OPEN
COMPLIANCE
SUMMIT

Generating a SPDX SBOM for your code in GitLab CI using OSS Review Toolkit

Photo by Pankaj Patel on Unsplash

About me

opensource@steenbe.nl
@tsteenbe
linkedin.com/in/tsteenbe

Head of Open Source

- HERE Open Source Office (OSO) is a team of 7 people
- Supporting 9k+ employees in 56 countries on all things Open Source together with our legal counsels

Contributing to:

OSS Review Toolkit
http://oss-review-toolkit.org

SPDX

Open Source
Tooling Group

OPENCHAIN

ClearlyDefined

TODO
European
Chapter

Why: Open Source Compliance Program

- **Know your obligations.** You should have a process for identifying and tracking Open Source components that are present in your software

- **Satisfy license obligations.** Your process should be capable of handling Open Source license obligations that arise from your organization's business practices

Benefits of a robust Open Source Compliance program include:

- Increased understanding of the benefits of Open Source and how it impacts your organization
- Increased understanding of the costs and risks associated with using Open Source
- Increased knowledge of available Open Source solutions
- Reduction and management of infringement risk, increased respect of Open Source developers/owners' licensing choices
- Fostering relationships with the Open Source community and Open Source organizations

Source: OpenChain training slides: https://github.com/OpenChain-Project/curriculum

What information do you need to gather?

When analyzing Open Source usage, collect information about the identity of the Open Source component, its origin, and how the Open Source component will be used. This may include:

- Package name
- Status of the community around the package (activity, diverse membership, responsiveness)
- Version
- Download or source code URL
- Copyright owner
- License and License URL
- Attribution and other notices and URLs
- Description of modifications intended to be made
- List of dependencies
- Intended use in your product
- First product release that will include the package
- Location where the source code will be maintained
- Possible previous approvals in another context
- If from an external vendor:
- Development team's point of contact
- Copyright notices, attribution, source code for vendor modifications if needed to satisfy license obligations

Source: OpenChain training slides: https://github.com/OpenChain-Project/curriculum

Questions to ask when comparing SCA tools

- **How are SW components detected?** Most build tools are meant to build code and not to produce an SBOM. As a result, software composition analysis tools on the market generally do a best effort approach thus their SBOMs may differ.
- **Can the tool provide proof for a finding?** A lot of SCA tools show for example OSS licenses in the SBOM but won't show how they came to that license e.g. show the exact files and lines in the source code.
- **Can it do my policy or risk decisions?** Default is to have 1) allow/deny list for licenses and 2) 'ignore' buttons for vulnerabilities but satisfy obligations is not black & white and ignoring issues does not solve them.
- **Can we run in CI with acceptable compliance levels?** Run the tool cost effectively at scale and speed whilst maintaining the right levels of compliance and not overload users with false-positives.
- **Can we edit SBOM?** No tool is 100% correct as real world is ugly, being able to manually fix things is a must-have.

See also: https://linuxfoundation.org/resources/publications/an-open-guide-to-evaluating-software-composition-analysis-tools/

Lesson learned…

Various SCA tools are **capable of generating a SBOM**
however, **many use best effort approaches**
thereby likely not usable to
automate your organization’s
OSS policies in CI/CD

Tooling Challenges

Our experience when trying to introduce automation:

Missing / incorrect metadata

Source location may not be defined or found. Declared ≠ detected license

No sources available

Simply missing in central repositories

MISSING DATA

Ways of working issues

Devs do not always follow best engineering practices
resulting host issues when trying to automate

Build/dependency tools issues

Not designed to support FOSS reviews
e.g. lacking methods or return inaccurate data

Different build/dependency tools

~30 common build/dependency tools

Large volume of scan results

No tooling is available to automate reviewing large amounts of scan results,
conclude obligations and determine any issues to be resolved within limited timeframe

MISSING TOOLING

Solving challenges is what engineers do

Photo by Mimi Thian on Unsplash

OSS
Review Toolkit

Our approach:

- Created with and by the OSS/OSPO community for the community

  Vibrant community of users (mostly OSPOs) using and contributing. ORT maintainers participate in SPDX, OpenChain and the TODO group.
- Use build tools to get SBOM info, otherwise use static analysis

  Albeit slower it matches what developer would get.
- Multiple methods to fix SBOM information

  Fix license findings or project metadata such as source code repository/paths, artifacts URLs.
- Policy rules as code

  Perform highly customizable policy checks against scan results
- Adaptable to your build/review workflow

  Toolkit is implemented as set of libraries (for programmatic use) and exposed via a command line interface (for scripted use)
- Built on top of open standards

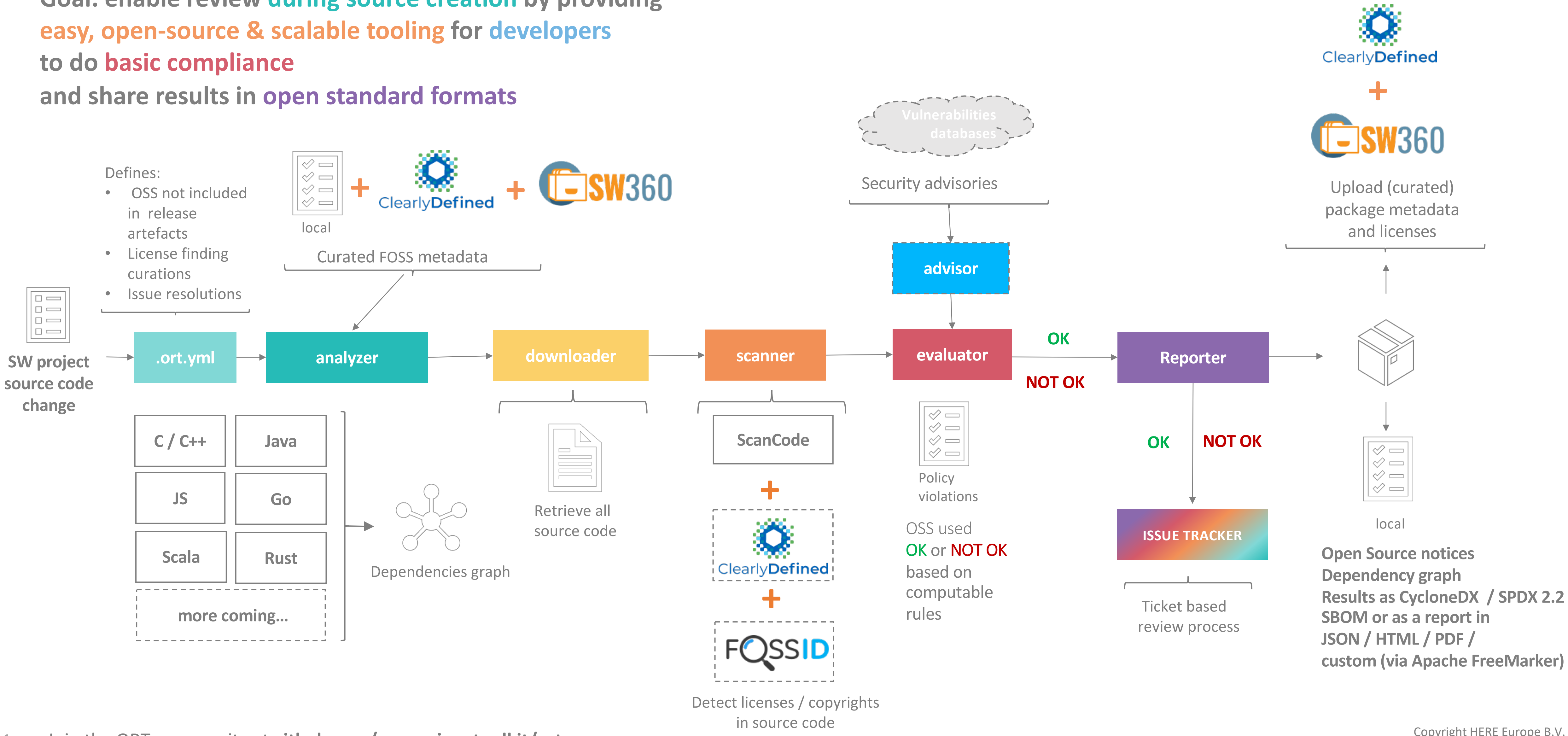
  Uses SPDX expressions for license handling and future proof by able to generate CycloneDX or SPDX

Collected Information

- Package name
- Version
- Source code repository URL
- Source and binary artifacts
- Copyright owner
- License and License URL
- Attribution and other notices and URLs
- List / tree of dependencies
- Location where the source code will be maintained

OSS Review Toolkit (Q4 2021)

Goal: enable review during source creation by providing
easy, open-source & scalable tooling for developers
to do basic compliance
and share results in open standard formats



SW project source code change

Defines:
- OSS not included in release artefacts
- License finding curations
- Issue resolutions

.ort.yml

local + ClearlyDefined + SW360

Curated FOSS metadata

analyzer

C / C++ | Java | JS | Go | Scala | Rust | more coming...

Dependencies graph

downloader

Retrieve all source code

scanner

ScanCode + ClearlyDefined + FOSSID

Detect licenses / copyrights in source code

Vulnerabilities databases

Security advisories

advisor

evaluator

Policy violations

OSS used OK or NOT OK based on computable rules

OK

NOT OK

Reporter

OK

NOT OK

ISSUE TRACKER

Ticket based review process

ClearlyDefined + SW360

Upload (curated) package metadata and licenses

local

Open Source notices
Dependency graph
Results as CycloneDX / SPDX 2.2 SBOM or as a report in JSON / HTML / PDF / custom (via Apache FreeMarker)

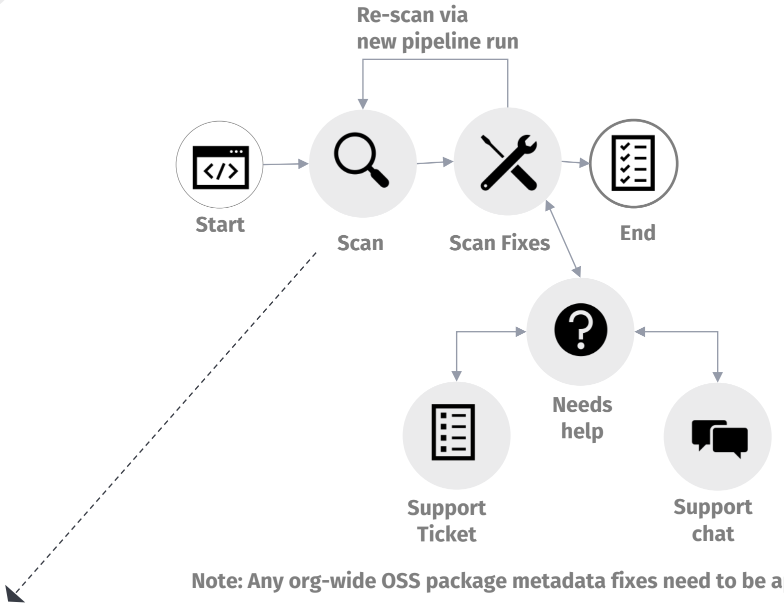
Join the ORT community at **github.com/oss-review-toolkit/ort**

OSS
Review Toolkit

Hands-on Demo

Generating a SPDX SBOM in GitLab

OSS Scan in GitHub/GitLab CI (continuous)



Re-scan via new pipeline run

Start

Scan

Scan Fixes

End

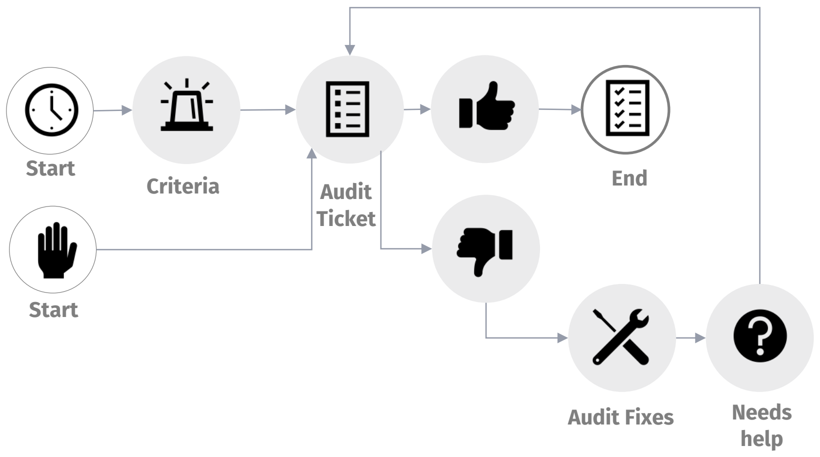
Needs help

Support Ticket

Support chat

Note: Any org-wide OSS package metadata fixes need to be approved by OSO

OSS Audit Process (on-demand / scheduled)



Start

Criteria

Audit Ticket

End

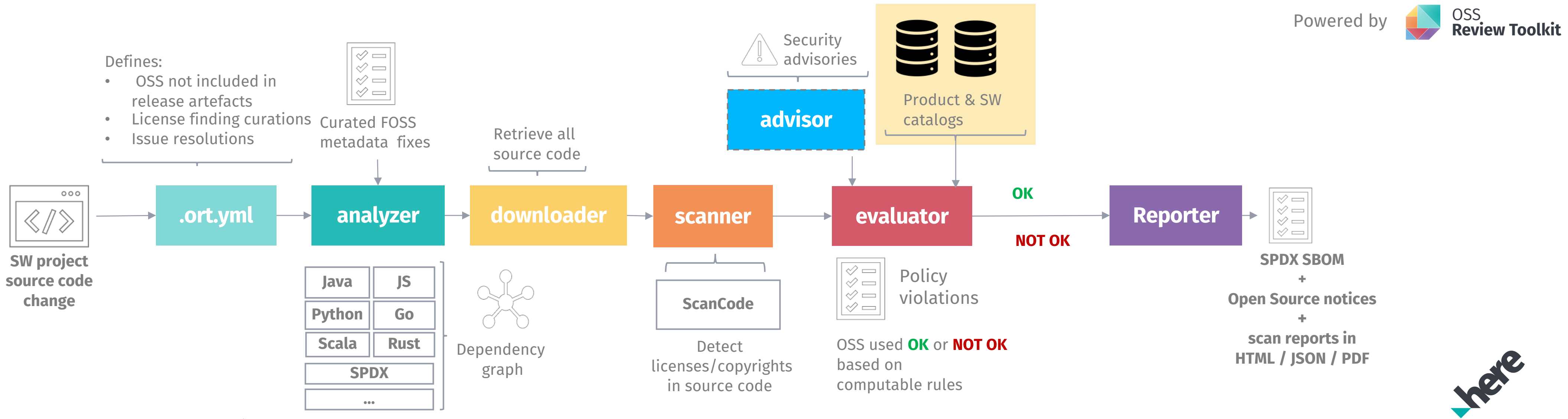
Start

Audit Fixes

Needs help

Note: An OSS audit ticket assigned to developer team will be included in CI OSS scan report

Scan



Powered by OSS Review Toolkit

Defines:

- OSS not included in release artefacts
- License finding curations
- Issue resolutions

Curated FOSS metadata fixes

Retrieve all source code

Security advisories

Product & SW catalogs

SW project source code change

.ort.yml

analyzer

downloader

scanner

advisor

evaluator

OK

NOT OK

Reporter

Java	JS
Python	Go
Scala	Rust
SPDX	
...	

Dependency graph

ScanCode

Detect licenses/copyrights in source code

Policy violations

OSS used OK or NOT OK based on computable rules

SPDX SBOM
+
Open Source notices
+
scan reports in HTML / JSON / PDF

OK / NOT OK = code context + license context + product context + security context

Thank you

Questions? Interested in collaborating? Chat with us using the ort-talk Slack channel, see **github.com/oss-review-toolkit/ort**

OSS Review Toolkit

https://github.com/oss-review-toolkit/ort

Starting and Scaling an Open Source Office

ORT Slack

Related OSS Projects

https://oss-compliance-tooling.org

https://clearlydefined.io

https://spdx.org

https://www.openchainproject.org

https://www.doubleopen.org

https://www.eclipse.org/sw360

Thomas Steenbergen

✉ opensource@steenbe.nl

@tsteenbe

linkedin.com/in/tsteenbe