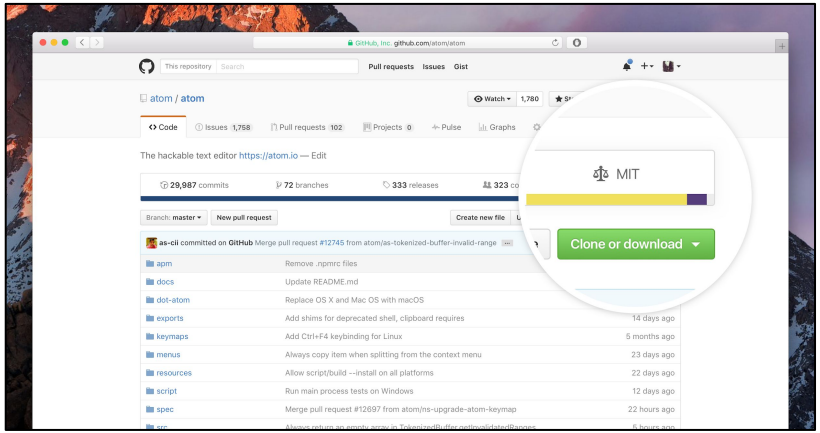
2018 SPDX Ecosystem Update

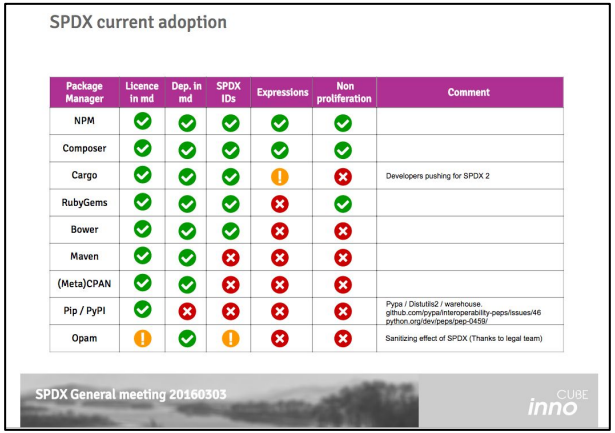
Kate Stewart, (@_kate_stewart)
Sr. Director of Strategic Programs, The Linux Foundation

April 5, 2018

SPDX License List Identifiers:

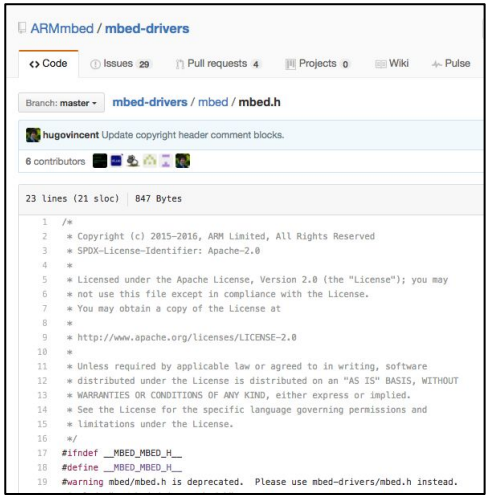
- Debian recognized since DEP5, Fedora considering transitioning.
- Linux Foundation & Eclipse projects transitioning.
- New project in Package Manager Repositories adopting
- Github adopted for projects in September 2016 (see Licenses API)!





SPDX License Identifiers in Source Files

- Developer initiated: in U-Boot in 2013 for efficiency and to help with automatic processing.
- Selective upstream projects adopt based on developer preferences.
- Ecosystem Adopting: Linux Foundation projects started adding to Linux in 2016, Eclipse in 2017, FreeBSD in 2017, REUSE.software guidelines
- "Open Government Partnership" created a best practices template for Open Source Policy that includes SPDX-License-Identifiers in December, France adopting "as is".



Examples:

```
Original software

<AUTHORS>,

Copyright © 2013-2015 ENTITY_NAME

SPDX-License-Identifier: GPL-3.0

v1.0
```

Why create an SPDX document?

Can be very descriptive

- Provides an accurate Bill of Materials (BOM) also known as a manifest of the software contents (copyrights, licensing, dependencies, provenance, CPE, …)

Can be easily translated to human readable forms

- Spreadsheet, grep’able text file (tag:value), viewers

Share data between open source and commercial tools

HELPS the downstream user understand and comply with the upstream developer’s licensing expectations!

SPDX

What is an SPDX Document?

tag:value

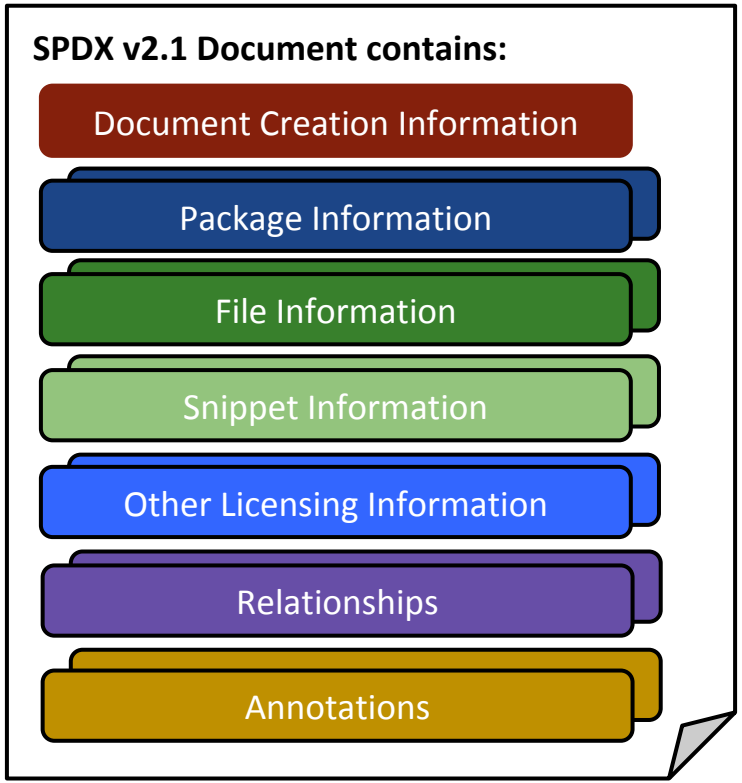
```
##-------------------------
## Package Information
##-------------------------

PackageName: time-1.7.tar.gz
PackageFileName: time-1.7.tar.gz
PackageDownloadLocation: NOASSERTION
PackageVerificationCode: dd5cf0b17bfef4284c6c22471b277de7beac407c
PackageChecksum: SHA1: dde0c28c7426960736933f3e763320680356cc6a
PackageLicenseConcluded: GPL-2.0+
PackageLicenseDeclared: GPL-2.0+
PackageLicenseInfoFromFiles: GPL-2.0
PackageLicenseInfoFromFiles: GPL-2.0+
PackageLicenseInfoFromFiles: MIT
PackageLicenseInfoFromFiles: LicenseRef-1
PackageLicenseInfoFromFiles: LicenseRef-2
PackageLicenseInfoFromFiles: LicenseRef-3
PackageCopyrightText: NOASSERTION

##-------------------------
```

RDF/XML

```
dolph Chung &lt;tausq@debian.org&gt;

://spdx.org/licenses/GPL-2.0+"/>
p://spdx.org/rdf/terms#noassertion"/>
g/rdf/terms#fileType_source"/>

eName>
g/rdf/terms#File"/>

- <rdf:Description rdf:nodeID="A2">
    <checksumValue>dc90a437e03f31ab04e7059d8da5f88b28cde77d</checksumValue>
    <algorithm rdf:resource="http://spdx.org/rdf/terms#checksumAlgorithm_sha1"/>
    <rdf:type rdf:resource="http://spdx.org/rdf/terms#Checksum"/>
  </rdf:Description>
- <rdf:Description rdf:nodeID="A3">
    <rdfs:comment/>
    <copyrightText rdf:resource="http://spdx.org/rdf/terms#none"/>
    <licenseComments/>
```

SPDX

What makes up an SPDX Document?



SPDX

Only subset of fields are mandatory

Document Creation Information

1 per document

- 2.1 SPDX Version.
- 2.2 Data License
- 2.3 SPDX Identifier
- 2.4 Document Name
- 2.5 SPDX Document Namespace
- 2.8 Creator
- 2.9 Created

1 per package in document

Package Information

- 3.1 Package Name
- 3.2 Package SPDX Identifier
- 3.7 Package Download Location
- 3.9 Package Verification Code
- 3.13 Concluded License
- 3.14 All Licenses Information from Files
- 3.15 Declared License
- 3.17 Copyright Text

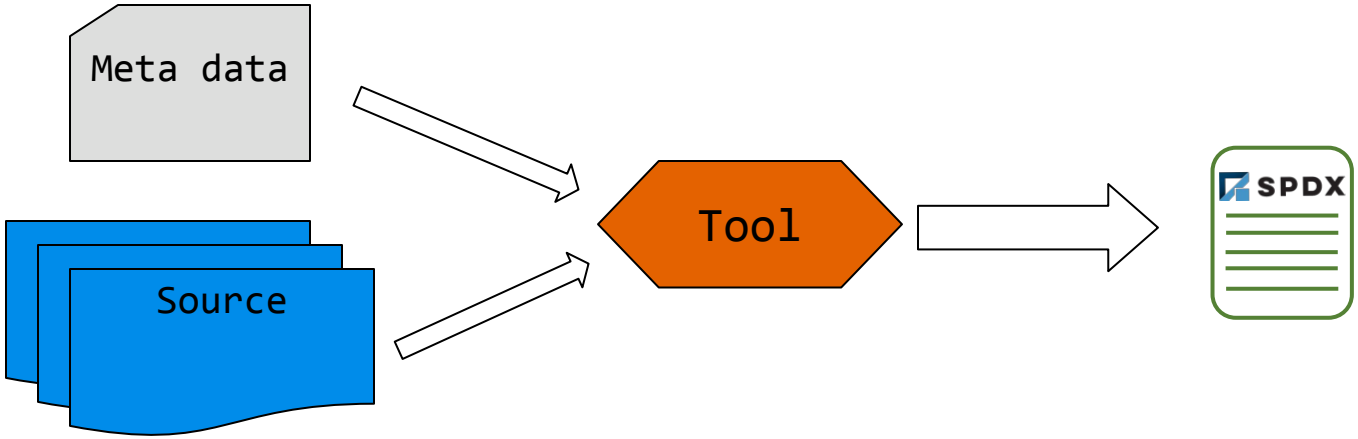
File Information

- 4.1 File Name
- 4.2 File SPDX Identifier
- 4.4 File Checksum
- 4.5 Concluded License
- 4.6 License Information in File
- 4.8 Copyright Text

1 per file in each package

SPDX

Creating an SPDX document



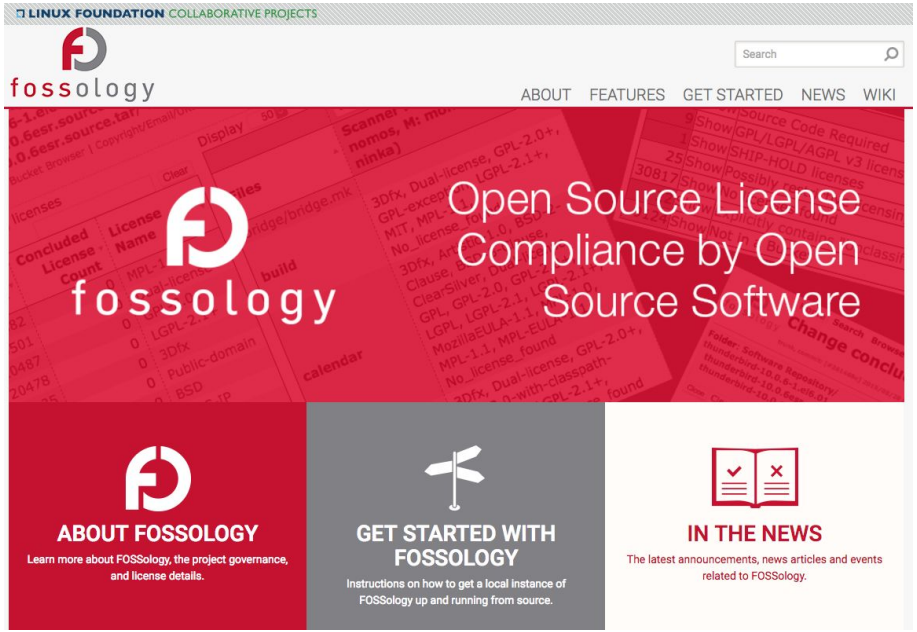
Requires tooling as the package verification code is generated by file **checksums** and the total **number of files** per useful package

SPDX

Open Source supporting SPDX Documents

- **SPDXTools:** Verify, Translate, Plugins, Summarizers See: https://github.com/spdx/tools
- **FOSSology:** FOSSology is a open source license compliance software system and toolkit. As a toolkit you can run license, copyright and export control scans from the command line. As a system, a database and web ui are provided to give you a compliance workflow. License, copyright and export scanners are tools available to help with your compliance activities.
  - Install instructions can be found in: https://wiki.fossology.org/handson
- **ScanCode:** ScanCode is a tool to scan code and detect licenses, copyrights, packages metadata & dependencies and more... to find, discover, inventory open source and third-party components used in your code.
  - Install instructions can be found at: https://github.com/nexB/scancode-toolkit

www.fossology.org



3.2 release generates and imports both SPDX tag:value & SPDX RDF documents.

ScanCode

Is able to generate SPDX documents!

In the current code** (2.9b1) you can use the options:

--output-spdx-rdf FILE Write scan output as SPDX RDF to FILE.

--output-spdx-tv FILE Write scan output as SPDX Tag/Value to FILE.

Both options can be used, and create both files from a single scan.

** 2.2.1 had the capability using --format spdx-rdf and --format spdx-tv

Commercial Tools supporting SPDX

- **Wind River:**
  - SPDX is standard output format for Software Manifests
- **Protecode:**
  - Creates SPDX document http://www.protecode.com/license-compliance-is-evolving-with-spdx/
- **Source Auditor:**
  - Creates SPDX documents http://sourceauditor.com/blog/source-auditor-supports-spdx-2/
- **TripleCheck:**
  - Creates SPDX documents http://triplecheck.net/what-we-do.html
- **WhiteSource:**
  - Uses SPDX license list http://docs.whitesourcesoftware.com
- **Black Duck:**
  - Uses SPDX license list https://www.blackducksoftware.com/products/spdx next version of Hub will support 2.1 documents.

SPDX

So how accurate is any scanning tool?

All are based on heuristics at some level

- Interprets “clues” from inside the source code
- Not all licenses have standard headers
- Standard headers get mangled routinely
- Tools are run much later than development
- Human review is still needed for confidence today

Significant projects have 1000’s of files

- Not fun for the human even with tool scanning.

⇒ Reviewer needs to be able to know tools are accurately reporting what is in the code!

Debian Detectable File Level Licensing

Files Coverage	Scanner #1	Scanner #2
80 - 100%	2038 packages (9%)	1970 packages (9%)
60 - 79%	3636 packages (17%)	3713 packages (17%)
40 - 59 %	3571 packages (17%)	3703 packages (17%)
20 - 39%	4357 packages (20%)	4303 packages (20%)
0 - 19%	7223 packages (34%)	7130 packages (34%)

Source: **Jessie** analysis by Thomas Gleixner, Philippe Ombredanne - Q12017

SPDX

Sample Projects from Debian Jessie

Project	#licenses detected	#relevant files of #total files	Scanner #1		Scanner #2	
			#licenses in relevant	#licenses in total	#licenses in relevant	#licenses in total
linux-3.16.39	80	**45,420** of 51,312	31,447 (69%)	31,617 (61%)	29,684 (65%)	29,922 (58%)
u-boot-2014.10	33	**8,574** of 10,764	7,108 (82%)	7,139 (66%)	7,028 (81%)	7,063 (65%)
git-2.1.4	21	**1,909** of 2,835	132 (6%)	141 (4%)	164 (8%)	180 (6%)

SPDX

Source: Thomas Gleixner, Philippe Ombredanne - Q12017

Test Scanning Tools to build Trust

Recommendation:

1) Check SPDX documents generated are correct format
2) Run a test suite derived from SPDX license list.
3) Run example projects and compare results with a curated and verified set of output files with.

How to: Test SPDX documents correct

https://github.com/spdx/tools

Use “Verify” function to confirm correct SPDX document

```
java -jar spdx-tools-jar-with-dependencies.jar Verify TestFiles/SPDXRdfExample.rdf
```

Other useful functions available: format converters, viewers, compare utilities… :-)

SPDX

Comparing SPDX documents

https://github.com/spdx/license-test-scans

- SPDX Tag-value Python diff tool - bin/spdxdiff.py
- Creates a CSV with diffs

https://github.com/spdx/tools

- SPDX Java Tools
- can diff docs and produce a spreadsheet of differences

SPDX

How to: Use License Test Files

https://github.com/spdx/license-test-files

- Version 1.0 of test suite generated using SPDX License List 2.6 JSON data files.
- Contains headers, full license text, SPDX short identifiers to match.
- Contributed by Jack Manbeck. **Thank you very much Jack!** :-D

- Download tarball.
- Unpack and run source files through your scanner.
- Check results are as you expect.

SPDX

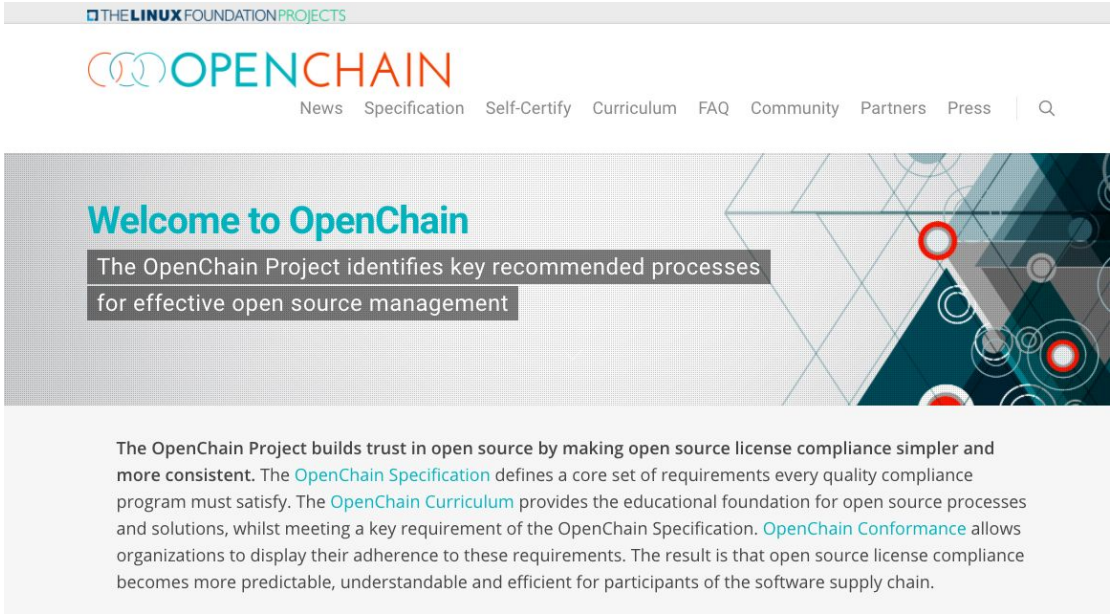
Open Source “Inside Organization” Workflow Tools

SPDX specification is used to structure internal **databases** in large companies (Samsung, TI, ARM, Intel, Siemens …) but we need open source tooling to share and access it.

- **SPDX online tools:** Utilities to validate and compare SPDX files. See: http://13.57.134.254/app/
- **SW360:** SW360 is an software component catalog application to maintain your projects / products and the software components within. It can send files to the open source license scanner **FOSSology** for checking the license conditions and maintain license information. See: https://github.com/sw360/
- **ORT:** suite of tools to assist with reviewing Open Source Software dependencies in your software deliverables. See: https://github.com/heremaps/oss-review-toolkit
- **More Open Source Tools emerging:** Quartermaster, ...

“Between Organization” Workflows

OpenChain project documents the processes to build trust between members of a software supply chain using open source software.



First Steps

1) Review the OpenChain specification at https://www.openchainproject.org/spec
2) Implement and document processes to meet the spec requirements. Use the curriculum slides as an easy starting point for training - https://www.openchainproject.org/curriculum
3) Certify conformance with the OpenChain specification at https://www.openchainproject.org/conformance
4) Get involved and help improve the project, see https://www.openchainproject.org/community

“Between Organization” Workflows

REUSE - developed by FSFE describes best practices for describing licensing information in open source software and making it suitable for automation.

See: https://reuse.software/

Practices: https://reuse.software/practices/2.0/

Overview: https://reuse.software/reuse/reuse-presentation.pdf

REUSE
SOFTWARE

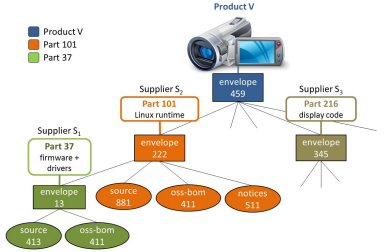
Best practices for license information in ways not only humans can read, but *computers* as well. Machine readable copyright and license information, simply put!

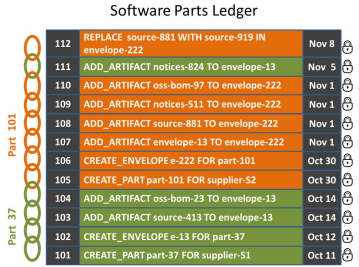
Emerging “Between Organization”

Software Parts Ledger - utilizes Blockchain to manage open source across the supply chain. Utililzes Hyperledger Sawtooth Platform & SPDX based BOM to conform to OpenChain best practices.

See: https://github.com/Wind-River/sparts

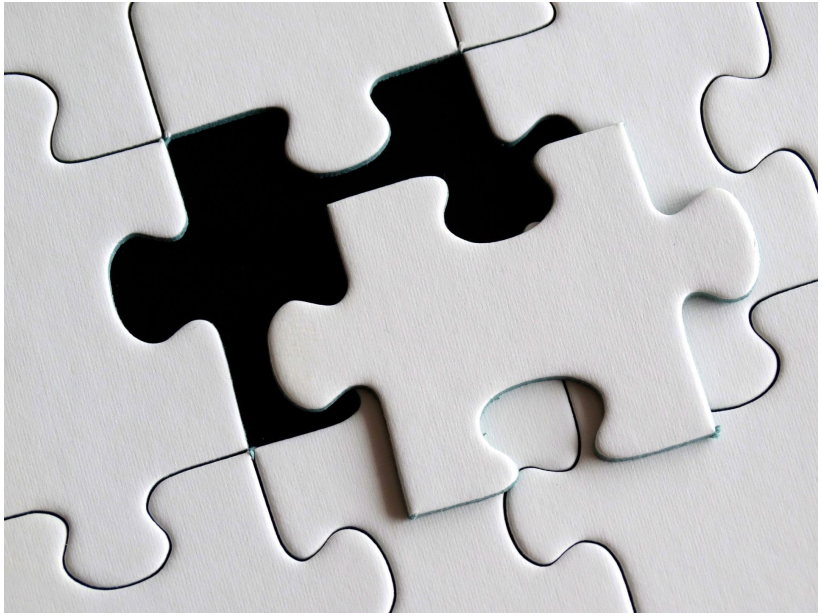
Accepted 2018/3 into Hyperledger Labs - https://github.com/hyperledger-labs/hyperledger-labs.github.io/blob/master/labs/SParts.md





Missing Pieces...

- Curated Real World Example Projects documented with SPDX
- End-to-End Reference Case Studies using SPDX Documents as Software BOMs
- CI/CD Build Tool Integration
- ?



Source: https://pixabay.com/en/puzzle-last-particles-piece-654956/

Want to help?

If you work with teams producing source code:

- Ask them to add SPDX license identifiers in the comments for each file.
- Follow the guidelines from REUSE.software on placement of information in projects to support automation.
- Generate SPDX documents for the projects they contribute to.

If you work with teams reviewing use and distribution of free/libre open source code:

- Ask them to contribute any curated SPDX documents they've generated to SPDX Outreach Team

kstewart@linuxfoundation.org