SPDX 3 in the Yocto Project
A case study of migrating from SPDX 2 to SPDX 3

Joshua Watt
SPDX General Meeting
June 6th, 2024
About Me

- Worked at Garmin since 2009
- Using OpenEmbedded & Yocto Project since 2016
- Member of the OpenEmbedded Technical Steering Committee (TSC)
- Joshua.Watt@garmin.com
- JPEWhacker@gmail.com
- IRC (OFTC or libera): JPEW
- X/Twitter: @JPEW_dev
- LinkedIn: joshua-watt-dev

All opinions are my own
OpenEmbedded and Yocto Project

OpenEmbedded
- Community project
- OpenEmbedded core layer
- Build system (bitbake)

Yocto Project
- Linux Foundation project
- Poky reference distribution
- Runs QA tests
- Manages release schedule
- Provides funding for personnel
- Documentation
Circa 2021: Yocto Project Needs SBoMs
How hard could this be?
A brief overview of how Yocto works

See https://youtu.be/Q5UQUM6zxVU
Build Images from Source Code

Source
Metadata
Policy

bitbake

Target Image

Widget
SPDX Generation
SPDX 2 Model

Package SPDX
 Runtime SPDX
 Recipe SPDX
 Package Files
 Source Code
 Image SPDX
 Image Index JSON

RUNTIME_DEPENDENCY_OF
AMENDS
CONTAINS
GENERATED_FROM
BUILD_DEPENDENCY_OF
GENERATED_FROM
Everything is Awesome… Right?

Tim Moreillon from Geelong, Australia, CC BY-SA 2.0, via Wikimedia Commons
Quirks

- SPDX 2 has no concept of a "build"
  - So we cheat a little
- Merging Documents is hard
  - So we have a tarball
- Data of this magnitude can't be easily validated
  - So it's best effort
- SPDX 2.2 has no vulnerability information
  - So we add custom information
- Tooling is all hand written
  - We have a "Zero Dependency" rule

So... Much... Data!
Unanswered Questions

- But what do I do with this data?
- Are there any tools to consume and manipulate it?
- Can I use it to figure out XYZ?

All compounded because our SPDX 2.2 format is unique

It's unlikely 3rd parties will support our "quirks"
A new contender emerges!

PDX 3.0
SPDX 3.0 Involvement

- My involvement in SPDX 3 has primarily been to address the shortcomings we discovered in SPDX 2
  - I'm not claiming credit for all these!
SPDX 3 Build Information
Build Elements

Build

File

hasInputs

dependsOn

hasOutputs

Build

File
Nested Builds

Top Level Invocation by user: `bitbake ...`

- **Build**
  - **Recipe Build**
  - **Recipe Build**
  - **Recipe Build**

ancestorOf
Build Invocation Information

Three new relationships to describe how a build was invoked:

- **hasHost** - The host where the build was run (potentially the full SPDX document that describes it)
- **invokedBy** - The user or agent that actually invoked the build
- **delegatedTo** - Relates a user to the `invokedBy` user to indicate the build was executed on their behalf (e.g. A build service building on behalf of a user)
Document Linking and Merging
SPDX 3 uses Linked Data

- SPDX 3 follows the principles of [Linked Data](#)
- Objects *can* have a globally unique SpdxId (IRI, URL-ish) which can be referenced by anyone (*mandatory* if the Element can be referenced)
- Since SpdxIds are globally unique (instead of scoped to a document), linking intermediate documents is *much* easier
- Merging documents is also much easier because there are no namespaces (other than anonymous objects)
Yocto SPDX 3 Generation

- Yocto builds have many discrete steps that generate SPDX files
- SPDX 3.0 makes it much easier for each step to generate valid intermediate output
- Intermediate documents can easily reference other objects by SpdxId
- A database that maps SpdxIds to files is kept as files are generated
- We make heavy use of extensions at this phase to track extra information needed later in the intermediate documents
- The final step is to aggregate all the documents into a final output document (and strip out the intermediate extension data), using the database to recursively resolve missing SpdxIds
  - I wrote a linker today....
Validation is Hard
SPDX 3 Validation

SPDX 3 has a formal SHACL model

- Validates the semantics of a document (but not necessarily the structure)
- [https://spdx.org/rdf/3.0.0/spdx-model.ttl](https://spdx.org/rdf/3.0.0/spdx-model.ttl)

SPDX 3 JSON-LD Documents must conform to a JSON Schema

- Validates the structure of a document (but not necessarily the semantics)
- [https://spdx.org/schema/3.0.0/spdx-json-schema.json](https://spdx.org/schema/3.0.0/spdx-json-schema.json)

Allows off-the-shelf tooling to be used in validation
Tooling Improvements
Automatic Generation of Code Bindings

- I wrote a tool to consume the formal SHACL model and generate code bindings:
  - [https://github.com/JPEWdev/shacl2code](https://github.com/JPEWdev/shacl2code)
- Python bindings can be generated this way
  - Importantly for use, these are "zero dependency"
  - Should be published to PyPi "soon"
- JSON Schema for validation is generated using this tool
- Other bindings are in the works (e.g. go, etc.)
  - If you want to help add more, please contact me!
- Keeps the code bindings up to date automatically as new versions of the spec come out
Makes tool writing easier

Being able to generate meaningful code bindings reduces the barrier to entry for writing simple tools

- [https://github.com/JPEWdev/spdx3query](https://github.com/JPEWdev/spdx3query)
  - CLI tool to navigate a SPDX 3 document
- [https://github.com/JPEWdev/spdx3merge](https://github.com/JPEWdev/spdx3merge)
  - Merge SPDX 3 documents together
- [https://github.com/maxhbr/spdx3ToGraph](https://github.com/maxhbr/spdx3ToGraph)
  - Generate a visualization graph from an SPDX 3 file
Vulnerability Information
SPDX 3 Vulnerability reporting

- SPDX 3 implements a complete VEX-compliant encoding for how Vulnerabilities have been addressed
- It's complex, but powerful
SPDX 3 VEX Example

hasAssociatedVulnerability

doesNotAffect
text:

vexVersion: 1.0.0

impactStatement: Code not compiled
Conclusion
Yocto SPDX 3 support is much improved

- Better adherence to the standard
- Better tooling to deal with output
- Much more likely to be usable by 3rd party tools
Still a lot of data!

- The basic testing image in Yocto is 79 MB
- The SPDX 3.0 JSON file that describes it is 306 MB

Consider yourself warned
Coming to a Yocto release near You! (October 2024)

Yocto 5.1, a.k.a. "Styhead (pass)"

Towards Styhead Pass by Richard Law, CC BY-SA 2.0, via Wikimedia Commons
Questions?