zvei
electrifying ideas

# Software Bill of Material (SBOM)

## Basics, assessments, prospects

The Software Bill of Material (SBOM) contains information about software components used and can be understood as an important building block of software supply management. It can help to achieve transparency about software components used and to improve cyber security along the supply chain. The topic of SBOM has therefore been discussed for several years as a possible instrument for improving cyber security and transparency within the (software) supply chain. In regulatory terms, the software bill of material is also increasingly finding its way into possible requirements for manufacturers or (re-)users of hardware or software products.

Before considering further use and implementation, a common understanding of SBOM should be developed by industry stakeholders as well as the regulator. This paper aims to contribute to this understanding from the perspective of the electrical and digital industries.

**Electro and Digital Industry Association**

# 1. Introduction

**Background**:

The topic of SBOM is not only gaining importance in the internal supply chain management of many companies but is also receiving greater attention through political decisions and regulatory developments. One is the US “Cyber Supply-Chain Management and Transparency Act” and Executive Order 14028 "Improving the Nation's Cybersecurity", which was published on 12 May 2021. The Executive Order defined the provision of an SBOM as a transparency requirement for the first time and required it for certain areas. As part of the Executive Order, a report by the responsible telecommunications and information authority NTIA[1] was published, which describes the minimum elements of an SBOM.

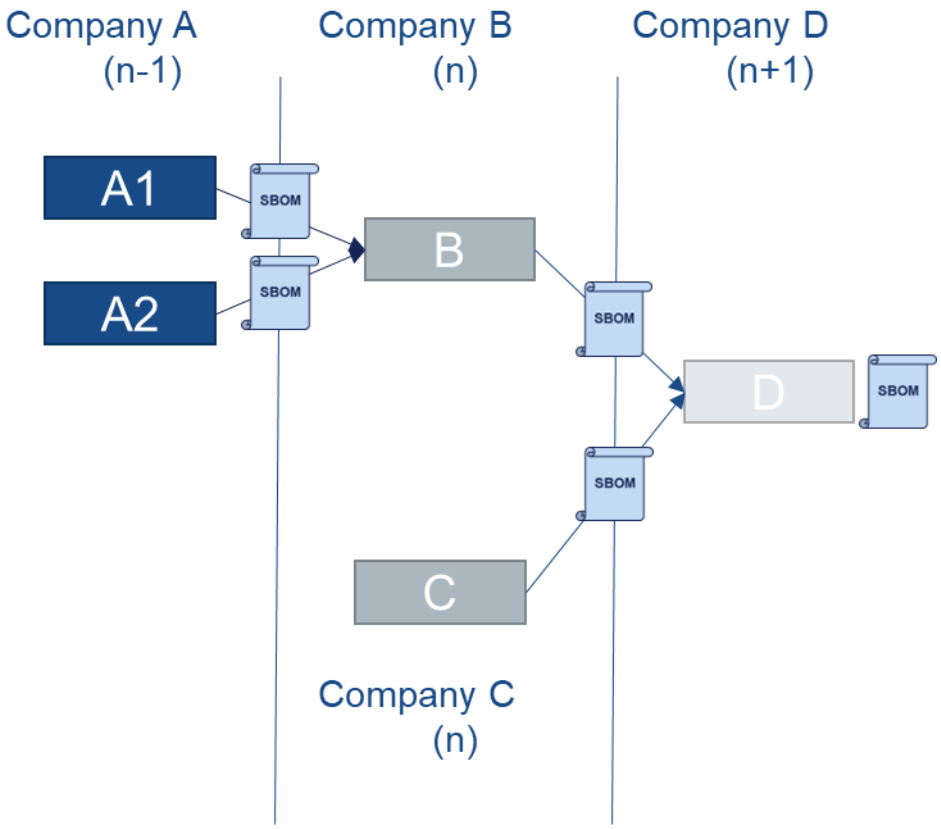
The concept gained further importance with the draft EU Cyber Resilience Act published on 15 September 2022 - the world's first regulation with horizontal product requirements for cyber security of hardware and software. There, SBOM is envisaged both as part of the manufacturer's "vulnerability handling requirements" and as a possible element of user information.

**Organisational problems**:

- Challenge of managing own assets in software development (Security Development Lifecycle, SDL)
- SBOM generation: availability of standardised formats and distribution mechanisms and building of content and tools
- Preparation and reprocessing of SBOM for already existing/sold software
- Clear, reproducible identification of software, especially in the case of open source
- Establishment and (step-by-step) improvement of processes for SBOM creation as well as gradual improvement (ramp-up) of the quality of an SBOM
- Availability of an SBOM; transparency (also with regard to 'outdated' software)
- How are corrections to an SBOM communicated?
- Pass SBOM 'enriching' throughout supply chain security chain; despite the limited scope of each BOM generation to levels "n-1", "n" & "n+1"
- Linking affectedness of vulnerabilities and SBOM (cf. chapter 6)

**Target**:
The aim of this paper is to develop a common understanding of SBOM and its meaningful use, to describe minimum elements and their scope, and to identify options for future development. However, the introduction of yet another burdensome requirement to provide information without added value for the implementing companies should be avoided at all costs. This document also aims to show the internal benefits of an SBOM for the manufacturer and the advantages and disadvantages for an external consumer of an SBOM. In doing so, the limitations of the scope of one's own SBOM (creation) should also be taken into account (cf. the levels "n-1, n, n+1 in the following chart). In addition, awareness should be raised that suppliers need to be identified, addressed and sensitised, and that a discussion should be initiated about which information should be provided to which customers.

[1] https://www.ntia.gov/files/ntia/publications/sbom_minimum_elements_report.pdf




Ideally, a SBOM should be generated for each step in the supply chain for the respective software product or component. By using upstream SBOM when generating SBOM downstream in the supply chain, all relevant information should come together at the end of the chain, despite the limited scope of each SBOM generation.

This is exemplified in the figure above: Company A provides products A1 and A2 including SBOM. Company B combines A1 and A2 into product B. The SBOM for product B therefore contains references to products A1 and A2. Company D carries out a further integration step and combines product B and product C into product D.

## 2. What is an SBOM?

A SBOM is a formal, ideally machine-readable data set that contains the traceable inventory of software components and their relationships that are contained in the product (the software or firmware). The SBOM is regarded as purely static information on the software of a particular product version:
An SBOM contains information about open source or proprietary software and could be widely available or restricted.[2] It can be assumed that an SBOM could have different levels of detail depending on its further use, internal or external (cf. chapter 9).

[2] Definition: https://www.cisa.gov/sbom

# 3. What are the elements of an SBOM?

The main purpose of an SBOM is to identify components and their relationships clearly and unambiguously to each other. To achieve this, a combination of basic component information is required. The attributes differ in how accurately they describe software components - certain attributes offer greater accuracy. The accuracy can be increased by having a larger number of attributes in an SBOM entry. The following minimum elements support many use cases. Additional attributes may be required for advanced use cases.

Minimum elements of an SBOM:

- Author of the **SBOM** (Author Name/Author of SBOM Data) of the organisation creating the SBOM.
- **Name of the software supplier** (Supplier Name) of the organisation providing the product.
- **Name of the software component** (Component Name), incl. product name of the product considered by the SBOM.
- **Version number/name** (Version of Component) of the component
- **Timestamp** of the creation of the SBOM
- **Unique identifier** of the component
- **Dependencies/relationships** with other software components ( Dependency Relationship; of direct third party components)

The minimum elements listed correspond to the "Baseline Component Information" of the NTIA, which already includes the most important elements.[3] In the opinion of the ZVEI, an SBOM should consist of at least these minimum elements.

Other possible elements of an SBOM:[4]

- Other Identifiers
- Version of SBOM
- Reference(s) to the source of vulnerability information (e.g. CSAF); reference to the possible sources/URL of vulnerability information.
- Licence Information (Copyright Information)
- Hash value of the software component (Component Hash)
- ...

**No** elements of an SBOM are:

- Vulnerability information, because those are dynamic, whereas the SBOM is static.
- Other information that cannot be transmitted by means of the elements described (machine-readable, if applicable). Such information should be provided externally by the manufacturer.
- Sensitive information, such as links to build systems or names (email addresses) of developers

[3] h ttps://www.ntia.gov/files/ntia/publications/sbom_minimum_elements_report.pdf
SBOM at a Glance (ntia.gov)

[4] The other elements listed are additional information which, from the point of view of the ZVEI, could be useful to take into account when preparing an SBOM.

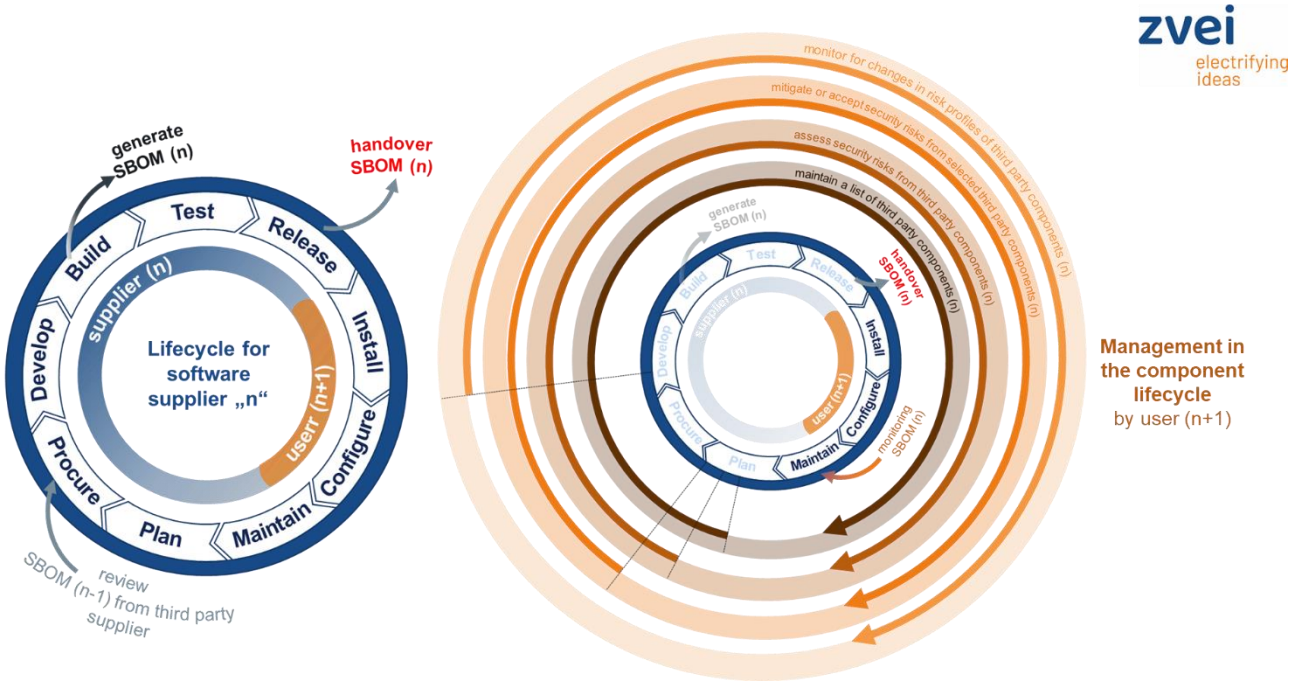
# 4. What and how can an SBOM be used for?

As a building block ...

… in software development, in the Security Development Lifecycle (SDL), (cf. "Supplier n" in graphic)
- for open source compliance management (management of licence information) & dependency management (end of life of libraries)
- in vulnerability management: linking and interconnecting SBOM and vulnerability management,

… in supply chain management (cf. "User n+1" in graphic)
- for identification and inventory of used components and sub-components; challenge change of SBOM in the product life cycle.

**Life cycles linked by SBOM:**




By means of SBOM, software components can be identified and inventoried on different levels. The respective individual SBOM helps as a data record in the communication between the respective supplier of an individual software component and the user of the corresponding component. The user usually uses this component together with other software components.

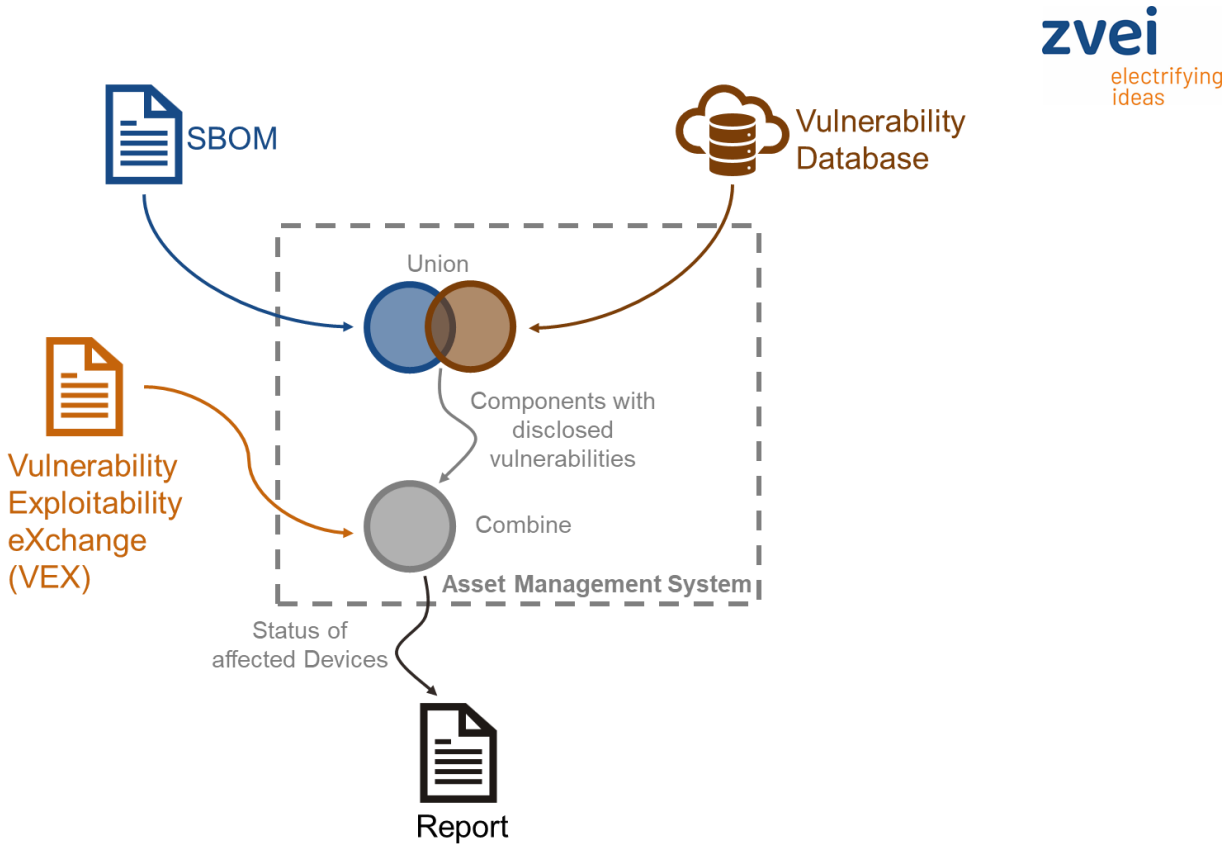
For example, the figure above shows on the left the creation of an SBOM from the perspective of the software supplier "n" on the life cycle of the software (component). On the other hand, the right-hand side, shows the use of this SBOM in the life cycle of the use of the components by the user "n+1" and steps in this life cycle that the user takes with the help of the SBOM.

# 5. Relationship between SBOM and vulnerability management

**Linking affectedness of vulnerabilities and SBOM**

SBOM remain static, except for the correction of content errors for the version of the product. A correction of content errors in the SBOM must not be confused with a dynamic enrichment or linkage with other information sources, e.g. VEX[5] (cf. graphic below). An SBOM does not contain any vulnerability information.

SBOM should also not be enriched with vulnerability information, as otherwise each new vulnerability information would be tied to a new creation and distribution of the SBOM of the corresponding product version. The SBOM should rather serve to identify the affectedness upon receipt of a vulnerability information.




The link will also place high demands on the quality assurance of the vulnerability reports and their processing: It is to be expected that considerably more negative vulnerability reports than reports of confirmed vulnerabilities ("real advisories") will be generated and published. Users therefore need processes and corresponding tools to be able to process this amount of data.

Therefore, the use of standards in Vulnerability Management is particularly important to be able to most extensively automate the inventorisation, the comparison of known vulnerabilities and the provision of necessary information to the user.

[5] CycloneDX - Vulnerability Exploitability eXchange (VEX)
Vulnerability Exploitability eXchange (VEX) - Use Cases (cisa.gov);
Source graphic: https://www.ntia.gov/files/ntia/publications/framing_2021-04-29_002.pdf

This way, vulnerability information, including information about affectedness (positive notification) and non-affectedness (negative notification) can be communicated via CSAF (Common Security Advisory Format)[6] and VEX (Vulnerability Exploitability eXchange).

Only if both SBOM and vulnerabilities are machine-readable is an automatic and thus accurate and fast evaluation possible. This also ensures that the user of the product is informed about which publicly known vulnerabilities are actually relevant for the product. This is not guaranteed if only the SBOM information is available, without any vulnerability information assessed by the manufacturer (in the context of the application).

[6] Link to the CSAF specification: https://docs.oasis-open.org/csaf/csaf/v2.0/csaf-v2.0.html.0/csaf-v2.0.html https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Empfehlungen-nach-Angriffszielen/Industrielle-Steuerungs-und-Automatisierungssysteme/CSAF/CSAF_node.html

# 6. Limits of use

The SBOM can be used for risk assessment but needs additional information for the assessment.

**No uncontrolled release of SBOM**:

An SBOM cannot be used directly by the end customer for vulnerability management without additional information.

The additional information is, for example, the use of the functions of the components integrated in the product. This is to be concretised by means of an example:

Different algorithms (A, B, C) are supported in an encryption library. The manufacturer integrates the library and only uses algorithm C. A vulnerability is then found in algorithm A, the error is fixed, and an update and a security advisory are published. Without information about the use, the SBOM gives rise to the suspicion that the product is affected by this vulnerability. In reality, however, this is not the case, as the affected functionality is not used.

This additional information can only be provided by the manufacturer, or the evaluation can only be carried out by the manufacturer. This additional information is not part of the SBOM but is essential for its use. Therefore, these two contents should only be maintained and published in combination, or this information must be available in reference in order to be able to use an SBOM.

This additional information is mainly the evaluation and assessment of vulnerability information (CVE) by the manufacturer on the product

**Processes for use must be in place:**

- Manufacturers must first have established the processes and tools so that SBOM can be created or received, maintained and (automatically) passed on. (Maintenance of SBOM over the life cycle: generation, updating, vulnerability management, provision of data).
- Processes for (automated) processing must also be in place on the user side in order to effectively use the curated edited SBOM.

Finally, it is important to emphasise that the mere existence of an SBOM does not qualify it for purposeful further use.

# 7. What added value does the SBOM represent?

An SBOM helps manufacturers to organise and structure their internal processes. Therefore, the unproblematic connection to existing company processes is particularly important in terms of increasing software quality.

Benefits include the reduction of costs, security risks, licensing and compliance risks. Use cases include improvements in software development, supply chain management, vulnerability management, asset management, procurement and "high assurance" processes.

**Traceability**:
An SBOM is a common data set for reporting to external or internal recipients (cf. chapter 2) in an addressable manner how software is built, selected and operated. It is an important part of the Security Development Lifecycle (SDL).

- The use of SBOM creates an inventory of third-party components in use. With this data pool, statements for the software operator and its eco-system about dependencies are possible quickly, promptly and completely. In the event of a security incident or a newly discovered vulnerability, this makes it possible to react quickly and take the necessary measures.
- Software developers who use open source components will already have a database for tracking the licence conditions at least for these components.

**Quality improvement of the processes in the supply chain:**
- To create an SBOM, supply chain processes are defined, implemented and improved. This improves the quality as well as the manageability of the complexity in the manufacturing process and thus of the products as a whole.
- SBOM should be viewed in the same way as software, that is, it is never bug-free. That is why there will be updates to SBOM. Ideally, an SBOM change is traceable in the software history.

**Speed**:
- Possibility of automated evaluation
- Products potentially affected by vulnerabilities can be identified more quickly.
- Risk assessment of a plant and implementation of mitigating measures can be done faster.

**Significance:**
- Qualified statements on the affectedness by vulnerabilities in a manufacturer's products via third-party components are supported by SBOM (e.g. affectedness by Log4Shell).

# 8. Which SBOM standards exist, and which are used in practice?

To take full advantage of SBOM, machine processing and automation is required. This requires extensive interoperability throughout the supply chain, which in turn requires standardised data formats and identification schemes.

The following three formats focus on the core problem of identifying software entities and transmitting associated metadata. They have the necessary fields to meet the needs for the minimum elements of an SBOM. Tools are available to create, consume and transform this SBOM.

| Format | Specification | Known tools |
|---|---|---|
| SPDX | https://spdx.github.io/spdx-spec/ | https://docs.google.com/document/d/1A1jFIYihB-IyT0gv7E_KoSjLbwNGmu_wOXBs6siemXA/edit |
| CycloneDX | https://cyclonedx.org | https://docs.google.com/document/d/1biwYXrtoRc_LF7Pw10TO2TGIhlM6jwkDG23nc9M_RiE/edit |
| SWID | ISO/IEC 19770-2:2015 | https://docs.google.com/document/d/1oebYvHcOhtMG8Uhnd5he0l_vhty7MsTjp6fYCOwUmwM/edit |

Thus, the following three standards in particular are under discussion for SBOM as formats (see above) for data exchange:

- **International Open Standard (ISO/IEC 5962:2021) - Software Package Data Exchange (SPDX)**[7]
  *An open standard for communicating software bill of material information, including components, licenses, copyrights, and security references.*

- **OWASP CycloneDX Software Bill of Material (SBOM) Standard**[8]
  *OWASP CycloneDX is a lightweight Software Bill of Material (SBOM) standard designed for use in application security contexts and supply chain component analysis.*

- **Software Identification (SWID) Tags**, *defined by the ISO/IEC 19770-2:2015 standard.*

In the internal application, the information is securely stored and managed in an application or database that enables evaluations and automatic assignments, for example of vulnerabilities. The term "software composition analysis" is often used here.
Component information in the SBOM and vulnerability information should be used together to communicate along the supply chain both the dependencies on third party components and the vulnerabilities in the components.

[7] https://spdx.dev/; https://github.com/spdx
[8] https://cyclonedx.org/

# 9. Who gets access to what information?

An SBOM is a common record of the inventory and supply chain relationships of the various components included in the final product (the software). It is also useful along the supply chain for meeting and achieving security requirements by providing traceability as to what has been integrated into a product. The implementation of the creation and use of SBOM should be carried out according to a step-by-step model. It should always be issued on a **need-to-know basis.** As noted in chapter 2, it can be assumed that SBOM could be designed differently according to their further use, internal or external. An internal SBOM that is only used internally under appropriate confidentiality could be more extensive than an SBOM that is issued externally, e.g. also for compliance with regulatory requirements.

**Step model for all stakeholders (manufacturers, integrators and operators):**

**From 2025 onwards,** in the context of the anticipated publication of the Cyber Resilience Act, provide a "best effort" and respond to feedback (e.g. from customers) as well as rectify errors:

**B2C-Customer:** Only information that the customer needs for the intended use

**B2B customer:** Contractually negotiated disclosure of information depending on the further use of the product and according to the customer's level of knowledge and need for information.

**B2G (manufacturer vis-à-vis regulatory authorities):**

**Reporting obligations:** Information from SBOM can be used to comply with the previously regulatory reporting obligations. However, the information should always be limited to that which is actually required by the supervisory authorities, so that information can be passed on in a targeted manner. (For example, vulnerability management: only information if products are actually affected).

**Reasoned request by market surveillance authorities:**

If required by regulation, e.g. by the CRA , the SBOM may have to be issued in response to a justified request by a market surveillance authority.

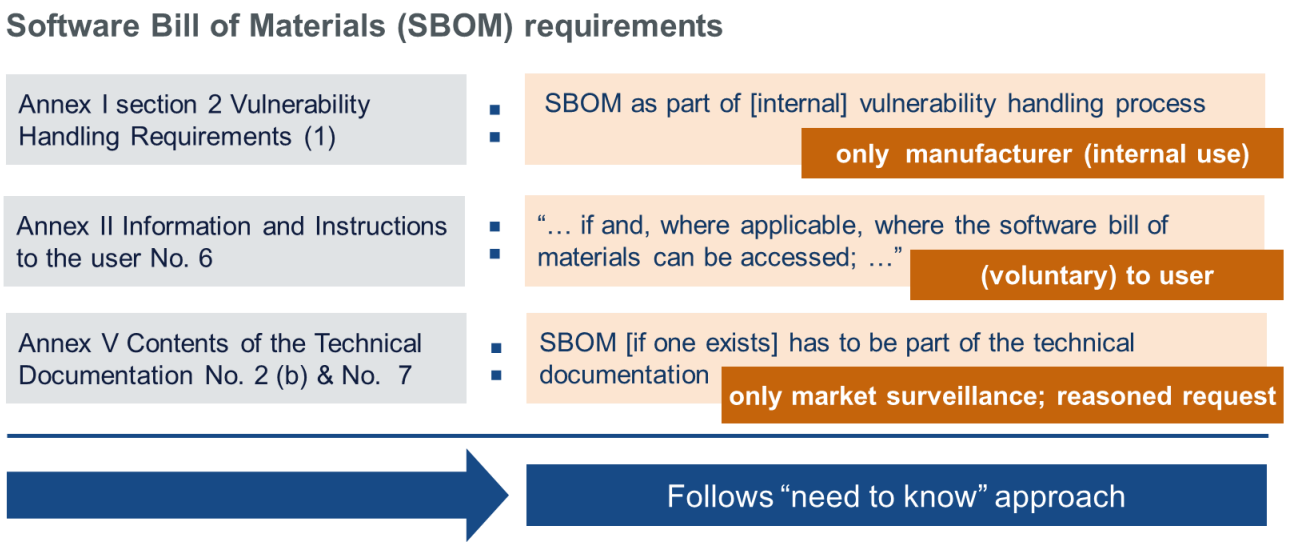
Over time, the increasing use of SBOM, the establishment of processes and the use of appropriate tools will improve software quality, understood as quality of the software itself, quality of the processes and quality of the SBOM. Standardised use and a focus on a few standards to establish their use between companies is important.

# 10. SBOM as basis/component of a declaration of conformity

Requirements for the use of an SBOM can currently only be found in the draft EU Cyber Resilience Act of 15.09.2022.

The draft EU CRA requires an SBOM (currently without providing details) as part of the manufacturer's own technical documentation. The customer documentation only has to state whether an SBOM is provided and how it can be accessed, where applicable.

In the view of ZVEI the "minimum elements " of an SBOM described in Chapter 3 should be included to meet the requirements set out in the draft CRA.




**Annex I, section 2:**
To meet the requirements of the CRA to establish a vulnerability management process that includes the creation of SBOM, the elements described in this paper can provide assistance.

**Annex II, No. 6:**
Within the framework of a security advisory, information from the internal SBOM could be released on a voluntary basis.

**Annex V, No. 2 (b) and No. 7:**
The CRA requires technical documentation for each product. If it is a product for which an SBOM has been prepared as part of the vulnerability management process, the SBOM must also be included in the technical documentation. In individual cases, the technical documentation must be issued to the competent market surveillance authority in response to a justified request from the latter.

# 11. ZVEI recommendations

- SBOM as an important means for the flow of information between stakeholders along the supply chain, the regulatory level should support the free flow of information
- SBOM is essential for efficient vulnerability management (monitoring/scanning).
- It makes sense to consider SBOM in the context of the software: Changes in the real software should be reflected in the SBOM at the same time so that the SBOM always reflects the current software status.
- SBOM should contain the third-party components used directly by the manufacturer, not their dependencies (level n-1), in order to take account of the depth of consideration described in Chapter 1.

**Contact**
Marcel Hug • Manager Cyber Security & Strategy • Digital and Innovation Policy •
Tel.: +49 69 6302 432 • Mobil: +49 162 2664 941 • E-Mail: Marcel.Hug@zvei.org

ZVEI e. V. • Electro and Digital Industry Association • Lyoner Straße 9 • 60528 Frankfurt am Main • Germany
Lobbying Register ID.: R002101 • EU Transparency Register ID: 94770746469-09 • www.zvei.org

10/2023