ETSI TR 104 034 V1.1.1 (2025-05)

Cyber Security (CYBER);
Software Bill of Materials (SBOM) Compendium

Reference
DTR/CYBER-00124

Keywords
cybersecurity, resilience

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI IPR online database.

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Cyber Security (CYBER).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

The SBOM ecosystem is a broad, diverse umbrella of related supply chain management mechanisms and mandates that include software and processes in various manifestations, including AI, cryptographic algorithms, and hardware. The perfection of BOMs is also one of the "grand challenges" in the broad field of information processing. Although it emerged early in the evolution of cyber security to deal with known vulnerabilities, it has achieved significant contemporary prominence due to the considerable complexity of both software and hardware, global open provisioning, the enabled vulnerabilities, high-profile information security incidents, and diverse regulatory and procurement mandates.

Pioneering work and leadership in several cybersecurity forums over the past decade have led to widespread global awareness, leadership in key communities (healthcare, cloud, finance, defence, and consumer), the emergence of several prominent standardized platforms and tools [i.34]. However, there is a lack of measurable data about the data, several significant challenges remain to fully enable the use of SBOM interoperability and end-to-end integration at scale. Software transparency does not solve all supply chain concerns, but it is necessary for all scalable solutions.

Introduction

A "Software Bill of Materials" (SBOM) has emerged as a key building block in software security and software supply chain risk management. An SBOM is a nested inventory, a list of ingredients or nested inventory that make up software components designed to prevent software supply chain attacks that compromise software through cyber attacks, insider threats, or other malign activities at any stage throughout its entire lifecycle [i.34]. SBOM exists at the intersection of multiple cybersecurity disciplines including the secure development process, risk management and vulnerability management across the multiple processes of producing, choosing, operating, and retiring software.

The recognition of software and device supply chain threats was identified by NSA senior scientist Bernard Peters and RAND's Willis Ware at the first major cybersecurity conference at Atlantic City in 1967 and been pursued as an essential component of cyber risk management since that time. Systemic mitigation of these threats was manifested in the 1980s as part of NSA's Secure Data Network System (SDNS) initiative where software code was treated as managed objects with associated identifier and trust mechanisms that were instantiated in an array of global standards and activities. However, it has only been since about 2014 that explicit SBOM specifications have emerged that allow providers to make sure open-source and third-party software components are up to date and respond quickly to new vulnerabilities and enable users to SBOMs to perform vulnerability or license analysis and to evaluate and manage risk in a product. A SBOM declares the inventory of components used to build a software artifact, including any open source and proprietary software components [i.14]. It is the software analogue to the hardware BOM used as part of any assembly or manufacturing supply chain management.

1 Scope

The present document identifies Software Bill of Material challenges, types, existing specifications, existing tools and cybersecurity uses, including compliance obligations (e.g. regulatory or contractual). The present document identifies gaps and makes recommendations for further work.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

[i.1] Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) No 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act) (Text with EEA relevance).

[i.2] UK NCSC: "How to assess and gain confidence in your supply chain cyber security".

[i.3] Netherlands NCSC: "Software Bill of Materials and Cybersecurity".

[i.4] Canada CFDIR: "Recommendations to Improve the Resilience of Canada's Digital Supply Chain".

[i.5] BSI Technical Guideline oneM2M TR-03183: "Cyber Resilience Requirements for Manufacturers and Products - Part 2: Software Bill of Materials (SBOM)".

[i.6] Regulation (EU) 2024/1781 of the European Parliament and of the Council of 13 June 2024 establishing a framework for the setting of ecodesign requirements for sustainable products, amending Directive (EU) 2020/1828 and Regulation (EU) 2023/1542 and repealing Directive 2009/125/EC (Text with EEA relevance).

[i.7] Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act) (Text with EEA relevance).

[i.8] Directive (EU) 2024/2853 of the European Parliament and of the Council of 23 October 2024 on liability for defective products and repealing Council Directive 85/374/EEC (Text with EEA relevance).

[i.9] Regulation (EU) 2022/2554 of the European Parliament and of the Council of 14 December 2022 on digital operational resilience for the financial sector and amending Regulations (EC) No 1060/2009, (EU) No 648/2012, (EU) No 600/2014, (EU) No 909/2014 and (EU) 2016/1011 (Text with EEA relevance).

[i.10] Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive) (Text with EEA relevance).

[i.11] Commission Implementing Regulation (EU) 2024/2690 of 17 October 2024 laying down rules for the application of Directive (EU) 2022/2555 as regards technical and methodological requirements of cybersecurity risk-management measures and further specification of the cases in which an incident is considered to be significant with regard to DNS service providers, TLD name registries, cloud computing service providers, data centre service providers, content delivery network providers, managed service providers, managed security service providers, providers of online market places, of online search engines and of social networking services platforms, and trust service providers (Text with EEA relevance).

[i.12] Commission Implementing Regulation (EU) 2024/482 of 31 January 2024 laying down rules for the application of Regulation (EU) 2019/881 of the European Parliament and of the Council as regards the adoption of the European Common Criteria-based cybersecurity certification scheme (EUCC).

[i.13] National Security Agency: "Recommendations for Software Bill of Materials (SBOM) Management", PP-23-4432, January 2024, Version 1.1.

[i.14] NTIA: "The Minimum Elements For a Software Bill of Materials (SBOM)", July 12, 2021.

[i.15] NTIA: "Framing Software Component Transparency: Establishing a Common Software Bill of Materials (SBOM)".

[i.16] CISA: "SBOM FAQ".

[i.17] CISA: "Software Transparency in SaaS Environments".

[i.18] CISA: "SBOM Sharing Primer".

[i.19] CISA: "SBOM Sharing Roles and Considerations".

[i.20] CISA: "Types of Software Bill of Material (SBOM) Documents".

[i.21] The Linux® Foundation: "System Package Data Exchange (SPDX®)".

NOTE: Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

[i.22] OWASP Foundation: "CycloneDX: Authoritative Guide to Attestations".

[i.23] CISA: "Securing the Software Supply Chain: Recommended Practices Guide for Suppliers".

[i.24] CISA: "Securing the Software Supply Chain: Recommended Practices for Software Bill of Materials Consumption".

[i.25] CISA: "Software Identification Ecosystem Option Analysis", October 2023.

[i.26] Ministry of Economy, Trade and Industry (METI): "Guide of Introduction of Software Bill of Materials (SBOM) for Software Management".

[i.27] India CERT: "Technical Guidelines on Software Bill of Materials (SBOM)".

[i.28] India Security and Exchange Board of India (SEBI): "Cybersecurity and Cyber Resilience Framework (CSCRF) for SEBI Regulated Entities (REs)".

[i.29] U.S. FDA: "Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket Submissions".

[i.30] U.S. Executive Order 14028: "Improving the nation's Cybersecurity".

[i.31] American Council for Technology-Industry Advisory Council (ACT-IAC): "Software Bill of Materials (SBOM) FAQ & Myth Buster: SBOM Guidance for the Acquisition Community".

[i.32] ENISA: "Cyber Resilience Act implementation via EUCC and its applicable technical elements".

[i.33] Recommendation ITU-T X.1250: "Baseline capabilities for enhanced global identity management and interoperability".

[i.34] ETSI: "The State of SBOM", Security Conference 2024, Friedman.

[i.35] US Department of Energy: "Software Bill of Materials (SBOM) Sharing Lifecycle Report".

[i.36] draft-ietf-scitt-architecture-12: "An Architecture for Trustworthy and Transparent Digital Supply Chains".

[i.37] AFIPS: "Conference Proceedings", Vol. 30, 1967 Spring Joint Computer Conference.

[i.38] Tater and Kerut: "The Secure Data Network System: An Overview", NCSC 10th Proceedings, pp. 150-152.

[i.39] OID-base: "OID Repository".

[i.40] IETF RFC 9393: "Concise Software Identification Tags".

[i.41] GitHub Docs: "About the dependency graph".

[i.42] Netherlands Nationaal Cyber Security Centrum (NCSC): "Software Bill of Materials Starter Guide".

[i.43] U.S. Federal Register: "Securing the Information and Communications Technology and Services Supply Chain: Connected Vehicles".

[i.44] IETF RFC 9472: "A YANG Data Model for Reporting Software Bills of Materials (SBOMs) and Vulnerability Information".

[i.45] Center for Internet Security: "CIS-CAT Pro".

[i.46] ETSI TR 103 305-1: "Cyber Security (CYBER); Critical Security Controls for Effective Cyber Defence; Part 1: The Critical Security Controls".

[i.47] OSV: "A distributed vulnerability database for Open Source".

[i.48] GitHub: GitHub Advisory Database.

[i.49] OpenSSF: SLSA Community.

[i.50] OpenSSF: "Graph for Understanding Artifact Composition (GUAC)".

[i.51] Google SAIF (Security AI Framework): "Securing the AI Software Supply Chain".

[i.52] OpenSSF: "Secure Supply Chain Consumption Framework".

[i.53] OpenSSF: bomctl.

[i.54] OpenSSF: protobom.

[i.55] OASIS: "Common Security Advisory Framework (CSAF)".

[i.56] OpenSSF: OpenVEX.

[i.57] OWASP: CycloneDX.

[i.58] OWASP: "Vulnerability Exploitability eXchange (VEX)".

[i.59] OWASP: "Authoritative Guide to SBOM". .

[i.60] ETSI: "SBOMs, Asset Management and Vulnerability Management, a vendor's view", Security Conference 2024, Ambrosini.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

Artificial Intelligence Bill of Materials (AI BOM): formal record containing the details of software components used in an artificial intelligence software system

Cryptographic Bill of Materials (CBOM): formal record containing the details of cryptographic software components used in a software system

SBOM access: control mechanisms used by the author or provider to regulate who can view or use an SBOM

SBOM author: creator of an SBOM

SBOM consumer: receiver of the transferred SBOM

EXAMPLE: Third parties, authors, integrators, distributors, and end users.

SBOM discovery: mechanism used by the consumer to know the SBOM exists and how to access it

SBOM distributor: receiver of SBOMs to share them with SBOM Consumers or other SBOM Distributors

NOTE: The role is introduced to capture the role of organizations that neither produce SBOMs nor make use of SBOM data.

SBOM enrichment: activities that leverage an SBOM to create a new product, which may include the antecedent SBOM

EXAMPLE: Author, consumer, provider, or other third party.

SBOM ingestion: process of parsing and loading data from a Software Bill of Materials (SBOM) into an enterprise's workflows or systems of record

Software Bill of Materials (SBOM): formal record containing the details and supply chain relationships of various components used in building software

NOTE: As defined in [i.15].

software supply chain attacks: compromising software through cyber attacks, insider threats or other malign activities at any stage throughout its entire lifecycle

NOTE: As defined in [i.17].

sophistication: relative amount of time, resources, subject-matter expertise, effort, and access to tooling needed to implement a phase of the SBOM sharing lifecycle, and can either be low, medium, or high

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AI BOM	Artificial Intelligence Bill of Materials
BOM	Bill of Materials
BOMctl	Bill of Materials command line interface (OpenSSF)
BSI	Bundesamt für Sicherheit in der Informationstechnik
CBOM	Cryptographic Bill of Materials
CDXA	CycloneDX Attestations

CISA	Cybersecurity and Infrastructure Security Agency (US)
CoSWID	Concise SWID
CRA	Cyber Resilience Act
CSAF	Common Security Advisory Framework
CVE	Common Vulnerabilities and Exposures
CVRF	Common Vulnerability Reporting Framework
CVSS	Common Vulnerability Scoring System
GHSA	GitHub Advisory Database
GUAC	Graph for Understanding Artifact Composition
HBOM	Hardware Bill of Materials
MBOM	Manufacturing Bill of Materials
ML-BOM	Machine Learning Bill of Materials
NCSC	Nationaal Cyber Security Centrum (NL)
NCSC	National Cyber Security Centre (FI, UK)
NSA	National Security Agency (US)
NVD	National Vulnerability Database (US)
OBOM	Operations Bill of Materials
OpenSSF	Open Source Security Foundation
OSS	Open Source Software
OSV	Open Source Vulnerability
OWASP	Open Worldwide Application Security Project
S2C2F	Secure Supply Chain Consumption Framework
SaaSBOM	Software-as-a-Service Bill of Materials
SBOM	Software Bill of Materials
SDNS	Secure Data Network System
SLSA	Safeguarding artifact integrity across any software supply chain
SPDX	Software Package Data eXchange
SWID	SoftWare IDentification
VDR	Vulnerability Disclosure Reports
VEX	Vulnerability Exploitability eXchange
xBOM	eXtensible Bill of Materials

4 SBOM ecosystem

4.1 History and venues

Common schemes for structured Bill of Material information have existed for almost as long as human societies have been created assembled objects and had use in manufacturing and commerce. The need in conjunction with software security appears to have emerged in national security communities in the 1960s as digital data networks became feasible and complex computer software was being transferred across computer-based information systems.

Significant initiatives subsequently emerged - especially in the 1970s and 80s as the networks and information systems become increasingly complex and openly available to diverse communities that created multiple persistent vulnerabilities [i.37]. NSA's Secure Data Network System (SDNS) in the 1980s was perhaps the most ambitious initiative [i.38]. It began driving international standards and arrangements to tag and discover digital objects, including code and processes, using hierarchical nested identifiers that is still in widespread use today [i.39]. However, as the ICT sector became ever more complex, global, open, and dynamic, vulnerabilities and incidents rose exponentially. This led to the emergence of DevSecOps and major ICT industry led initiatives.

Increasingly the Open Source community has scaled up SBOM and AI BOM standards, database, and tools activity in open source venues - especially OpenSSF and GitHub - as very large sets of developers engage in the development of SBOM products and support services [i.41].

4.2 Core specifications

4.2.0 Minimum Elements

The core specification for SBOM is the Minimal Elements standard published by the U.S. Dept. of Commerce that emerged from an industry collaboration initiative [i.14]. The Minimum elements are enumerated in Table 4.2.0-1, below.

Table 4.2.0-1: SBOM minimum elements

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.

The baseline component information of an SBOM is enumerated in Table 4.2.0-2, below, and described in detail in the Minimum Elements standard.

Table 4.2.0-2: SBOM 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.

Other capabilities specified and described in the Minimum Elements specification are:

- automation support;
- practices and processes (frequency, depth, known unknowns, distribution and delivery, access control, accommodation of mistakes);

- enabling broader SBOM use cases by component hashes, lifecycle phases, other component relationships and license information;
- facilitating cloud-based software and software-as-a-service;
- SBOM integrity and authenticity;
- vulnerability and SBOM relationships;
- vulnerability and exploitability in dependencies;
- legacy software and binary analysis;
- flexibility and uniformity in information.

The Minimum Elements specification underscores that SBOM is an emerging technology, and suppliers are still learning how to share this data with their customers. Many suppliers already have trusted channels with their downstream users, including for software updates and support, although not all of these are automated or flexible.

A companion specification to the Minimum Elements document treats further details of component transparency to bring about a common SBOM [i.15]. As the document notes, to scale the SBOM model globally, it is necessary to address the difficult problem of universally identifying and defining certain aspects of software components. Thus, a subsidiary goal was to select a core, baseline set of attributes necessary to identify components with sufficient relative uniqueness. Another goal was to capture SBOM applications and consider what additional, optional attributes and external elements might be needed beyond the baseline set.

The CISA maintained FAQ provides pointers concerning the creation of SBOMs, the benefits, common misconceptions, and distribution or sharing [i.16]. It currently sees four essential SBOM requirements [i.34]:

1) attest that an SBOM exists;
2) the SBOM can be accessed on demand;
3) an SBOM is delivered to the customer in advance of acquisition; and
4) an updated SBOM is shared with the customer.

Four needs are articulated:

a) demonstrating that suppliers know what they have;
b) enable the ability to manage ad hoc responses;
c) integrate the SBOM into existing and emerging security functions including cybersecurity supply chain risk management and operational vulnerability management (DevDecOps); and
d) holistic risk awareness from aggregate data [i.34].

4.2.1 SBOM expression types

An SBOM may contain different forms of the minimum information sourced from different product artifacts. Given the disparate ways SBOM data can be collected, tool outputs may vary and provide value in different use cases. The common types of SBOMs that tools may create today, along with the data typically presented for each type of SBOM is shown in Table 4.2.1-1, below [i.20]. An SBOM document may combine information for multiple SBOM types. The referenced specification further elaborates on the benefits and limitations of these SBOM typed.

Table 4.2.1-1: SBOM Type Definition and Composition [i.20]

SBOM Type	Definition	Data Description
Design	SBOM of intended, planned software project or product with included components (some of which may not yet exist) for a new software artifact.	Typically derived from a design specification, RFP, or initial concept.
Source	SBOM created directly from the development environment, source files, and included dependencies used to build an product artifact.	Typically generated from software composition analysis (SCA) tooling, with manual clarifications.
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.	Typically generated as part of a build process. May consist of integrated intermediate Build and Source SBOMs for a final release artifact SBOM.
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.	Typically generated through analysis of artifacts by 3rd party tooling.
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.	Typically generated by recording the SBOMs and configuration information of artifacts that have been installed on systems.
Runtime	SBOM generated through instrumenting the system running the software, to capture only components present in the system, 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.	Typically generated from tooling interacting with a system to record the artifacts present in a running environment and/or that have been executed.

4.2.2 SBOM interoperability and sharing properties

A Software Bill of Materials exists to enable sharing of structured information among different actors in the software supply chain. A focus on the processes and mechanisms of sharing is provided in a pair of documents - the SBOM Sharing Primer [i.18] and SBOM Sharing Roles and Considerations [i.19].

The Sharing Primer defines and describes sharing lifecycle phases and sophistications [i.18]. It provides examples of SBOM sharing processes in use today: for proprietary software shared via Email or a vendor portal with and without pre-vetting, open-source software shared via tooling, OSS shared via a platform, and proprietary software shared along a supply chain. The primer concludes that key areas for continued community collaboration include consistency in SBOM formats, identifiers, and storage locations as well as transport protocols and federated services. It further notes that maturation of SBOM sharing practices will be crucial to realizing the full benefits of software transparency across the entire software supply chain.

The SBOM Sharing Roles and Considerations publication describes the sharing lifecycle phases from the perspectives of three SBOM specific roles: author, consumer and distributor [i.19]. Three phases are described and roles identified: discovery, access and transport.

4.2.3 Additional SBOM implementation properties

In addition to interoperability and sharing properties, the effective implementation of SBOMs at scale requires a number of properties that are ideally reflected in the standards and tools employed enumerated in Table 4.2.3-1, below [i.23], [i.24] and [i.35].

Table 4.2.3-1: SBOM additional implementation properties

Property	Description
Flexibility	The considerable diversity among the suppliers and users of digital objects as well as actors and communities that provide and consume SBOMs, as well as the objects themselves and the manner in which information is captured and exchanged, necessitates the use of expressions and tools that accommodate that diversity.
Enrichment	Enrichment activities and information leverage an SBOM to create a new product, which may include the antecedent SBOM. This may be done by an author, consumer, provider, or other third party. Acquisition and provisioning of enrichment is an important SBOM property [i.35].
Verifiable trust/risk levels	Suppliers should provide a mechanism for verifying software release integrity by digitally signing the code throughout the software lifecycle. Digitally signed code enables recipients to positively verify and trust the provenance and integrity of the code. Some implementations may require a Zero-Trust Architecture [i.35].
Roles	Communication between these three different roles and among cybersecurity professionals that may facilitate increased resiliency and security in the software supply chain process is necessary. These include developers, suppliers, and customers (or the organization acquiring a software product).
Formats	SBOMs can be represented in different formats - the most common being XML, JSON, and YAM. ASN.1 was once used, but does not appear to be in use. RDF/XML, XLSX, tag-value, and protobuf are also used. Software identifiers can include any globally unique and discoverable format [i.25], although IETF RFC 9393 [i.40] CoSWID is encouraged. See also SCITT [i.36].
Dependencies and depth	The complexity of software dependency is frequently depicted in graphic form. There are two forms of dependency: 1) dependencies, the ecosystems and packages it depends on, and 2) dependents, the repositories and packages that depend on it [i.22]. The depth level relates to the recursive tiers depicted [i.34] and [i.42].
Lifecycle/end of life	Throughout their lifecycle, software applications often undergo changes, such as bug fixes, security patches, new features, etc. *"Every time the software is altered, the software supplier must also supply a new SBOM that includes the changes."* In theory, the software configuration carried out by the software customer can also be added to the SBOM. In this way, it is possible to distinguish between different configurations which may also have different runtime dependencies [i.42]. At some point in time, software support and patches may cease - known as end-of-life.

It is useful to examine the principal SBOM standard-based tools to understand how all of the SBOM/BOM properties in the NTIA/CISA requirements specification that have being widely required in regulatory mandates and procurement specifications. Among the SBOM tool implementations, the Linux Foundation's Software Package Data Exchange (SPDX) describes its functionalities as shown in Figure 4.2.3-1, below.

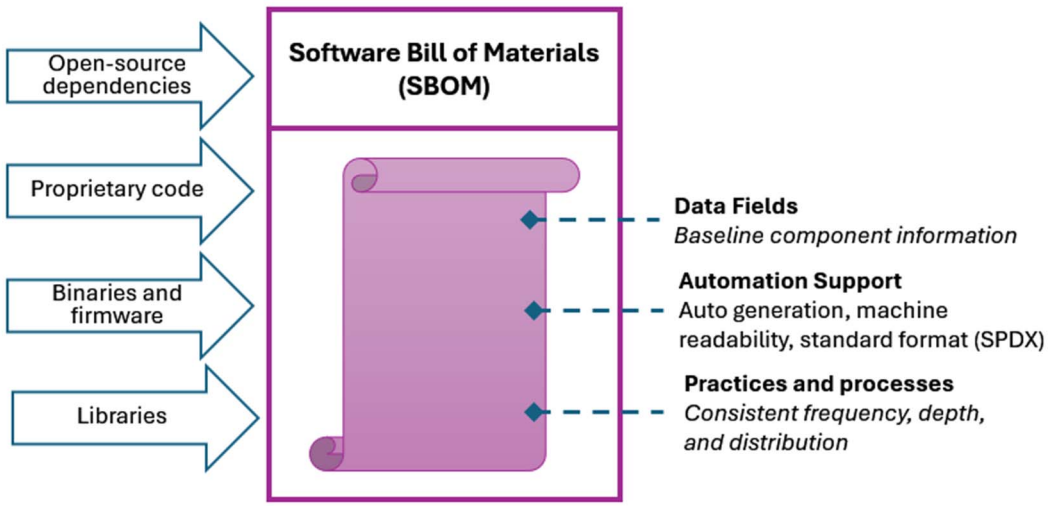

Figure 4.2.3-1: SPDX SBOM functionality model [i.21]

4.2.4 SBOM categories

Four broad categories of SBOMs exist.

Table 4.2.4-1: SBOM categories

Category	Description
Proprietary	Proprietary SBOMs can be software bill of materials for a commercial proprietary product, and/or software bill of materials prepared by a commercial concern.
Open source	Open source SBOMs can be software bill of materials for an open source non-commercial proprietary product, and/or software bill of materials prepared by an open-source non-commercial concern.
Non-SaaS	Non-SaaS SBOMs are software bill of materials for products that does not reside on an online server.
SaaS	SaaS SBOMs are software bill of materials for products that reside on an online server [i.17].

4.3 SBOM information exchange platforms

Numerous generic as well as specialized SBOM information exchange platforms have emerged.

Table 4.3-1: SBOM tools

Tool	Description
Cyclone DX	OWASP Foundation CycloneDX is a full-stack Bill of Materials (BOM) standard that provides advanced supply chain capabilities for cyber risk reduction [i.57]. The specification supports: • Software Bill of Materials (SBOM) • Software-as-a-Service Bill of Materials (SaaSBOM) • Hardware Bill of Materials (HBOM) • Machine Learning Bill of Materials (ML-BOM) • Cryptography Bill of Materials (CBOM) • Manufacturing Bill of Materials (MBOM) • Operations Bill of Materials (OBOM) • Vulnerability Disclosure Reports (VDR) • Vulnerability Exploitability eXchange (VEX) • CycloneDX Attestations (CDXA)
SPDX	Software Package Data Exchange (SPDX) is an open standard (or format) for communicating software Bill of Materials (SBOM) information including components, licenses, copyrights, and security references. Using a standardized format for presenting this information ensures that it is consistent across industries and companies, which helps reduce reformatting efforts, makes it easier to share information, and streamlines compliance activities [i.21].

Tool	Description
VEX	Vulnerability Exploitability eXchange (VEX) is a form of a security advisory where the goal is to communicate the exploitability of components with known vulnerabilities in the context of the product in which they are used. Often products are not affected by a vulnerability simply by including an otherwise vulnerable component. VEX allows software vendors and other parties to communicate the exploitability status of vulnerabilities, providing clarity on the vulnerabilities that pose risk and the ones that do not [i.58]. VEX is a useful capability to operationalize SBOM. "VEX information communicates the vulnerability details, exploitability, and detailed analysis, and informs software consumers on actions the vendor has taken, or which the consumer must take, to reduce risk" [i.59]. This capability also highlights the limitation of SBOM for the Vulnerability Management use case, since VEX is necessary to confirm the status of a potential vulnerability identified from an SBOM [i.60]. VEX has also been implemented in non-SBOM standard such as CSAF (with the VEX profile). Generally, a manufacturer would not update an SBOM each time there is a VEX, but provide the VEX information separately, most likely through other channels than an SBOM. It is possible to provide VEX or CSAF/VEX data without referring to any SBOM, since a VEX object can provide a reference to a product version directly. Therefore, asset owners can focus on asset management and requiring their suppliers to provide CSAF/VEX associated to product versions, instead of managing SBOMs. An illustration of component management and asset management in relation to vulnerability management through the supply chain can be found in [i.60].
OpenVEX	OpenVEX is a simplified implementation of the Vulnerability Exploitability Exchange (VEX for short) that is designed to be minimal, compliant, interoperable, and embeddable and maintained on GitHub by OpenSSF [i.56].
CSAF	Common Security Advisory Framework (CSAF) is a language to exchange Security Advisories by OASIS. It plays a crucial role in the cybersecurity arena since it allows stakeholders to automate the creation and consumption of security vulnerability information and remediation [i.55].
Protobom	OpenSSF Protobom is a protocol buffers representation of SBOM data able to ingest documents in modern SPDX and CycloneDX versions without loss. It has an accompanying Go library generated from the protocol buffers definition that also implements ingesters for those formats. Standard SBOMs are read by a reader using parsers that understand the common formats. Parsers create a neutral protobom from data read from CycloneDX or SPDX documents [i.54].
BOMctl	OpenSSF BOMctl is format-agnostic Software Bill of Materials (SBOM) tooling, which is intended to bridge the gap between SBOM generation and SBOM analysis tools. It focuses on supporting more complex SBOM operations by being opinionated on only supporting the minimum fields or other fields supported by protobom. It is intended to help developers who need to manipulate SBOMs at the CLI or within a workflow. Example operations would be merging in project specific SBOM data that would not be detected by a SBOM generation tool [i.53].
OSV	The Open-Source vulnerability reporting community maintains OSV as both an expression standard and database of software vulnerabilities used to maintain SBOMs [i.47]. It includes GitHub code vulnerabilities that use GHSA (GitHub Advisory Database) identifiers that can be used for SBOM expressions [i.48].
GUAC	Graph for Understanding Artifact Composition (GUAC) is a tool that ingests software metadata like SBOMs and maps out relationships between software [i.50]. It thereby enables knowing how one piece of software affects another including related vulnerabilities. GUAC is being used for AI BOMs by the Google SAIF [i.51].
SLSA	Safeguarding artifact integrity across any software supply chain (SLSA) is a specification for describing and incrementally improving supply chain security, established by open-source industry consensus. It is organized into a series of levels that describe increasing security guarantees [i.49].
S2C2F	Secure Supply Chain Consumption Framework is a guide which outlines and defines how to securely consume Open Source Software (OSS) dependencies into the developer's workflow [i.52].

4.4 SBOM implementations

A number of tailored Bill of Material implementations have emerged [i.22].

Table 4.4-1: SBOM implementations

Implementations	Description
SBOM	Software Bill of Materials inventory software components and services and the dependency relationships between them.
HBOM	HBOMs describe the many types of components, including hardware devices, consumer electronics, IoT, ICS, and other types of embedded devices that may be referenced by SBOMs.
AI BOM	Protobom is a protocol buffers representation of SBOM data able to ingest documents in modern SPDX and CycloneDX versions without loss. It has an accompanying Go library generated from the protocol buffers definition that also implements ingesters for those formats. Standard SBOMs are read by a reader using parsers that understand the common formats. Parsers create a neutral protobom from data read from CycloneDX or SPDX documents.
ML-BOM	ML-BOMs provide transparency for machine learning models and datasets, which provide visibility into possible security, privacy, safety, and ethical considerations.
CBOM	A Cryptography Bill of Materials (CBOM) describes cryptographic assets and their dependencies. Discovering, managing, and reporting on cryptographic assets is necessary as the first step on the migration journey to quantum-safe systems and applications. Cryptography is typically buried deep within components used to compose and build systems and applications. As part of an agile cryptographic approach, organizations should seek to understand what cryptographic assets they are using and facilitate the assessment of the risk posture to provide a starting point for mitigation.
SAASBOM	SaaSBOMs complement Infrastructure-as-Code (IaC) by providing a logical representation of a complex system, complete with inventory of all services, their reliance on other services, endpoint URLs, data classifications, and the directional flow of data between services. Optionally, SaaSBOMs may also include the software components that make up each service.
xBOM	eXtensible BOMs are generic BOM formats that can be adapted to a broad array of supply chains for risk reduction.
MBOM	Manufacturing BOMs describe declared and observed formulations for reproducibility throughout the product lifecycle of components and service.

4.5 SBOM enhancements

4.5.1 SBOM and NVD content

It is also possible for industry groups and sector ISACs to play roles in SBOM enrichment processes that can be captured in SBOM expressions. The IETF, for example, standardized a YANG data model that allows CVRF reports to be added as vulnerability enrichment [i.44].

4.5.2 SBOM and OSV content

The OpenSSF SBOM community has developed an open-source vulnerability capability known as OSV [i.56]. OSV enables the implementation of an open source vulnerability database and a broader array of software code and associated identifiers, including especially for AI BOMs. Open-source tools such as GUAC have also developed to facilitate use [i.50]. GUAC provides directed, actionable insights into the security of a software supply chain.

4.5.3 SBOM Critical Security Control safeguards and mappings

The Critical Security Control Safeguards [i.46] are widely deployed worldwide to effect continuing cybersecurity capabilities in enterprises of all kinds and meet diverse obligations. A companion tool to maintain Safeguard implementations known as the CIS-CAT Pro Dashboard has been augmented to enable the production of SBOMs of software inventories deployed in enterprise infrastructure in JSON and XML [i.45].

4.6 SBOM enactments

4.6.1 Legislative provisions

Obligations and guidance have emerged in several venues [i.1] to [i.12]. The most prominent recent enactments that contain explicit SBOM requirements are the European Commission Cyber Resilience Act [i.1] including CRA implementation via EUCC [i.12] and [i.32], and Product Liability Act [i.8] instruments. Under the CRA, 1) the SBOM is used for identifying vulnerabilities, 2) manufacturers have no obligation to share the SBOM to third parties other than the Market Surveillance Authorities, and this only in specific circumstances. A key change in final CRA enactments clarifies that: a) the SBOM is used for identifying vulnerabilities, b) manufacturers have no obligation to share the SBOM to third parties other than the Market Surveillance Authorities, and this only in specific circumstances.

Numerous other countries have also published SBOM implementation guidance as an important means for enhancing risk managements, and include the UK NCSC [i.2], Netherlands NCSC [i.3], Canadian CFDIR [i.4], Germany BSI [i.5], U.S. NSA [i.13], Japan METI [i.26], India's CERT [i.27] and Securities Exchange Board [i.28], and the U.S. Food and Drug Administration for medical devices [i.29].

However, the provisions for sharing SBOM across the supply chain introduces various risks and breaks the responsibility boundaries between the supply chain stakeholders. As a result, SBOMs may be excluded from NIS2 requirements.

4.6.2 Procurement and contract

The use of procurement and contract requirements in increasingly being used as a non-regulatory means of implementing SBOM requirements [i.30] and [i.31]. Contracts are part of supplier-customer negotiation. In addition, not all customers in downstream will require SBOM from their upstream suppliers because of the potential liability associated with SBOM information leak, and many suppliers will back-off from SBOM sharing requirements or demand strict measures, such as liability, for the protection of intellectual property.

5 BOM Challenges and further work

5.1 Unique identifiers with global discovery capability

The development of globally unique identifiers for computer code and processes - referred to as digital objects - that can be globally resolved to persistent information has remained an identity management challenge over the decades with only sub-optimal, pragmatic solutions such as UUIDs [i.33]. The problem set applies to all objects with a digital tag. Further articulation of definitive scalable solutions will assist BOM implementations.

5.2 Level of dependencies

Objects with digital tags may consist of recursive, tiered object assemblies that have considerable dependency depth. Depth adds to the value of BOMs at the same time as it adds potentially considerable complexity and the inability to fully express the dependency relationships and exponentially increases the process dynamics when changes occur.

Regarding complexity, neither the tools nor the industry is able to handle in-depth dependency declarations in an SBOM. Comments related to connected automobile BOM regulations in U.S. underscore the challenges [i.43].

Declaring several depths of components in an SBOM exposes the manufacturer's supply chain information to information leakage risks, which may help attackers to target the manufacturer's supply chain and can lead to risks to intellectual property. The concerns are reflected in CRA and the limitations on distribution of this information strictly for market surveillance purposes and anonymized supply chain dependency analysis [i.1].

5.3 Lifecycle and change dynamics

Software-based objects and services - especially those which exist entirely as online cloud-based processes - have potentially highly dynamic instantiations during their lifecycles. Widely used products often change monthly or even more frequently with new replaced code components. These SBOM forcing-functions get reflected back into SBOM expressions that could result in constant updating to reflect the new dependencies as well as the challenges of distributing the information at the global level. However, SBOM best practice dictates updates should only happen with a version update of the product. Failing to do so would contradict the principle of component status tracking, which is implicit in an SBOM approach.

As a best practice, a manufacturer in general needs to abandon practices such as silent patching and silent backporting (updating a component without updating its version) if it wants to implement an SBOM-based / component-based approach. This can represent a significant change of practice and costs for manufacturers that are not mature in their SDLC.

5.4 Trust mechanisms, data quality, replicability and access

Trust in the authenticity of digital objects and BOM constructions, the quality, comparability and replicability of information are all perennial challenges that have multiple technical solutions revolving around structured expression and cryptographic techniques including digital certificates and licenses. However, the authentication and auditing processes attendant to these techniques is a difficult, costly process. In the case of some digital objects - especially open-source objects - the information may not be available, and authentication and auditing may not be possible. The continuing evolution of these capabilities will enhance the value of BOMs.

5.5 Interoperability and harmonisation

All the elements of BOMs - identifiers, discovery, information capture and expressions - have on a global scale potentially varied implementation. The development and implementation BOM standards that provide for interoperability and facilitate harmonisation among the diverse solutions, enhances their utility. There is no universal SBOM.

Annex A: Bibliography

- Bellsoft: "U.S. and EU regulations are demanding a software bill of materials (SBOM)".
- Daniele Bifolco et al.: "On the Accuracy of GitHub's Dependency Graph", EASE '24.
- Finite State: "The SBOM is Coming, with Allan Friedman".
- ITU-T Work Item X.st-ssc: "Security threats of software supply chain".
- ITU-T Work Item X.ss-cti: "Guidelines on Security Capabilities for Software Supply Chain in the Telecommunications Industry".
- ITU-T Work Item X.ssc-sa: "Guidelines for software supply chain security audit".
- Slashdot: "Best Software Bill of Materials (SBOM) Tools in China".
- MITRE: "Data Normalization Challenges and Mitigations in Software Bill of Materials (SBOM) Processing".

History

Document history		
V1.1.1	May 2025	Publication