

# SPDX 3.0 - a migration journey

---

Gary O’Neill, co-lead SPDX tech, maintainer SPDX Java libs



# Agenda

---

## SPDX 2 to SPDX 3

- Why SPDX 3?
- The approach to creating the spec
- Overview of the changes

## Impact on the Java Libraries

- Current (soon to be previous) Java Library architecture
- Breaking changes in the model causing breaking changes in the library
- Reducing the impact of the breakage
- Reducing errors translating from the spec



# Why SPDX 3.0?

Simplify - profiles

Flexibility - relationship structure

Interest in SPDX for non-licensing scenarios

- supporting **security** and **safety critical** application compliance requirements.
- **AI/ML** and **datasets** increasing need for transparency

Consolidate efforts between the SPDX community & OMG/CISQ efforts

# Timeline of SPDX Evolution - Use Case by Use Case

I AM THE  
**Cavalry**

**Legislation:**  
proposed software transparency, updatability & bill of material as reqts in safety critical sectors (automotive & healthcare)



**NTIA:**  
Software Transparency begins



**3T-SBOM:**  
OMG/CISQ begins w/ CONOPs for Tool-to-Tool SBOM

SPDX 3.0 initial draft  
3T-SBOM initial draft



**Transition of SBOM work to DHS**



**Executive Order 14028**



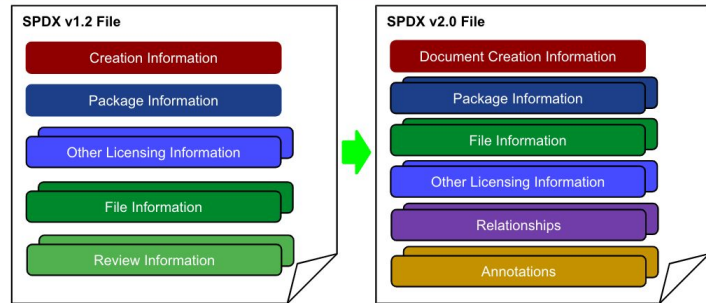
**EU Cyber Resilience Act**



# Specification Evolution



## The SPDX Document



Search Tools

Copyright

Foreword

Introduction

Clause 1: Scope

Clause 2: Normative references

Clause 3: Terms and definitions

Clause 4: Conformance

Clause 5: Composition of an SPDX document

Clause 6: Document Creation Information

Clause 7: Package Information

Clause 8: File Information

Clause 9: Spdxid Information

Clause 10: Other Licensing Information Detected

Clause 11: Relationship between SPDX Elements Information

Clause 12: Annotation Information

Clause 13: Review Information

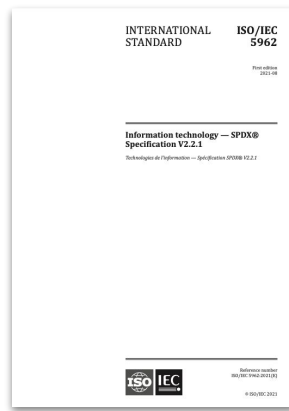
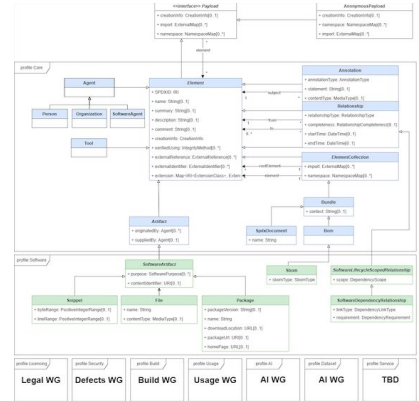
Annex A: SPDX License List

Annex B: License Matching Guidelines and Templates

### The Software Package Data Exchange® (SPDX®) Specification Version 2.2.2

Copyright © 2010-2022 Linux Foundation and its Contributors. This work is licensed under the Creative Commons Attribution License 3.0 Unported (CC-BY-3.0) reproduced in its entirety in Annex G herein. All other rights are expressly reserved.

With thanks to Adam Cohn, Alan Tan, Alexios Zavas, Andrew Black, Ann Thornton, Basil Perez, Bill Scheller, Braden Ederington, Bruno Gomes, Casari Farini, Daniel Gennari, David Edstrom, David Kemp, David A. Wheeler, Debra McGlad, Dennis Clark, Dick Brooks, Ed Worsick, Eran Srod, Eric Thomas, Emban Richter, Gary O'Neil, Guillaume Rozencas, Haasb Kraussler, Hank Blahnik, Henadi Fakhri, Jack Markovic, Jaime Garcia, Jeff Lopez, James Knodys, John Ellis, Jonas Ober, Kinnang Salina, Karen Coppenbaker, Kate Stewart, Kevin Mitchell, Kim Vahns, Kirsten Newcomer, Kris Reeves, Liang Cai, Lon Holboven, Marc Elmore, Margaux, Matt Cui, Marshall Cox, Martin Michnyager, Martin von Willbrand, Mark Alwood, Matjaz Sulic, Matt Gomezogor, Maximilian Haber, Michael J. Herzig, Michel Ruffer, Nikh Kumar, Norio Kobata, Nuno Brito, Oliver Fendt, Paul Madala, Peter Williams, Phil Babin, Philip Kotler, Philo Oelene, Philippe Oudenaerde, Pierre Lapointe, Rana Rahal, Robert Martin, Robin Gandhi, Ross Judge, Sam Ellis, Sameer Ahmed, Scott K Peterson, Scott Lintott, Scott Shilling, Sean Barron, Sebastian Cravea, Shree Coughlin, Shree Cragger, Steve Winlow, Stuart Hughes, Thomas F. Inoué, Thomas Steenbergen, Tom Callaway, Tom Vidas, Tony Taina, Venkata Krishna, W. Trevor King, William Bartholomew, Wie Bronsthor, Yuhua Chen, Yoshitaka Ito, Yui Nemura and Zachary McIninch for their contributions and assistance.



Serialization into common formats  
 .json/.json-ld/.yaml/.xml/.rdf/.spdx/.xls



Online Specification



PAS ISO Submission



# SPDX 3.0 Specification Infrastructure

Specification is being transformed into markdown describing

- Classes, Properties, Enumerations
- Metadata (type & cardinality) and description for each element.
- Will be able to automatically generate schema from this version (for JSON, YAML, RDF, XML, tag-value, etc.) and reduce errors.

Profiles can add their own Classes and Properties and may also restrict other profiles (e.g. values, cardinalities, ...)

See: <https://github.com/spdx/spdx-3-model>



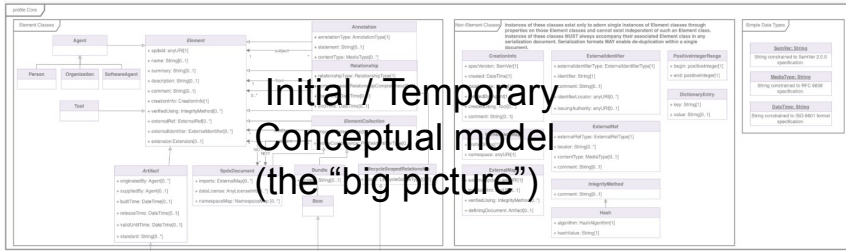








# Approach to Creating the Spec



Preview Code 11:25 AM (7:24) - 1.12.18

SPDX-License-Identifier: Community-Spec-1.0

### Element

Summary

**Write the spec in Markdown using a specific formation**

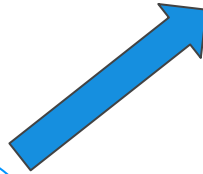
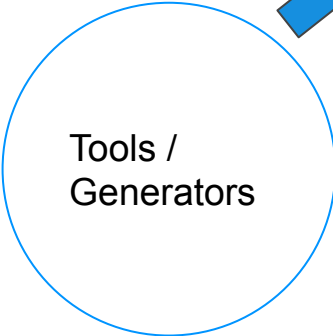
Description

Metadata

- name: Element
- SubclassOf: none
- Instability: Abstract

Properties

- spdxid



### 7.13 Concluded license field

#### 7.13.1 Description

7.13.1.1 Overview

7.13.1.2 Examples

7.13.1.3 Licenses information from file field

7.13.1.4 Declared license field

7.13.1.5 Comments on license field

7.13.1.6 Copyright text field

7.13.1.7 Package summary description field

7.13.1.8 Package detailed description field

7.13.1.9 Package comment field

7.13.1.10 External reference field

7.13.1.11 External reference comment field

7.13.1.12 Package attribution text field

7.13.1.13 Primary Package Purpose field

7.13.1.14 Release Date

7.13.1.15 Bulk Date

7.13.1.16 Valid Until Date

7.13.1.17 File Information

7.13.1.18 Other Licensing Information

7.13.1.19 Previous

7.13.1.20 Next

7.13.1.21 Required

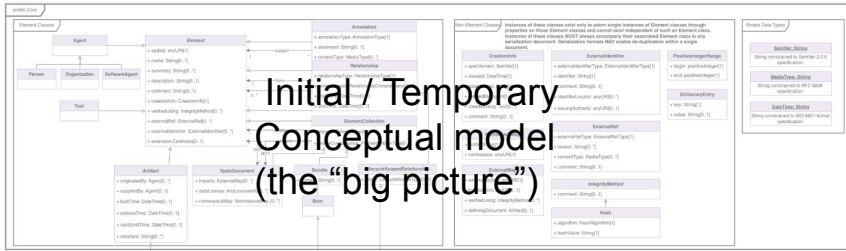
7.13.1.22 No

### Schema OWL/SHACL



### Serialization Schemas

# Approach to Creating the Spec



SPDX-License-Identifier: Community-Spec-1.0

### Element

Summary

Description

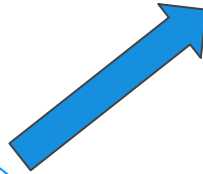
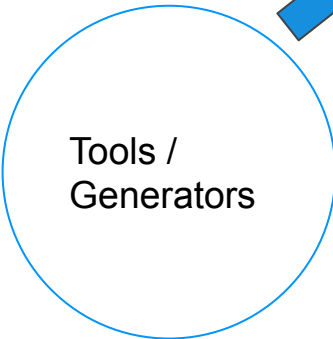
Metadata

- name: Element
- SubclassOf: none
- Instability: Abstract

Properties

- spdxid

Write the spec in Markdown using a specific formation



7.13 Concluded license field

7.13.1 Description

7.13.1.1 Example

7.13.1.2 Example

7.13.1.3 Example

7.13.1.4 Licenses information from file field

7.13.1.5 Declared license field

7.13.1.6 Comments on license field

7.13.1.7 Copyright text field

7.13.1.8 Package summary description field

7.13.1.9 Package detailed description field

7.13.1.10 Package comment field

7.13.1.11 External reference field

7.13.1.12 External reference comment field

7.13.1.13 Package attribution text field

7.13.1.14 Primary Package Purpose field

7.13.1.15 Release Date

7.13.1.16 Bulk Date

7.13.1.17 Valid Until Date

7.13.1.18 File Information

7.13.1.19 Snippet Information

7.13.1.20 Other Licensing Information

Attribute	Value
Required	No

Schema OWL/SHACL



Serialization Schemas

SPDX-License-Identifier: CC-BY-4.0

Ensures Consistency

# Why SHACL / OWL?

- Captures semantic constraints as well as syntax constraints
- A superset of serialization schema functionality
  - Can be used to generate any of the schemas we've identified
- Tools exist in most language ecosystems to validate

# Structural Changes

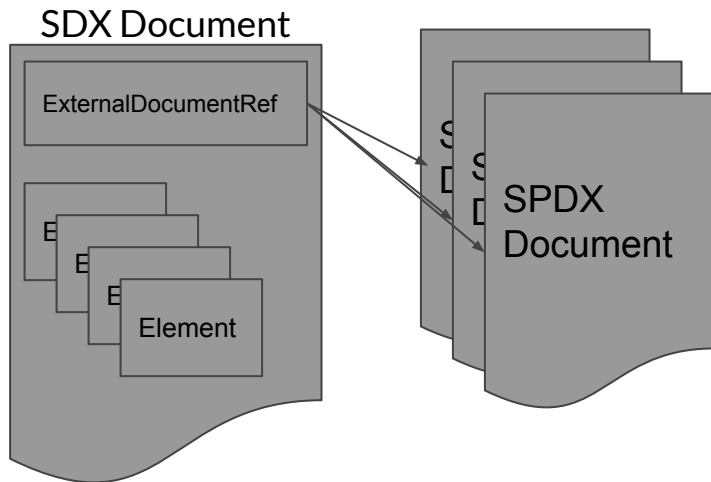
## Profiles

- Conformance Requirements
  - Additional restrictions on properties (e.g. required license information in the licensing profile)
- Namespace
  - Organizes the vocabulary into more logical digestible units (e.g. you don't have to know all the licensing terms if you're only interested in security)
- Organization
  - SPDX work groups are organized around profiles

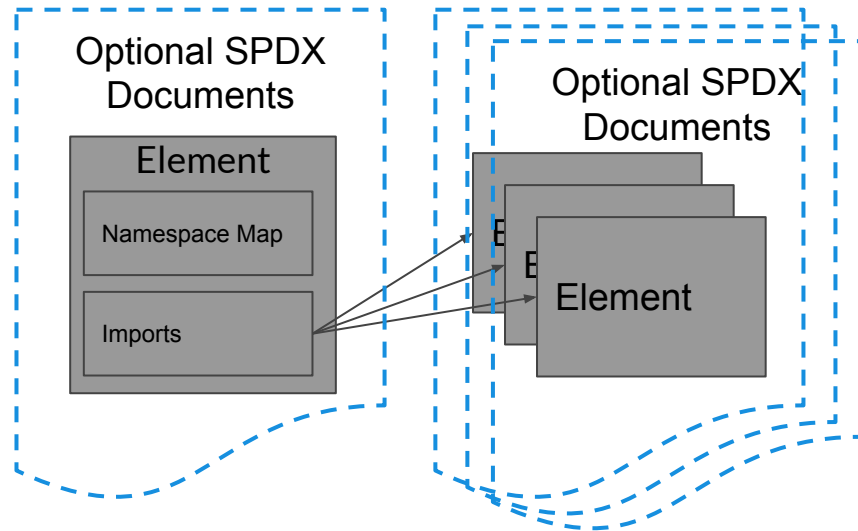
# Structural Changes

## External Document References

### SPDX 2.3



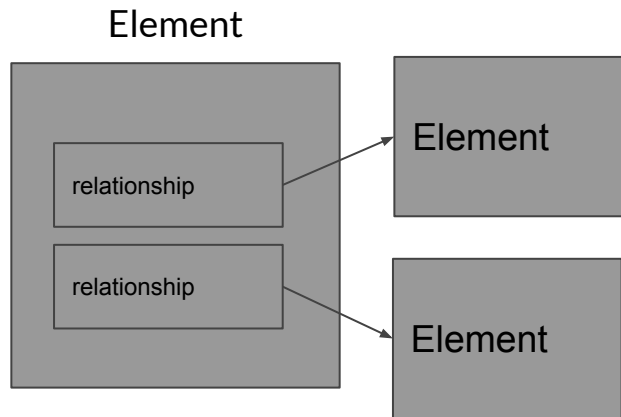
### SPDX 3.0



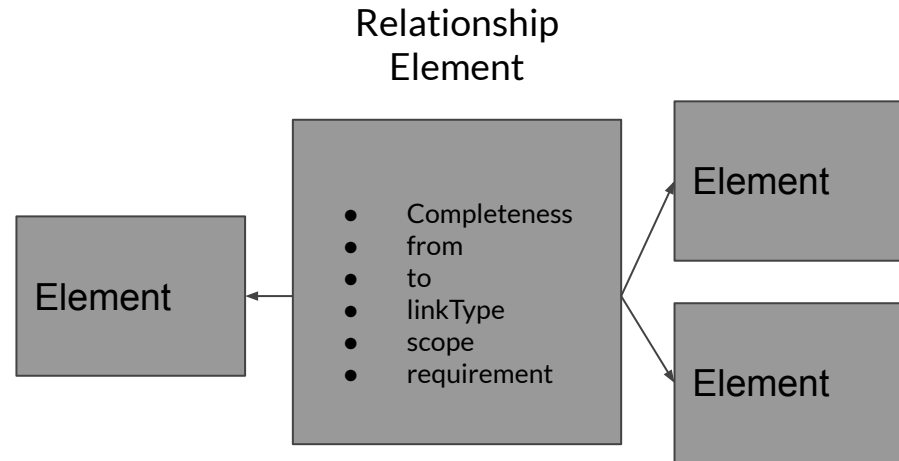
# Structural Changes

## Relationships

### SPDX 2.3



### SPDX 3.0





# Other SPDX 3.0 changes

- Better model for “Entities” (person, organization, tools)
- Renamed or removed confusing properties (e.g. filesAnalyzed)
- Added some useful classes and properties (e.g. Artifact, packageUrl)
- Profile specific classes and properties
- Details: [https://docs.google.com/document/d/1-olHRnX1CssUS67Psv\\_sAq9Vd-pc81HF8MM0hA7M0hg](https://docs.google.com/document/d/1-olHRnX1CssUS67Psv_sAq9Vd-pc81HF8MM0hA7M0hg)

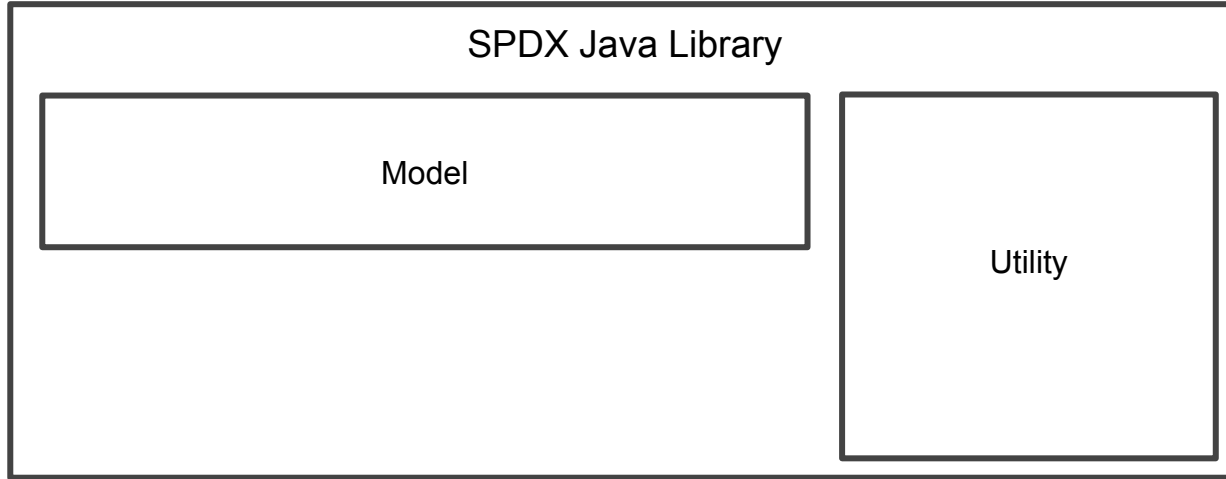
# Changes “Big Picture”

- More flexibility with profiles, new relationship structure, independent annotations, external map structure
- Simpler with profiles, simpler snippets, no more filesAnalyzed, clearer naming
- More use cases with profiles
- For the details - see the Migration Document:  
[https://docs.google.com/document/d/1-oHRnX1CssUS67Psv\\_sAq9Vd-pc81HF8MM0hA7M0hg/edit?usp=sharing](https://docs.google.com/document/d/1-oHRnX1CssUS67Psv_sAq9Vd-pc81HF8MM0hA7M0hg/edit?usp=sharing)

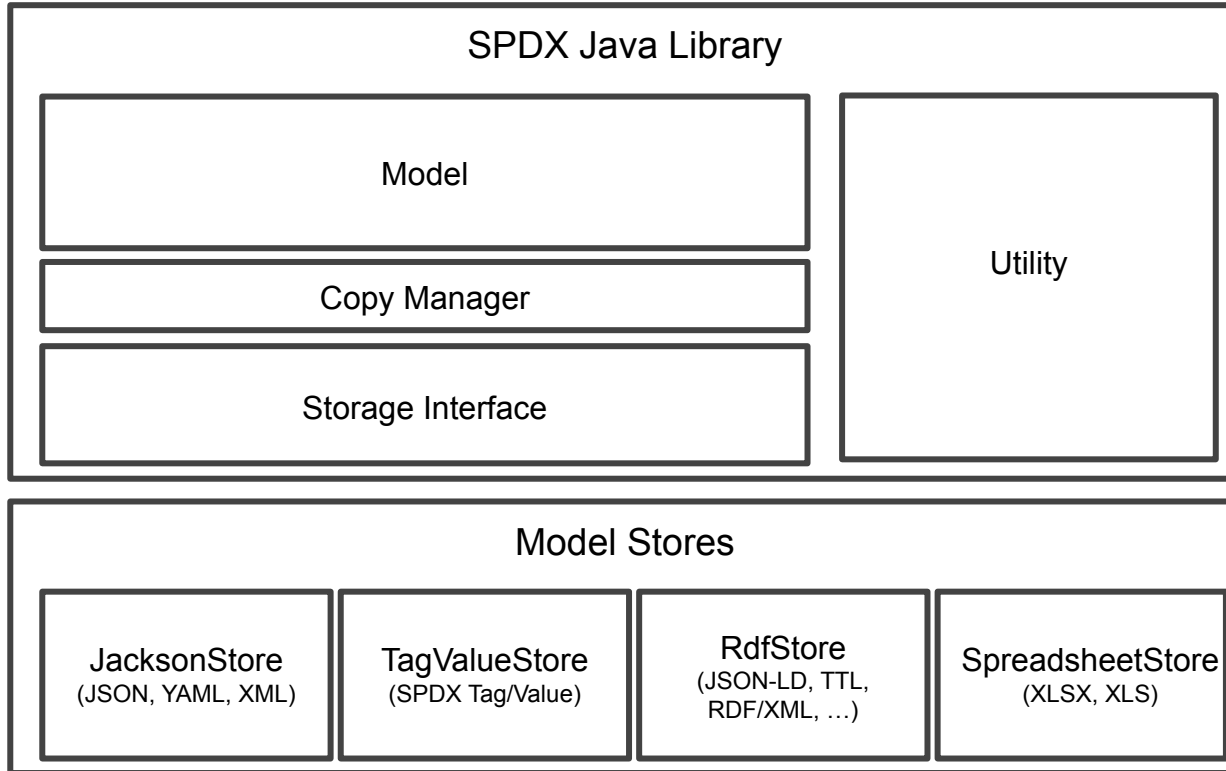
# Impact on Java Libraries



# Current Java Library Architecture



# Current Java Library Architecture



# Breaking Changes in the Java Library

## Storage Interface

- Expanded namespaces in profiles
  - Can no longer assume property names are unique
  - Added a “namespace” parameter in addition to the property name

```
public Optional<Object> getValue(String documentUri, String id, String propertyName)
```



```
public Optional<Object> getValue(String documentUri, String id, String namespace, String propertyName)
```

# Breaking Changes

## Model

- Compatibility package if you're dealing with an SPDX 2.X document
- Relationships and Annotation structure - independent elements
- ExternalDocumentRef structure
- Replace Person/Organization strings with Agent
- Snippet simplifications
- Properties -> Relationships (e.g. packageFileName, concludedLicense)
- Plus a few other miscellaneous changes (see the migration document for details)

# Reducing the Breakage

- Compatibility package with the version 2.X model
  - “CopyManager” handles upgrades from 2.X to 3.0
- SpdxModelFactory to direct you to the correct model based on SPDX spec version
  - getElement, getModelObject (by type), createModelObject (by type)
- “Relationship” list properties
  - Treats a relationship as a list property within a class
  - Can also access the relationships independently
  - Example: getConcludedLicense
- Compatible storage interface which looks up the namespace for property names



# Reducing the Errors

## Generating the Model Files

- Generated from the OWL/SHACL file
  - Currently implemented in Java using Mustache templates
  - Investigating switching to the Python generator written by Joshua Watt
  - Consistency from the spec markdown files all the way through the code
- Generate serialization specific schemas from the OWL/SHACL file
  - JSON Schema
  - XML Schema
- Generate verification code based on the OWL/SHACL schema
- For RDF serializations, verify directly against the OWL/SHACL schema
  - Not implemented with other serialization formats due to the “weight” of the RDF dependencies

# SPDX 3.0 Java Library Architecture

