NTIA Software Component Transparency
April 15, 2020

Formats & Tooling Workgroup

JC Herz
Steve Springett
Kate Stewart

Agenda

- Workgroup Goals
- Overview Document
- Tooling Information being collected per Format
  - Template and Example
- Gap Analysis
- Feedback Request

Formats and Tooling Workgroup Goal

Wrapping up from phase I, we identified for the need for:

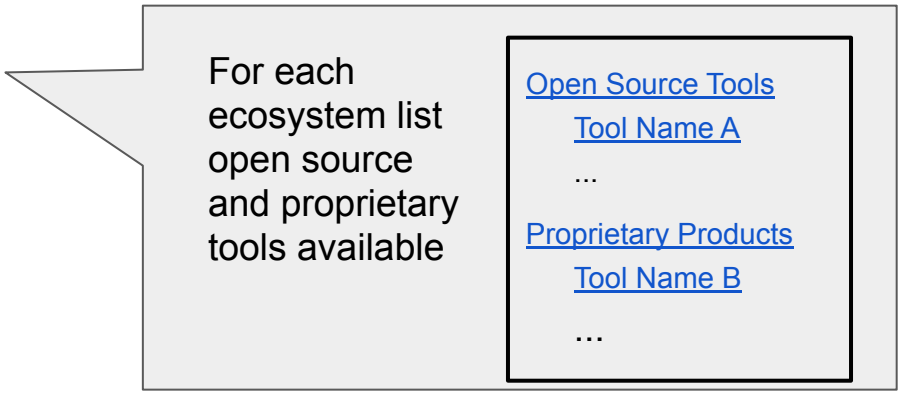
- Tooling
  - Documenting tooling
  - Identifying tooling gaps
  - Documenting processes
  - Turnkey universal translation tools

Formats and Tooling workgroup is focusing on addressing these items.

Overview Document

Overview of Tooling that supports Automation working with Software Bill of Materials Formats.

- Introduction
- Definitions
- Tooling Ecosystem for Key Formats
  - SWID
  - SPDX
  - CycloneDX
- Conclusions



Updated Taxonomy used for classifying tools

Category	Type	Description
Author during Build	Build	Document is automatically created as part of building an artifact and contains information about the build.
Author after Creation	Manual	A person will manually fill in the information
	Audit Tool	A source code analysis or audit tool will generate the document by inspection of the artifact and any associated sources.
Consume	View	Be able to understand the contents in human readable form (picture, figures, tables, text.). Use to support decision making & business processes.
	Diff	Be able to compare two documents of a given formation and clearly see the differences. For instance, comparing between two versions of a piece of software.
	Analyze	Be able to import a document into
Transform	Translate	Change from one file type to another file type while preserving the same information.
	Merge	Multiple sources of documents can be merged together for analysis and audit puproses
	Tool integration	Support use in other tools by APIs, libraries.

Information to Collect per Tool (updated)

Tool Template

Support	Author during Build, Author after Creation, Consume, Transform
Functionality	
Location	Website: Source:
Installation instructions	
How to use	
Versions Supported	

Example: FOSSology

Support	Author after Creation (Audit tool, Manual), Consume(View,Diff,Analyze), Transform(Translate, Merge, Tool Integration)
Functionality	FOSSology is an open source license compliance software system and toolkit allowing users to run license, copyright and export control scans from a REST API. As a system, a database and web UI are provided to provide a compliance workflow. As part of the toolkit multiple license scanners, copyright and export scanners are tools available to help with compliance activities.
Location	Website: https://www.fossology.org/ Source: https://github.com/fossology
Installation instructions	https://www.fossology.org/get-started/
How to use	https://www.fossology.org/get-started/basic-workflow/
Versions Supported:	SPDX 2.1, SPDX 2.2 (WIP)

Tooling Surveys to be Reviewed:

SWID

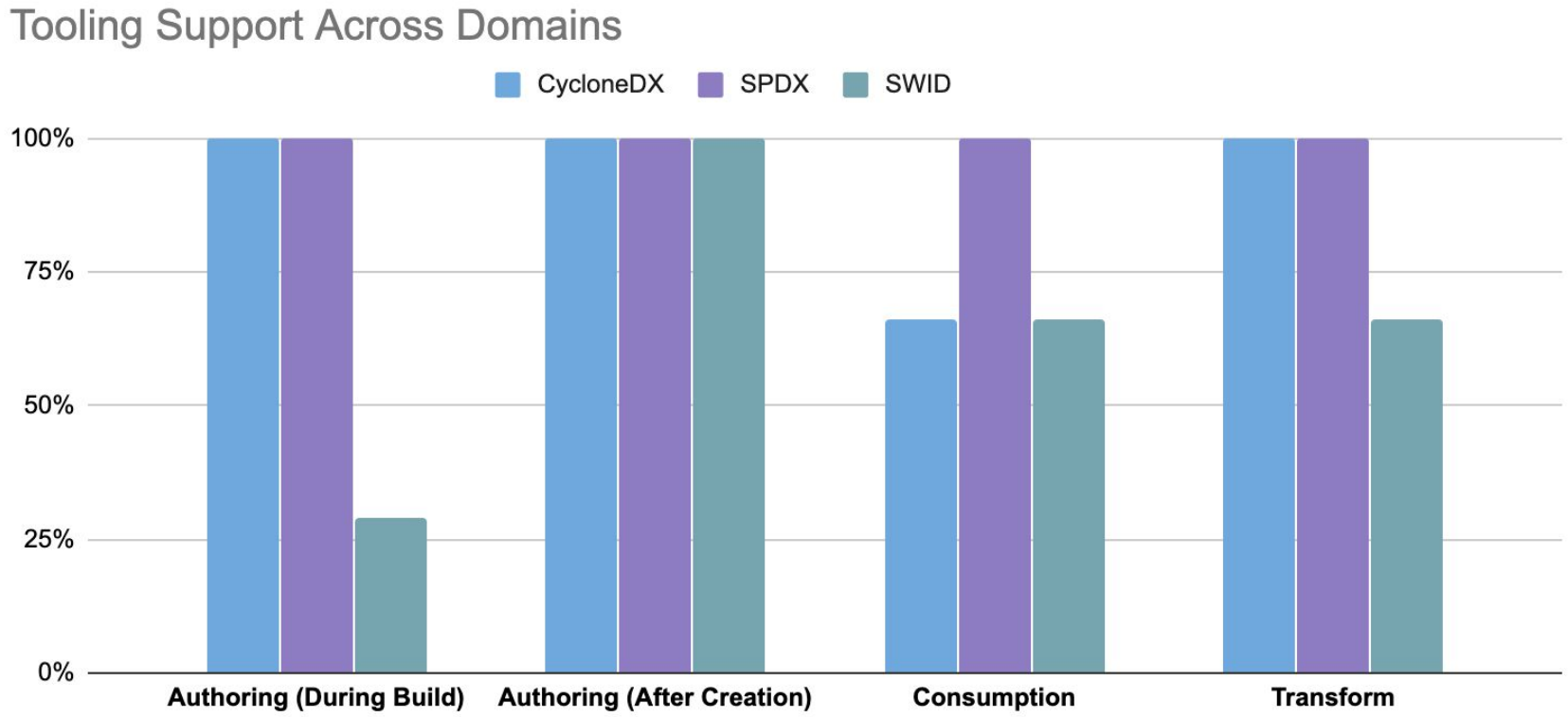
SPDX

CycloneDX

Gap Analysis between the Different Formats

		CycloneDX	SPDX	SWID
Author during Build				
	Javascript	✓	✓	
	Python	✓	✓	
	Java	✓	✓	✓
	PHP	✓	✓	
	C#	✓	✓	
	C/C++	✓	✓	✓
	Go	✓	✓	
Author after Creation				
	Manual	✓	✓	✓
	Audit Tool	✓	✓	✓
Consumption				
	View	✓	✓	✓
	Diff		✓	
	Analyze	✓	✓	✓
Transform				
	Translate	✓	✓	✓
	Merge	✓	✓	
	Tool Integration (Libraries, APIs, etc)	✓	✓	✓

Gap Analysis between the Different Formats



Feedback Request

- Feedback on proposed approach? taxonomy?
- Know a tool to be added to each ecosystem document?
  Put a comment in the document, so it can be added.
  - SWID: http://tiny.cc/SWID
  - SPDX: http://tiny.cc/SPDX
  - CycloneDX: http://tiny.cc/CycloneDX
- Priorities for next steps?
  - Crop circles mapping
  - Software Life cycle mapping

Questions?

More Info...

Mailing List: ntia-sbom-formats@linuxfoundation.org

Subscribe at: https://lists.linuxfoundation.org/mailman/listinfo/ntia-sbom-formats

Shared Drive:
https://drive.google.com/drive/folders/1KAQ7AWlWMKcSFnRc_S-7XB76xFRRWLmT