Sharing and reusing SBOMs with

OSSelot

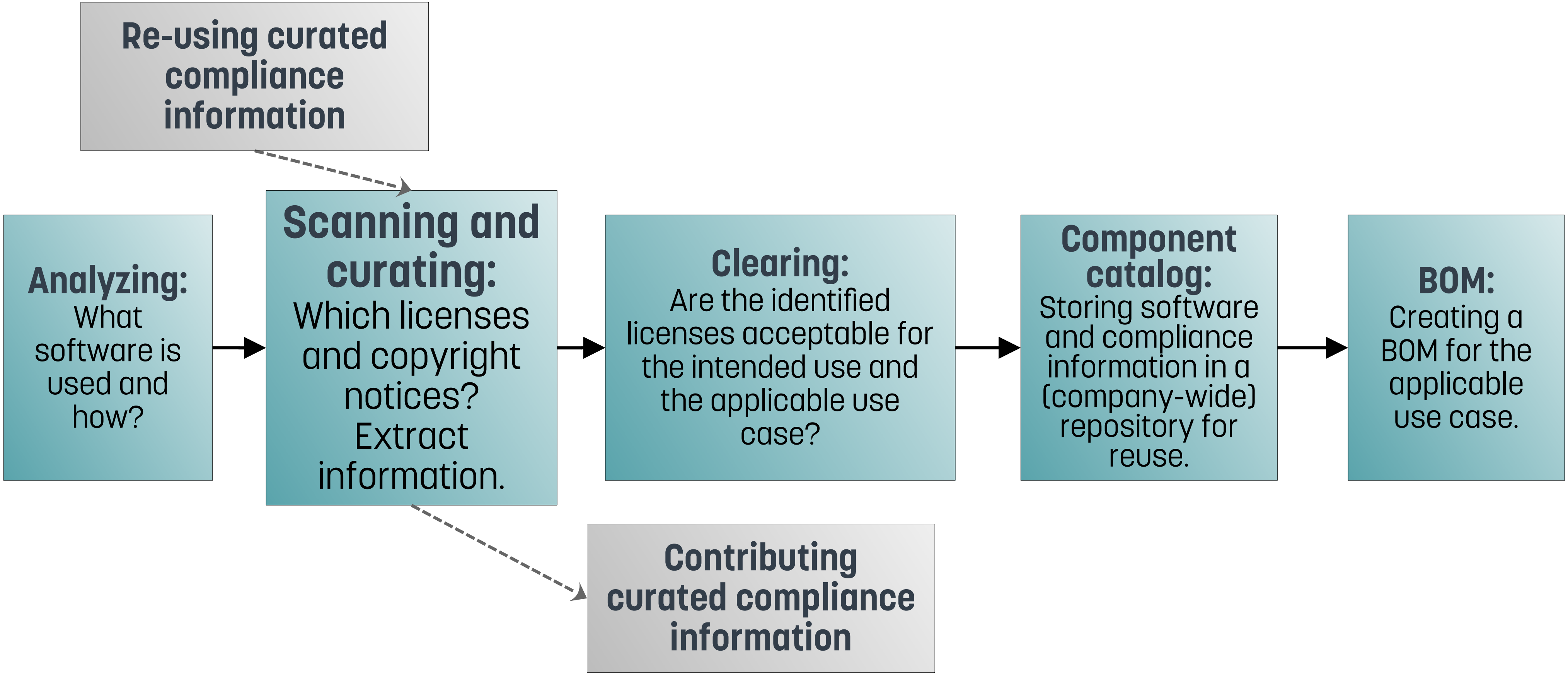
THE OPEN SOURCE CURATION DATABASE

Caren Kresse
Open Source Automation Development Lab (OSADL) eG

Why re-using compliance data?

- The success of FOSS is partly due to the fact that development resources and efforts are reduced by **re-using existing software components**.
- Some of the reduction is canceled out by the effort required to scan and clear FOSS for **license compliant** use.
- There are many FOSS components that are deployed unmodified by a large number of users.
- Why not **share compliance tasks** in the same way as the software development?

Re-using compliance information



Re-using curated compliance information

Analyzing: What software is used and how?

Scanning and curating: Which licenses and copyright notices? Extract information.

Clearing: Are the identified licenses acceptable for the intended use and the applicable use case?

Component catalog: Storing software and compliance information in a (company-wide) repository for reuse.

BOM: Creating a BOM for the applicable use case.

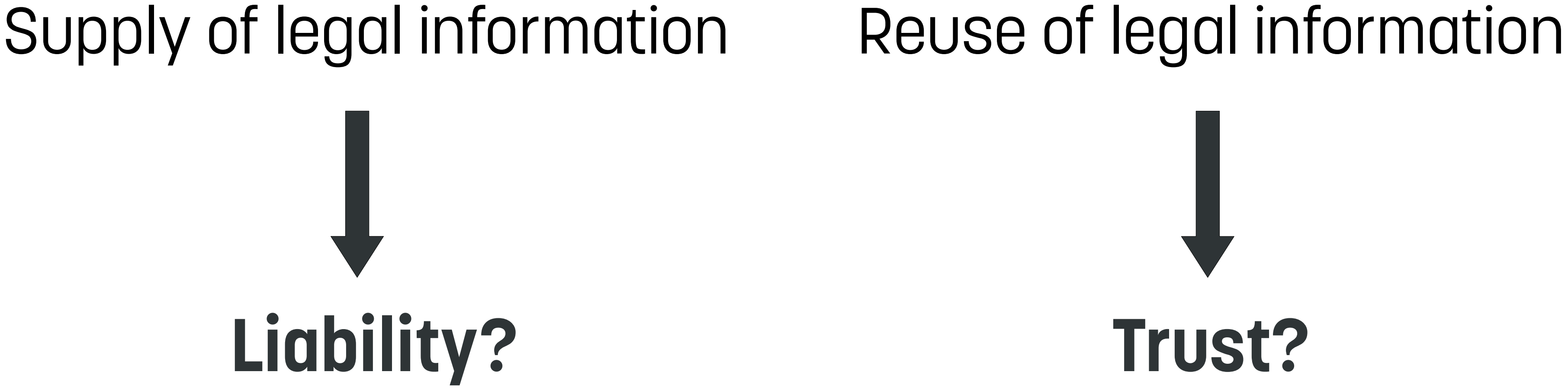
Contributing curated compliance information

The OSADL project:
OSSelot The Open Source Curation Database

https://www.osselot.org
https://github.com/Open-Source-Compliance/package-analysis

- Contains **license and copyright analysis results** for various packages (approx. **320 (200 unique) packages** and 1.5 Mio. files):
  - **Readme with metadata** of the package, *e.g.* download location, purl, reviews, comments.
  - **SPDX** Tag:Value, JSON, RDF-XML and YAML files with concluded licenses, copyright notices and comments on decisions per file.
  - **Disclosure document** with aggregated license texts, copyright notices and acknowledgments.

Liability and trust



Supply of legal information

Liability?

Reuse of legal information

Trust?

Liability and trust

Limiting liability

- Licensing to give **maximal rights** (*e.g.* CC0-1.0). Then, **gift regulations** apply and liability applies only for willful intent and gross negligence.
- There are **no known lawsuits** pursuant to supply of incorrect legal information for FOSS so that reservations have generally lessened.

Establishing trust

- Data are curated **conscientiously** and **diligently**.
- Compliance artifacts are created by named **persons with expertise**.
- There is a **transparent review process**.
- An organization stands with its **reputation** for the quality of the project.

OSSelot curation guidelines: General

- Scanning and curating currently with **Fossology** with integrated Scancode.
- Download of source code **as upstream as possible**, *e.g.* directly from the project page.
- **Curating** license findings and copyright findings **manually**.

OSSelot curation guidelines: Copyrights

- Findings that were **incorrectly identified** as a copyright statement (*e.g.* license texts, code, *etc.*) are removed.
- Content that is not part of the copyright notice (*e.g.* formatting signs, license notices, comments on content, code, *etc.*) is removed from the copyright notice.
- If the source code tree contains an **AUTHORS** file, the content of this is given as value to the SPDX tag "PackageCopyrightText".

OSSelot curation guidelines: Licenses

- Review is done on **file level**, *i.e.* every file in the source code tree for which at least one scanner found a result is analyzed and **confirmed or corrected.**
- **Acknowledgments and individual license texts** are added, if applicable.
- Only if there is a LICENSE or COPYING or similar file in the root directory that states a **main license for the package**, this information is given as value to the SPDX tag "PackageLicenseDeclared".
- In case a license conclusion is not obvious, **the decision is explained** in the "Comments" section which maps to the SPDX tag **"LicenseComments"**.

License comments

- License comments are made with the following heuristic:

```
The information in the file is:
"[Quote licensing information in the source code file]"
[Give reason for conclusion] Therefore, [license] is concluded.
```

License comments

- License comments are made with the following heuristic:

```
The information in the file is:
"[Quote licensing information in the source code file]"
[Give reason for conclusion] Therefore, [license] is concluded.
```

- **Example 1: No version of the license is given**

```
The information in the file is:
"This file is GPL'd."
As no version of the GPL is given, GPL-1.0-or-later is concluded.
```

License comments

- License comments are made with the following heuristic:

```
The information in the file is:
"[Quote licensing information in the source code file]"
[Give reason for conclusion] Therefore, [license] is concluded.
```

- **Example 2: URL for license text**

```
The information in the file is:
"This file is licensed under License A. You can find the license
text at https://www.LicenseTextOfLicenseA.com."
The URL contains the license text of License A, therefore License
A is concluded. The information was retrieved on [date].
```

Common corrections of scanner findings (1)

- **Not a license:** The scanner concludes a license from an expression in a file that is not actually a license expression at all.
- **Not the file's license:** The scanner concludes a license from a license expression that is part of the file’s content but not the license of the file itself.
- **License text:** Files that contain only a license text (e.g. COPYING) are concluded by the scanners to be licensed under the respective license. This is usually not correct. Most license texts are not explicitly licensed. The GNU licenses contain a license statement for the license text itself which is concluded for these cases.

Common corrections of scanner findings (2)

- **Generic license texts:** If an individual text differs from the generic text, the individual license text is provided (especially BSD-type licenses).
- **Imprecise finding:** The scanner finding might be imprecise, *e.g. w.r.t.* to the version of a license. This is concluded *e.g.* as v1.0-or-later.
- **Dual licensing:** If the context requires to chose one specific license, this choice must be noted. In any case, all applicable licenses must be concluded.
- **License exceptions:** Fossology notes the license and the exception as separate findings. This is corrected to one finding using the SPDX license expression [License] WITH [exception], *e.g.* GPL-2.0-or-later WITH GCC-exception-2.0.

Common corrections of scanner findings (3)

- **External references:** If a file does not contain the name or text of a license but references an external resource, such as a COPYRIGHT file in the root directory or a URL, the external reference is checked, the detected license is concluded and the process is documented as a LicenseComment (in case of a URL, the date of access is noted).
- **(Partially) global license assignment:** License assignments from a README file or similar to several files within the source tree (*e.g.* all files in a specific directory) are not generally used to conclude licenses as such information is often outdated or does not account for individual licensing.

SPDX in OSSelot

- Currently on SPDX 2.2; Upgrade is under way.
- OSSelot SPDX reports serve as **stand-alone documents** to fulfill information obligations, *i.e.* **all license texts are included** and not only referenced.

SPDX tag:value report – Creation information

```
##---------------------------
## Creation Information
##---------------------------

Creator: Tool: spdx2
Creator: Person: Oliver Fendt
CreatorComment: <text>
This document was created using license information and a generator
from Fossology.
It contains the license and copyright analysis of OpenSSL 3.0.5
Please check "LicenseComments" for explanations of concluded licenses
</text>
Created: 2022-07-06T14:58:22Z
LicenseListVersion: 2.6
```

Curator’s name

SPDX tag:value report – Package information

```
##-------------------------
## Package Information
##-------------------------

PackageName: openssl-openssl-3.0.5.tar.gz
[...]
PackageChecksum: SHA1: edc3465a8a43ce580268e726b6f7b827f4a6261e
PackageChecksum: SHA256:
b6363cf1bca88f0a46a768883a225e644135432d6a51ab1c4660ab58af541078
PackageChecksum: MD5: 22733b9187548b735201fd9f7aa12e71
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: LicenseRef-Apache-2.0
PackageLicenseComments: <text> Additional information found in: [File name of third
party notices]
[Third party notices]
[...]</text>
[...]
```

Only if there is a main license for the package (LICENSE/COPYING file in root directory)

If the package contains any meta information on licensing, *e.g.* third party notices.

SPDX tag:value report – File information

```
##File

FileName: openssl-3.0.5.tar.gz/openssl-3.0.5.tar/openssl-openssl-3.0.5/crypto/LPdir_wince.c
SPDXID: SPDXRef-item158856105
FileChecksum: SHA1: dc3e4bb9f2cf76426da9ad5dbc8ad4a2356c3359
FileChecksum: SHA256: fd878a5b569cd41d63ba673420a4d95adfac9ad3048ea0fb4854504ba55572d8
FileChecksum: MD5: 62fbea2db5fb486d537d869af119135b
LicenseConcluded: LicenseRef-Apache-2.0 OR LicenseRef-BSD-2-Clause-3185f2587757a9c63eaa83143f7c0386
LicenseComments: <text>The information in the file is:
Besides the Apache-2.0 header the following information is in the file:
 This file is dual-licensed and is also available under the following terms:
Followed by the BSD-2-clause license text. Thus dual licensing was concluded. </text>
LicenseInfoInFile: LicenseRef-Apache-2.0
LicenseInfoInFile: LicenseRef-OpenSSL
LicenseInfoInFile: LicenseRef-Dual-license
LicenseInfoInFile: LicenseRef-BSD-2-Clause_REGENTS-AND-CONTRIBUTORS
FileCopyrightText: <text> Copyright 2004-2016 The OpenSSL Project Authors.
Copyright (c) 2004, Richard Levitte <richard@levitte.org></text>
```

Final license decision

If not obvious, explanation of license decision

Scanner findings

SPDX tag:value report – License information

```
##-------------------------
## License Information
##-------------------------

LicenseID: LicenseRef-Apache-2.0
LicenseName: Apache License 2.0
ExtractedText: <text> Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

[...]
```

Also include template licenses to make the SPDX report stand-alone.

Add “LicenseRef-” prefix for SPDX validity.

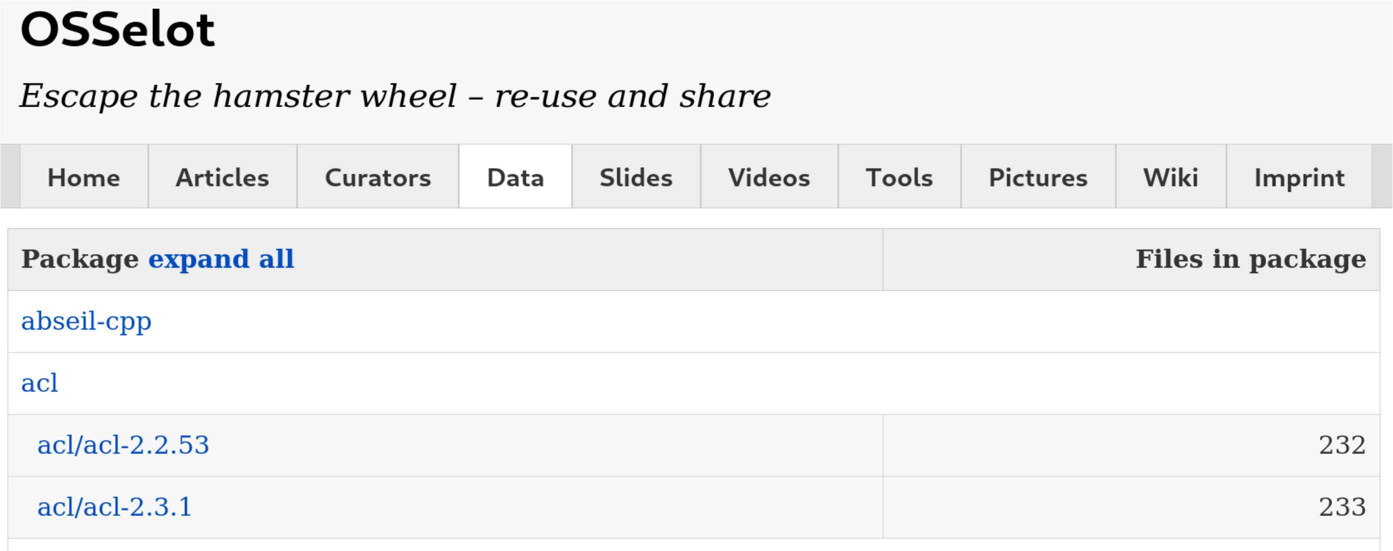
SPDX tag:value report – License information

```
LicenseID: LicenseRef-BSD-2-Clause-3185f2587757a9c63eaa83143f7c0386
LicenseName: BSD-2-Clause-3185f2587757a9c63eaa83143f7c0386
ExtractedText: <text> Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. </text>

[...]
```

How to reuse OSSelot data

Use case 1: Package is available in the required version

1. Go to www.osselot.org → Data and check whether the package is available in the required version.

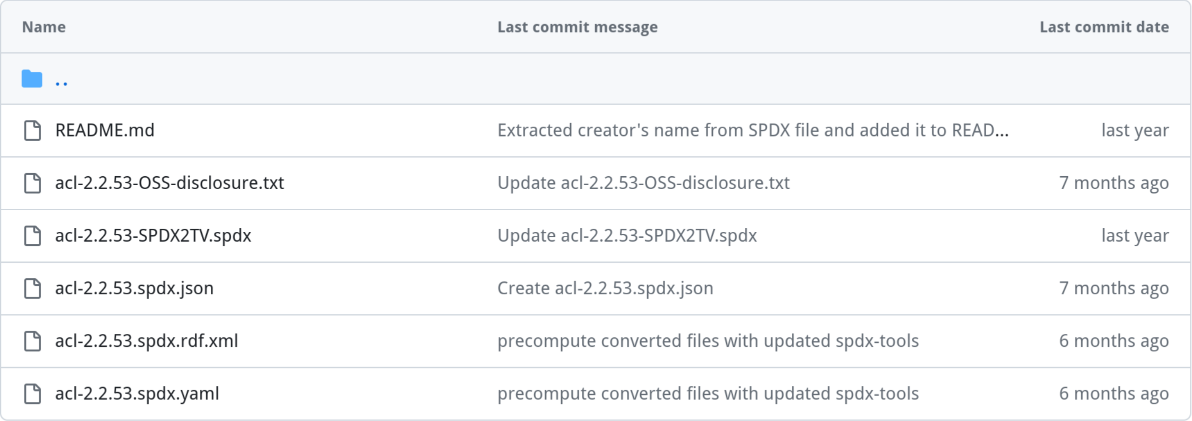


Package **expand all**	Files in package
abseil-cpp	
acl	
acl/acl-2.2.53	232
acl/acl-2.3.1	233

How to reuse OSSelot data

Use case 1: Package is available in the required version

1. Go to www.osselot.org → Data and check whether the package is available in the required version.
2. Clicking on the required version redirects to the OSSelot repository where the data can be downloaded.



Name	Last commit message	Last commit date
..		
README.md	Extracted creator's name from SPDX file and added it to READ...	last year
acl-2.2.53-OSS-disclosure.txt	Update acl-2.2.53-OSS-disclosure.txt	7 months ago
acl-2.2.53-SPDX2TV.spdx	Update acl-2.2.53-SPDX2TV.spdx	last year
acl-2.2.53.spdx.json	Create acl-2.2.53.spdx.json	7 months ago
acl-2.2.53.spdx.rdf.xml	precompute converted files with updated spdx-tools	6 months ago
acl-2.2.53.spdx.yaml	precompute converted files with updated spdx-tools	6 months ago

How to reuse OSSelot data

Use case 2: Package is available in a different version

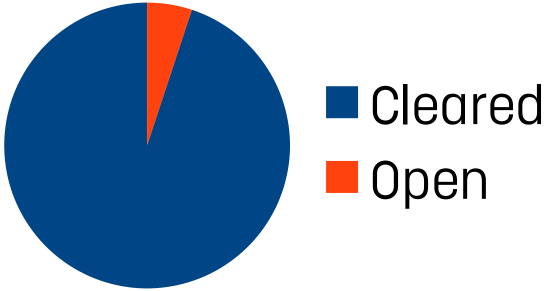
1. Find the **closest version** of the required package that is available in OSSelot and **upload the source code** into Fossology without running any scans.
2. Download the **curated RDF SPDX report** from the OSSelot repository and import into the source code upload in Fossology (**"Import Report"**) → This automatically clears the package.
3. Upload the source code of the required version into FOSSology, running license scanners and selecting **"Reuse"** referencing the already cleared package.
4. Manually clear the remaining files.

How to reuse OSSelot data

Use case 2: Package is available in a different version

An example: Clearing bash v5.2.21

1. The closest available version in OSSelot is v5.2.15.
2. Importing the OSSelot SPDX file clears 993 files immediately.
3. The scanners find 1042 files with license information in v5.2.21. Reusing v5.2.15 clears 95 % of files automatically.



4. The remaining 53 Files have to be cleared manually.

How to reuse OSSelot data

Use case 3: Automated use of curated material

1. Create a list of required packages and their versions in the form pkg/ver.
2. Use the OSSelot REST API:

```
https://rest.osselot.org/<format>/<package>/<version>
```

to download curated RDF SPDX reports of available packages of required versions and list alternative versions for available packages:

```
#!/bin/bash
for i in $(cat list); do
 pkg=$(echo $i | cut -d/ -f1)
 if ! wget -O $i.rdf https://rest.osselot.org/xml/$i; then
  rm -f $i.rdf
  wget -qO - https://www.osselot.org/curated.php?$pkg | grep -v "<" \
                                             >>version-mismatch.list
 fi
```

How to reuse OSSelot data

Use case 3: Automated use of curated material

An OSSelot layer for OpenEmbedded

- Watch the "Tools" section of www.osselot.org for the meta-osselot project, a layer for OpenEmbedded and Yocto to **automatically integrate and manage** OSSelot curation data.

Conclusion

- The OSSelot project provides a **publicly available database** with **curated compliance information** licensed under CC0-1.0 for frequently used FOSS components.
- The database contains **stand-alone SPDX reports** and disclosure documents.
- Some manual input and review is still indispensable, but:

The reuse of curated licensing and copyright information can tremendously reduce the time required to clear a software package!

- https://www.osselot.org is the official project page with news, explanations, example use cases, tools and to dos.
- https://wiki.osselot.org contains Tools for automated use of the curation data.
- The curated material is available via GitHub https://github.com/Open-Source-Compliance/package-analysis
- Contact: info@osselot.org

Questions, requests and contributions are very welcome!