Tab 1

Improving Risk Management Decisions with SBOM Data (April 4, 2025)

Citation (until it is published) is:

SBOM Community, SBOM Operations Working Group. *Improving Risk Management Decisions with SBOM Data.* (2025, April 4). [Draft under review]. https://docs.google.com/document/d/1vFVbWEJmNsAbNPRAtHclC89YQlLUt6xYIvKmFGRkcQA/edit?tab=t.0#heading=h.uweqougndqvk

Disclaimer

This document was drafted in an open process by a community of Software Bill of Materials (SBOM) experts, facilitated by the Cybersecurity and Infrastructure Security Agency (CISA). CISA did not draft and is not the author of this document, nor does this document represent an official CISA and/or U.S. Government policy. CISA and the U.S. Government do not specifically adopt or endorse the views expressed in this document. [1]

Nothing in this document should be considered binding on any organization and instead should be viewed as forming a basis for future requirements covering SBOM consumption and usage.

DRAFT

Table of Contents

DRAFT

Executive Summary

The purpose of this document is to demonstrate the benefits of Software Bills of Materials (SBOMs) to software Producers and Consumers. It strives to answer the questions: “Once I generate or receive an SBOM, what do I do with it?” and “What additional insights or intelligence can I gain from the SBOM that will benefit my organization?”

The document answers these questions in two major ways: 1) It defines the SBOM Lifecycle by explaining and depicting what happens to an SBOM from the point after its generation by the software Producer to its analysis and consumption by the Consumer. 2) It provides thirteen practical use cases that exemplify how the SBOM can be used by a variety of stakeholders to benefit their organizations.

The SBOM Lifecycle depicts operations that occur through three major phases of an SBOM’s life: Production, Sharing, and Consumption. It further categorizes these operations into three levels of maturity:

- Basic SBOM Operations cover the SBOM’s generation, verification, publishing, storage, and consumption.
- Advanced SBOM Operations describe how Producers or Consumers work with SBOMs to derive further value; for example, they can compare, enrich, or merge SBOMs and analyze SBOMs for a variety of risks.
- Continuous Vulnerability Management operations are the most mature, such as regular post-release monitoring of SBOMs for newly discovered risks.

Thirteen use cases describe real situations in which Producers or Consumers utilize SBOMs to extract information of value to their organizations. These use cases span the SBOM’s Lifecycle and cover topics such as how SBOMs can be used to: pinpoint differences between software versions; identify security, licensing or compliance risks; alert organizations when software components are nearing their end of support; inform incident responders of impacted software; and support procurement or mergers and acquisition (M&A) decisions. For each use, the document provides a brief narrative description followed by a table that summarizes the use cases’ actors, business motivation, objectives, steps to achieve the objectives, National Telecommunications and Information Administration (NTIA) minimum elements used, other supplementary data, and benefits achieved.

Key takeaways from the use cases are identified at the end of the document.

DRAFT

1.0 Introduction

Software Bill of Materials (SBOM) adoption has increased and diversified since the Department of Commerce National Telecommunications and Information Administration (NTIA) initiated the Multistakeholder Process on Software Component Transparency in 2018. The implementation of SBOMs has also matured to reflect technological advancements and in response to new use cases. One driver for adoption and implementation of SBOM is the increased transparency gained from analyzing SBOM data to understand and manage modern applications, code reuse, and use of open source libraries. This document describes specific operations that software Producers and Consumers can perform on SBOM data to draw valuable insights that can be applied to improve security risk decisions for software.

1.1 Motivation

Beyond serving as a software inventory, the utility of SBOMs for both the software Producers and Consumers is not well understood. The major questions this document answers are: “Once I generate or receive an SBOM, what do I do with it?” and “ What additional insights or intelligence can I gain from the SBOM that will benefit my organization?”

The motivation for this document is to demonstrate the benefits of SBOMs to both the software Producers who generate them and the Consumers who receive them from a vendor and/or partner. It does this by providing use cases that are already implemented, or could be implemented in the future; for example, by cross referencing SBOM data with other datasets to help an organization proactively address security, licensing, or other supply chain risks. These use cases provide organizations with actionable tasks to perform on their collective SBOMs to extract intelligence and insight into their software.

1.2 Goals

This document was created by the SBOM Operations Working Group, a community-driven workstream facilitated by the Cybersecurity and Infrastructure Security Agency (CISA) [1]. The ultimate goal of this document is to lay a foundation for how practitioners can use an SBOM to make more informed technical and business decisions. A secondary goal is to spur conversation about external datasets[1], enabled through the use of SBOMs, that can further improve software transparency for the industry as a whole. For the purposes of identifying use cases, this document introduces the concept of an “SBOM Lifecycle”, which identifies processes that may be followed by a person, organization, and/or tool to enrich, analyze, and securely share SBOM data. In order to time-bound and focus the discussion and use cases submitted by the stakeholders, we limited the scope of the document to the highlighted sections of the SBOM lifecycle depicted in Figure 1.

[1] An example of such a dataset is https://endoflife.date

DRAFT

This document focuses on the operations that can be performed on an SBOM that has been provided to a Consumer by a Producer. The Consumer/Producer relationship isn't limited to SBOM-sharing across organizational entities; it can also be within the same organization. To focus on the utility of the SBOM, the Producer-supplied data is assumed to be accurate and complete; and it's assumed the Producer actively verified the contents and structure of the SBOM prior to sharing it with the Consumer. Data quality issues, including accuracy and completeness, require focused, in-depth discussions that are outside the scope of this document. In addition, this document does not discuss SBOM storage, transport, or sharing practices within/across organizations. For the purposes of data extraction, the document assumes the SBOM is machine readable and in a widely-used format such as SPDX or CycloneDX. To ensure that Consumer workflows operate on approved SBOM information, it's assumed that any SBOM shared by a Producer will have an associated signature allowing Consumers to determine its integrity and that any changes to SBOM data or file format results in a new uniquely identifiable SBOM.

This document does not use the specific organizational roles outlined in previous SBOM documents [2] and instead uses more general references to diverse roles commonly found in organizations that consume, operate, and produce software. These roles may include architects, senior engineers, security teams, supply chain risk management practitioners, acquisition authorities, security executives, legal teams, and risk officers. The use cases are also not focused on any specific software system. They describe generic processes that can be applied to many types of software in different industries and governments.

1.3 Analysis within the SBOM Lifecycle

1.3.1 Definition of SBOM Lifecycle and SBOM Lifecycle Management

The SBOM Operations Working Group defines the SBOM Lifecycle as what happens to an SBOM from the point that it has been generated to its analysis and consumption by the Consumers of the software described by the SBOM. We define SBOM Lifecycle Management as the actions and operations that are performed to achieve a business objective or achieve some benefit to the organization. Examples of these benefits are: greater insight into security, licensing, or supportability issues in the components of a software system; fulfillment of regulatory compliance; and accelerated identification of vulnerable components during incident response.

The SBOM Lifecycle described in this document starts with the premise that an SBOM has already been created following a robust SBOM Authoring workflow or imported from an SBOM provided by a Producer. It specifically excludes the actions taken by developers or software vendors to technically generate the SBOM and maintain it; academic researchers [3] have already described this lifecycle.

DRAFT

1.3.2 Genesis of the SBOM Lifecycle diagram

Figure 1 provides a notional diagram of the SBOM Lifecycle, starting from its point of initial creation by a Producer to its consumption by at least one Consumer. This diagram was created for two major reasons: 1) to answer frequently-asked questions such as: “What happens to an SBOM after it’s created?” or “What do people or organizations do with SBOMs?”; and 2) to provide a mental model of the SBOM Lifecycle that can help Producers and Consumers of SBOMs to frame their discussions, problems, requirements, and solutions. Producing, sharing, and consuming SBOMs is an emerging domain of expertise, with nascent technologies or practices. Therefore, an SBOM Lifecycle diagram provides a common notional model and terminology, which can facilitate stakeholder communication and alignment of emerging technologies with needs.

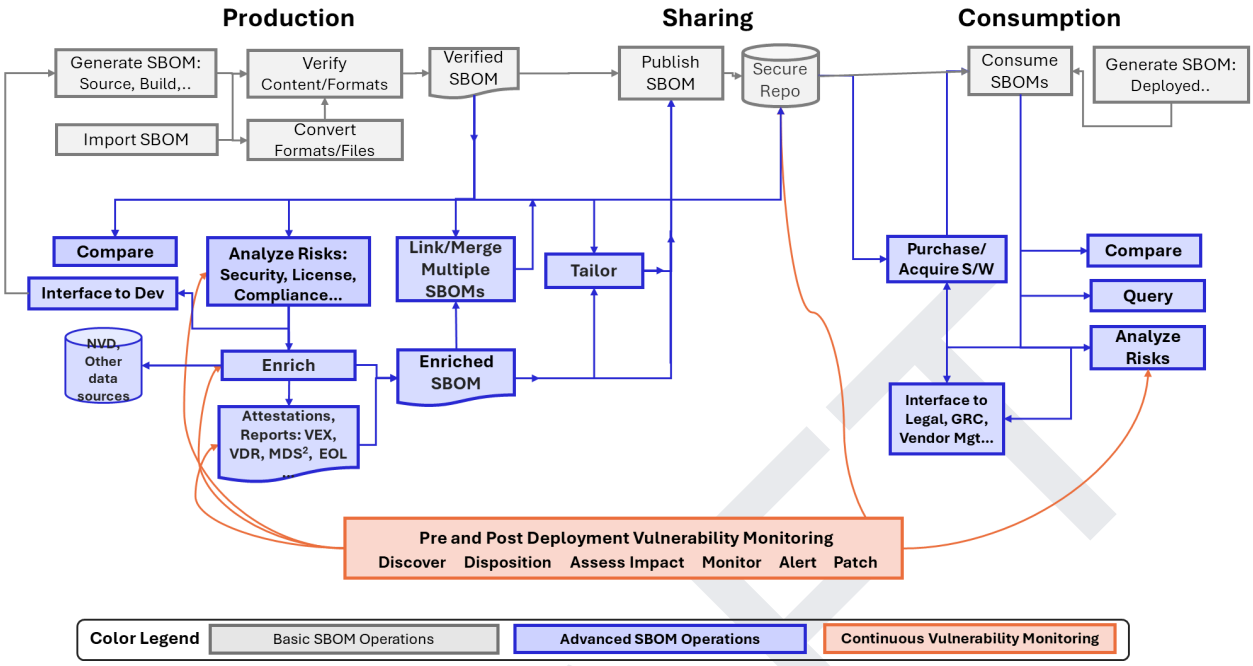
To draft the initial content of the workflow diagram, we synthesized information about how people can use SBOMs from documents produced by the SBOM community as convened by CISA [4] [5] and NTIA [6], as well as guidance from the National Security Agency (NSA) [7], and from requirements and stories we heard from industry SBOM users . We further refined the diagram based on information gathered from use cases presented in Section 2 of this document, regular feedback from members of the SBOM Operations Working Group, and response to a conference presentation that included the refined diagram [8]. The resulting workflow is identified in Figure 1.

This SBOM Lifecycle depicted in the diagram is not a “once and done;” it will undoubtedly go through subsequent revisions. As organizations increasingly adopt and use SBOMs, the SBOM Lifecycle will evolve, likely expanding additional details,insights and technologies.

Finally, individual organizations are unlikely to engage in every process of the SBOM Lifecycle. Each organization will place itself at its own position on the path, based on their roles, business objectives, contractual obligations, regulatory requirements, and supply chain maturity level. The diagram is designed to help with this positioning and with understanding what precedes and follows that position.

DRAFT

Figure 1 - SBOM Lifecycle Diagram



1.3.3 The SBOM Lifecycle Diagram

The SBOM Lifecycle diagram, shown in Figure 1, is *aligned from left to right to the three major phases of an SBOM's life*: **Production, Sharing, and Consumption**. It is *aligned from top to bottom based on three levels of maturity of SBOM operations:* **Basic** (in gray)**, Advanced** (in blue), and **Continuous Monitoring** (in orange). Because SBOM Lifecycle Management is an emerging domain, its operations or analyses vary in sophistication and maturity. For example, the SBOM Sharing Primer [9] describes three different levels of sophistication of the technologies and processes applied to SBOM dissemination. For SBOM Lifecycle Management, the SBOM Operations Working Group clustered lifecycle operations into the three general categories of maturity in the diagram.

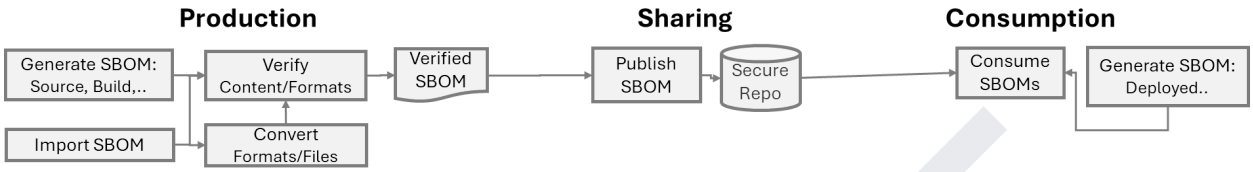
In the most basic form of its lifecycle, an SBOM is either generated (via tooling or human audit) or imported into an SBOM management solution; verified for content (e.g., meeting NTIA SBOM Minimum Elements [10]) and formats (e.g., CycloneDX or SPDX); stored in a secure location; and accessed by a Consumer. In the diagram, these basic SBOM management operations are represented by gray boxes and are not covered by the use cases in this document. As described in Section 1.2, the SBOM Operations Working Group excluded from our analysis the basic operations of SBOM generation, storage, or transport to Consumers. Our focus is on what happens to an SBOM *after* it has been generated and has moved on to analysis and monitoring by Producers and Consumers.

DRAFT

1.3.3.1 Basic SBOM Operations

Basic SBOM Operations are depicted in Figure 2.

Figure 2 - Basic SBOM Operations



Software Producers **Generate SBOMs** of many different types, as shown in the upper left of the diagram. Consumers may also generate SBOMs of deployed software. Organizations also **Import SBOMs**, and according to the United States (U.S.) NSA's Recommendations for SBOM Management [7], SBOMs should follow either the SPDX [11] or CycloneDX [12] specification and be importable in a file format appropriate to the specification, such as JSON, XML, or CSV. Producers or Consumers (in the case of imported SBOMs) **Verify the Content and Formats** to ensure that they contain the elements required by legislation, regulation, industry standards, and/or contract; and that they conform to the appropriate SBOM specification. In the U.S., an SBOM's minimum elements are often those defined in 2021 by the Department of Commerce National Telecommunications and Information Administration (NTIA) [10] or a successor by the relevant U.S. governmental organization. The NSA (page 8 of [7]) recommends import and support for both SPDX and CycloneDX. The Verification process is a critical step that, if not done properly, creates difficulty for Consumers importing and analyzing SBOMs later in the lifecycle. As part of this Verification process, Producers may need to **Convert from one SBOM format into another**. Consumers may also need to convert SBOM formats upon receipt of an SBOM.

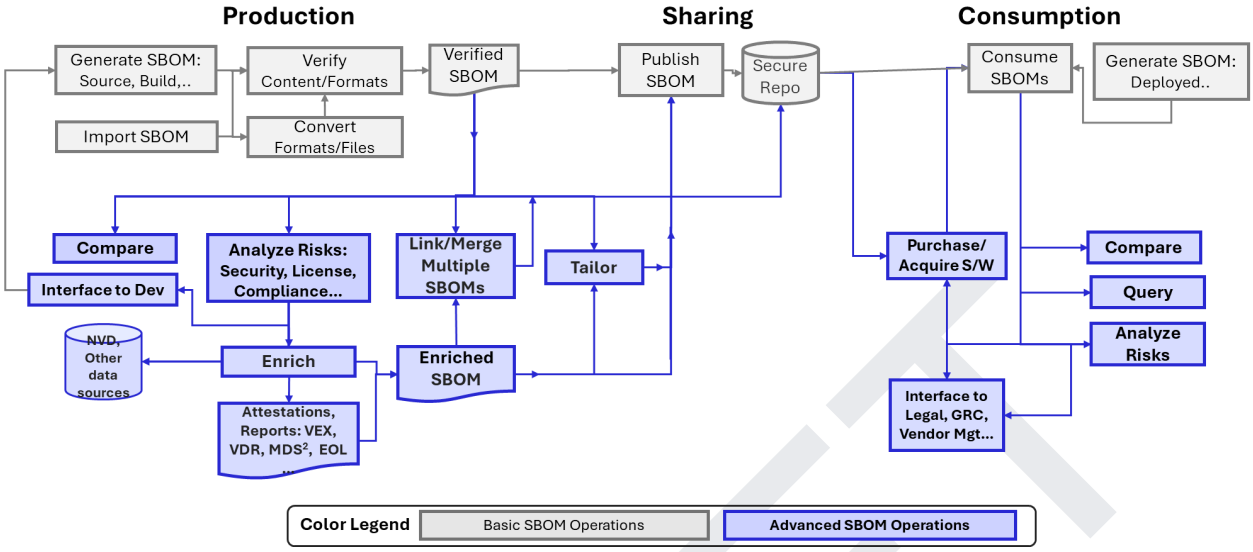
At the basic level of the lifecycle, the Producer **Publishes** the SBOM and stores it into a **Secure Store or** common exchange point for dissemination for others to **Consume**. The Consumers [5], who receive the SBOM, may be end customers who will be deploying the software, third parties who hold the SBOM for distribution, or software Producers who use the SBOM as part of their software development process.

1.3.3.2 Advanced SBOM Operations

The use cases described in this document largely depend on advanced SBOM operations, which are depicted in blue in Figure 3. The operations in this part of the lifecycle rely on an SBOM that is verified for conformance to specifications.

DRAFT

Figure 3 - Advanced SBOM Operations



The **Compare** operation, which is shown on both the Production and Consumption side of the diagram, is performed to clearly see the differences between builds or versions of the same software.

Analyze Risks is a critical operation engaged in by both Producers and Consumers [13], and includes risks associated with security, licensing, non-compliance, and maintainability [2] [14] [15]. Security risks associated with incorporating vulnerable components into software is one of the most frequently cited rationales for analyzing SBOMs [15]. In security risk analysis, the components identified in the SBOM are cross-referenced against lists of known vulnerabilities in third-party components accessible from the NIST National Vulnerability Database (**NVD**) or other sources. The results of the risk analyses may be sent to others internally within the organization. For example, software Producers can use SBOM security analysis to discover vulnerable components in their own software and notify their software development team for remediation. Development teams, acting as Consumers of SBOMs [3] from downstream suppliers, can alert both suppliers and their teams about discovered vulnerabilities. Risk analysis reports may also be shared with external Consumers who can use them to inform purchasing decisions, legal compliance, or vendor and supply chain risk management.

Processes to **Enrich** SBOMs are used to supplement the SBOM with information needed for additional **Reports** or attestations [16]. For example, the Producer may include a statement or advisory about why the software is not impacted by a specific vulnerability. Security advisories may be communicated in a variety of formats, such as Common Security Advisory Framework (CSAF) and Vulnerability Exploitability eXchange (VEX) [17]. In some industries, software Producers add End of Life (EOL) or End of Support (EOS) data to meet regulatory compliance requirements. For example, the U.S.Food and Drug Administration (FDA) Center for Devices and Radiological Health (CDRH) wants Support Status, and End of Support / End of Life dates [18] to be provided, above and beyond the NTIA minimal SBOM elements. Enrichment can occur throughout the lifecycle and may be done by the Producer who authors the SBOM, the

third-party Provider who stores and shares the SBOM, or the Consumer who receives the SBOM [5], or by third party organizations offering SBOM analytical services.

An Enriched SBOM may be directly stored for sharing with Consumers, or may be part of additional operations such as Tailoring or Merging with Other SBOMs. Each SBOM is a snapshot in time of the software. As SBOMs are immutable once created, any processing of an SBOM that changes its content results in a new, discrete SBOM with the appropriate modifications rather than an updated SBOM. It is the latest snapshot in time.

Tailoring SBOMs involves customizing the content and format of the SBOM for the intended audience. It may impact content such as supplemental data, level of transitive dependencies that are included in the Enriched SBOM prior to dissemination, or sensitive data [14]. Tailoring may be done for several reasons. Contractual requirements might dictate specific fields and format. Regulations may dictate level of detail, e.g., the EU's Cyber Resilience Act (CRA) requires only direct dependencies. An internally-used SBOM, which contains a superset of information useful for Internal processes, may be redacted by Legal prior to publication.

Linking/Merging SBOMs (NSA calls this aggregating) can be relatively simple such as tagging SBOMs as related to each other and part of a common system. Or it can be complex, in which a system SBOM is created from a hierarchy of all the SBOMs of the subsystems that comprise it. For example, an automotive infotainment system merged SBOM may include a navigation SBOM, media SBOM, and vehicle function SBOM, and each of those three SBOMs may be composed of many "children" SBOMs. A hierarchically Merged SBOM may also be referred to as a "system SBOM," "SysBOM," or "Product SBOM."

SBOMs are shared through a variety of mechanisms ranging from something simple such as the Producer emails the SBOM to the Consumer to more sophisticated solutions such as publishing the SBOM to a **Secure Repository**, administered by a third-party Distributor [19], for access by properly authenticated Consumers [20].

Consumers obtain value from SBOMs in a number of ways. Their earliest interaction may be when making a decision to **Purchase or Acquire Software**. At this point the Consumer may **Analyze Risks**–e.g., from security vulnerabilities, software component licensing, provenance, or supportability—to the organization from the targeted software. This information may be exchanged with the organization's **Legal, Governance, Vendor Management,** or other departments for use in the current purchase or stored as reference material for future purchases from the software's supplier, or authorized resellers. If this is a new or updated version of previously acquired software, the Consumer will **Compare** the new SBOM to the older SBOM and note any changes.

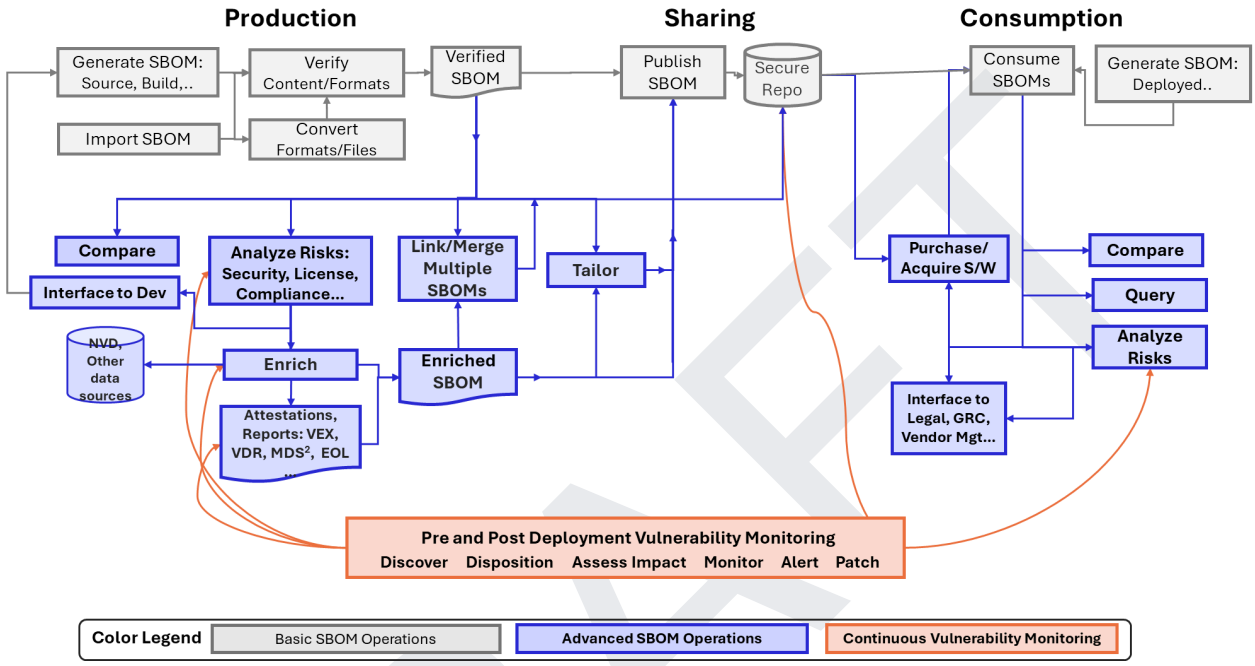
After purchase, the SBOM becomes part of the Consumer's software inventory, where the Consumer may **Query** the SBOM to obtain valuable information, e.g., the presence of specific software components as part of incident response; upcoming End of Support of components; vulnerability exploitability status; or provenance data for legal compliance.

DRAFT

1.3.3.3 Continuous Monitoring

The most advanced operations within SBOM Lifecycle Management, shown in orange in Figure 4, are those associated with **Continuous Monitoring** of components identified in the SBOM.

Figure 4 - Continuous Vulnerability Monitoring



Armed with an SBOM that contains all the software's components and dependencies, a security team can regularly assess if the risks presented by those components have changed. Among the operations they perform are: **Discovery** of new vulnerabilities in components identified in the SBOM through daily updates from NVD; **Dispositioning** of these vulnerabilities to determine their exploitability, impact and any required remediation; **Patching** of components with updates to reduce risks; **Monitoring** SBOM repositories for the existence of emergent vulnerabilities; and **Alerting** software users of new vulnerabilities. NSA describes many of the vulnerability tracking and analysis operations that are foundational to continuous monitoring in their Recommendations for SBOM Management [7].

Continuous Monitoring may be engaged in by Producers of SBOMs, by Distributors who store and disseminate SBOMs, and by Consumers who are using software represented by SBOMs.

DRAFT

2.0 Use Cases

The SBOM Operations Working Group brainstormed many use cases in which people or organizations extracted value from SBOMs to help manage risks. To be considered as a use case, it had to meet two main criteria:

- It describes activities that are done to or with an SBOM after its initial generation and verification against standards. It answers the question: “I have an SBOM, now what?”
- These activities are done for the purposes of providing benefit, value or insight to an organization.

To meet the second criterion, there are some key assumptions on Producers:

- Producers have verified that the SBOM conforms to the appropriate specification for its format (e.g. SPDX or CycloneDx file)
- Data expressed within the SBOM is correctly formatted based on the requirements of the data element (e.g. a pUrl identifier is syntactically correct)
- Producers provide a means for Consumers to validate the integrity of an SBOM, such as an SBOM signature.
- Once an SBOM is provided to a Consumer, that the SBOM becomes immutable.

We realize that, currently, SBOM verification occurs inconsistently and that the reliable accuracy of SBOMs remains an unsolved problem. However, we made the assumptions of reliable verification, immutability, and accuracy so that we could focus on use cases.

Where an SBOM is used in a workflow requiring vulnerability information, it’s critical that any vulnerability identifier used by a Producer uniquely identifies the same vulnerability for the Consumer. In the use cases presented, the term “**Vulnerability Information Sources**” is used to indicate information, such as a CVE identifier, that might be stored in a shared location such as the National Vulnerability Database (NVD) or country-specific NVDs, or where a private entity publishes a shared information source, such as GitHub Security Advisory (GHSA) identifiers. These identifiers often serve as an index into additional or augmented information sources such as the CISA Known Exploitable Vulnerability (KEV) catalogue or the Exploit Prediction Scoring System (EPSS) information. The term “Vulnerability Information” as used in the use cases is meant to be an expansive term covering all potential information required to contextually meet the use case objectives.

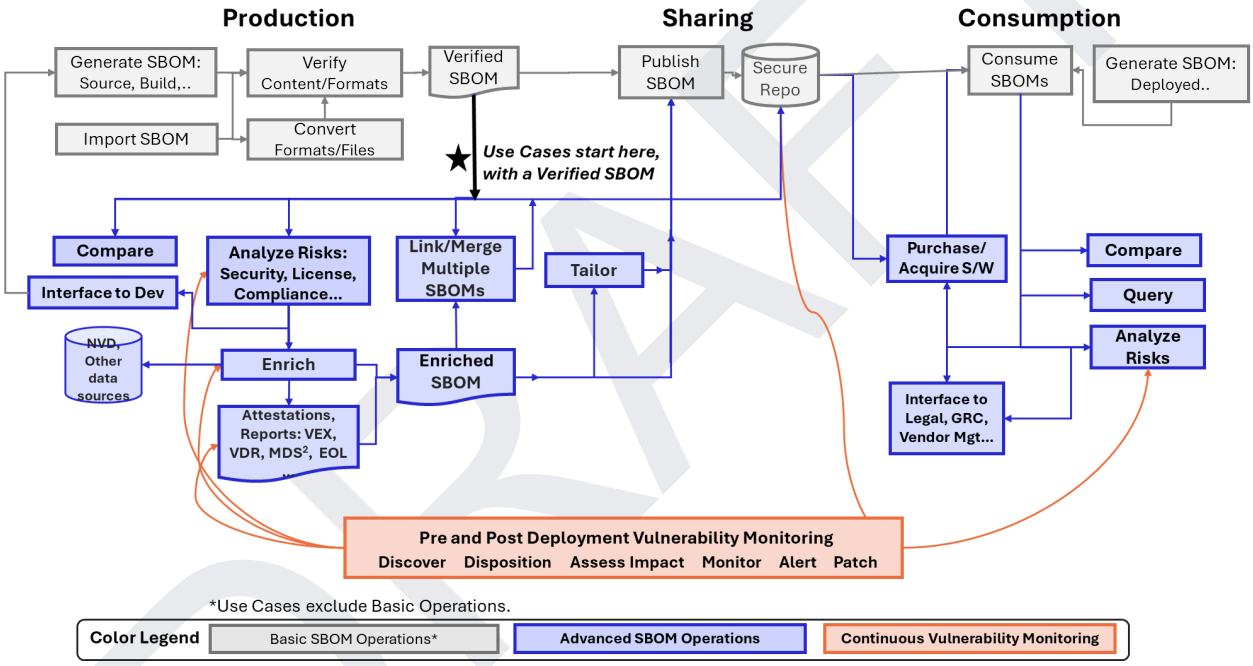
Where third-party vulnerability information sources are valuable when determining the potential scope for a given vulnerability, they are no substitute for assertions by suppliers as to the exploitability or potential mitigations within the context of specific applications. In the use cases presented, the term “**Supplier Security Assertions**” is used to indicate assertions made by a supplier, such as VEX statements or product errata. Assertions differ from security or software “**Attestations**” where an attestation is made by a supplier indicating that the software complies with certain standards or regulatory expectations.

DRAFT

While it's common to think of the Producer and Consumer roles as occurring between independent organizations, such an approach artificially limits the utility of SBOMs. Many of the use cases presented could occur within a single organization in order to fulfill unique requirements. One key to successful SBOM usage is a recognition that SBOMs can be transformed, enriched, or tailored to meet specific requirements and that any modification of the contents of an SBOM results in a new SBOM meeting the assumptions of a Producer and the expectations of a Consumer.

The working group also excluded any use cases related to the transmission or storage of SBOMs.Within the SBOM Lifecycle Diagram, the use cases start at the point of receiving a Verified SBOM, as shown in Figure 5, and exclude Basic SBOM operations shown in that figure.

Figure 5 - Use Cases Start with a Verified SBOM and Exclude Basic SBOM Operations



From the broader set, the group narrowed the use cases down to a curated set of thirteen within the context of the SBOM Lifecycle, which are listed below. This is not a comprehensive list of all the possible use cases; rather it reflects the use cases most familiar to the experts on the SBOM Operations Working Group.

The use cases are grouped into three categories based on how likely a reader is to encounter the use case in their organization. This likelihood is affected by the maturity of solutions to address the use case as well as the breadth of applicability. For example, the Pre-Deployment CVE Vulnerabilities use case is in the first category because there are solutions already in use today to cross-link CVE data with SBOM fields, and because CVE vulnerabilities affect almost every government and commercial organization, regardless of vertical. The SBOM Support for Field Services Software-enabled Devices use case is in the third category because the

processes/technologies for comparing a device's build SBOM to information collected directly from the fielded device (such as a remotely-generated runtime SBOM) are less mature, and because fewer verticals engage in this use case (e.g., medical technology, electronics, energy sensors).

Most Mature / Broadest Applicability

1. **Pre-deployment Common Vulnerabilities and Exposures** (**CVE) vulnerabilities**: Discover vulnerabilities in software products before release.

2. **Post-deployment CVE vulnerabilities**: Discover vulnerabilities in software products after release.

3. **Open source (OS) licensing risks**: Determine if open source licensing of components presents risks to an organization.

4. **EOL and non-maintained component alerting**: Identify software packages near End of Life to plan upgrades or replacement.

5. **Pre-purchase risk assessment**: Assess software for risks prior to purchase or acquisition.

6. **Component usage across an organization**: Identify all software components used and their prevalence in an organization.

Moderately Mature / Moderate Applicability

7. **Incident response**: Identify all applications that depend on a component involved in a security incident.

8. **Mergers and Acquisitions (M&A) and Investment risk assessment**: Assess risks in target software prior to mergers, acquisitions, or investment by a third party.

9. **Verification of accessory software:** Verify that all accessory components are included with core software's SBOMs, and analyze accessories for security, licensing and compliance risks.

10. **Differences in components between builds or versions**: Discover how components differ between software builds or software versions.

Least Mature / Focused Applicability

11. **Conformance with disparate Governance, Regulatory, and Compliance (GRC) specifications**: Comply with disparate regulations and contract requirements for SBOMs or software inventories.

DRAFT

12. **Integrity and threat management for Operational Technology (OT) and isolated networks**: Standardize and streamline version and dependency management across network boundaries to minimize attack surface and other risks

13. **Field servicing of software-enabled devices**. To assist maintenance and troubleshooting, field service representatives compare a previously- generated SBOM of a device to data collected from an operationally deployed device.

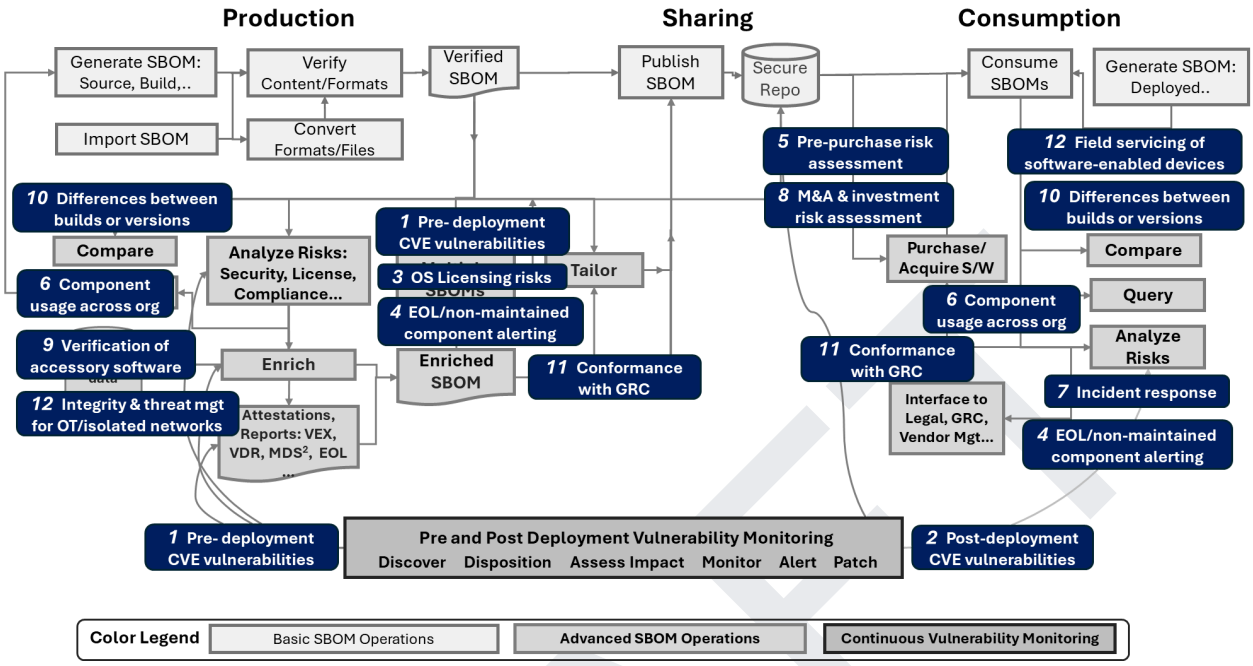
Figure 6 organizes the use cases by these three categories and shows their relevance to either the Producer, Consumer or both.

Figure 6 - Use Cases Grouped by Maturity/Applicability and Lifecycle Phase

Use Case #	Maturity / Applicability	Lifecycle Phase	
	Most Mature / Broadest Applicability	**Produce**	**Consume**
1	Pre-Deployment CVE Vulnerabilities	x	
2	Post-Deployment CVE Vulnerabilities		x
3	Open Source Licensing Risks	x	x
4	EOL and Non-maintained Component Alerting	x	x
5	Pre-purchase Risk Assessment		x
6	Component Usage Across an Organization	x	x
	Moderately Mature / Moderate Applicability		
7	Incident Response		x
8	M&A and Investment Risk Assessment		x
9	Verification of Accessory Software		x
10	Differences in Components Between Builds or Versions	x	x
	Least Mature / Focused Applicability		
11	Conformance with Disparate GRC Specifications	x	x
12	Integrity and Threat Management for OT and isolated networks	x	
13	Field Servicing of Software-enabled Devices	x	x

These use cases are also mapped onto the SBOM Lifecycle diagram as shown in Figure 7.

Figure 7- Use Cases Mapped onto SBOM Lifecycle Diagram



For each of these use cases, the group prepared a short narrative description and a table comprising seven attributes that further describe the use case. These attributes are: Actors, Business Motivation, Functional Objectives,Steps to Achieve Objectives, NTIA Fields Used [10], Added or Cross-linked Data, and Benefits Achieved.

While each use case is presented on its own, the group recognizes that many are interconnected; for example, the use case regarding Post-deployment CVE Vulnerabilities can stand on its own, or also be a part of Purchasing Decisions or M&A and Investment Risk Assessment.

DRAFT

2.1 Use Case: Pre-deployment CVE vulnerabilities

Software Producers must ensure or attest that their products are free of known and addressable security risks (e.g., attest compliance with Secure Software Development Framework (SSDF) prior to their release) [21]. In the pre-market setting, an SBOM serves as a foundational tool for managing cybersecurity risks in software and firmware. By using SBOMs in conjunction with vulnerability information, software Producers can systematically identify vulnerabilities which are then analyzed against the planned product. This analysis drives potential actions such as new design requirements, supplier controls and testing to minimize or eliminate the potential for exploits once the product is released into the market. This activity also supports regulatory submissions, eases market entry and provides a level of confidence and trust for buyers. The documentation of this analysis can serve as evidence to support security attestations. By proactively sharing relevant security information, software producers can enhance transparency, demonstrate accountability, and support informed decision-making across the supply chain.

Table 1 describes the key attributes of this use case.

Table 1 - Use Case: Pre-deployment CVE vulnerabilities

Actors	Producer’s Procurement Office for Components, Regulatory, Engineering, Product Security Teams
Business Motivation	Minimize risks and liabilities to the software Producer from unaddressed vulnerabilities in a software product
Functional Objectives	Discover vulnerabilities and address risks to the software product prior to the software's release
Processes or Steps to Achieve Objectives	Cross reference SBOM components with vulnerability information from various sources, including NVD, GitHub Security Advisory, and other trusted repositories Identify any associated CVEs for each component and document for further analysis Review supplier-provided vulnerability assessment reports of third-party software integrated into the product Assess risk and prioritize vulnerabilities based on acceptance criteria and risk scoring factors based on vulnerability information and other factors like component dependencies Remediate vulnerabilities by implementing compensating security controls, patching or removing unused or non-essential components If remediation requires substantial changes, reassess the product design or implementation to ensure vulnerabilities are effectively addressed without introducing new risks

DRAFT

	Perform vulnerability assessment post-remediation to ensure all identified vulnerabilities are mitigated and no new vulnerabilities have been introduced Record all steps taken to address vulnerabilities, including the rationale for risk acceptance if specific vulnerabilities remain unaddressed If required, provide updates to stakeholders, customers, or regulatory authorities regarding vulnerability management actions and risk mitigation measures taken
NTIA Fields Used	Supplier, Component Name, Version of the Component, Other Unique Identifiers, Dependency Relationship
Added or Cross-linked Data	Vulnerability Information Sources Supplier Security Assertions (e.g., VEX), and threat model insights
Benefits Achieved	Ensures the product software is as secure as possible, based on the discovered vulnerabilities, prior to deployment and field use. Minimizes liability for the Producer. Supports security attestations and security advisories that may be required by software procurers.

DRAFT

2.2 Use Case: Post-deployment CVE vulnerabilities

After software is deployed, ongoing vulnerability monitoring becomes essential for both Producers and Consumers to reduce security risks throughout the software's lifecycle. New vulnerabilities are continuously discovered, making the periodic scanning of SBOM-identified software components against CVEs stored in the NVD a crucial proactive action that Consumers can take prior to exploitation.

Beyond periodic scanning, the Consumers' security, IT, and compliance teams can actively integrate SBOM monitoring into their post-deployment security operations. This includes automating alerts for newly discovered vulnerabilities, assessing the potential impact on deployed systems, and prioritizing remediation efforts based on real-world risk factors such as exploitability, software dependencies, and critical system exposure.

The Producer's engineering or product security teams may be assigned to engage in an ongoing process to detect and mitigate these risks throughout the software product's lifecycle. By leveraging a properly maintained SBOM, Producers can trace vulnerabilities to specific components, enabling targeted remediation efforts and efficient resource allocation. When the SBOM is paired with a Coordinated Vulnerability Disclosure system, the Producer can notify Consumers and provide updates which can be applied to reduce or eliminate the chance of exploit in the software or firmware.

Both software Producers and Consumers reduce risks from new vulnerabilities in deployed software products and build trust in deployed software through periodic correlation of SBOMs against the CVE database and consistent communications.

By maintaining a continuous feedback loop with Producers, Consumers can quickly receive security advisories, patches, or mitigations, ensuring that deployed software remains secure, compliant, and resilient against emerging threats.

Table 2 describes the key attributes of this use case.

Table 2 - Use Case: Post-deployment CVE vulnerabilities	
Actors	Risk/Compliance Officer, Regulatory, Engineering, Security Teams, Product Security Teams (PSIRT), Consumer Security Teams (CSIRT)
Business Motivation	Maintain secure products in the market, Maintain regulatory compliance, Avoid costly/embarrassing incidents
Functional Objectives	Discover how new CVEs impact software components in deployed software

DRAFT

	Assess security and compliance risks of emerging vulnerabilities on deployed software
Processes or Steps to Achieve Objectives	Maintain an accurate SBOM Producer - Ensure your organization generates and maintains an accurate SBOM for all products Consumer - If you rely on software providers, require them to deliver accurate SBOMs with each release, update, or patch Review vulnerability assessment reports Analyze vulnerability reports provided by suppliers of third-party software integrated into the product Supplement this with results from third-party scanning tools Perform regular vulnerability monitoring Regularly monitor vulnerability information sources to identify new vulnerabilities relevant to components in the SBOM Monitor cybersecurity signals from Information Sharing and Analysis Organizations (ISAOs), threat intelligence feeds, and security advisories to stay informed about vulnerabilities and threat trends Map newly identified CVEs to SBOM components Assess discovered vulnerabilities against the acceptance criteria for the product If required, make decisions related to containment, updates/patches, compensating controls, and reporting or recall decisions
NTIA Fields Used	Supplier, Component Name, Version of the Component, Other Unique Identifiers, Dependency Relationship
Added or Cross-linked Data	Vulnerability Information Sources Supplier Security Assertions (e.g., VEX) Inventory systems that include runtime data
Benefits Achieved	Software Producer and Consumer maintain the security and compliance of deployed software products that may be impacted by new CVEs posted in the NVD. Producer and Consumer supporting functions can take appropriate actions to contain, mitigate and report new vulnerabilities before exploitation.

DRAFT

2.3 Use Case: Open source licensing risks

Open source software (OSS) is a critical component in the development of modern applications, powering everything from infrastructure to end-user features; but its use introduces diverse licensing requirements that can impose significant legal and operational obligations, especially for redistributed software. This use case demonstrates how an SBOM can be used to effectively manage open source licensing risks by serving as a comprehensive inventory of all software components within an application, along with their respective licenses. An enriched SBOM that includes license details, full license text, and copyright information—elements that may be absent from an NTIA-minimum-elements compliant SBOM—can provide additional artifacts needed for open source license management.

By integrating SBOMs with open source license databases and aligning with organizational OSS license policies, organizations can proactively identify and mitigate licensing conflicts, track compliance obligations, and address changes in licensing terms that may introduce legal risks. This structured approach empowers legal, development, and compliance teams to collaborate effectively, ensuring adherence to licensing requirements, minimizing legal exposure, and fostering a secure and compliant software supply chain.

Table 3 describes the key attributes of this use case.

Table 3 - Use Case: Open source licensing risks

Actors	Legal and Compliance Teams, Open Source Program Office (OSPO), Engineering/Development Teams, Procurement Office, Security Teams, Executive/Management Teams
Business Motivation	Protect company from legal risks associated with improperly licensed open source components
Functional Objectives	Empower all actors to collaboratively manage open source software in a legal manner. Ensure adherence and compliance to the various open source licenses in use, avoiding costly legal penalties or violations. Provide visibility into the software supply chain to identify licensing conflicts.

DRAFT

Processes or Steps to Achieve Objectives	Determine the specific requirements for the organization's OSS license policy, including acceptable license types, usage restrictions, and compliance obligations. Determine the compatibility of various open source licenses used across different components in the software supply chain. This step helps prevent conflicts that may arise from combining components with incompatible license terms. Determine the need to update or replace open source components based on changes in licensing terms, discovery of legal risks tied to specific licenses. Identify the license being used for the distribution of a software release version. Identify the software licenses associated with each component listed in the SBOM by leveraging an open source license database. Identify potential licensing risks, such as restrictive licenses that could impose obligations on proprietary code or licenses requiring source code disclosure. Identify the legal and compliance obligations for any software that incorporates open source components, particularly for products that will be redistributed..
NTIA Fields Used	Component Name
Added or Cross-linked Data	Open source license database such as those maintained by the SPDX [22], OSI [23], or ecosystem.ms [24] databases
Benefits Achieved	Provides specific actions legal teams should take based on the SBOM's license data to protect the organization from potential legal exposure. This includes setting policies for component updates, replacements, or license conflict resolution. Ensures that software vendors and internal teams comply with established OSS license requirements and policies.

DRAFT

2.4 Use Case: End of Life (EOL) and non-maintained component alerting

End of Life (EOL) is a supplier-determined designation that reflects a formal and explicit organizational decision to cease maintenance of a particular product or version. This designation must be understood in the context of legal and contractual obligations that exist between vendors (including both product and professional services firms) and customers.

While EOL data sources are not publicly consolidated (i.e. this information exists on corporate web sites and security advisories), organizations are responsible for creating and monitoring this data on their own. Once the EOL information about a component and its version is obtained (e.g., by using sources such as https://endoflife.date/), an organization can use this information in conjunction with the SBOM to identify software components that are close to EOL status and plan for upgrade, migration or self-maintenance.

More colloquially and especially with regard to open source packages and software components, “EOL” is often used to mean “non-maintained,” i.e. a package has been abandoned. There is a critical functional difference between “non-maintained” and “EOL,” which is: a non-maintained open source package can be maintained by new community members or even by a Consumer if it makes economic sense to do so.

Non-maintenance *is important* for software security, operational risk and integration cost. Unlike EOL, thresholds for active maintenance can be defined by Consumers and governed in the context of SBOM analysis.

For a Producer’s development teams, SBOM analysis of the maintenance status of components provides early feedback to the teams requesting usage of these components to either minimize usage or update the package early on. For Consumers, SBOM analysis to identify EOL or non-maintenance triggers in their software can be used to plan for replacement of unsupported software or allocation of resources for self-maintenance.

Table 4 describes the key attributes of this use case.

Table 4 - Use Case: EOL and non-maintained component alerting	
Actors	Security/Compliance/Risk management, Engineering/Development Teams, Program Managers, Open Source Program Office (OSPO), Procurement Office (for commercial components)
Business Motivation	Plan for deprecation of use and Risk Mitigation
Functional Objectives	Alert when a software component is reaching End Of Life (EOL) early enough for products and services to upgrade or replace the component

DRAFT

	before the event. Alert when an open source component is not actively maintained, early enough to either update the component or participate in the active maintenance of a component, either via upstream contributions or by forking and/or backporting.
Processes or Steps to Achieve Objectives	Set EOL dates for specific components based on communication from suppliers, release cycles, or general observations. Plan for EOL events within product or service release cycles.process Plan for issues that may arise from upgrades. Determine thresholds for active maintenance of open source components.
NTIA Fields Used	Supplier, Component Name, Version of the Component, Dependency Relationship, Timestamp
Added or Cross-linked Data	Product EOL status data Vulnerability Information Sources
Benefits Achieved	Reduce unexpected downtime due to software upgrade Increase resilience of software products

DRAFT

2.5 Use Case: Pre-purchase risk assessment

Prior to purchasing or acquiring software, various stakeholders in an organization–purchasing agents, contract officers, risk officers, network defenders and legal–may need to determine if the target software exposes the organization to risks. Analysis of the software's SBOM, and cross-referencing the SBOM data with other common sources, can reveal potential security, licensing, compliance or maintainability risks that the organization will inherit upon deploying the software. This analysis provides the organization with an opportunity to mitigate these risks prior to purchase, thereby reducing their risk exposure.

In addition, an organization's security, risk or vendor compliance teams can use information from their analysis of multiple SBOMs supplied by the same vendor (from different software applications or different versions of the same application) to assess the evolving risk level of the vendor's software and whether that risk is increasing or decreasing over time and whether additional mitigations are required.

Table 5 describes the key attributes of this use case.

Table 5 - Use Case: Pre-purchase risk assessment

Actors	Procurement, Purchasing, Contracting Officer, Risk/Compliance Officer, Legal, Security Teams, Network Defenders
Business Motivation	Avoid introduction of new security, compliance or supportability risks to the organization from software that is being considered for purchase or acquisition
Functional Objectives	Assess security risks of software to be acquired, Assess licensing risks of software to be acquired, Conform to regulations requiring SBOMs of suppliers Assess vendor risk based on multiple SBOMs supplied by the same vendor. Identify mitigations to reduce risk of new software being acquired
Processes or Steps to Achieve Objectives	Use components identified in SBOM to discover vulnerabilities in target software Determine if vulnerabilities pose risks to our organization Determine mitigations to reduce risk Identify licenses associated with components in target software Determine any risks associated with software licenses Determine if there are end of life (EOL) considerations for software components that introduce future risks related to maintainability,

DRAFT

	reliability or compatibility Determine if any components originate from sanctioned or prohibited suppliers Conduct risk scoring of potential acquired software leveraging the SBOM to assess its level of risk Use information from multiple SBOMs provided by the same vendor as input into a vendor risk score.
NTIA Fields Used	Supplier, Component Name, Version of the Component, Other Unique Identifiers, Dependency Relationship, Author of SBOM Data, Timestamp
Added or Cross-linked Data	EOL Licenses Information about Ownership and Controlling Interest Sanctions lists such as the U.S.Federal Communications Commission Covered Entities list [25] Vulnerability Information Sources
Benefits Achieved	Provides purchasing actors with information to assess risks from target software, and engage in risk-mitigating discussions with suppliers. Provides information for purchasing actors to disseminate to legal and risk/compliance officers to inform the risk assessment process. Identify risk mitigation actions

DRAFT

2.6 Use Case: Component usage across an organization

Organizations maintain software asset inventories to support engineering, security, compliance, legal and other functions. SBOMs extend these asset inventories down to the level of the components that comprise each software application. Organizations can use this component-level inventory to identify commonly-used software components across an organization and assess the impact of their prevalence on the organization's security and operational efficiency.

For example, if a commonly-used component is now associated with a zero-day or exploited vulnerability, its remediation by the security and engineering functions requires knowledge of every software application within the organization that depends on that vulnerable component. In another example, an organization that discovers a commonly-used component has unreliable supportability, may proactively allocate engineering resources for the component's maintenance.

An SBOM provides the initial component inventory that Engineering, Security, Incident Response, Compliance and other organizational functions can analyze and correlate with information such as vulnerabilities, licenses, End of Life (EOL) or End of Support (EOS) to assess risks from commonly-used components and efficiently plan for their mitigation, remediation, and ongoing support.

Table 6 describes the key attributes of this use case.

Table 6 - Use Case: Component usage across an organization

Actors	Engineering Management, Security Incident Response, Audit, Compliance, Governance, and Risk Management
Business Motivation	Asset management, evaluate long-term risk
Functional Objectives	Assess security and license risks across multiple software systems or an entire organization. Assess compliance and risk mitigation progress for an entire organization. Assess long-term risk incurred by adopting specific components or systems within an organization.
Processes or Steps to Achieve Objectives	Identify SBOMs that have been ingested for all software applications in use within the organization Identify the most commonly used software components across an organization. Identify security, licensing or EOL issues in software components that impact multiple software systems.

DRAFT

	Identify the prevalence of software components with different versions. Identify the prevalence of different components that serve similar purposes. Identify proliferation of vulnerable components across a complex system. Identify critical software components within an organization. Identify components which are EOL
NTIA Fields Used	Supplier, Component Name, Version of the Component, Other Unique Identifiers, Dependency Relationship
Added or Cross-linked Data	EOL Licenses Vulnerability Information Sources Inventory systems that include runtime data
Benefits Achieved	Provides engineering management with the ability to understand components that affect multiple software systems under their management Provides engineering management with insights to plan where more resources are needed to maintain pervasive software components. Provides audit and compliance personnel with an understanding of the organizational impact of findings about a software component. Provide Incident Response personnel information about the scale and impact of an incident associated with a vulnerable component.

DRAFT

2.7 Use Case: Incident response

SBOMs (Software Bill of Materials) can significantly enhance incident response processes by enabling faster identification, containment, and remediation of incidents, ultimately improving the overall security posture and resilience of organizations. Incident response teams can analyze their portfolio of SBOMs, e.g, by importing SBOMs into configuration management databases (CMDB) or Security Incident Event Management (SIEM) tools, to identify which systems and applications within their environment are affected by a vulnerable component or identify where shared weaknesses across suppliers is indicated in supplier cyber security attestations.

For instance, if a security incident is due to a vulnerability in a specific component, a repository of SBOMs could proactively alert for the new vulnerability and can easily be scanned or queried to identify all the applications actively using the vulnerable component. This process simplifies the identification and remediation effort, significantly reducing the mean time to detect (MTTD) for analysis and investigation. By streamlining these tasks, organizations can lower resource costs and allocate efforts more effectively, prioritizing and addressing affected vulnerable applications with greater efficiency.

Additionally, organizations can use SBOMs to implement immediate actionable steps as part of the incident response process, such as isolating affected systems or deploying temporary safeguards to mitigate the vulnerability's impact.

SBOMs can further improve incident response actions by providing a standardized framework for collaboration among Security Researchers, Software Publishers, and Vulnerability Coordinators. This common language facilitates the effective communication and disclosure of vulnerabilities, including sharing CVE status, enabling faster and more accurate resolution to security issues.

Table 7 describes the key attributes of this use case.

Table 7 - Use Case: Incident Response	
Actors	Teams performing: Incident Response, Engineering/Development, DevSecOps and IT, Security, Legal and Regulatory
Business Motivation	Respond swiftly and intelligently to cybersecurity incidents, Minimize business impacts from the incident, Minimize risks through proactive and effective actions, Improve the mean time to detect and address vulnerabilities, Reduce financial costs associated with incident response, Conserve resources by streamlining response efforts.

DRAFT

Functional Objectives	Empower the security incident response team to rapidly identify vulnerable components and the systems impacted by their usage, Enable software engineers or cyber defenders to take swift and effective actions to remediate the vulnerabilities.
Processes or Steps to Achieve Objectives	Identify root cause components and versions from SBOMs of the affected system. Use SBOMs associated with the affected system to evaluate incident impact. Compare the intended SBOM of the affected system to the current state of the system.
NTIA Fields Used	Supplier, Component Name, Version of the Component, Other Unique Identifiers, Author of SBOM Data, Timestamp
Added or Cross-linked Data	Multiple SBOM types (design, build, runtime) Software Attestations Vulnerability Information Sources Supplier Security Assertions (e.g., VEX)
Benefits Achieved	Reduction of mean time to detect (MTTD) is directly linked to resiliency of the organization and positively impacts vendor reputation and revenue.

DRAFT

2.8 Use Case: M&A and investment risk assessment

Organizations that acquire or invest in businesses must perform due diligence to identify risks in software developed by the acquisition target that might make the acquisition or investment problematic. For example, acquirers don't want to take on Intellectual Property (IP) or licensing compliance risks and may view unpatched source code as too risky. SBOMs provide a method to understand which components are used within the target software, their suppliers, and each component's license. Based on the suppliers and disclosed licenses, the acquisition team can determine if there are any IP conflicts or unresolved obligations and from there determine the cost to resolve or cure those issues.

Acquirers and investors also need to assess and manage risks from unpatched vulnerabilities in the target software and determine if the development team's dependency list is current. SBOMs containing components with many outstanding updates are potential signs of an immature update process. If the only components being kept up to date are those with vulnerabilities published in the NVD, then that implies the team prioritizes patching of public vulnerabilities above maintaining a current codebase.

Acquisition targets also benefit from sharing an SBOM in the early stages of an M&A effort, rather than source code, thereby reducing the. intellectual property risk of releasing source code too soon. An acquirer can verify the integrity of the target-supplied SBOM by using tools such as binary Software Composition Analysis (SCA) to extract an SBOM from applications and then compare that extracted SBOM against what was provided by the target.

Acquirers and investors can use information gathered through analysis of SBOMs to identify risks in the target software associated with its maintenance,security, licensing, IP, or regulatory compliance and use that risk assessment in an evaluation of the target business.

Table 8 describes the key attributes of this use case.

Table 8 - Use Case: M&A and investment risk assessment

Actors	Technical due diligence teams, Risk/Compliance Officer, Legal
Business Motivation	Ensure that the company targeted for acquisition or investment is free of liabilities from software security, licensing, IP or regulatory compliance issues. Ensure target's dependence on commercial, Original Equipment Manufacturer (OEM) or other third-party software will be maintained without interruption after acquisition. Determine acceptability of the target software's maintenance and updating practices.

DRAFT

Functional Objectives	Identify dependencies on open source, commercial and other third-party software which are critical to continued development and deployment of the target company's software products Determine potential software product liabilities related to target software's security, licensing, or commercial/OEM relationships Determine conformance to regulations, legislation or industry standards requesting "software inventory" information Assess adequacy of software maintenance practices
Processes or Steps to Achieve Objectives	Critical Dependencies: Analyze target company's SBOMs for inclusion of all components including open source, proprietary, commercial, and contracted software Analyze SBOMs to identify most commonly used open source, commercial and other third-party software in the target software For each commercial/OEM developer relationship, identify transferability of licenses Potential Liabilities: Analyze target company's SBOMs for CVEs Analyze target software's security advisories and security release documentation Determine if there are incompatible IP licenses in target software based on jurisdiction in the target software For each commercial/OEM supplier, analyze the SBOMs of their software for unpatched vulnerabilities and their security advisories For each commercial/OEM supplier, analyze the SBOMs of their software for appropriate licensing Regulatory Compliance: Analyze target software's SBOM to identify potential sanctioned or prohibited suppliers Analyze all commercial/OEM supplier software suppliers for supply chain risk assessments Many regulations pre-date SBOMs becoming mainstream and instead reference maintaining an inventory of software dependencies. SBOMs provide a method for compliance aligned with the spirit of such regulations. Software Maintenance: Determine average age of components Determine quantity of unpatched CVEs Determine SBOM generation process Determine frequency of new software versions

DRAFT

NTIA Fields Used	Supplier, Component Name, Version of the Component, Other Unique Identifiers, Dependency Relationship, Author of SBOM Data, Timestamp
Added or Cross-linked Data	SBOM should include all components, proprietary, commercial, contracted, COTS, and not just OSS Cross linked data should include provenance data, component licenses, conformance statements for 3rd party components, and any export/import compliance documentation for 3rd party components Where additional SBOM data is present it should be used as part of an existing process
Benefits Achieved	Reduce the cost of integrating a newly acquired company through transparency of development processes. Focus tech due-diligence efforts on areas with greatest risk to ongoing or future business operations Identify deal breaking or deal modifying risk elements early

DRAFT

2.9 Use Case: Verification of accessory software

Both Producers and Consumers must assess security and compliance risks from the total software product they produce or deploy. Many software products include *accessory software*–such as installers, download managers, runtime dependencies, or Software Development Kits (SDKs)–that facilitate installation, updates, or integration with other systems. Because these components are not directly compiled into core executables, they are often excluded from the core product SBOM. While accessory software may have their own SBOMs, these may be overlooked during security and compliance reviews of a core product's SBOM, thereby creating significant risks to the organization.

From a security perspective, installers or SDKs run with elevated privileges or fetch external code, making accessory software an attractive target for attackers. Unverified accessory code introduces an expanded attack surface, increasing the risk of exploitation. From a compliance standpoint, organizations must verify that no accessory software originates from sanctioned or prohibited suppliers. Even if the primary vendor is not sanctioned, the inclusion of prohibited components can result in non-compliance.

To address these concerns, SBOM analysis of both core and accessory software is required. This comprehensive approach ensures that all components with elevated privileges or external integrations are programmatically vetted–reducing the risk of false assurance, and helping maintain security, licensing, and compliance. Ultimately, the use cases for SBOM-driven verification of accessory software mirror those of the core product, reinforcing a holistic, end-to-end risk management strategy.

Table 9 describes the key attributes of this use case.

Table 9 - Use Case: Verification of accessory software

Actors	Security Teams, Incident Response Teams, Engineering Management, Regulatory, Governance, Vendor Management, Audit and Compliance
Business Motivation	Maintain organizational security posture and regulatory compliance Ensure vendor compliance with contractual terms and conditions Ensure compliance with terms and conditions of cyber insurance policies
Functional Objectives	Assess security risks of accessory software Assess licensing risks of accessory software Conform to regulations requiring SBOMs of suppliers Assess vendor risk based on multiple SBOMs supplied by the same vendor.

DRAFT

Processes or Steps to Achieve Objectives	Request complete inventory of all accessory software packaged with a software capability Confirm that the supplier has delivered SBOMs for each piece of accessory software and all relevant updates and service packs Analyze SBOMs to identify remote access utilities in accessory code for subsequent removal or monitoring
NTIA Fields Used	Supplier, Component Name, Version of the Component, Other Unique Identifiers, Dependency Relationship, Author of SBOM Data, Timestamp
Added or Cross-linked Data	Vulnerability Information Sources, Licenses, and other compliance-relevant databases of designated or precluded software.
Benefits Achieved	Supply chain visibility on all packaged components of an installed software product

DRAFT

2.10 Use Case: Differences in components between builds or versions

Producers, Distributors and Consumers of software need to understand, communicate and respond to risks emanating from software vulnerabilities, inappropriate licensing, non-compliance, and lack of support. However, risks involving software are not static; they change over time and need to be reassessed with new builds or versions of the software. Since an SBOM is a snapshot in time of the software, an SBOM produced for each major build or version of a software application represents a new snapshot of time. These new SBOMs can be analyzed for important changes in the risks posed by the software's components over time.

For example, a component with no known vulnerabilities at the time of the software's initial version may be associated with a newly discovered vulnerability in a future version. Another third-party or open source component that was adequately supported upon initial release may have diminishing support over time. Such changes can be tracked by analyzing SBOMs across builds or versions of the software. Information from the SBOMs can then be correlated with vulnerability or other information to understand when risks were introduced or identified as well as when those risks were mitigated or remediated.

Security, Engineering, Compliance, and Governance functions within both Producer and Consumer organizations can use these analyses to track progress on the introduction and remediation of risks, plan for reductions in software supportability, and maintain compliance over time.

Table 10 describes the key attributes of this use case.

Table 10 - Use case: Differences in components between builds or versions

Actors	Engineering (Development, Operations, Security), Engineering Management, Governance, Audit and Compliance
Business Motivation	Reduce exposure to vulnerabilities, ensure license compliance, compliance to internal policies, and maintain regulatory adherence.
Functional Objectives	Assess security and license risks over time. Assess compliance and risk mitigation progress over time.
Processes or Steps to Achieve Objectives	Identify changes in software components between builds or versions. Identify if changes in software components have fixed prior security, licensing or supportability issues or introduced new ones. Identify persistent or recurring security, licensing or supportability issues between builds and versions.

DRAFT

	Track software composition to correlate new risks with existing software. Track issues over time for metric based performance tracking, and compliance.
NTIA Fields Used	Supplier, Component Name, Version of the Component, Other Unique Identifiers, Dependency Relationship, Author of SBOM Data, Timestamp
Added or Cross-linked Data	EOL Licenses Vulnerability Information Sources Inventory systems that include runtime data
Benefits Achieved	Provides engineering with the ability to track progress on fixing vulnerabilities, addressing license concerns, and assessing supportability. Provides management with the ability to track progress in mitigating risks across builds and versions. Provides engineering management with insight into allocation of resources to address changes in risk. Provides audit and compliance personnel with the ability to pinpoint compliance findings and track progress towards compliance for a given software project/product. Provides Consumers with what vulnerabilities have been fixed between versions. Provides Consumers with information to update risks to their organization from new software versions and update vendor risk scores.

DRAFT

2.11 Use Case: Conformance with disparate Governance, Regulatory, and Compliance (GRC) specifications

Many software delivery contracts now require the delivery of an SBOM, each with particular specifications for content and detail. Producers must meet these requirements upon delivery of the software and Consumers must ensure they have been met prior to software acquisition and deployment.

Further, emerging regulations require SBOMs or similar software inventories to be delivered to Consumers and/or Regulators where the expectation is that the Producer and Consumer are using the SBOM for internal risk management. For example:

- The U.S.Food and Drug Administration (FDA) implemented changes made by Congress to the Food, Drug, and Cosmetic Act, creating section 524b [26] which requires medical device manufacturers to provide SBOMs for specified FDA regulatory filings. The FDA "Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket Submissions" [27] details the SBOM requirements.

- Article 13 (24) of the European Cyber Resilience Act (EU-CRA) (EU Regulation 2024/2847) [28], references an SBOM as an obligation of a manufacturer, and Annex I Part II (1) requires an SBOM be created with a minimum detail level comprising top level components. Details on SBOM format or elements will be published by December 2025.

- The U.S.Department of Defense (US DoD) Systems Engineering Standards and Specifications in DI-SESS-82433 as requested under the Department of the Army memo covering "Software Bill of Materials Policy" [29] detail SBOM data elements that go beyond that of both the NTIA minimum fields and available fields from the most recent respective SPDX or CycloneDX specifications.

Producers' internal technical, legal, and compliance teams collaborate on assessing, enriching and tailoring SBOMs to ensure conformance with various contractual or regulatory requirements. Data that is not covered in the initial, automatically-generated SBOM may be brought in from other sources to supplement or enrich the SBOM consistent with regulatory specifications.

Consumers, subject to their own GRC constraints, use the enriched Producer-supplied SBOM to assess the software's conformance with distinct regulatory and contractual requirements prior to deployment of the Producer's software or its inclusion in a Consumer's product..

Table 11 describes the key attributes of this use case.

DRAFT

Table 11 - Use Case: Conformance with disparate GRC specifications	
Actors	Risk/Compliance Officer, Auditors (internal and external), Legal, Board
Business Motivation	Meet internal governance processes and practices associated with legal and regulatory requirements within specific jurisdictions, market segments, or vertical sectors.
Functional Objectives	Provide an SBOM meeting specific requirements established by the jurisdiction that a Producer operates in, or provides their products or services in. Assess the risks documented within the Supplier SBOM as they relate to specific regulations and whether those risks present regulatory challenges to the Consumer
Processes or Steps to Achieve Objectives	Identify regulations that specify an SBOM requirement or reference a need for a software inventory Determine the level of SBOM data required (e.g., only direct dependencies) Identify if requisite data fields require supplementation from external data sources (e.g. date the commercial support agreement expires) For SBOMs from direct suppliers, identify if requisite information required for compliance is present, and implement policies for missing information or non-conformant components Determine disclosure requirements, including timeframes and locations for disclosure (e.g. within x days of a new release an SBOM must be uploaded to a designated repository) Based on the requirements of individual contracts and regulations, enrich with supplemental information or tailor the disclosed SBOM to meet the specific requirements of the regulation without excess disclosure Identify SBOM sharing constraints and the processes to ensure they are followed
NTIA Fields Used	SBOM should include data elements specified by the regulation, which may not reference NTIA minimum fields. Non-US based regulatory efforts may not reference NTIA elements. An SBOM can facilitate compliance with these regulations after required fields have been identified through a comprehensive review of the relevant laws and regulations.
Added or Cross-linked Data	Varies by regulation. To meet varied GRC requirements, Producers may be required to

DRAFT

	include all components, proprietary, commercial, contracted, COTS, and not just open source Cross linked data available for inclusion in a GRC oriented SBOM may also include ownership and control information, conformance statements for 3rd party components, support statements, incident response process documentation, and any export/import compliance documentation for 3rd party components
Benefits Achieved	Ensure that contractual and regulatory requirements are met by Producer and Consumer of software.

DRAFT

2.12 Use Case: Integrity and threat management for Operational Technology (OT) and isolated networks

Software deployed for critical infrastructure, industrial use, and high security environments, or on isolated networks typically doesn't follow a cloud-native or DevSecOps operations model where software updates might be pushed to systems to address security issues. While the software might be isolated, isolation doesn't preclude a need to understand dependencies and known security issues. Such knowledge becomes quite valuable for both update/upgrade process management, but also as part of threat management and compliance efforts.

As an example, suppliers of cyber-physical devices often manufacture those devices in batches and sell them over an extended period of time. As unsold devices are manufacturing inventory, they are not kept current with patches while they sit in inventory. Once sold and installed in an isolated environment, without an SBOM, it is difficult to determine if the newly installed device presents increased risk due to unresolved vulnerabilities or through weaknesses in end-of-life or obsolete dependencies.

An SBOM provides transparency of software dependencies by providing a comprehensive list of components, libraries, and potentially runtime tools used within and by an application. With appropriate transparency, SBOMs enable teams to standardize and streamline version and dependency management across network boundaries to minimize attack surface and other risks. This approach simplifies compliance and security review and ensures traceability, enabling repeatable packaging for software transfers between environments. For instance, artifacts and dependencies can be collected from sources like container image registries/repositories, source repositories, and local files, then bundled into an artifact repository for seamless deployment.

The process of using an SBOM to standardize dependencies within isolated network environments ensures the integrity of target environments by verifying that only declared components are gathered and distributed. By computing and including secure hashes for each component, the SBOM provides an additional layer of metadata that enables Consumers to verify the authenticity and integrity of the software bundle, safeguarding it against tampering or unauthorized modifications.

Table 12 describes the key attributes of this use case.

Table 12 - Use Case: Integrity and threat management for OT and isolated networks

Actors	Software Developers, Deployment Engineers, Security Engineers, DevSecOps, Maintenance, Compliance, and Legal
Business Motivation.	Enable secure software distribution in air-gapped environments. Facilitate controlled patching and deployment without Internet

DRAFT

	dependency. Reduce costs, streamline operations, and enhance security. Ensure legal and compliance for bundled software like container images or embedded systems. Simplify audits, mitigate risks, and build stakeholder trust. Maintain compliance with security and regulatory requirements. Faster incident response, and stronger resilience. Improve trust and traceability for mission-critical applications.
Functional Objectives	Declare components to be included in a data transfer between systems or networks Track configuration of systems post-transfer between systems and networks Identify security controls that are met prior to transfer as part of government's or organization's "Authority to Operate" (ATO) decision.
Processes or Steps to Achieve Objectives	Collect - Using component unique identifiers like PURLs, download components from package repositories, git repositories, and container registries. Process - Complete any validation checks on the downloaded components like antivirus checks, CVE lookup, or signature validation. Bundle - Bundle the downloaded and processed components to be moved across a network gap. Expand - Once the transferred bundle is available on the isolated network, expand and move the components to isolated package repositories or container registries.
NTIA Fields Used	Supplier, Component Name, Version of the Component, Other Unique Identifiers, Dependency Relationship, Author of SBOM Data, Timestamp
Added or Cross-linked Data	Licenses Component hashes Sanctioned or prohibited supplier lists Vulnerability Information Sources
Benefits Achieved	Users who develop or deploy on networks not connected to the public Internet have to certify that all components meet security controls as part of a government's or organization's Authority to Operate. SBOMs provide a clear path to understand what was introduced and allow you to verify everything that was included.

DRAFT

2.13 Use Case: Field servicing of software-enabled devices

Producers of software enabled devices such as medical devices, security sensors, and heating controls issue SBOMs upon product release. These devices are then deployed in an operational technology (OT) environment, where they are subject to field servicing and maintenance over the lifespan of the device - often measured in years or decades. Over that lifespan,, replacement hardware with associated firmware might be installed necessitating updated firmware, however, the device technician might not have full access to perform the necessary maintenance..

As part of routine maintenance and troubleshooting, field service representatives compare the source SBOM with data from operational devices, such as SBOMs, component hashes, or firmware revision comparisons, for maintenance and troubleshooting. They can also collect security and error logs to detect intrusions or failures before escalation.. For example, a medical device manufacturer managing patient monitors in hospitals has encountered unauthorized software on these devices, risking patient safety. A reference SBOM provides a baseline for identifying expected software, aiding maintenance and repairs.

By comparing the source SBOM and its expected contents, to data gathered from the deployed device, field service reps can identify changes in software components since the device's deployment. These differences may occur for a variety of reasons, such as: repair or replacement of essential components needed for the device's operation, addition of non-essential software to the device at the Consumer site, or malicious insertion of components. Identifying these differences can help a field service rep to diagnose problems in device operation and perform maintenance, and can be valuable input to the manufacturer's product security team.

Field service representatives can also compare the patch levels of components in the deployed device to the manufacturer-recommended patch levels to determine what needs to be upgraded in the deployed devices.

Since cyber-physical devices have long lifespans, SBOMs offer a means to reduce maintenance costs for both Producers and Consumers by ensuring that an accurate record of expected software within a device is maintained. Where field customization of device software is possible, SBOMs enable field service reps to determine whether the device is operating within normal or expected risk profiles and determine necessary mitigations.

Table 13 describes the key attributes of this use case.

Table 13 - Use case: Field servicing of software-enabled devices	
Actors	Producer Field Service Representatives, Product Security Team, Consumer IT Team, Consumer Security Team

DRAFT

Business Motivation	Maintain reliability of software-enabled devices Maintain supportability, performance and availability of software-enabled devices Reduce exposure to vulnerabilities from unnecessary components
Functional Objectives	Compare software inventory on a deployed device to the software inventory provided at the time of the device's release by the manufacturer
Processes or Steps to Achieve Objectives	Obtain most recently verified deployed device SBOM Generate SBOM for deployed device Identify differences between two SBOMs Determine potential reasons, both intended and unintended, for the differences Compare deployed device's patch level of components to those recommended by manufacturer Inform Producer's Product Security Team and Consumer's Security Team and IT team of any unintended additions to the device's software and any security issues in deployed device
NTIA Fields Used	Supplier, Component Name, Version of the Component, Other Unique Identifiers, Dependency Relationship, Author of SBOM Data, Timestamp.
Added or Cross-linked Data	Manufacturer's recommended patch levels for components Manufacturer's installation instructions Component hash
Benefits Achieved	Assures Consumer that software-enabled devices are performing reliably and securely Assures Consumer that device's software is at latest patch level Provides Producer's field service representatives with software status needed for maintenance

DRAFT

3.0 Key Takeaways

The following key takeaways distill the most salient points drawn from the thirteen use cases, as well as the supporting data referenced in Tables 15 and 16. They capture high-level insights, conclusions, or guiding principles arising from this document's analysis and are meant to inform an organization's strategic and operational practices in a way that is both broadly applicable and immediately actionable.

SBOM data, combined with external intelligence, greatly improves security and vulnerability management. By linking SBOM fields (such as Supplier, Component, Version, and Dependency Relationship) with vulnerability information sources, organizations can quickly assess and remediate new security issues. This continuous cross-linking accelerates patch decisions, enhances prioritization, and reduces mean-time-to-detect (MTTD) during incidents.

SBOM-driven workflows reduce compliance and licensing risks across disparate software environments. Whether it is open source license obligations or ensuring that a component's provenance does not violate policy, SBOMs help unify multiple compliance checks. By comparing the listed software components (in the SBOM) against license databases or sanctioned-entity lists, organizations can prevent costly legal risks and avoid supply-chain blind spots.

Operational efficiency gains arise from using SBOMs as a centralized inventory for multiple use cases. Many of the document's thirteen use cases (from pre-deployment CVE checks to post-deployment vulnerability monitoring) re-use the same SBOM fields for different analyses. Consolidating an organization's SBOM data (and the added/cross-linked data) cuts down on re-work, fosters consistent risk scoring, and speeds up vendor or acquisition-related decisions.

Comparing SBOM snapshots across builds or versions drives more effective maintenance and lifecycle planning. Several use cases and Tables 15–16 illustrate how "Version" and "Author" fields, coupled with EOL data, highlight differences across new builds, product lines, or major releases. This clarity helps teams track how risks evolve (e.g., newly introduced vulnerabilities or license changes) and plan upgrades or refactoring efforts more systematically. Each SBOM goes through a lifecycle where actors extracting intelligence and value from the SBOM may analyze, enrich, cross-link, merge, or do some other type of operation with it, depending on the use case.

SBOM lifecycle management Is becoming a specialized practice—but also a shared responsibility. As revealed by the breadth of data columns in Tables 15 and 16 and SBOM Lifecycle Management illustration, SBOM-driven risk management relies on cross-functional stakeholder inputs (Security, Engineering, Legal, Procurement, etc.). Tools and processes that facilitate data enrichment, merging of multiple SBOMs, or tailoring to GRC requirements will become increasingly central to an organization's software governance strategy. This is a promising field with many opportunities for automation and expansion in the future. For

DRAFT

example, several of the use cases use common operations such as enrichment, cross-linking, or merging, as part of the steps to extract value.

SBOMs require additional information to enable effective correlation. Many use cases require the SBOMs to have more data than the NTIA minimum elements list. They require the ability to cross-reference data in the SBOM with other datasets. This operation already takes place for vulnerability analysis and can be used for other use cases such as licensing compliance or alerting on End Of Life and non-maintained software components. However, in order to accurately perform the cross-reference, there should be some common naming convention or set of known naming conventions for each of the NTIA minimum elements.

Supply chain transparency and trust increase when SBOMs are enriched with key operational data. The tables show that risk assessments and licensing checks frequently draw on additional data beyond the SBOM's basic fields—namely EOL, legal attestations, and sanctioned-entity checks. Enrichment with this operationally relevant data creates a more trustworthy supply chain by making it clear where components come from and whether they are actively maintained.

Meeting regulatory and contractual requirements requires more than the minimum NTIA fields. As shown in Tables 15 and 16, NTIA's "minimum elements" (e.g., Supplier, Component, Version, Other Unique Identifiers) form a solid baseline but often need to be supplemented with added or cross-linked data—such as End of Life (EOL) dates, vulnerability exploitability (VEX) information, and ownership details—to satisfy emerging regulations (e.g., FDA, EU Cyber Resilience Act) or GRC specifications in specific industries.

In conclusion, this document illustrates why SBOM data should be viewed not as a static artifact but as a dynamic, multi-purpose tool for modern software risk management. Each takeaway emerges from real-world examples showing how enriched SBOM data can reduce costs, speed up incident response, inform licensing and compliance strategies, and ultimately foster deeper trust and reliability throughout the software ecosystem.

DRAFT

Table 15 - NTIA fields used by use case							
	NTIA Fields Used						
Use Case	Supplier	Component	Version	Other Unique Identifiers	Dependency Relationship	Author	Timestamp
2.1 Pre-deployment CVE vulnerabilities	✓	✓	✓	✓	✓		
2.2 Post-deployment CVE vulnerabilities	✓	✓	✓	✓	✓		
2.3 Open source licensing risks		✓					
2.4 EOL and non-maintained component alerting	✓	✓	✓		✓		✓
2.5 Pre-purchase risk assessment	✓	✓	✓	✓	✓	✓	✓
2.6 Component usage across an organization	✓	✓	✓	✓	✓		
2.7 Incident response	✓	✓	✓	✓		✓	✓
2.8 M&A and investment risk assessment	✓	✓	✓	✓	✓	✓	✓
2.9 Verification of accessory software	✓	✓	✓	✓	✓	✓	✓
2.10: Differences in components between builds or versions	✓	✓	✓	✓	✓	✓	✓
2.11 Conformance with disparate GRC specifications	✓	✓	✓	✓		✓	
2.12 Integrity and threat management for OT and isolated networks	✓	✓	✓	✓	✓	✓	✓
2.13 Field servicing of software-enabled devices	✓	✓	✓	✓	✓	✓	✓

Table 16 - Data added or cross-linked to SBOMs to meet use case objectives											
	Added or Cross-linked Data										
	Vulnerability Source										
Use Case	Database	Supplier	Licenses	EOL	Attestations	Runtime Data	Multiple SBOM Types	Ownership	Sanctioned Entity	Integrity	Vendor documenta-tion
2.1 Pre-deployment CVE vulnerabilities	✓	✓									
2.2 Post-deployment CVE vulnerabilities	✓	✓				✓					
2.3 Open source licensing risks			✓								
2.4 EOL and non-maintained component alerting	✓			✓							
2.5 Pre-purchase risk assessment	✓		✓	✓				✓	✓		
2.6 Component usage across an organization	✓		✓	✓		✓					
2.7 Incident response	✓	✓					✓				
2.8 M&A and investment risk assessment					✓				✓		✓
2.9 Verification of accessory software	✓		✓								
2.10: Differences in components between builds or versions	✓		✓	✓		✓					
2.11 Conformance with disparate GRC specifications					✓			✓	✓		✓
2.12 Integrity and threat management for OT and isolated networks	✓		✓						✓	✓	
2.13 Field servicing of software-enabled devices										✓	✓

4.0 References

[1] Cybersecurity and Infrastructure Security Agency. (2024, January). SBOM Community Legal Explanation.
https://www.cisa.gov/sites/default/files/2024-01/SBOM-Community-Legal-Explanation_508c.pdf

[2] National Telecommunications and Information Administration. (2019, November). Roles and Benefits for SBOM Across the Supply Chain.
https://www.ntia.gov/sites/default/files/publications/ntia_sbom_use_cases_roles_benefits-nov2019_0.pdf

[3] Bi, T., Xia, B., Xing, Z., Lu, Q., & Zhu, L. (2024). *On the way to SBOMs: Investigating design issues and solutions in practice*. ACM Transactions on Software Engineering and Methodology, 33(6), 1-25. https://doi.org/10.1145/3654442

[4] Cybersecurity and Infrastructure Security Agency. (2023, April 21) *Types of Software Bill of Material (SBOM) Documents*
https://www.cisa.gov/sites/default/files/2023-04/sbom-types-document-508c.pdf

[5] Cybersecurity and Infrastructure Security Agency. (2023, April 17). *Software Bill of Materials (SBOM) Sharing Lifecycle Report*.
https://www.cisa.gov/resources-tools/resources/software-bill-materials-sbom-sharing-lifecycle-report

[6] National Telecommunications and Information Administration, NTIA SBOM Formats & Tooling Working Group. (2021, November). *SBOM Tool Classification Taxonomy*
Ntia_sbom_tooling_taxonomy-2021mar30_0.pdf

[7] National Security Agency. (2023, December). *Recommendations for Software Bill of Materials (SBOM) Management*.CSI-SCRM-SBOM-MANAGEMENT.PDF (defense.gov)

[8] D'Amico, Anita and Zalevsky, Ken. (2024, October). *The Life of an SBOM: Where does it go and what do people do to it and with it?* BSidesNYC October 19, 2024. Slides posted at
https://www.linkedin.com/feed/update/urn:li:activity:7276685538156244993/

[9] Cybersecurity and Infrastructure Security Agency. (2024, February 5) *SBOM Sharing Primer.*
https://www.cisa.gov/resources-tools/resources/sbom-sharing-primer

[10] U.S. Dept of Commerce and National Telecommunications and Information Administration (2021, July). *The Minimum Elements for a Software Bill of Materials (SBOM).*
https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf

[11] SPDX is an ISO standard : https://www.iso.org/standard/81870.html

[12] CycloneDX is an ECMA Standard:
https://ecma-international.org/publications-and-standards/standards/ecma-424/

DRAFT

[13] Woody, C. (2024, February 5). Applying the SEI SBOM framework. *Software Engineering Institute*. https://doi.org/10.58012/5eh5-5862

[14] Indian Computer Emergency Response Team (CERT-In), Ministry of Electronics and Information Technology, Government of India. (2024, October 3). *Technical Guidelines on Software Bill of Materials (SBOM) Version 1.0*. https://www.interlynk.io/post/sbom-technical-guidance-for-india

[15] Berend Kloeg, Aaron Yi Ding, Sjoerd Pellegrom, Yury Zhauniarovich. (2024). *Charting the Path to SBOM Adoption: A Business Stakeholder-Centric Approach*. ASIA CCS '24: Proceedings of the 19th ACM Asia Conference on Computer and Communications Security Pages 1770 - 1783 https://doi.org/10.1145/3634737.363765

[16] National Telecommunications and Information Administration. NTIA Multistakeholder Process on Software Component Transparency. (2021, April 27). *SBOM Options and Decision Points*. https://www.ntia.gov/sites/default/files/publications/sbom_options_and_decision_points_20210427-1_0.pdf

[17] Cybersecurity and Infrastructure Security Agency. SBOM VEX Working Group. (2023, April). *Minimum Requirements for Vulnerability Exploitability eXchange (VEX)*. https://www.cisa.gov/sites/default/files/2023-04/minimum-requirements-for-vex-508c.pdf

[18] Wilkerson, J. (2023, May 31). FDA's Medical Device Cybersecurity Program and SBOM. Food and Drug Administration. https://csrc.nist.gov/csrc/media/Presentations/2023/fda-s-medical-device-program-and-sbom/images-media/JWilkerson-ssca-forum-053123.pdf

[19] CISA Community SBOM Sharing & Exchanging Working Group. (2024, March 22). SBOM Sharing Roles and Considerations. https://www.cisa.gov/sites/default/files/2024-03/SBOM%20Sharing%20Roles%20and%20Considerations.pdf

[20] National Telecommunications and Information Administration Multistakeholder Process on Software Component Transparency. (2021, February 10). *Sharing and Exchanging SBOMs*. https://www.ntia.gov/files/ntia/publications/ntia_sbom_sharing_exchanging_sboms-10feb2021.pdf

[21] General Services Administration (GSA) Office of Government-wide Policy. (2024, May 14). *Acquisition Letter MV-2023-02 Supplement 2.* https://www.gsa.gov/system/files/MV-2023-02%20w%20sups%201-2.pdf

[22] Linux Foundation. *SPDX License List*.https://spdx.org/licenses

[23] Open Source Initiative (OSI). *OSI Approved Licenses*. https://opensource.org/licenses

[24] Ecosyste.ms. *Licenses.* https://licenses.ecos__yste.ms

DRAFT

[25] Federal Communications Commission. *List of Equipment and Services Covered By Section 2 of The Secure Networks Act*. https://www.fcc.gov/supplychain/coveredlist

[26] U.S. Food and Drug Administration. (2024, March). *Select Updates for the Premarket Cybersecurity Guidance: Section 524B of the FD&C Act: Draft Guidance for Industry and Food and Drug Administration Staff.* https://www.fda.gov/regulatory-information/search-fda-guidance-documents/select-updates-premarket-cybersecurity-guidance-section-524b-fdc-act

[27] U.S. Food and Drug Administration. (2023, September). *Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket Submissions: Guidance for Industry and Food and Drug Administration Staff.* https://www.fda.gov/regulatory-information/search-fda-guidance-documents/cybersecurity-medical-devices-quality-system-considerations-and-content-premarket-submissions

[28] European Union. (2024, October 23). Regulation (EU) 2024/2847 on horizontal cybersecurity requirements for products with digital elements. https://eur-lex.europa.eu/eli/reg/2024/2847/oj/eng

[29] U.S. Department of Army memo. (2024, October 17). *Software Bill of Materials Policy.* https://api.army.mil/e2/c/downloads/2024/10/17/4072ab1e/asaalt-software-bill-of-materials-policy-signed.pdf

[30] Cybersecurity and Infrastructure Security Agency. SBOM Community Tooling and Implementation Working Group. (2024, October 15) *Framing Software Component Transparency: Establishing a Common Software Bill of Materials (SBOM) Third Edition.* https://www.cisa.gov/sites/default/files/2024-10/SBOM%20Framing%20Software%20Component%20Transparency%202024.pdf

DRAFT

5.0 Abbreviations

API - Application Programming Interface
ATO - Authority to Operate
CDRH - Center for Devices and Radiological Health
CDX - CycloneDX
CISA - Cybersecurity and Infrastructure Security Agency
CSAF - Common Security Advisory Framework
CSIRT - Consumer Security Team
CSV - Comma Separated Values
CVE - Common Vulnerabilities and Exposures
CVSS - Common Vulnerability Scoring System
DevSecOps - Development, Security, and Operations
EO - Executive Order
EOL - End of Life
EOS - End of Support
EPSS - Exploit Prediction Scoring System
FDA - Food and Drug Administration
FOCI - Foreign Ownership, Control, or Influence
GRC - Governance, Regulatory, and Compliance
GSA - General Services Administration
IP - Intellectual Property
JSON - JavaScript Object Notation
KEV - Known Exploited Vulnerabilities
M&A - Mergers and Acquisitions
MDS2 - Manufacturer Disclosure Statement for Medical Device Security
MTTD - Mean time to detect
NSA - National Security Agency
NIST - National Institute of Standards and Technology
NTIA - National Telecommunications and Information Administration
NVD - National Vulnerability Database
OEM - Original equipment manufacturer
OSPO - Open Source Program Office
OSI - Open Source Initiative
OSS - Open Source Software
OT - Operational Technology
PSIRT - Product Security Team
PURL - Package URL
SBOM - Software Bill of Materials
SCRM - Supply chain risk management
SDK - Software Development Kit
SDLC - Software Development Life Cycle
SPDX - Software Package Data Exchange
SSDF - Secure Software Development Framework

DRAFT

VDR - Vulnerability Disclosure Report
VEX - Vulnerability Exploitability eXchange
XML - eXtensible Markup Language

DRAFT

6.0 Terminology

Author - The Author reflects the source of the metadata, which could come from the creator of the software being described in the SBOM, the upstream Component Supplier, or some third-party analysis tool. Note that this is not the Author of the software itself, just the source of the descriptive data. [10]

Chooser - The Chooser is the person/organization that decides the software/products/Suppliers for use. [30] In this document the Chooser of a software/product/Supplier appears as the Purchaser, Procurement or Contracting Officer in Use Case #5 Pre-purchase risk assessment and the Acquirer or Investor in Use Case #8 M&A and investment risk assessment.

Component - A Component is a unit of software defined by a Supplier at the time it is built, packaged, or delivered. Many Components contain subcomponents, or upstream Components. Examples of Components include a software product, a library, or a single file.Depending on the perspective in the supply chain, a Component (often the Primary Component) can be considered to be a product, intermediate good, final good, or final assembled good. [30]

Consumer - The Consumer receives the transferred SBOM. This could include roles such as third parties, authors, integrators, and end users. [5] [19]

CycloneDX - A widely used, machine-readable, open-source SBOM format. The ECMA-424 CycloneDX Bill of materials specification can be found at https://ecma-international.org/publications-and-standards/standards/ecma-424/

Dependency - A Dependency is the relationship between two Components.

Distributor - A Distributor receives SBOMs for the purpose of sharing them with SBOM Consumers or other Distributors. [5] [19]

NTIA Minimum Elements - The minimum constituent parts of an overall SBOM that include Data Fields, Automation Support, and Practices and Processes. The minimum Data Fields are: Supplier Name, Component Name, Version of the Component, Other Unique Identifiers, Dependency Relationship, Author of SBOM Data, and Timestamp. [10]

Operator - An Operator is a person/organization that operates the software Component. [2] In this document the term Consumer can be viewed as an Operator.

SBOM Lifecycle - How an SBOM progresses from initial generation to consumption. It includes all phases in which an SBOM might be created, verified, analyzed, enriched, shared, and monitored.

DRAFT

Secure Store - A repository or system that houses SBOMs in a secure manner, ensuring integrity and limiting access to authorized parties.

Security Advisory - Information from a Supplier or security researcher about vulnerabilities or patches related to specific software. SBOMs can help identify which versions or components are affected.

Software Bill of Materials (SBOM) - An SBOM is a formal, machine-readable inventory of software Components and Dependencies, information about those Components, and their relationships. [30]

SPDX - A widely used, machine-readable, open-source SBOM format. The ISO/IEC 5962:2021 SPDX Specification V2.2.1 can be found at https://www.iso.org/standard/81870.html

Supplier - The Supplier refers to the originator or manufacturer of the software Component. [10]

System SBOM - A top-level SBOM that represents an interconnected collection of SBOMs. It links or merges multiple SBOMs (e.g., for different modules or microservices) to represent an entire product or solution.

Tailoring - Modifying an SBOM (e.g., by redacting proprietary data or adding GRC-specific fields) to meet contractual or regulatory requirements, or to address audience-specific needs. Typically performed prior to sharing the SBOM.

Use Case - A scenario illustrating how SBOMs can be used by a variety of stakeholders to benefit their organizations (e.g., for vulnerability scanning, license compliance). This document presents thirteen use cases.

Verification - Process to ensure the SBOM contains the elements required by legislation, regulation, industry standards, and/or contract, and that it conforms to the appropriate specification.

VEX (Vulnerability Exploitability eXchange) - A standardized format (often JSON, XML, or CSAF) used by a Supplier to clarify exploitability of known vulnerabilities for their specific product. Frequently cross-referenced with the SBOM to confirm if a CVE truly impacts a given component.

Vulnerability Information - The term “Vulnerability Information” as used in this document’s use cases is an expansive term covering all potential information required to contextually meet the use case objectives. For example, vulnerability information could be a CVE identifier that is stored in a shared location such as the National Vulnerability Database (NVD) or country-specific NVDs, or where a private entity publishes a shared information source, such as GitHub Security Advisory (GHSA) identifiers. These identifiers often serve as an index into additional or augmented information sources such as the CISA Known Exploitable Vulnerability (KEV) catalogue or the Exploit Prediction Scoring System (EPSS) information.

DRAFT

7.0 Acknowledgments

The acknowledgments provided in this document do not imply endorsement of its content.

The leaders of the SBOM Operations Working Group express their sincere gratitude to all members of the group for their valuable time, expertise, and commitment to achieving the team's objectives. We also extend our thanks to our hosts, the CISA SBOM team including Allan Friedman and Victoria Ontiveros, for their support in convening this industry-led working group and their active participation in many sessions.

This effort would not have commenced without the original leadership of Nisha Kumar (Oracle), who established the group, alongside her co-leads Deanna Medina (United Airlines) and Ricardo A. Reyes (Chainguard). Leading community groups requires significant time and energy, and this responsibility was later shared by Bunny Hernández Banowsky (SHE BASH) and Anita D'Amico (Cotopaxi Consulting and Vigilant Ops).

The drafting, outlining, and technical review of this document—from its inception to the final version—demanded a unique ability to organize and inspire. Anita D'Amico took on this pivotal role, effectively "herding cats" to encourage and guide the SBOM Operations experts in producing clear, specific, and actionable content addressing the central question: "How can an SBOM bring value to my organization?"

Below, we acknowledge the considerable contributions of the working group members who served as primary authors of the document sections or served as designated technical reviewers to critically review specific sections. We also acknowledge SBOM Community contributors who provided significant feedback during the open comment period. These comments offered perspectives and insights that enriched the quality of the final document.

DRAFT

Authors	Designated Technical Reviewers	Other Contributors
Bunny Hernández Banowsky (SHE BASH) Anita D'Amico (Cotopaxi Consulting, Vigilant Ops) Ian Dunbar-Hall (Lockheed Martin) Bill Hansen (Hansen Enterprises LLC)) JC Herz (Exiger) Nisha Kumar (Oracle) Tim Mackey (Black Duck) Mike Lieberman (Kusari) Victoria Ontiveros (CISA) Anusha Penumacha (Splunk) Ricardo Reyes (Chainguard) Ken Zalevsky (Vigilant Ops)	Bunny Hernández Banowsky (SHE BASH) Cassie Crossley (Schneider Electric) Anita D'Amico (Cotopaxi Consulting, Vigilant Ops) JC Herz (Exiger) Nisha Kumar (Oracle) Tim Mackey (Black Duck) Mike Lieberman (Kusari) Bob Martin (MITRE) John Nuckles (ODNI) Victoria Ontiveros (CISA) Kayra Otaner (Roche) Animesh Pattanayak (PNNL) Vijaya Ramamurthi (Accenture Federal Services) Ricardo Reyes (Chainguard) Ria Schalnat (HPE) Anant Shrivastava (Cyfinoid Research) Gaurav Srivastava (Siemens) Ken Zalevsky (Vigilant Ops)	Ralph Bean (Red Hat) John Cavanaugh (ProCap360) Brindusa Curcaneanu (NeuroPace) Anthony Harrison (APH10) Charlie Hart (Hitachi America, Ltd.) Syed Zaeem Hosain (Aeris Communications, Inc.) Philippe Ombredanne (AboutCode.org, Package-URL, and nexB Inc.) Melissa Rhodes (Medtronic) Duncan Sparrell (sFractal)

DRAFT