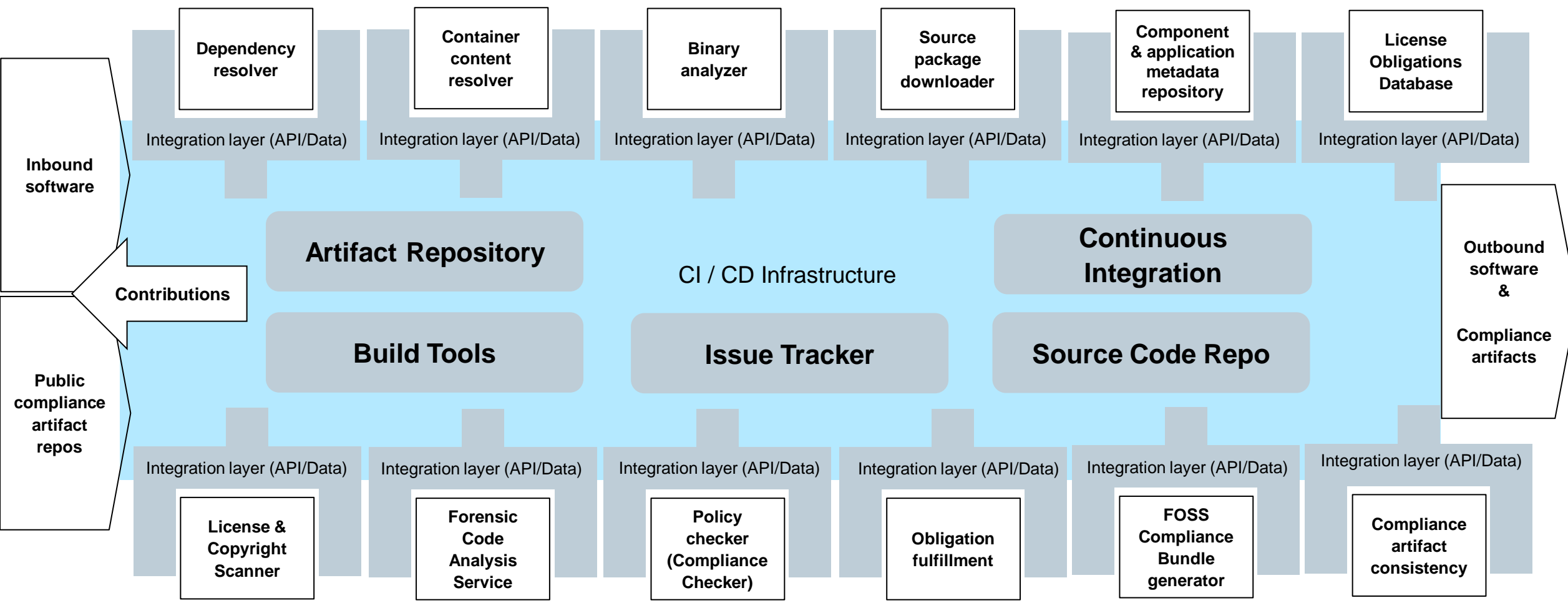
The Approach

The stakeholders / users of the toolchain are identified and their interaction with it are described
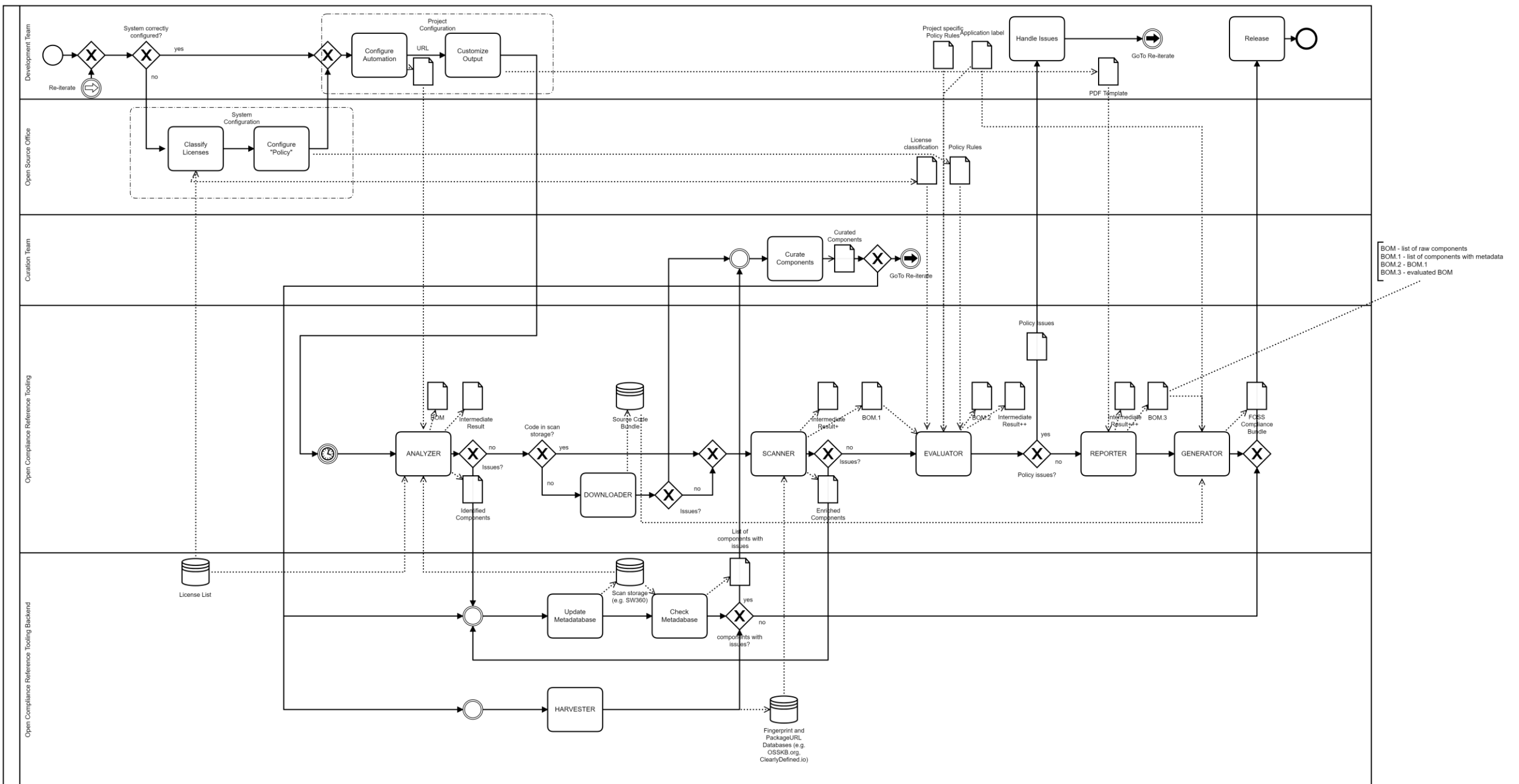
The functional building blocks of an integrated compliance toolchain are identified and described

The workflow and the flow of information is identified

First version of a data model is available

 Oliver Fendt

Big Picture – Integrated Compliance Toolchain

Dependency resolver

Container content resolver

Binary analyzer

Source package downloader

Component & application metadata repository

License Obligations Database

Integration layer (API/Data)

Inbound software

Contributions

Public compliance artifact repos

Artifact Repository

CI / CD Infrastructure

Continuous Integration

Outbound software & Compliance artifacts

Build Tools

Issue Tracker

Source Code Repo

Integration layer (API/Data)

License & Copyright Scanner

Forensic Code Analysis Service

Policy checker (Compliance Checker)

Obligation fulfillment

FOSS Compliance Bundle generator

Compliance artifact consistency

Development Team

System correctly configured?

yes

no

Re-iterate

Project Configuration

Configure Automation

URL

Customize Output

Project specific Policy Rules

Application label

Handle Issues

GoTo Re-iterate

Release

PDF Template

Open Source Office

System Configuration

Classify Licenses

Configure "Policy"

License classification

Policy Rules

Curation Team

Curate Components

Curated Components

GoTo Re-iterate

BOM - list of raw components
BOM.1 - list of components with metadata
BOM.2 - BOM.1
BOM.3 - evaluated BOM

Open Compliance Reference Tooling

BOM

Intermediate Result

Code in scan storage?

no

no

Issues?

Identified Components

yes

DOWNLOADER

no

Issues?

Source Code Bundle

ANALYZER

Enriched Components

SCANNER

no

Issues?

Intermediate Result+

BOM.1

Policy Issues

EVALUATOR

BOM.2

Intermediate Result++

yes

no

Policy issues?

Intermediate Result+++

BOM.3

REPORTER

GENERATOR

FOSS Compliance Bundle

Open Compliance Reference Tooling Backend

License List

List of components with issues

Update Metadatabase

Scan storage (e.g. SW360)

Check Metadatabase

yes

no

components with issues?

HARVESTER

Fingerprint and PackageURL Databases (e.g. OSSKB.org, ClearlyDefined.io)

Example Automation Implementation Using Open Source Tools

SPDX-License-Identifier: CC-BY-SA-4.0

TDOSCA – Test Driven Open Source Compliance Automation

TDOSCA wants to deliver different test cases to evaluate whether a tool or a combination of tools can create the required artifacts in a suited quality. Therefor testcases will be identified and implemented. They will be offered as autonomous repositories by which Open Source Compliance tools can
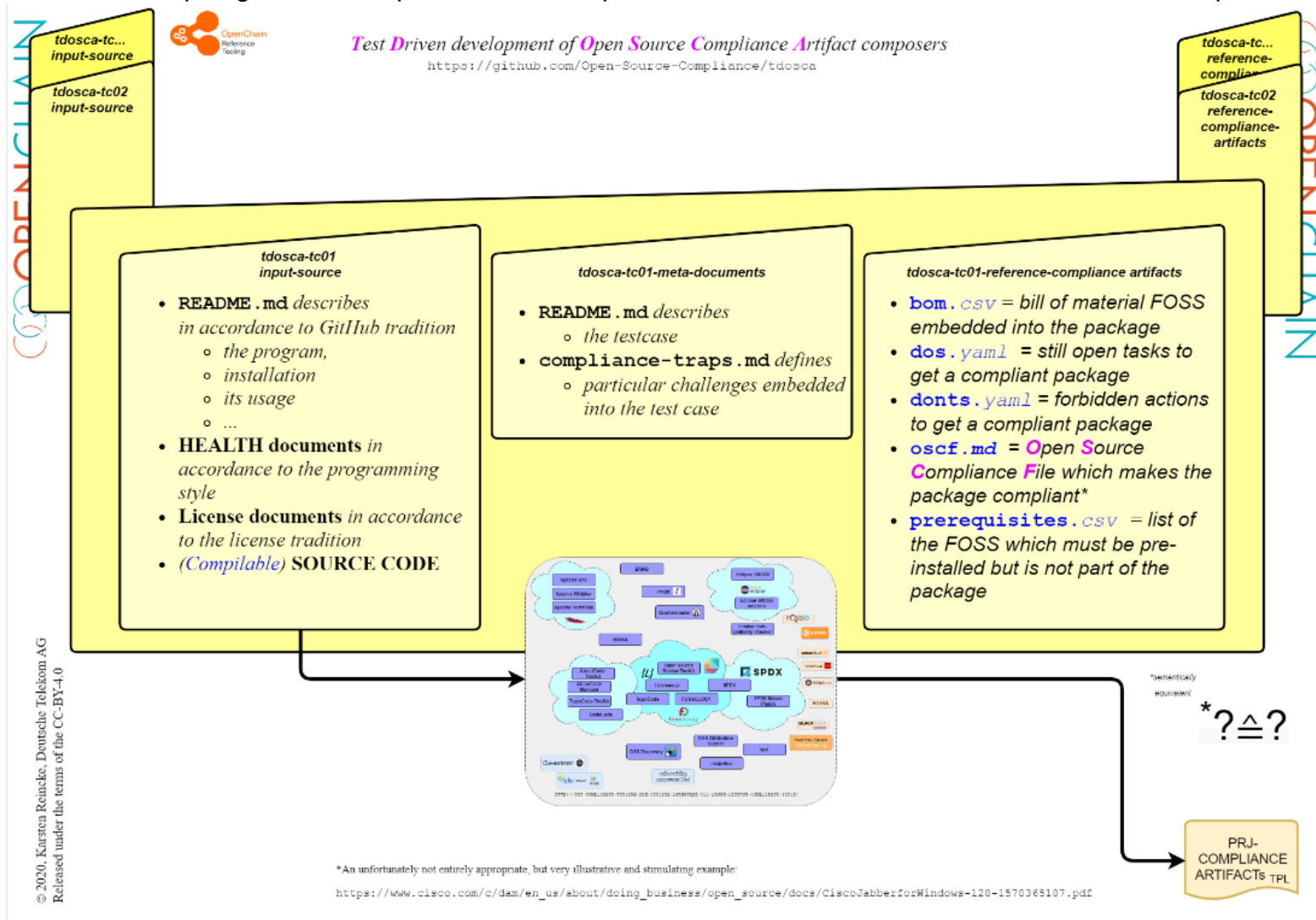
- either prove that they already successfully generate such required artifacts

- or show which parts of an imaginary compliance chain they successfully process.

For doing this job successfully, these tools must gather information about the embedded sub components on which the main component depends or they must have access to the information gathered by other tools. For finding such information the gathering tools do not only evaluate the programming language specific 'include' statements of the software itself, but the statements of respective package manager.

https://github.com/Open-Source-Compliance/tdosca

 Oliver Fendt

TDOSCA - overview

https://github.com/Open-Source-Compliance/tdosca/blob/master/doc/tdcosca-motivation.pdf

Test Driven development of Open Source Compliance Artifact composers
https://github.com/Open-Source-Compliance/tdosca

tdosca-tc...
input-source

tdosca-tc02
input-source

tdosca-tc...
reference-complia...

tdosca-tc02
reference-compliance-artifacts

File originates from the TDOSCA project (https://github.com/Open-Source-Compliance/tdosca) initiated by Deutsche Telekom AG" License CC-BY-4.0

tdosca-tc01 input-source

- **README.md** *describes in accordance to GitHub tradition*
  - *the program,*
  - *installation*
  - *its usage*
  - *...*
- **HEALTH documents** *in accordance to the programming style*
- **License documents** *in accordance to the license tradition*
- *(Compilable)* **SOURCE CODE**

tdosca-tc01-meta-documents

- **README.md** *describes*
  - *the testcase*
- **compliance-traps.md** *defines*
  - *particular challenges embedded into the test case*

tdosca-tc01-reference-compliance artifacts

- **bom.**`csv` = *bill of material FOSS embedded into the package*
- **dos.**`yaml` = *still open tasks to get a compliant package*
- **donts.**`yaml` = *forbidden actions to get a compliant package*
- **oscf.md** = *Open Source Compliance File which makes the package compliant**
- **prerequisites.**`csv` = *list of the FOSS which must be pre-installed but is not part of the package*

semantically equivalent

$*? \triangleq ?$

PRJ-COMPLIANCE ARTIFACTs TPL

*An unfortunately not entirely appropriate, but very illustrative and stimulating example:
https://www.cisco.com/c/dam/en_us/about/doing_business/open_source/docs/CiscoJabberforWindows-120-1570365187.pdf

OSS compliance toolchain things to do

Data model needs to be completed

APIs need to be defined and described

APIs need to be implemented

Tutorials to set up and interact with the functional building blocks need to be created

Reference set up needs to be done and maintained

The OpenChain Reference Tooling Work Group

World-Wide Collaboration, World-Wide Availability

In short words

OSS license compliance for everyone

This is what we do

Your help and expertise is highly welcome

Join us

Reference Tooling Work Group

We are building an open source compliance toolchain ecosystem with open source tools as an open source project.
To accomplish this we:

- Use existing independent tooling projects

- Provide reference workflows to allow their adoption

- Provide the concepts and glue to ensure easy interoperability and integration in existing environments

- Provide reference turnkey toolchains that can be used without fees by anybody

World-Wide Collaboration, World-Wide Availability

Example Automation Implementation Using Open Source Tools
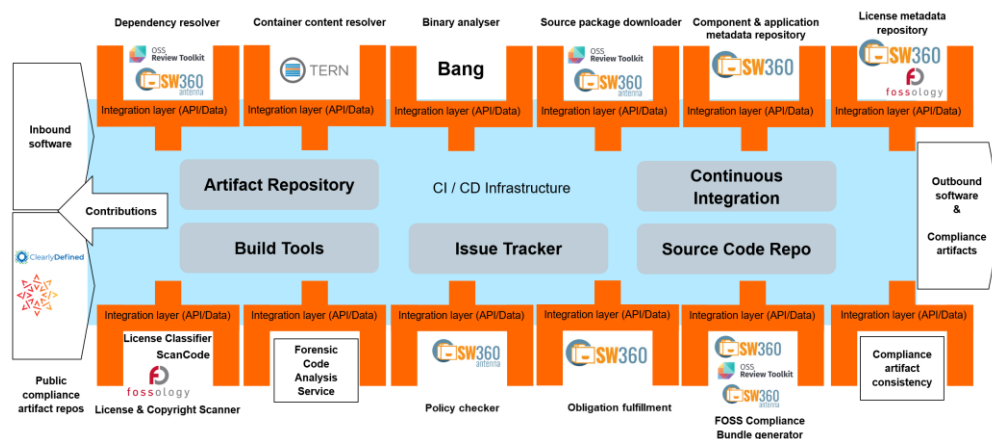
Join Us in Creating a New Era for Open Source Compliance

Mailing List: oss-based-compliance-tooling@groups.io

Subscription page: https://groups.io/g/oss-based-compliance-tooling

Online meetings: Bi-weekly - Invitations are sent to the mailing list

Website: https://oss-compliance-tooling.org/

And of course we are on GitHub:

https://github.com/Open-Source-Compliance/Sharing-creates-value

How to get involved

Github:

https://github.com/Open-Source-Compliance/Sharing-creates-value

Slack:

https://join.slack.com/t/ossbasedcompl-bhx9742/shared_invite/enQtODE2MTMxNzUyNDY1LWQyNWVlNzkyMjhhOWUyNDdjNDJlMzk0YzU0NDUwNzQ2YzY0Mzc1N2Y2NjhhZGEyN2JmNDE0ZTg2MTBjYmM3MWI3MWI

Mailing List:

Subscription page: https://groups.io/g/oss-based-compliance-tooling

Email address: oss-based-compliance-tooling@groups.io

Thank you