DATA NORMALIZATION CHALLENGES AND MITIGATIONS IN SOFTWARE BILL OF MATERIALS (SBOM) PROCESSING

A WHITE PAPER FOR MEDICAL DEVICE MANUFACTURERS

October 2024

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD®

This white paper was prepared by The MITRE Corporation under contract with the U.S. Food and Drug Administration. The views, opinions, and findings contained in this white paper do not constitute agency guidance, policy, recommendations or legally enforceable requirements. Utilizing the information presented in this document does not constitute compliance with any requirements of the Federal Food, Drug, and Cosmetic Act, or any other applicable law.

Contents

List of Tables

1 Introduction

Software Bills of Materials (SBOMs) have emerged as key building blocks in software security and software supply chain risk management, as they allow for comprehensive risk management, including management of the risks that those software components pose to the device. They enable taking proactive actions to mitigate risks in the device during development and reactive actions to expeditiously control emerging risks in fielded devices. SBOM "is effectively a nested inventory, a list of ingredients that make up software components. An SBOM identifies and lists software components, information about those components, and supply chain relationships between them [1]."

However, the process of developing robust SBOMs has non-technical (e.g., process and governance) and technical content-creation challenges. Non-technical challenges include developing processes for obtaining SBOMs from third-party components, both commercial and open source; managing SBOMs over the software lifecycle (including the need to maintain SBOMs for all currently supported versions of the software); and selecting appropriate tools for generating and exchanging SBOMs. Technical challenges include interoperability between different SBOM standards, handling missing information, imprecise definitions of SBOM elements, multiple formats for SBOM elements (e.g., component name, version), and difficulties with ingesting/parsing data in producing SBOM elements. Generating SBOMs at scale requires automation, which in turn requires the ability to ingest information from build systems, SBOM-generation tools, and SBOMs delivered by component vendors and open source projects. A major challenge in ingesting this information is data normalization, using a standard nomenclature and formats to ensure that data from various sources is consistent.

This white paper is directed to medical device sector stakeholders discussing considerations of data normalization for producing SBOMs, SBOM ingestion at scale, and related issues.

1.1 Approach

MITRE conducted a landscape analysis to understand the data normalization challenges in generating SBOMs and identify potential mitigations.

MITRE reviewed the products developed by the National Telecommunications and Infrastructure Administration (NTIA) and Cybersecurity and Infrastructure Security Agency (CISA) community-led initiatives to define SBOMS and how they are generated and used.[1]

MITRE also conducted interviews with a broad sample of stakeholders, including CISA, the U.S. Food and Drug Administration (FDA), large Medical Device Manufacturers (MDMs), small MDMs, MDM trade associations, cybersecurity and regulatory consultants, participants in SBOM standardization efforts, and SBOM tool vendors.

In addition, we surveyed the underlying technical infrastructure. We reviewed the specifications of the two widely used SBOM standards, Software Package Data Exchange (SPDX) [2] and CycloneDX [3] and standards that may be used in creating

[1] See Section 2 for a discussion of these initiatives.

SBOM content, such as Common Platform Enumeration (CPE) [4] and Package Uniform Resource Locator (PURL) [5] for unique identifiers, and Semantic Versioning for component version. We examined the SBOM tools listed at the SPDX[2] and CycloneDX[3] websites to categorize the tools and their capabilities. We also reviewed approaches to data normalization used in various technologies, including databases and data science.

1.2 Structure

The following sections describe the NTIA and CISA initiatives in software transparency and SBOMs, challenges in generating SBOMs at scale, data normalization issues, and recommendations for addressing these challenges.

2 NTIA and CISA Initiatives

Since 2018, NTIA and CISA have convened community-led efforts to define SBOMs; understand approaches to generating, consuming, and managing SBOMs; and promote adoption. The work products of these efforts are foundational to understanding the challenges of generating SBOMs at scale, including data normalization. They define the baseline SBOM attributes, which is where data normalization issues arise; characterize SBOM tooling, which both contributes to normalization issues and offers potential solutions; and address some of the ambiguity in the standards, which contributes to data normalization issues. This section describes the NTIA Multistakeholder Process on Software Component Transparency and the CISA SBOM initiative, in order to provide a foundation for the remainder of the paper.

2.1 NTIA Multistakeholder Process on Software Component Transparency

In 2018, NTIA launched the Multistakeholder Process on Software Component Transparency. The goal of this effort was to “explore how manufacturers and vendors can communicate useful and actionable information about the third-party software components that comprise modern software and IoT devices, and how this data can be used by enterprises to foster better security decisions and practices [6].”

The NTIA Multistakeholder Process developed an SBOM model defined a set of minimum elements and additional elements, explored different use cases, and investigated SBOM formats and translations between them. In addition, MDMs and healthcare delivery organizations (HDOs) established a Healthcare Proof of Concept working group to demonstrate producing and consuming SBOMs for different use cases in the healthcare context.

[2] https://spdx.dev/use/tools/

[3] https://cyclonedx.org/tool-center/

The NTIA Multistakeholder Process produced several documents for advancing software component transparency,[4] including:

- *Framing Software Component Transparency: Establishing a Common Software Bill of Materials (SBOM)* [1], which defines SBOM concepts, provides an SBOM model, and describes processes for creating and sharing SBOMs (henceforth called "NTIA Framing Document").
- *Software Identity: Challenges and Guidance* [7], which describes the challenges of uniquely identifying software components.
- *SBOM Tool Classification Taxonomy* [8], which categorizes the various types of SBOM tools.
- *How-To Guide for SBOM Generation* [9], which is the Healthcare Proof of Concept's playbook for generating SBOMs.

2.2 CISA

In 2021, CISA assumed the lead for advancing "the SBOM work by facilitating community engagement, development, and progress, with a focus on scaling and operationalization, as well as tools, new technologies, and new use cases" and is bringing together resources on their SBOM website.[5]

Initially several workstreams were established to enable community-driven evolution and refinement of SBOMs, focusing on the Vulnerability Exploitability eXchange (VEX) model, sharing and exchanging SBOMs, promoting SBOM adoption, tooling and implementation, and SBOMs in cloud environments. After these workstreams produced initial products, a more streamlined approach to SBOM evolution was adopted. Members of the community can propose short-term tiger teams focused on very specific problems. A weekly community meeting was set up to share information across workstreams and tiger teams.

Work products developed by these workstreams have been reviewed in developing this paper, including:[6]

- Amendment to SBOM NTIA Framing Document.
- CISA SBOM Contents - Pragmatic Expectations.
- Draft of the Third Edition of the NTIA Framing Document, which incorporates the Pragmatic Expectations working document.
- SBOM Tool Criteria spreadsheet.

[4] All the work products produced by the NTIA Multistakeholder Process on Software Transparency can be found at https://www.ntia.gov/page/software-bill-materials

[5] This section summarizes the CISA-facilitated community-led activities documented on the CISA SBOM website (https://www.cisa.gov/sbom).

[6] Final workstream products are published on CISA's SBOM Resource Library (https://www.cisa.gov/topics/cyber-threats-and-advisories/sbom/sbomresourceslibrary). For information about workstreams and draft products, contact SBOM@cisa.dhs.gov.

- Criteria for evaluation and cataloging of Software Bill of Materials Tooling document.

In addition to the workstreams, CISA facilitates SBOM-a-Ramas, which are meetings (in-person, virtual, or hybrid) to help the broader community understand the current state of SBOM and the current SBOM community efforts, including the "CISA-facilitated community-led work."

2.3 SBOM Baseline Attributes

The NTIA Framing Document defines a set of baseline attributes "that can be used to identify components and their relationships [1]." Data normalization issues arise because of inconsistencies in the content and format of these attributes. For reference, Table 1 lists the baseline attributes described in the second edition of the document.[7] Some of the attributes provide meta-information, while other attributes apply to the software components; some attributes are required, and others are recommended.[8]

Table 1: SBOM Baseline Attributes

Attribute Name	Attribute Type	Description
Author Name	Meta-information	Author of the SBOM
Timestamp	Meta-information	Date and time when the SBOM was last updated
Supplier Name	Component Attribute (required)	Name or other identifier of the supplier of a component in an SBOM entry
Component Name	Component Attribute (required)	Name or other identifier of a component
Version String	Component Attribute (required)	Version of a component
Component Hash	Component Attribute (recommended)	Cryptographic hash of a component
Unique Identifier	Component Attribute (required)	Additional information to help uniquely define a component
Relationship	Component Attribute (required)	Association between SBOM components

The third edition of the NTIA Framing Document was released in late 2024. The differences in the baseline attributes across the two editions are described in Table 4.

3 FDA Policy

The U.S. FDA has recognized the importance of SBOMs in managing postmarket software vulnerabilities in medical devices and providing transparency to the users of these devices since the 2018 *Medical Device Safety Action Plan* [10], including

[7] The *Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket Submissions* guidance document (issued in 2023) [11] recommends that SBOMs include the NTIA baseline elements from version 2 of the NTIA Framing Document.

[8] Executive Order 14028 on Improving the Nation's Cybersecurity (issued May 12, 2021) directed the Department of Commerce, in coordination with the Assistant Secretary for Communications and Information and the NTIA Administrator, to issue a minimum set of elements for an SBOM. The necessary Data Field elements in the *Minimum Elements for a Software Bill of Materials (SBOM)* [22] are a subset of the Framing Document's baseline attributes.

considering the need for additional regulatory authorities in this space. These authorities were granted in Section 3305 in the Consolidated Appropriations Act, 2023, which added Section 524B "Ensuring Cybersecurity of Medical Devices" to the Federal Food, Drug, and Cosmetic (FD&C) Act. This provision, among other requirements, requires SBOMs (Section 524B(b)(3)) as part of premarket submissions for cyber devices. The 2023 guidance, *Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket Submissions* (henceforth called "premarket cybersecurity guidance") [11], provides, among other things, FDA's recommendations on using SBOMs to manage cybersecurity risks.

The premarket cybersecurity guidance discusses SBOMs as a "tool to help manage supply chain risk" by "providing a mechanism to identify devices and the systems in which they operate that might be affected by vulnerabilities in the software components [11]."

The premarket cybersecurity guidance recommends manufacturers provide machine-readable SBOMs consistent with the NTIA baseline attributes from version 2 of the NTIA Framing Document described in Section 2 of this paper.

In addition, the FDA recommends including additional information for each software component, which the FDA states may be provided separately from the SBOM. This includes:

- The software level of support from the component manufacturer (e.g., actively maintained, no longer maintained, etc.) and
- The end-of-support date.

Furthermore, FDA recommends a vulnerability assessment of software components. For components with known vulnerabilities, FDA recommends manufacturers provide:

- Safety and security risk assessment of each known vulnerability (including device and system impacts); and
- Details of applicable safety and security risk controls to address the vulnerability.

4 Data Normalization Challenges

During our landscape analysis, we identified several data normalization challenges in generating SBOMs, especially at scale. In this section we first discuss the underlying factors that contribute to data normalization challenges, then we discuss the challenges and considerations that are common across SBOM elements, and finally we discuss the challenges and considerations specific to individual SBOM elements (both the baseline attributes and the additional information recommended in the FDA premarket cybersecurity guidance).

4.1 Factors Leading to Challenges

4.1.1 Maturity of SBOM Adoption

Although Bills of Materials (BOMs) are widely used in industrial supply change management, creating BOMs for software is relatively recent. There are multiple machine-readable data formats (i.e., SPDX, CycloneDX, and SoftWare IDentification [SWID]) and various tools to help generate, share, and process SBOMs. The initiatives

described in Section 2 are developing frameworks for defining SBOM content, translating between formats, evaluating tooling, and defining levels of maturity, all of which will help users navigate the challenges of producing and managing SBOMs. With increasing emphasis on SBOMs to help mitigate supply chain and software development lifecycle risks and aid in vulnerability management, standards, tools, and processes are rapidly evolving.

Organizations that produce SBOMs are at various stages of maturity. Some are just starting to pilot processes, while others may be on their second or third approach. Companies may have product lines at different levels of maturity. MDMs need to be aware that they may be receiving SBOMs from third-party suppliers at different levels of maturity, which will contribute to the challenges of normalizing the SBOM attributes from these different sources.

4.1.2 Complexity of SBOM Generation Processes

The components of a software system may be open source software, commercial off the shelf (COTS) software purchased from third parties, internally developed code, etc. The SBOM data about these multiple artifacts, in turn, may be collected in various ways:

- Received from supplier.
- Manual creation from product development documentation and product teams.
- Created by build tools during development.
- Produced by package managers.
- Running software composition analysis tools against source code or binaries.

This diversity of information can lead to data normalization issues since different suppliers, internal product teams, and tools may use different nomenclature, data formats, and conventions. See the following subsections in Section 4.

In some cases, such as with Software of Unknown Pedigree (SOUP), the same component could be in multiple products, but different tools or techniques may be used for each product, causing the same component to be treated differently.

4.1.3 Complexity of MDM Product Lines

Multiple product lines may also introduce challenges in managing SBOMs that cause data normalization issues. Different products may include the same third-party components. However, product development teams may acquire SBOM data through different mechanisms, use different tools and repositories to manage the SBOM information, and use different naming and formatting conventions. In addition, larger MDMs might use common proprietary code across multiple product lines, in essence becoming a third-party supplier for the product lines, with all the associated data normalization challenges.

Even smaller MDMs with a single product may need to manage multiple SBOMs, leading to potential data normalization challenges. This product may consist of multiple components, such as software/firmware installed on the device, a cloud application, and a mobile application. These components are developed in different development

environments that may generate SBOMs with different tools or manual processes, which may lead to the use of different nomenclature and data formats.

4.1.4 Limitations of Standards

The different SBOM standards (i.e., SPDX and CycloneDX) specify the elements that can appear in SBOMs, and the NTIA Framing Document identifies a set of baseline attributes and maps them to fields in SPDX and CycloneDX. The content and formats for these attributes are not fully specified or described, which contributes to the use of multiple naming and formatting conventions. In some cases, the NTIA Framing Document offers more specific suggestions for attribute content.

4.2 Issues Affecting Multiple SBOM Elements

This section covers issues that can occur across many different SBOM elements. The subsequent section identifies specific issues within individual elements.

4.2.1 Content/Format

SBOMs can vary widely depending on factors such as how the SBOMs were generated and shared, which can introduce inconsistencies that cause normalization problems. Contents might vary simply because of the format being used. Table 2 includes items that introduce inconsistencies and contribute to normalization issues.

Table 2: Normalization Issues for Multiple SBOM Elements

Item	Description
Case Sensitivity	There can be variations in data with respect to case – uppercase, lowercase, or mixed, such as: • UPPERCASE (only capital letters). • lowercase (only small letters). • Mixed Case (mixture of capital and small letters). While case variations typically do not pose problems for human readers, it is important that these be accounted for in automated processing. Unless SBOM-processing code is written or directed to ignore case differences, even simple string comparisons can cause mismatches. In some situations, differences in case may identify different items (e.g., “abc.txt” and “ABC.TXT” point to different files in Unix-oriented file systems).
Abbreviations	There can be data variations in how (and whether) abbreviations are used. For example: • “Win2K” vs. “Windows 2000” vs. “Windows 2K.” • “Corp.” vs. “Corporation.” • “NFC” vs. “Near Field Communication.”
Word Separators	Some data elements might have variations in whether distinct words are combined or separated by spaces or punctuation. This appears to happen most frequently for supplier names, product names, and some versions. For example: • “Product Name” (space separated). • “Product-Name” (hyphen separated). • “ProductName” (combined with camel case).

Item	Description
Punctuation	Data could contain variations in when punctuation marks are used, and which ones are used. For example: • Microsoft smart quotes. • Single quotes vs. double quotes. • Em dashes (—), en dashes (–) , hyphens (-), etc. In some cases, punctuation marks might be omitted in one data item element and used in an equivalent data element.
Non-Latin Characters	For data that may contain non-Latin characters, there may be variations in whether (and how) the data is converted to Latin-only characters. These strings do not appear to match based on strict comparison. A conversion routine might preserve the character as-is; remove the character and use a visually similar character; or convert to a multi-character sequence that does not contain the original character. For example, “Brücke” (bridge) might handle the umlaut (ü) as “Bruecke” or “Brucke.”
Trademark or Copyright Symbols	The symbols that signify trademarks or other protections can vary in how they are represented, such as “(tm)” vs. “™”. In other cases, they might be omitted entirely.
Directory Separators	In cases in which file or directory resources are used, the separators can vary (e.g., “/” on *nix vs. “\” on Windows). This can cause incorrect mismatches.
Acronyms	Acronyms might be spelled out, used alone, or combined (e.g., “ACME Bow Company [ABC]).”

4.2.2 Encoding

SBOMs do not all use the same universal encoding, such as the widely adopted UTF-8. At each point in the processing or transfer of an SBOM, inconsistencies can arise with respect to which encoding is being used—or which encoding is assumed to be in use. Encoding inconsistencies can cause data to be inadvertently transformed in ways that do not preserve the initial data, often producing what appears to be garbage characters in the middle of data that otherwise seems normal. The implications for matching and normalization are clear, since it can make it easier to mismatch data that should be treated as equivalent. In addition, normalization techniques might attempt to reverse bad or unexpected encoding, but these techniques would need to be programmed or manually performed, neither of which is efficient at scale.

While differences between encodings were not described during interviews, this problem is frequently encountered during data sharing between multiple organizations or data sources, and it appears likely for any large-scale SBOM processing.

4.2.3 Date/Time

Dates or time can be represented or parsed in different ways. This can lead to inconsistencies in date representation or parsing that may prevent tools from matching dates that are otherwise equivalent. This can also cause incorrect conversions to produce inaccurate or nonsensical dates.

Many different date/time formats can be used in SBOMs. For example, *March 17, 2024* could be represented many different ways: “2024-03-17” (International Organization for

Standardization [ISO] 8601 style); “March 17, 2024;” “2024 Mar 17;” “03/17/24;” “3/17/24;” “03/17/2024;” “17/03/2024;” “2024/03/17;” “03-17-2024;” etc. Month-only or year-only dates may be provided (e.g., “March 2024,” “2024,” etc.).

While ISO 8601 is a widely adopted standard for date and time formats, it is not universally used. It is not necessarily required by commonly used SBOM formats or recommendation documents. For example, NTIA’s “Framing Software Component Transparency” cites ISO 8601 as an example of a “common international format” for timestamps. In CycloneDX, the (eXtensible Markup Language (XML) schema definition (XSD) supports a standard string-based date representation, but its JavaScript Object Notation (JSON) specification represents timestamps using strings without specifying their format, since JSON itself does not directly support dates or times.

The granularity of dates or times can vary widely as well. Some SBOM data might only report a year and month, or even just a year, whereas other data might include full times down to the millisecond. While such precision might not be necessary for most logic that operates on SBOMs, the differences in text strings will further complicate processing.

When the names of months or days are used instead of numbers, additional conversion challenges may arise based on the language used (e.g., “Avril” [French] instead of “April”).

4.2.4 Missing Data

While this document’s focus is not on “quality” considerations within SBOMs (e.g., missing data), quality needs to be considered when addressing normalization challenges. For example, when multiple component SBOMs need to be analyzed so that they can be combined into a larger SBOM, there might be a technique to detect duplicate components; however, if there is missing data, then the duplicates might not be detectable.

4.2.5 Multiple “Sources of Truth”

Each tool that produces or consumes SBOMs may act as a “source of truth” because its own outputs are self-consistent. However, it is essential for users who process SBOMs that were created by different tools to address variations across the different tools. That is, there is no single “source of truth” for many data elements, so variations can arise depending on which source is used.

- Proprietary components that will never be publicly available (e.g., shared code across multiple business lines).
- Automated mechanisms will generate different results (e.g., use of machine learning to match product names will vary based on training sets and the resulting models).

4.2.6 Changes Over Time

SBOMs and the elements they include can change regularly over time, in ways that can introduce normalization problems.

If these changes are not closely monitored and addressed, they may hinder attempts to identify and remove duplicate information.

For example, names can change over time for:

- Organizations (due to rebranding, mergers and acquisitions, etc.).
- Products or components (due to rebranding, forks in open source, etc.).

Products whose development has been abandoned ("abandonware") might be separately maintained by different sources in minimal ways over time, causing small variations that cannot be as easily tracked as forked repositories.

Other elements or data can change over time, such as format, encoding, date formats, etc. The same source of an SBOM could make changes that downstream tools do not recognize, causing inaccurate results or processing errors, potentially forcing additional development to handle the changes.

4.3 Normalization Problems for Specific SBOM Elements

This section identifies other difficulties that are specific to individual baseline attributes.

4.3.1 Component Name

With different standards in place for representing components (e.g., CPE and PURL) the same product and version could have a CPE representation in one SBOM, and a PURL representation in another SBOM. This equivalence would need to be resolved by a normalization task; otherwise, the same component in an SBOM could be listed multiple times.

MDMs need to track the components that they have built in-house or outsourced to external developers. For MDMs with multiple product lines, this can require dedicated staff and/or a centralized team to ensure that component information remains consistent across all products.

4.3.2 Supplier Name

The same supplier could be represented with different names that will need to be unified when processing multiple SBOMs.

There can be variations in how an organization name is provided, even when the spelling and encoding are correct and consistent, such as:

- Suffixes indicating the type of organization may be present (e.g., "Co." or Limited Liability Company "LLC").
- Some suffixes might include full words or abbreviations (e.g., "Co." or "Company").
- The organization's legal name might include a prefix that is commonly omitted (e.g., "The XYZ Company" might be a legal name that is commonly listed only as "XYZ Company").
- Affiliations might be included or excluded (e.g., an organization X might be a subunit of a parent organization Y, but the organization name could be listed as "X," "Y," "X, a Y Company," etc.).

There can be multiple distinct supplier names for the same product. For example, with open source, there could be a legal entity that protects the product, which may be separate from the “project” or “foundation” that develops and distributes the product. This might happen more often in open source than in COTS. There are also trade names (i.e., “DBA” or “doing business as”) that are different from the legal name. Whether the legal name or an alternate name is reported in SBOMs can vary based on the tools or methodologies used to generate them.

4.3.3 Version

There are many variations in how product versions are named, identified, and cited, as illustrated in Table 3.

Table 3: Normalization Issues for Version Attribute

Data	Description
Numbers	Various numeric schemes can be used. For example: • Semantic versioning (e.g., “4.10.8”). • Ad hoc schemes.
Dates	Some versions are closely tied to dates, whose format can vary widely (see Section 4.2.3).
Code Names	Project code names are frequently used and might be linked with numeric versions, such as “Jaunty Jackalope” for Ubuntu Linux 9.04 or “Monterey” for Apple MacOS 12.
Version Indicators	Version numbers are frequently indicated with labels or prefixes such as “version,” “v,” “v.,” etc. This happens especially within free text and might not be removed or standardized during automated conversion to structured data. The presence or absence of these version indicators can introduce important inconsistencies.
Git Hashes/Tags	Many open source packages are managed using the git version control system. Git hashes and tags act as labels that can be tied to “versions” of code, even as the code is being actively maintained in a central repository.

4.3.3.1 Wildcards vs. Specific Versions

Version information might be recorded at different levels of granularity, which can cause problems when attempting to match version information from multiple sources. For example, one source could include a long list of specific versions, whereas another source could use a wildcard, which could be interpreted as a distinct difference even if they were semantically identifying the same set of versions.

There can also be variations in how versions are phrased that can lead to different interpretations (e.g., “6.x before 6.10” and “all versions before 5.8”).

Versions might be represented or reported differently within dependency lists as recorded by package managers.

In some cases, it is possible that the exact version is unknown, and only an estimate is available, especially with abandonware, SOUP, or acquired components without source code.

Automated tools such as Software Composition Analysis might vary in how they handle cases in which the specific version cannot be unambiguously identified, possibly by inferring the major version and only reporting that, whereas there might be more authoritative information about the same component in a separate SBOM.

Inconsistencies in versions can lead to multiple entries for the same component or cause multiple components to be treated as only one component.

4.3.3.2 Timestamps

Versions can sometimes include timestamps or dates, which can be subject to the same differences as listed in Section 4.2.3.

4.3.4 Dependency Relationships

Even if multiple sources report the same dependency information for a component, the representation can vary based on the tool or methodology that was used. This can make it difficult to merge dependency information from multiple sources or to determine when there are inconsistent results that need further analysis to resolve.

Dependency data typically identifies relationships between different components or packages using structured data, generally symbolic in nature, with names for these relationships that can differ based on the tools or other methods used to generate the dependency data. Relationships might be implicitly stated. While translations between well-known formats such as CycloneDX and SPDX may significantly reduce the number of variations in how dependencies are represented, users still need to account for other data representations that they might encounter in SBOMs that were not generated by such tools.

Besides relationship names or indicators, there can be variations in how dependency trees or graphs are represented. They could be structured with nested syntax in which components are implied by the nesting, or they could have a flatter structure where dependencies are explicitly identified, for example, by using references to components that are fully defined elsewhere in the SBOM. Again, while formats such as CycloneDX and SPDX likely have translation tools that account for these differences, some SBOMs may use other representations that can be dealt with on an ad hoc basis.

Variations in depth might also lead to difficulties in normalization. For example, if different tools are used to generate SBOMs for components that include the same sub-component, there might be inconsistencies in how the sub-component is identified. This can be further exacerbated in more complex software such as operating systems, in which the same sub-component could be reported differently for different parts of the operating system although it could be argued that differences in completeness in SBOMs pose a separate fundamental problem.

4.3.5 Other Issues with Identifiers

For identifiers involving supplier and component names, standards like CPE and PURL have limitations that cause complications in SBOM processing. Simply put, no standard provides a single, unique identifier for all components that might be covered by an SBOM. This leads to gaps and normalization challenges as some components may lack the necessary identifiers.

New CPE identifiers are only provided by National Institute of Standards and Technology (NIST) as needed, when vulnerabilities are discovered in components. As a result, CPE identifiers may not be available for certain public components without known vulnerabilities, or private, custom in-house components. To address this gap, organizations may maintain some private identifiers that follow the CPE format but are not NIST-provided CPEs.

With PURL, there is a chance that the same component could have multiple names.

Even though there is likely significant overlap between CPE and PURL in the range of components that can be represented, users who process SBOMs from multiple sources need to contend with integrating both identifier schemes, as well as any other ad hoc schemes.

4.3.6 Additional Elements

The following SBOM elements are not recommended in the NTIA Framing Document baseline attributes. However, since the FDA premarket cybersecurity guidance recommends including them in premarket submissions and labeling, SBOM suppliers and consumers might encounter difficulties in normalization for these elements.

4.3.6.1 Support Level

In the premarket cybersecurity guidance, FDA recommends that MDMs provide "the software level of support provided through monitoring and maintenance from the software component manufacturer (e.g., the software is actively maintained, no longer maintained, abandoned) [11]." The level of support for a product and/or its components may use inconsistent language that varies based on the supplier and the support level (e.g., some MDMs might use "support" and other might use "maintenance"). The level of support might be described using tier level with different definitions. These differences would need to be accounted for when collecting this information from suppliers and providing it to the FDA.

4.3.6.2 End-of-Life and End-of-Support

FDA recommends that MDMs include both End-of-Life (EOL) and (End-of-Support) EOS in Section V.A. Security Risk Management and Section VI. Cybersecurity Transparency in their premarket submission and labeling. However, both may be affected by variations in time and date, as described in more detail in Section 4.2.3. If the same product or component is listed multiple times with variations (e.g., when combining SBOMs), then MDMs might want to investigate the apparent inconsistencies, as they could turn out to be false positives. In addition, the terms "end-of-life" and "end-of-support" themselves might vary based on the SBOM supplier, which can prevent MDMs from being able to automatically find and extract this data.

4.3.7 Vulnerability Information

As discussed in Section 3, FDA recommends that MDMs conduct a vulnerability assessment of software components and provide safety and risk assessments, along with applicable controls, for each vulnerability. Normalization issues in elements such as product names, versions, or component names that are critical for vulnerability management (i.e., recording) and exchanging vulnerability information, can contribute to

inaccuracies in this vulnerability assessment and in overall vulnerability management. Namely, normalization issues could lead to false negatives (in which an asset has a vulnerability, but it is not reported) or false positives (in which an asset is incorrectly claimed to have a vulnerability when it does not).

5 Mitigations

Section 4 describes the types of data normalization issues that can arise when generating SBOMs, as well as some of the sources leading to these issues. This section provides general recommendations to address these challenges, including technical mitigations for SBOM attributes, policies, and processes to help an organization manage SBOM generation, and recommendations to advance the SBOM ecosystem as a whole.

5.1 Technical Mitigations

This section describes technical mitigations to address data normalization challenges, which may be implemented by available commercial or open-source SBOM tools, or through ad hoc scripts or parsers developed by the MDM (see Section 5.1.2). Sections 4.2 and 4.3 provide specific examples of data normalization issues that may help guide the development of in-house scripts and parsers, or the acquisition of third party SBOM tools.

5.1.1 Use Canonical Names and Representations

A common approach to handling multiple names/representations for data elements, used in pre-processing data for data analysis and natural language processing (NLP),[9] is to create a set of canonical names/representations and map the actual names/representations appearing in the raw data against them. The mapping can be done by maintaining an alias database, parsing inputs (e.g., using regular expressions), fuzzy matching, and other techniques. The canonical names are then used in the processing required for specific use cases to provide consistency.

When defining canonical names, authoritative sources, both internal to the MDM and external, can be leveraged. For example, external corporate registration databases or internal contracting or lifecycle management databases can be used to identify the legal corporate name of a supplier. Additional development effort may be necessary to export identifier data from these databases, and it is recommended that the quality of these sources be validated prior to integrating them.

The NTIA Framing Document suggests different formats and sources for obtaining the content for the baseline attributes. MDMs may wish to consider these suggestions as they define the canonical names/representations in their SBOMs. The normalization challenges discussed in Section 4 above describe some of the issues to consider when developing canonical names and representations. This includes multiple SBOM elements related to content and format—case sensitivity, abbreviations, word separators, punctuation, non-Latin characters, trademark/copyright symbols, directory separators, and acronyms (Section 4.2.1); encoding (Section 4.2.2); date/time

[9] For example, see [18] and [19] for normalization in NLP and [22] for normalization in information retrieval.

representation (Section 4.2.3); changes in data over time (Section 4.2.6); component/supplier names (Sections 4.3.1 and 4.3.2); versions (Section 4.3.3) including number schemes, dates, code names, version indicators, git hashes or tags, timestamps, and handling of wildcards; dependency relationships (Section 4.3.4); support level (Section 4.3.6); and issues in sharing of vulnerability information (Section 4.3.7).

Automated mechanisms for identifying aliases and producing matches could save significant manual effort and provide consistency. Some MDMs and other SBOM producers use “fuzzy matching” techniques to minimize necessary labor. However, any such mechanism would need to address false positives (i.e., matching two items that are actually distinct) and false negatives (i.e., failing to match two items when they are referring to the same thing).

5.1.2 Tooling

MDMs may be able to acquire or develop tools that help them with normalization issues, including matching data items; detecting and resolving encoding differences; and translating between different formats. Parsers, regular expressions, and ad hoc scripts can be useful in matching data elements and normalizing formats, but they may require human analysis to handle when they fail or produce different results. Various “fuzzy matching” techniques can be powerful, but they can be subject to false positives and false negatives. MDMs may need to develop ad hoc scripts to address limitations of available tooling (e.g., insufficient tooling for embedded C/C++ software). Some of the considerations in Section 4 can be used to guide the development of parsers and ad hoc scripts.

It may be necessary to plan for manual resolution and review to address the high likelihood of false positives or false negatives in any solution. Internal metadata could be developed to assist the automation and ensure reproducibility. For example, for hashes, metadata could identify the object that was hashed (source file, binary, tar file, disk image) and the algorithm used. MDMs could design processes with feedback loops in which automation errors are detected and catalogued; humans diagnose the causes of the errors; and the tools are subsequently fixed to avoid such errors in future runs.

Some existing tools may help identify issues for resolution, such as software-analysis services that are regularly executed automatically as part of development and build processes. Such tools might already be used in the MDM’s development/build processes, but they might not be explicitly advertised as supporting SBOM production and consumption. The CISA Tooling working group is developing documents defining attributes for evaluating and categorizing tools for generating and processing SBOMs, which will be able to help MDMs assess the capabilities of different tools and the impacts of using these tools on data normalization.

5.1.3 Baseline Attributes and FDA Additional Information

This section provides suggestions for mitigations for specific SBOM baseline attributes identified in the NTIA Framing Document and the additional information recommended in the FDA premarket cybersecurity guidance.

Software Identifiers

- As previously described, CPEs might not be available for components or products that are internal to the MDM or has never had a Common Vulnerabilities and Exposures (CVE) assigned, since the National Vulnerability Database (NVD) only assigns CPE IDs for products and components that have known vulnerabilities. To address this gap, the MDM could create an internal list of pseudo-CPEs that follow the CPE format and semantics.
- PURLs could be useful since they support a flexible hierarchical namespace.
- SoftWare Heritage persistent IDentifiers (SWHIDs)[10] support creation of "core identifiers... that are guaranteed to remain stable (persistent) over time."
- OmniBOR[11] (Universal Bill Of Receipts) can be used by build tools to produce dependency graphs and unique identifiers.

MDMs do not have to bear the burden of addressing identifier gaps alone. Industry-led organizations or services could stand up their own shareable repository that covers industry-wide gaps in identifier schemes such as CPE. Such industry-wide repositories could be used by the MDM and other healthcare stakeholders and tool providers, thus helping solve normalization challenges. For example, the Health-ISAC and CyBeats recently stood up an SBOM repository available to Health-ISAC members, including MDMs and HDOs.[12] This repository might be able to produce the CPEs and other software identifiers from the SBOMs in its database and provide a central listing of these identifiers that can be used by MDMs in generating their SBOMs. Another possible solution for addressing identifier gaps would be to adopt a federated approach, similar to how the CVE Numbering Authorities operate, and have the suppliers of software components publish the identifiers for their software when it is released. Then some organization, industry led or government (e.g., CISA, NIST), could "roll up" these identifiers into a federated search portal and make it available to all SBOM generators.

MDMs might be able to take advantage of public mappings between ID schemes (e.g., between PURLs and CPEs). However, it would likely still be important for them to maintain their own identifiers for their own components, especially private components that do not have any significant external identity to consumers.

Versions

A strategy for resolving discrepancies between versions will help minimize normalization problems. Differences such as punctuation/spacing separators may be easy to resolve from a technical perspective, but it may be more difficult to resolve cases in which different sources claim different sets of versions. For any logic that resolves version discrepancies, it is recommended to prefer the supplier's version information and map back to the source. However, a mechanism to note the discrepancy may be necessary, since some information may be incomplete or out-of-date from the supplier (e.g., in the case of unsupported versions, SOUP, or newly-discovered vulnerabilities).

[10] https://docs.softwareheritage.org/devel/swh-model/persistent-identifiers.html

[11] https://omnibor.io/

[12] https://h-isac.org/cybeats/

Supplier Names

Logic will need to account for cases in which supplier names change over time (e.g., by recording aliases). For example, the URGENT/11 set of vulnerabilities occurred in an older component that was acquired from the original manufacturer by a supplier.[13] However, there was confusion because the vulnerabilities were also detected in products that did not use that supplier. Recording aliases for supplier names can help address these discrepancies, although please note that automated "search and replace" may be a naive solution that generates too many incorrect equivalences (e.g., when only part of a supplier is acquired or sold to another supplier, or when an open source project is forked so that there are two seemingly-equivalent but distinct suppliers).

EOL and EOS dates

In addition to the NTIA baseline attributes, FDA recommends MDMs include end-of-life and end-of-support dates in their submissions. Since suppliers may use different names for "end-of-life (EOL)" and "end-of-support (EOS)," it is recommended to adopt the *Health Industry Cybersecurity – Managing Legacy Technology Security* (HIC-MaLTS) [12] and International Medical Device Regulators Forum (IMDRF) [13] terminology and map supplier-specific terms against them.

5.2 Policy and Process Mitigations

Although technical solutions are important in mitigating data normalization challenges, MDMs, especially those with multiple business and product lines, may want to consider adopting policies and processes to help mitigate data normalization issues.

5.2.1 Centralized Services and Repositories

MDMs are likely to consume and generate multiple SBOMs, either because their medical device includes multiple products, or it includes one product with multiple sub-components. For example, a device could contain an embedded component that is monitored with a mobile application in which data is transmitted and managed using cloud services. MDMs may also need to manage sub-components, both third-party and internally developed, that may be used by multiple business lines. One solution for ensuring consistency across these multiple SBOMs and minimizing data normalization issues is to develop centralized services and repositories across business lines:

- A "Central Alias Database" (CAD) could be maintained to map component names, supplier names, and other names from different sources to the single primary source.
- Internal sources of canonical names can be integrated into the CAD or an application programming interface to retrieve the information from those internal sources can be developed.

[13] [22] provides an overview of URGENT/11 and its supply-chain complexities.

- The ability to extend a central source to include all internal and third-party components is important.
- External sources of aliases might be available to help populate this database. These sources would have their own formats, so development would need to account for these formats; however, there would be a significant reduction in labor compared to populating the database from scratch. It is suggested that the quality of external sources be validated before using them.

It is recommended that MDMs develop consistent naming conventions for their own proprietary software (including software developed in-house or by contractors), and consistent processes for the baseline attributes. If multiple business lines produce or consume SBOMs in different ways, this can introduce normalization issues within the MDM itself, because the MDM might use different names or aliases, even for its own products. The use of pseudo-CPEs or other internal identifier schemes could help maintain consistency. When creating internal identifiers, it is recommended that MDMs ensure consistency across attributes, (e.g., that the component name in a pseudo-CPE [unique identifier baseline attribute] aligns with the component name baseline attribute).

5.2.2 Include SBOM Expectations in Contracting Language

It is suggested that MDMs update contracting language with suppliers to require provision of SBOMs in machine-readable formats. This will help the MDM to produce a better picture of all dependencies instead of having significant gaps in coverage or using resource-intensive methods of creating SBOMs for the third-party components themselves.

The timing for updating contracting language may depend on the nature of the relationship with the supplier. With new suppliers, contracting language could be updated immediately and included as boilerplate. For existing suppliers, the language could be adjusted when the contract needs to be renewed.

MDMs could include specific requirements for a well-defined SBOM format that is easy for machines to parse, use of good “sources of truth” for different baseline attributes that are relatively comprehensive, etc. Note that requirements that minimize normalization problems are not necessarily part of existing sample contract language. For example, provision of SBOM and minimum elements is covered in *Model Contract-language for Medtech Cybersecurity* by the Healthcare & Public Health Sector Coordinating Council [14], but it does not include language related to normalization.

5.2.3 Evolve SBOM Processes

As discussed in Section 4, the generation and use of SBOMs are relatively immature and evolving. MDMs may want to consider their approaches to responding to SBOM evolution, as well as evolving their internal SBOM processes.

Since SBOM processes and tooling are evolving rapidly, vigilant monitoring of the product space may help to anticipate and manage changes at scale.

- Design SBOM processes and procedures to reduce dependency on specific tool vendors and solutions. As tool capabilities evolve, they might address—or worsen—normalization challenges.

- Define criteria for tool capabilities, leveraging NTIA and CISA frameworks, and include how tools perform or support normalization in those criteria.[14]
- Ensure that strategic planning includes active monitoring and adoption of new standards or authoritative sources. Consistent internal mappings can facilitate mapping to new authoritative names.
- Define consistent processes and document them to facilitate adapting to changes.

Some considerations for MDMs evolving their SBOM generation, use, and management processes include:

- Regularly evaluate SBOM generation processes to identify gaps and issues needing resolution. Use these lessons learned to inform tool acquisition requirements and process enhancements.
- Develop a roadmap for SBOM adoption to scale generation of SBOMs. Consider guidelines that present options according to levels of maturity, such as the Third Edition of the NTIA Framing Document, and include support for data normalization as one aspect of maturity.
- Start producing SBOMs early in product development. It may be necessary to modify the SBOM during the development lifecycle if components are swapped out, but it will be easier to produce the product's SBOM than if one waits until the end of development. Early SBOM production will also allow more time to identify and address normalization issues.

5.3 Evolve the SBOM Ecosystem

Sections 5.1 and 5.2 present technical and process-oriented mitigations to help an individual MDM address data normalization issues arising when generating SBOMs. Section 4.1 discusses some of the underlying factors contributing to data normalization challenges, including the current immaturity of SBOM creation and management, interoperability issues with SBOM tools, ambiguity in standards, and the rapid development of the SBOM technologies and processes. This section describes some of the steps that can be taken to evolve the SBOM ecosystem and address some of the underlying systemic data normalization challenges.

Tools make it possible to produce SBOMs at scale, but also are a major source of data normalization issues because each tool may produce data elements that are incompatible and misaligned with the data elements produced by other tools. Further, whether tools are developed in-house or purchased, it is expected that they will evolve rapidly. The SBOM ecosystem might consider conducting tool "bake-offs" or competitions, similar to the approach taken by the Defense Advanced Research Projects Agency (DARPA) to improve natural language processing, the National Institute of Standards and Technology (NIST) to improve text retrieval technology, and Health Level Seven International (HL7) to foster Electronic Health Record (EHR)

[14] See the NTIA *SBOM Tool Classification Taxonomy* [8] and current work by the CISA Tooling work group.

interoperability and exchange.[15] These bake-offs would have different tools generate SBOMs for the same products, and the SBOM outputs could be compared and analyzed with respect to desired requirements, interoperability, and potential normalization issues. For example, the resulting SBOMs could be converted to a normalized output for comparison, either natively by the tools or through customized tooling to support the bake-off evaluation. These bake-offs could be conducted by individual MDMs; or MDMs could encourage the CISA SBOM Community or membership organizations, such as Health-ISAC and the Healthcare and Public Health Sector Coordinating Council (HSCC) Cybersecurity Working Group, to conduct them to broadly benefit the ecosystem.

Since there are industry-wide gaps in tooling and capabilities—including normalization challenges produced by tools—industry-led organizations and other stakeholders could proactively encourage improvements to tooling. Two tooling gaps were identified while researching this paper:

- Embedded systems implemented in C/C++ (most SBOMs for this technical environment are generated manually).
- Manual SBOM authoring and editing tools; for example, the component might be a type of technology that does not have mature tooling support or there may be cases when the component is no longer supported, but the component is part of essential performance and cannot be immediately replaced. In addition, manual editing may be needed to resolve issues with tool-generated SBOMs.

These industry-led organizations may work with government partners, such as CISA, NIST, Advanced Research Projects Agency for Health (ARPA-H), and others, to try to influence industry to develop such tools.

Although improving SBOM tooling is critical, there are other activities that can evolve the ecosystem to address some of the other underlying challenges. The content of SBOM data fields may not be sufficiently specified in the various standards, for the SBOM data formats and for the content of specific fields. The ambiguities in these standards can be addressed in new versions or in supplementary documentation and examples. MDMs who wish to influence the direction of these ongoing standards efforts may consider joining the standards groups working in these areas. More generally, the CISA SBOM Community tiger teams are addressing a wide range of issues in the creation, use, sharing, and management of SBOMs, some of which may intersect with data normalization challenges. Again, MDMs who want to influence the direction of these tiger teams, or propose new tiger teams to address other gaps, may wish to consider participating in the CISA SBOM Community.

Finally, industry-led organizations can provide centralized sources of information and services to address key data normalization challenges. Section 5.1.3 described this in the context of gaps in identifiers, but other challenges can be addressed by leveraging existing organizations, such as Health-ISAC and HSCC.

[15] DARPA sponsored the Message Understanding Conference, NIST sponsors the Text Retrieval Conference, and HL7 organizes the HL7/(Fast Healthcare Interoperability Resources (FHIR) Connectathon.

6 Conclusion

SBOMs are a powerful tool in software security and software supply chain risk management of medical devices. However, as detailed throughout the paper, normalization issues hinder the effectiveness of the generation and consumption of SBOMs. To be used effectively, SBOM data, especially the baseline attributes and additional data recommended in the FDA premarket cybersecurity guidance, needs to be normalized using a consistent nomenclature and data formats.

This paper has discussed the factors that contribute to data normalization challenges and specific data normalization issues, both those common across baseline attributes and those involving individual attributes.

To address those challenges and promote effective use of SBOMs, the paper also provides an overview of mitigations, including technical recommendations (i.e., building upon the discussion of data normalization issues for the various data elements) and process and policy recommendations.

Finally, the paper includes suggestions for evolving the SBOM ecosystem, including ways to improve tooling and mature SBOM standards. It encourages MDMs and software developers to take an active role in influencing industry-led organizations to provide information sources and services that can make it easier for individual organizations to address their data normalization challenges.

7 References

[1] NTIA Multistakeholder Process on Software Component Transparency Framing Working Group, "Framing Software Component Transparency: Establishing a Common Software Bill of Materials (SBOM), 2nd edition," 21 October 2021. [Online]. Available: https://www.ntia.gov/sites/default/files/publications/ntia_sbom_framing_2nd_edition_20211021_0.pdf.

[2] The Linux Foundation's SPDX Working Group, "The Software Package Data Exchange® (SPDX®) Specification Version 2.2.2 (ISO/IEC 5962)," August 2021. [Online]. Available: https://spdx.github.io/spdx-spec/v2.2.2/.

[3] OWASP Foundation CycloneDX Core Working Group, "CycloneDX Specification (Version 1.5)," June 2023. [Online]. Available: https://github.com/CycloneDX/specification/tree/1.5.

[4] NIST, "Four New Reports Update Security Content Automation Protocol," September 2011. [Online]. Available: https://www.nist.gov/news-events/news/2011/09/four-new-reports-update-security-content-automation-protocol.

[5] package-url project, "Package URL specification v1.0.X," [Online]. Available: https://github.com/package-url/purl-spec/blob/master/PURL-SPECIFICATION.rst.

[6] NTIA, "NTIA Software Component Transparency," [Online]. Available: https://www.ntia.gov/other-publication/2021/ntia-software-component-transparency.

[7] NTIA Multistakeholder Process on Software Component Transparency Framing Working Group, "Software Identification Challenges and Guidance," 30 March 2021. [Online]. Available: https://www.ntia.gov/sites/default/files/publications/ntia_sbom_software_identity-2021mar30_0.pdf.

[8] NTIA SBOM Formats & Tooling Working Group, "SBOM Tool Classification Taxonomy," 30 March 2021. [Online]. Available: https://www.ntia.gov/sites/default/files/publications/ntia_sbom_tooling_taxonomy-2021mar30_0.pdf.

[9] NTIA Software Transparency Healthcare POC, "How-To Guide for SBOM Generation," 2021. [Online]. Available: https://www.ntia.gov/sites/default/files/publications/howto_guide_for_sbom_generation_v1_0.pdf.

[10] U.S. FDA Center for Devices and Radiological Health, "Medical Device Safety Action Plan: Protecting Patients, Promoting Public Health," 16 April 2018. [Online]. Available: https://www.fda.gov/about-fda/cdrh-reports/medical-device-safety-action-plan-protecting-patients-promoting-public-health.

[11] U.S. FDA Center for Devices and Radiological Health, "Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket

Submissions," 27 September 2023. [Online]. Available: https://www.fda.gov/media/119933/download.

[12] HSCC Cybersecurity Working Group, "Health Industry Cybersecurity: Managing Legacy Technology Security (HIC-MaLTS)," March 2023. [Online]. Available: https://healthsectorcouncil.org/wp-content/uploads/2023/03/Health-Industry-Cybersecurity-Managing-Legacy-Technology-Security-HIC-MaLTS.pdf.

[13] IMDRF, "Principles and Practices of Cybersecurity for Legacy Medical Devices (IMDRF/Cyber WG/N70Final:2023)," 11 April 2023. [Online]. Available: https://www.imdrf.org/documents/principles-and-practices-cybersecurity-legacy-medical-devices.

[14] HSCC Cybersecurity Working Group, "Health Industry Cybersecurity - Model Contract-language for Medtech Cybersecurity," March 2022. [Online]. Available: https://healthsectorcouncil.org/wp-content/uploads/2022/05/HSCC-Model-Contract-language-for-Medtech-Cybersecurity-2022.pdf.

[15] NTIA, "Software Bill of Materials," [Online]. Available: https://www.ntia.gov/page/software-bill-materials.

[16] CISA, "Software Bill of Materials (SBOM)," [Online]. Available: https://www.cisa.gov/sbom.

[17] T. Duque, "Text Normalization: Why, what, and how," 2 April 2020. [Online]. Available: https://towardsdatascience.com/text-normalization-7ecc8e084e31.

[18] D. L. Yse, "Text Normalization for Natural Language Processing (NLP)," 17 February 2021. [Online]. Available: https://towardsdatascience.com/text-normalization-for-natural-language-processing-nlp-70a314bfa646.

[19] U.S. FDA Center for Devices and Radiological Health, "Select Updates for the Premarket Cybersecurity Guidance: Section 524B of the FD&C Act," 13 March 2024. [Online]. Available: https://www.fda.gov/regulatory-information/search-fda-guidance-documents/select-updates-premarket-cybersecurity-guidance-section-524b-fdc-act.

[20] L. H. Newman, "Decades-Old Code Is Putting Millions of Critical Devices at Risk," 1 October 2019. [Online]. Available: https://www.wired.com/story/urgent-11-ipnet-vulnerable-devices/.

[21] C. D. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[22] U.S. Department of Commerce, "The Minimum Elements for a Software Bill of Materials Pursuant to Executive Order 14028 on Improving the Nation's Cybersecurity," 12 July 2021. [Online]. Available: https://www.ntia.gov/sites/default/files/publications/sbom_minimum_elements_report_0.pdf.

Appendix A NTIA Framing Document Baseline Attributes

The FDA premarket cybersecurity guidance references *Framing Software Component Transparency: Establishing a Common Software Bill of Materials (SBOM)*, Second Edition, October 21, 2021. The Third Edition was published in late 2024. Table 4 shows the differences in the baseline attributes across the two versions. The Third Edition contains additional attributes, which are not included in the SBOM recommendations in the premarket cybersecurity guidance. Two of the attributes have slightly different names, but readily map to each other, and the two editions of the Framing Document contain tables mapping the attributes to CycloneDx and SPDX. It will be straightforward to use the baseline attributes defined in the Second Edition in generating SBOMs even if tools start to use those defined in the Third Edition.

Table 4: Comparison of Baseline Attributes in 2nd and 3rd Editions

2nd Edition Name	3rd Edition Name
Author Name	Author Name
Timestamp	Timestamp
	Type
	Primary Component
Supplier Name	Supplier Name
Component Name	Component Name
Version String	Version
Component Hash (optional)	Cryptographic Hash
Unique Identifier	Unique Identifier
Relationship	Relationship
	License
	Copyright Holder

Even though the FDA premarket cybersecurity guidance refers to the Second Edition, the Third Edition addresses some of the ambiguity found in the earlier editions and suggestions for content of the various baseline attributes, which may aid in mitigating some of the data normalization challenges.

Appendix B Abbreviations and Acronyms

Term	Definition
ARPA-H	Advanced Research Projects Agency for Health
BOM	Bill of Materials
CAD	Central Alias Database
CISA	Cybersecurity & Infrastructure Security Agency
COTS	Commercial Off the Shelf
CPE	Common Platform Enumeration
CVE	Common Vulnerabilities and Exposures
DARPA	Defense Advanced Research Projects Agency
DBA	Doing Business As
EHR	Electronic Health Record
EOL	End of Life
EOS	End of Support
FDA	Food and Drug Administration
FD&C	Federal Food, Drug, and Cosmetic Act
FHIR	Fast Healthcare Interoperability Resources
HDO	Healthcare Delivery Organization
HIC-MaLTS	Healthcare Industry Cyber - Managing Legacy Technology Security
HL7	Health Level Seven International
HSCC	Healthcare Sector Coordinating Council
IMDRF	International Medical Device Regulators Forum
ISAC	Information Sharing and Analysis Center
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
LLC	Limited Liability Company
MDM	Medical Device Manufacturer
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NTIA	National Telecommunications and Infrastructure Administration
NVD	National Vulnerability Database
OmniBOR	Universal Bill of Receipts
PURL	Package Uniform Resource Locator
SBOM	Software Bill of Materials
SOUP	Software of Unknown Pedigree
SPDX	Software Package Data Exchange
SWHID	Software Heritage persistent Identifier
SWID	Software Identification
VEX	Vulnerability Exploitability eXchange
XML	eXtensible Markup Language
XSD	XML Schema Definition