Piloting SPDX in Samsung: Case Studies and Experiences

김영택  Young Taek Kim
ytaek.kim@samsung.com
Open Source Initiative. AA Lab. SE Team
1. SPDX Introduction

2. Piloting SPDX
   1. Motivation in open source development process
      1. Open source development process
      2. Verification details and problem
      3. Release detail and problem
   2. Solutions with SPDX
      1. Reducing the cost of verifying open source licenses
      2. license notice web system
   3. Future work

3. Feedback to SPDX

4. Q&A
Introduction to SPDX
Demands on introducing open source to enterprise side

- Need a standardized, adopted format for a S/W Bill of Materials

Our suppliers aren’t giving us complete licensing information for open source packages.

I don’t mind vetting our code, but I’m sure this imported package has been analyzed a dozen times before.

Every customer wants a bill of materials in a different form.

software in

software out
License Complexity when adopting OSS (1)

- Products have mixed code from many different sources...

Diagram showing the flow of code from various sources including Open Source Software, Outsourced Code, Commercial 3rd-Party Code, Internally Developed Code, and Obligations.
License Complexity when adopting OSS (2)

- An OSS Package can contain multiple OSS packages, multiple licenses.

- License info for OSS is not provided in a consistent, easy-to-use format.

- One OSS Package = Many Licenses

Diagram:
- New code
- License A
- License B
- License C
- License D
- License E
- OSS Package
- License A
- License B
- License C
- License D
- License E
Software Package Data eXchange

- A standard format for communicating the components, licenses and copyrights associated with a software package.
- Hosted at the Linux Foundation whose goal is to facilitate compliance with free/open source software licenses and the exchange of such information between companies.
- 30 participants
SPDX introduction (2)

- SPDX as Software Bill of Materials

Companies combine OSS with other software

Creating an accurate bill of materials
Benefit

Benefits of standardizing
- Allows easy exchange of license information between companies, reducing the burden of both suppliers and consumers
- Avoids due diligence redundancy where the same source code package is analyzed multiple times by different receivers
- Provides a unified method for exchanging license information
List of “Standard Licenses”

- License repository
  - List of most common licenses (100+)
  - Include common exceptions
  - **Standardized license names**
  - Exact text of licenses
  - Available on SPDX® website – **URLs won’t change**
  - Short names adopted by OSI

<table>
<thead>
<tr>
<th>License Identifier</th>
<th>Recognized Exceptions</th>
<th>Full name of License</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFL-3.0</td>
<td></td>
<td>Academic Free License 3.0</td>
</tr>
<tr>
<td>AGPL-3.0</td>
<td></td>
<td>(GNU) Affero General Public License v3</td>
</tr>
<tr>
<td>APL</td>
<td></td>
<td>Adoptive Public License</td>
</tr>
<tr>
<td>ASL-2.0</td>
<td></td>
<td>Apache License, 2.0</td>
</tr>
<tr>
<td>APSL-2.0</td>
<td></td>
<td>Apple Public Source License 2.0</td>
</tr>
<tr>
<td>Artistic-2.0</td>
<td></td>
<td>Artistic license 2.0</td>
</tr>
<tr>
<td>AAL</td>
<td></td>
<td>Attribution Assurance License</td>
</tr>
<tr>
<td>BSD-4-Clause</td>
<td>BSD 4-clause “Original” or “Old” License</td>
<td></td>
</tr>
<tr>
<td>BSD-3-Clause</td>
<td>BSD 3-clause “New” or “Revised” License</td>
<td></td>
</tr>
<tr>
<td>BSD-2-Clause</td>
<td>BSD 2-clause “Simplified” or “FreeBSD” License</td>
<td></td>
</tr>
<tr>
<td>BSL-1.0</td>
<td>Boost Software License 1.0</td>
<td></td>
</tr>
<tr>
<td>CATOSL-1.1</td>
<td>Computer Associates Trusted Open Source License 1.1</td>
<td></td>
</tr>
<tr>
<td>CC-BY-1.0</td>
<td>Creative Commons Attribution 1.0</td>
<td></td>
</tr>
<tr>
<td>CC-BY-NC-1.0</td>
<td>Creative Commons Attribution Non Commercial 1.0</td>
<td></td>
</tr>
<tr>
<td>CC-BY-ND-1.0</td>
<td>Creative Commons Attribution No Derivatives 1.0</td>
<td></td>
</tr>
<tr>
<td>CC-BY-SA-1.0</td>
<td>Creative Commons Attribution Share Alike 1.0</td>
<td></td>
</tr>
<tr>
<td>CC-BY-NC-ND-1.0</td>
<td>Creative Commons Attribution Non Commercial No Derivatives 1.0</td>
<td></td>
</tr>
<tr>
<td>CC-BY-NC-SA-1.0</td>
<td>Creative Commons Attribution Non Commercial Share Alike 1.0</td>
<td></td>
</tr>
<tr>
<td>CC-BY-2.0</td>
<td>Creative Commons Attribution 2.0</td>
<td></td>
</tr>
<tr>
<td>CC-BY-NC-2.0</td>
<td>Creative Commons Attribution Non Commercial 2.0</td>
<td></td>
</tr>
<tr>
<td>CC-BY-ND-2.0</td>
<td>Creative Commons Attribution No Derivatives 2.0</td>
<td></td>
</tr>
<tr>
<td>CC-BY-SA-2.0</td>
<td>Creative Commons Attribution Share Alike 2.0</td>
<td></td>
</tr>
<tr>
<td>CC-BY-NC-ND-2.0</td>
<td>Creative Commons Attribution Non Commercial No Derivatives 2.0</td>
<td></td>
</tr>
<tr>
<td>CC-BY-NC-SA-2.0</td>
<td>Creative Commons Attribution Non Commercial Share Alike 2.0</td>
<td></td>
</tr>
</tbody>
</table>

Support permanent URL [https://spdx.org/licenses/](https://spdx.org/licenses/)
SPDX Specification (1)

Document Information

Creation Information

Package Information

File Information

Licensing Information

Review Information

SPDX Version and Licensing

How and when created

Package identification, copyright and licensing

File by file identification, copyright and licensing

Text of licenses that are not in SPDX™ standard list

Log of 3rd party reviews

File is in RDF/XML or tag value form and can be converted to/from spreadsheets.
SPDX Specification (2)

Document Information

Creation Information

Package Information

File Information

Licensing Information

Review Information

- Identification
  - Formal Name of Package (Full name given by originator and version information)
  - Unique ID (to unambiguously map a file to a package)
  - Package Download Location (download URL)

- Package Supplier and Originator

- Licensing for Package
  - Declared License- License that has been asserted for the package
  - Concluded License- License that Creator concluded
  - List of file licenses

- Copyright Text

Package identification, copyright and licensing

File by file identification, copyright and licensing

- File Name
- File Type (source, binary, archive)
- File CheckSum
- Declared License
- Concluded License (license determined by SPDX file creator)
- License Text in File
- Copyright Text
- Artifact of Project Name (from which project it came)
Tools and Version of SPDX

- **Open Source Tools** (hosted on SPDX Git Repo. [http://spdx.org/tools](http://spdx.org/tools))
  - SPDX Compare Utility
  - License RDFa Generator
  - RDF to HTML Pretty Printer
  - SPDX Viewer
  - Spreadsheet to RDF/Tag Value xlator
  - RDF/Tag Value to Spreadsheet xlator
  - License file generator (from Spreadsheet)
  - Spreadsheet template

- **Versions**
  - Current version: 1.1
  - Working on version 2.0
Piloting SPDX
- Motivation through open source development
Open source development process

OSS DISCOVERY
Search proper OSS package
License checking
Open source Package
Check from * Hosting Site * License Notice file
check out

Development
Develop using OSS packages

#include <stdio.h>
#include <stdlib.h>
int main() {
    printf("Hello, SPDX!");
    funcA();
    exit(0);
}

#include <stdio.h>
#include <stdlib.h>
int main() {
    printf("Hello, SPDX!");
    funcA();
    exit(0);
}

void funcA() {
    printf("funcA called");
}

Verification
Verify with Verification Tool

Open source Package

#include <stdio.h>
#include <stdlib.h>
int main() {
    printf("Hello, SPDX!");
    funcA();
    exit(0);
}

void funcA() {
    printf("funcA called");
}

Source Code Disclosure
License Notice

License checking
Open source Package
Check from * Hosting Site * License Notice file

Identify(Audit) by Reviewer

SPDX Piloting scope
“Identify” Effort

- Confirm the original copyright & license of source codes.
  - Verification tool can produce multiple matched result → reviewer need to audit them.
  - The work is labor-intensive and time-consuming for reviewers.

Multiple results are matched. (Each results can have different license obligation)

Example

GPL → source code needs to be released, and need to put up a GPL License notice
BSD → license notice needed but source code may not be released to public

It is GPL? BSD? → originator tracing..

Reviewer’s TODO

- Trace the originality of source code
  (web surfing using MD5 value of source file, visiting release site, or compare/analyzing various versions)
- Labor intensive, Time-consuming
Problems of identification

- Redundancy of identification.
  - Identification data cannot be shared with other company.
  - The Reviewer has to repeat identification effort.

- Inconsistency of identification result
Release process details

- **Source code disclosure**
- **License Notice**
  - Included in pages of the manual
  - Included in the product itself.
Problems of license notice

- Cost of printing manual pages.
- Cost of changing license info
  - Firmware upgrade or License Modification (by human error, etc...)
  - Re-printing license notice, new binary release
Piloting SPDX usage
Reducing the cost of verifying open source licenses

- Reducing re-verification cost
  - Auto Identify using SPDX format
AIRS (Auto Identify Using SPDX) Introduction (1)

- **AIRS**
  - Standalone program including Auto identify functionality
  - Interface for legacy system
    - JAVA library type or CLI

- **Main function**
  - **SPDX Export Function**
    - Export identification data as SPDX file format
    - Use file comment field to include identification data
  - **Auto Identify Function**
    - Identify exported identification data to new system

[ content of exported SPDX file ]

```
< ceasefire rdf:nodeID="A2464"/>
```

Identification data is written in SPDX File

This file is identified by STRING SEARCH AUTOIDENTIFICATION. - GPL 2.0 [Reviewed and Rejected]<rdf:comment>
"<licenseInfoInFile rdf:resource="http://spdx.org/licenses/GPL-2.0">"
<fileComment>tools/spicelo.c</fileComment>
<licenseComments></licenseComments>
<licenseConcluded rdf:nodeID="A269"/>

22/31
Pros of Auto Identify

- Identify when SHA1 checksum is identical whether absolute file path is not same.
  - It can be identified even if the directory structure is changed,
- Various options
  - Select the most recent reviewed file in case multiple identical files exists. (or not)
  - Overwrite existing identification data (or not)
  - Identify only when absolute file path is the same.(or not)

[Auto identify flow chart]

[Auto identify Rules]
Running AIRS

- Auto-identify target project with source project’s SPDX file.

[ before ]

Execute AIRS

ex)
# java -jar airs.jar ai -h http://127.0.0.1 -u developer@samsung.com -p passwd -proxy-host 127.0.0.1 --proxy-port 8080 --project-id c_13_swc_developer_ai_demo_130826 --spdx-files source.rdf

[ after ]

※ OSI: Open source Self Inspector (samsung in-house tool)
license notice web system

- Reduce license notice cost
  - Only QR Code/URL printed on the pages of the manual
  - No need to change binary or manual whenever license notice needs to be changed.

- Automate license notice by using standard form

AIRS

SPDX Export

Upload SPDX, insert product data

http://opensource.samsung.com

Provide QR Code/URL on product or the pages of the manual

Product Binary

Manual Pages

CD
[ License Notice on web site using SPDX ]

Write product information

QR Code/URL included as part of the product documentation (in manual, on box, in device with written offer, etc.)

End user visits specified Web site(OSRC) using QR Code (or connect to URL directly)

Admin page in opensource.samsung.com

License Notice and QR Code(URL) created

End user receives license info in the product including source code, license, and compliance contact

[ End User Scenario ]

Ex) QRCode on box
Future work (1)

- **Build Auto-identify Database in Samsung**
  - Collection of certified identification data.
  - Data contents
    - Frequently used open source package for every products.
    - Packages of main branch module in specific platform.

**Identification Data**
- RDF(SPDX) Format

[ IT & mobile division ]

[ Visual Display division ]

[ Printing Solution division ]
Future work (2)

- **Support various verification tool**
  - Currently AIRS supports only Protex.
  - Designed to be compatible with various tools (abstract layer)

- **Improve auto-identify function**
  - Identify when code snippet is same even if the file checksum is different
  - More detail options.

- **FOSS friendly**
  - Using SPDX Parser(from SPDX-tools)
  - Planning contribution of AIRS
Feedback to Standard
Feedback to SPDX

- Modification of “Artifact of Project” fields
  - Cardinality: “Optional, One” ➔ “Optional, one or many”
  - Specific derived-file information is necessary
    • Currently, DOAP is used ➔ less detailed.

- Package Hierarchy
  - Deal with sub-packages (no specification)
  - Connectivity to other packages

- Patternize license comment
  - Provide license comment patterns for better-automation
  - ex) This file is derived from APACHE-COMMONS.jar
    ➔ <derivedFrom>apache-commons.jar</derivedFrom>

- Force common rule for file path
  - File path is important for comparing/auto-identifying.
    • Path can be started from “./”, “/”, “<package_file_name>/”
  - Common rule is necessary.

- Summary of Derived project information in package info
  - When details of derived project are needed, only file info section can give them, however this way can produce highly redundant data.
Q&A