Welcome to Software Bill of Materials (SBOM) devroom

FOSDEM 2023

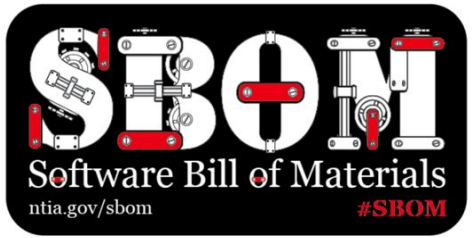
Schedule

Welcome to the SBOM devroom! *Introduction to the devroom*	Alexios Zavras, Kate Stewart, Adolfo García Veytia	09:00	09:05
Generating SBOM made easy with ORT	Thomas Steenbergen	09:05	09:30
Understanding and Managing the Dependency in SBOM with the New Feature of SW360	Kouki Hama	09:30	10:00
Automated SBoM generation with OpenEmbedded and the Yocto Project *A case study of automated SBoM generation in meta build systems*	Joshua Watt	10:00	10:30
SBOM with the Yocto Project for Automotive Grade Linux *Intro and lessons learned*	Jan-Simon Möller	10:30	10:45
Hermine: converting SBOMS into legal obligations	Nicolas Toussaint, Camille Moulin	10:45	11:15
A standard BOM for Siemens	Thomas Graf, Thomas Jensen, Alexander Gschrei	11:15	11:45
FOSSology and SPDX *How FOSSology works with SPDX*	Gaurav Mishra, Mohammed Shaheem Azmal Madanapalli	11:45	12:00
Build recorder: a system to capture detailed information	Alexios Zavras, Fotios Valasiadis	12:00	12:30
Discussion on SBOM contents	Arnout Vandecappelle	12:30	13:00

Schedule

Using SPDX for functional safety	Nicole Pappler	13:00	13:30
REUSE *The gold standard of communicating licensing and copyright information*	Linus Sehn	13:30	13:45
A complete compliance toolchain for Yocto projects *(even very large ones, yes)*	Carlo Piana, Alberto Pianon	13:45	14:00
In SBOMs We Trust: How Accurate, Complete, and Actionable Are They?	Joseph Hejderup, Henrik Plate	14:00	14:30
The 7 key ingredients of a great SBOM *Ensuring your SBOM includes enough data to be actionable*	Adolfo García Veytia	14:30	15:00
Panel discussion: SBOM content, usefulness, and caveats	Bradley M. Kuhn, Alexios Zavras, Anthony Harrison, Julian Coccia, Paul Novarese	15:00	16:30
General Q&A on SBOMs	Kate Stewart, Adolfo García Veytia	16:30	16:55
SBOM devroom closing	Alexios Zavras, Kate Stewart, Adolfo García Veytia	16:55	17:00

Common Understanding of “SBOM”



“An SBOM is a formal record containing the details and supply chain **relationships** of various **components** used in **building software.**

These components, including libraries and modules, can be open source or proprietary, free or paid, and the data can be widely available or access-restricted.”

Source: NTIA’s SBOM FAQ

NTIA SBOM Guidance

Minimum Elements	
Data Fields	Document baseline information about each component that should be tracked: Supplier, Component Name, Version of the Component, Other Unique Identifiers, Dependency Relationship, Author of SBOM Data, and Timestamp.
Automation Support	Support automation, including via automatic generation and machine-readability to allow for scaling across the software ecosystem. Data formats used to generate and consume SBOMs include SPDX, CycloneDX, and SWID tags.
Practices and Processes	Define the operations of SBOM requests, generation and use including: Frequency, Depth, Known Unknowns, Distribution and Delivery, Access Control, and Accommodation of Mistakes.

Source:https://www.ntia.gov/files/ntia/publications/sbom_minimum_elements_report.pdf.

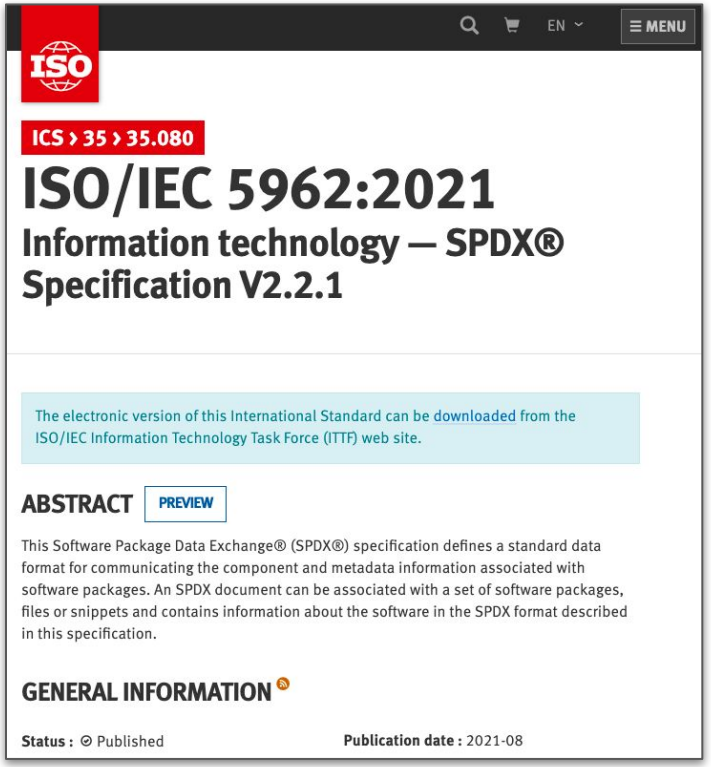
NTIA SBOM Guidance - Minimum Elements

Data Field	Description
Supplier Name	The name of an entity that creates, defines, and identifies components.
Component Name	Designation assigned to a unit of software defined by the original supplier.
Version of the Component	Identifier used by the supplier to specify a change in software from a previously identified version.
Other Unique Identifiers	Other identifiers that are used to identify a component, or serve as a look-up key for relevant databases.
Dependency Relationship	Characterizing the relationship that an upstream component X is included in software Y.
Author of SBOM Data	The name of the entity that creates the SBOM data for this component.
Timestamp	Record of the date and time of the SBOM data assembly.

Source:https://www.ntia.gov/files/ntia/publications/sbom_minimum_elements_report.pdf.
See also US Executive Order on Cybersecurity section 4(f)

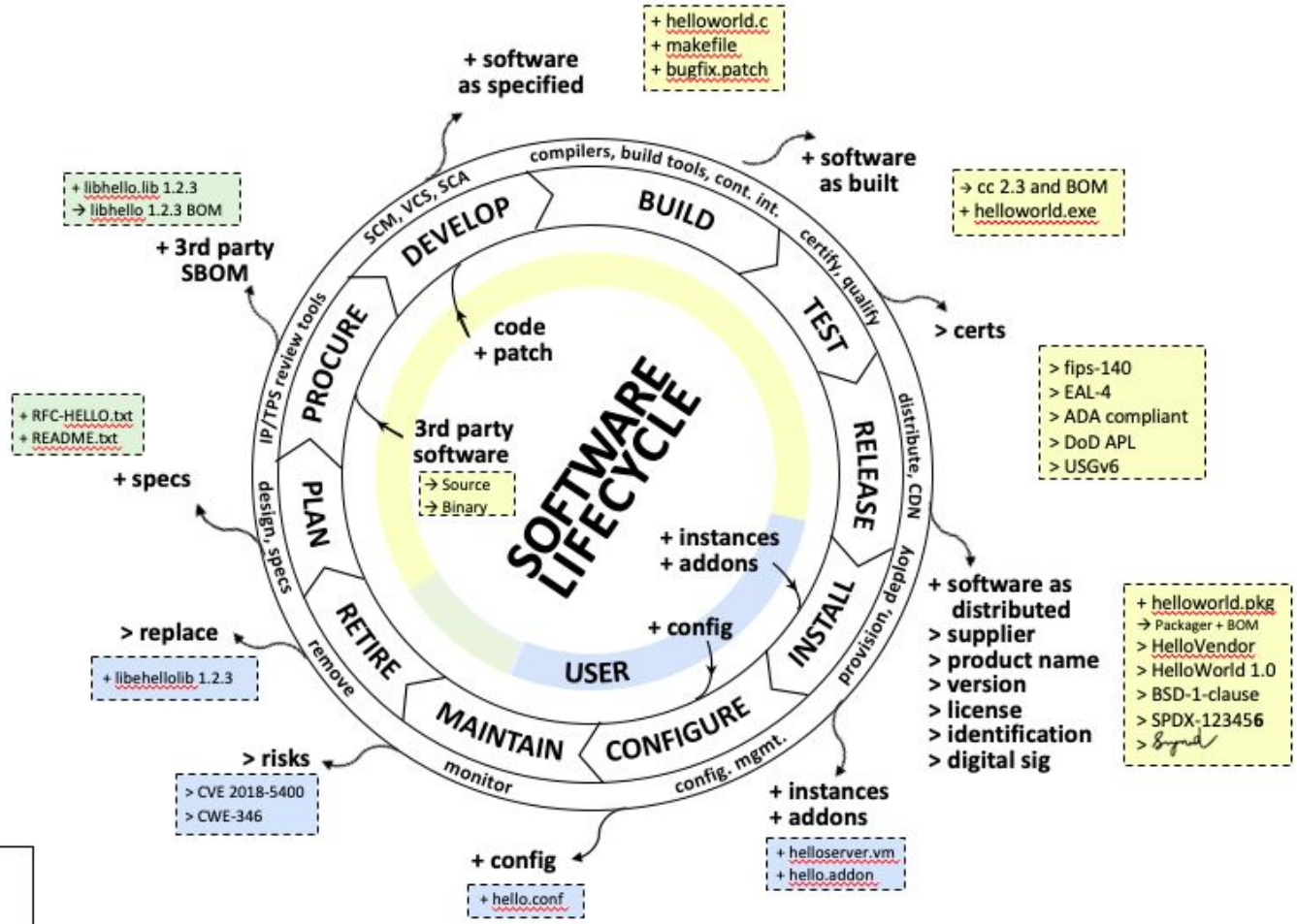
ISO/IEC 5962:2021 extends Minimum SBOM

- SPDX is an open ISO/IEC standard! Available for free.
- Able to represent SBOMs at the package level and track back to the source files and snippets.
- Specification is freely available from ITTF site
- Future updates are live tracked at: https://spdx.github.io/spdx-spec and work on satisfying safety requirements is being included. Use cases and help is welcome.
- More information at spdx.dev



Source: https://www.iso.org/standard/81870.html accessed on 2021/11/19

Software Lifecycle & Bill of Materials Generation



LEGEND
+ material
> metadata
→ reference
example BOM fragment
supplier consumer other

Derived from : NTIA's [Survey of Existing SBOM Formats and Standards]

SBOM Type	Definition
Design	SBOM of intended design of included components (some of which may not exist) for a new software artifact.
Source	SBOM created directly from the development environment, source files, and included dependencies used to build an product artifact.
Build	SBOM generated as part of the process of building the software to create a releasable artifact (e.g., executable or package) from data such as source files, dependencies, built components, build process ephemeral data, and other SBOMs.
Analyzed	SBOM generated through analysis of artifacts (e.g., executables, packages, containers, and virtual machine images) after its build. Such analysis generally requires a variety of heuristics. In some contexts, this may also be referred to as a “3rd party” SBOM.
Deployed	SBOM provides an inventory of software that is present on a system. This may be an assembly of other SBOMs that combines analysis of configuration options, and examination of execution behavior in a (potentially simulated) deployment environment.
Runtime	SBOM generated through instrumenting the system running the software, to capture only components present in the system what is loaded and executing in memory, as well as external call-outs or dynamically loaded components. In some contexts, this may also be referred to as an “Instrumented” or “Dynamic” SBOM.

Schedule

Welcome to the SBOM devroom! *Introduction to the devroom*	Alexios Zavras, Kate Stewart, Adolfo García Veytia	09:00	09:05
Generating SBOM made easy with ORT	Thomas Steenbergen	09:05	09:30
Understanding and Managing the Dependency in SBOM with the New Feature of SW360	Kouki Hama	09:30	10:00
Automated SBoM generation with OpenEmbedded and the Yocto Project *A case study of automated SBoM generation in meta build systems*	Joshua Watt	10:00	10:30
SBOM with the Yocto Project for Automotive Grade Linux *Intro and lessons learned*	Jan-Simon Möller	10:30	10:45
Hermine: converting SBOMS into legal obligations	Nicolas Toussaint, Camille Moulin	10:45	11:15
A standard BOM for Siemens	Thomas Graf, Thomas Jensen, Alexander Gschrei	11:15	11:45
FOSSology and SPDX *How FOSSology works with SPDX*	Gaurav Mishra, Mohammed Shaheem Azmal Madanapalli	11:45	12:00
Build recorder: a system to capture detailed information	Alexios Zavras, Fotios Valasiadis	12:00	12:30
Discussion on SBOM contents	Arnout Vandecappelle	12:30	13:00

Schedule

Using SPDX for functional safety	Nicole Pappler	13:00	13:30
REUSE *The gold standard of communicating licensing and copyright information*	Linus Sehn	13:30	13:45
A complete compliance toolchain for Yocto projects *(even very large ones, yes)*	Carlo Piana, Alberto Pianon	13:45	14:00
In SBOMs We Trust: How Accurate, Complete, and Actionable Are They?	Joseph Hejderup, Henrik Plate	14:00	14:30
The 7 key ingredients of a great SBOM *Ensuring your SBOM includes enough data to be actionable*	Adolfo García Veytia	14:30	15:00
Panel discussion: SBOM content, usefulness, and caveats	Bradley M. Kuhn, Alexios Zavras, Anthony Harrison, Julian Coccia, Paul Novarese	15:00	16:30
General Q&A on SBOMs	Kate Stewart, Adolfo García Veytia	16:30	16:55
SBOM devroom closing	Alexios Zavras, Kate Stewart, Adolfo García Veytia	16:55	17:00

Other Questions?

Please feel free to reach out to the devroom coordinators:

Alexios Zavras

Adolfo Garcia Vetyia

Kate Stewart

(we’re the ones in the Blue Shirts. :-))