FOSSID
What's in your software?

Beyond the Static SBOM

Operationalizing Software Bill of Materials Across the Full Lifecycle

Ibrahim Haddad, Ph.D.

Technology Executive and Open Source Expert

March 2026

Abstract

A Software Bill of Materials (SBOM) has evolved from a best practice to a regulatory requirement across multiple jurisdictions. Many organizations treat SBOMs as static documents to be generated and archived, missing the operational reality that an SBOM is a living asset that moves through a defined lifecycle following the evolution of the product's software stack.

This report maps the complete SBOM journey from creation or intake through verification, security review, license analysis, risk sign-off, distribution, maintenance, and archival. At each stage, we examine how regulations from the United States (National Telecommunications and Information Administration guidance and the Executive Order 14028), the European Union (Cyber Resilience Act), and China's Cybersecurity Law and associated regulations create specific expectations and obligations.

The report concludes with practical, actionable recommendations for organizations seeking to operationalize their SBOM workflows, reduce legal exposure, and build defensible compliance postures.

Foreword

The global adoption of the Software Bill of Materials (SBOM) is accelerating. Regulations and government guidance across the EU, US, and China, alongside rising customer expectations, are pushing organizations to produce and share SBOMs. However, generating SBOMs is only the beginning. The real challenge is operationalizing them.

At FossID, we recognize two capabilities that consistently separate organizations that merely produce SBOMs from those that use them effectively.

The first capability is generating complete and accurate SBOMs. Modern software development rarely consists solely of well-structured dependencies managed through package managers. Engineers frequently incorporate code through manual copy-paste, vendor SDKs, legacy codebases, and increasingly through AI-assisted development. This development process and ways-of-working model introduces unmanaged code and fragments that exist outside traditional dependency tracking systems. Without detecting this code at the source level, SBOMs risk becoming incomplete representations of what actually ships in a product.

The second capability is managing the SBOM lifecycle across complex software supply chains. In regulated industries such as automotive, medical devices, aerospace, and critical infrastructure, organizations are not only producing SBOMs for their own software, they are also receiving them from suppliers, validating their quality, consolidating them, and delivering them downstream to customers and regulators. SBOMs, therefore, become operational artifacts that must be governed throughout the product lifecycle.

In “Beyond the Static SBOM: Operationalizing Software Bill of Materials Across the Full Lifecycle”, Dr. Haddad draws on his deep experience in open source R&D and governance. and software supply chain integrity to outline how organizations can move beyond static SBOM generation toward a practical, operational model. This report provides a practical blueprint for making SBOMs actionable, helping organizations maintain development velocity and software integrity in an environment where supply chain scrutiny is only increasing.

Aaron Branson
Head of Go-to-Market, FossID

Table of Content

Introduction

When Marc Andreessen declared in 2011 that software is eating the world, he was describing a shift in economic power. What he could not have fully anticipated was the supply chain that would come to underpin it. Today, every industry that software has consumed runs and depends on stacks of open source and third-party components, each with its own dependencies, licenses, and security considerations. That invisible foundation is now a primary target for attackers, a source of regulatory scrutiny, and a legal liability for organizations that cannot account for what is running in their products and infrastructures. For years, organizations managed this complexity through informal tracking and audits. That era has ended.

The response to this challenge is the Software Bill of Materials.

The Software Bill of Materials (SBOM), a formal, machine-readable inventory of software components, has emerged as the foundation of software supply chain transparency. What began as a community-led initiative through the National Telecommunications and Information Administration (NTIA) has become embedded in law and regulation across major economies.

- The European Union's Cyber Resilience Act (CRA) requires manufacturers of products with digital elements to generate and maintain SBOMs and track vulnerabilities throughout the supported life of their products.
- The United States Executive Order 14028 directs federal agencies to demand SBOMs from suppliers providing software to the federal government.
- China's Cybersecurity Law and associated regulations impose traceability and supply chain security obligations on operators of critical information infrastructure.

These regulations share a common theme: they transform the SBOM from a voluntary transparency tool into a compliance artifact with legal weight. An inaccurate SBOM is now a potential liability.

Most organizations lack a systematic approach to managing SBOMs throughout their lifecycle. An SBOM arrives from a supplier or is generated during a build. Someone files it away where it sits on a drive, untouched, until an auditor requests it or a security incident forces a retrospective analysis. This reactive posture creates risk at every stage: unverified components pass into products, license obligations go unnoticed, security vulnerabilities remain unaddressed, and regulators see gaps in due diligence.

This report argues for two main points. The first is that SBOMs have a lifecycle, and the second is that organizations must treat SBOMs as living assets that move through their defined

lifecycle with clear controls at each phase. We map that lifecycle across eight stages, from initial creation or intake through long-term archival. At each stage, we examine the regulatory expectations taking effect in the US, EU, and China, because the requirements differ in ways that matter for global software supply chains.

Our goal from this report is to help engineering teams, open source program offices (OSPOs), and compliance functions build workflows that turn SBOMs from static documents into active risk management tools. The report concludes with a set of concrete recommendations for implementation drawn from real-world deployment patterns.

SBOM Basics

At its core, an SBOM serves as a comprehensive, machine-readable inventory detailing the constituent software components within an application, a system, or a software stack. Analogous to a bill of materials in traditional manufacturing, an SBOM offers transparency and visibility into the software supply chain, facilitating robust software governance and risk management practices. The following key functional categories characterize SBOM content in practice, distinct from the NTIA minimum data fields defined in the Legislative Context section below:

1. **Component Inventory:** The component inventory is a comprehensive list of all software components, including both open source and proprietary components, along with their respective versions and dependencies.
2. **Provenance Information:** The provenance information includes metadata covering the origin and ownership of each software component, including licensing terms, copyright attributions, and contributors.
3. **Dependency Relationships:** The dependency relationships are hierarchical relationships that delineate dependencies between software components, facilitating traceability and impact analysis.
4. **Vulnerability Intelligence:** Vulnerability intelligence encompasses detailed information regarding known security vulnerabilities associated with each software component in the component inventory. The goal of this intelligence is to enable proactive risk mitigation and vulnerability management.
5. **Metadata and Annotations:** Metadata and annotations are additional contextual information that enrich the SBOM, such as build instructions, release notes, and compliance attestations.

Together, these five functional categories form a structured, machine-readable foundation that gives organizations the visibility they need to manage software components with precision, accountability, and auditability across the entire development lifecycle.

This is the starting point: understanding what an SBOM contains. The next section examines how we got here, how the SBOM concept has evolved to become a regulatory requirement across major economies.

Historical Perspective

The concept of SBOMs has steadily evolved over the years in response to the growing complexity of software supply chains and the increasing importance of transparency and accountability in software development practices. Initially, SBOMs gained traction within highly regulated industries, such as aerospace, automotive, and defense, where stringent compliance requirements necessitated meticulous documentation of software (and hardware) components (also called parts, depending on the industry). However, it wasn't until the proliferation of open source software and the advent of cloud computing that SBOMs began to garner broader attention across diverse sectors. As software ecosystems expanded and dependencies proliferated, the need for standardized SBOM frameworks became increasingly apparent, laying the groundwork for industry-driven initiatives aimed at promoting SBOM adoption.

The Linux Foundation launched the SPDX (Software Package Data Exchange) project in 2009, a significant milestone towards SBOM standardization. SPDX aimed to develop a common standard for documenting and sharing SBOMs, fostering interoperability and collaboration.

Concurrently, CycloneDX emerged under the Open Worldwide Application Security Project (OWASP) as a lightweight, machine-readable SBOM format designed to meet the needs of modern software supply chains. Originally an OWASP project, CycloneDX has since evolved into an independently governed standard maintained by the CycloneDX community, and today stands alongside SPDX as one of the two dominant SBOM formats in industry use.

SBOM momentum received a boost with the issuance of the United States' EO 14028 on Improving the Nation's Cybersecurity in May 2021. This EO underscored the critical importance of enhancing software supply chain security in light of escalating cyber threats and directed federal agencies to adopt SBOMs for software procurement. It called for the establishment of minimum elements for SBOMs (discussed in a later section) and laid the groundwork for broader adoption through collaborative efforts. As a result, SBOMs have transitioned from being used in specific, highly regulated industries within a narrow context to a broader component of legislative and regulatory efforts aimed at safeguarding software supply chains and bolstering national cybersecurity posture, regardless of industry type or technology domain.

In the next section, we will discuss why SBOMs matter and what drives organizations to act on that information consistently and at scale.

Criticality of SBOMs

SBOMs play an important role in both license compliance and cybersecurity. From ensuring adherence to applicable licenses (including both commercial and open source licenses) to improving defenses against cyber threats, SBOMs provide organizations with insights into their software components. In the following subsections, we will go further into detail on the role SBOMs play from both a license compliance and security perspective.

License Compliance

From the standpoint of license compliance, SBOMs offer a strategic advantage by providing comprehensive insights into the web of software components that underpin organizational operations.

In today's environment, studies consistently show that open source components constitute the majority of most modern software stacks. It is therefore critical to catalog open source and proprietary software assets, understand license obligations, and have a plan to execute and fulfill these obligations. SBOMs empower license compliance teams to mitigate legal, reputational, technical, and financial risks associated with license violations. The absence of SBOMs exposes organizations to many license compliance pitfalls, necessitating manual and error-prone methods of software asset tracking that undermine operational efficiency and expose organizations to unnecessary risks.

Security

In the domain of cybersecurity, SBOMs emerge as a critical asset in improving organizational resilience against evolving threats within the software supply chain. By proactively identifying and addressing vulnerabilities in third-party components, SBOMs serve as early warning systems, enabling organizations to preemptively mitigate security risks before they escalate into full-blown breaches. SBOMs facilitate streamlined incident response and patch management efforts, equipping organizations with the agility and foresight needed to safeguard critical assets and avoid exploits of security vulnerabilities.

Legislative Context

Against the backdrop of escalating cyber threats and regulatory scrutiny, several legislative and regulatory efforts stand out for their direct impact on SBOM obligations.

The first is the United States' EO 14028, which directed federal agencies to demand SBOMs from software suppliers and tasked NTIA with defining the minimum elements a conforming

SBOM must contain. NTIA published those minimum elements in its 2021 guidance document, establishing the baseline data fields that now underpin SBOM practice across both government procurement and private sector adoption. While EO 14028 was instrumental in putting SBOMs on the map, its scope is limited to federal procurement, and it carries no direct enforcement mechanism for the broader private sector.

The second, and arguably more consequential instrument, is the EU's CRA. Unlike EO 14028, the CRA is binding legislation that applies to any manufacturer placing a product with digital elements on the EU market, regardless of where that manufacturer is headquartered. The CRA explicitly requires manufacturers to generate and maintain SBOMs covering at a minimum the top-level dependencies of the product, for all products with digital elements placed on the EU market, with conformity assessment requirements varying by risk category. In addition, the CRA requires tracking vulnerabilities for the entire supported life of a product and reporting actively exploited vulnerabilities and severe security incidents to authorities. The CRA entered into force on December 10, 2024, with most obligations, including SBOM requirements, applying from December 11, 2027, and vulnerability reporting obligations applying from September 11, 2026. Non-compliance can result in market access restrictions and significant financial penalties. For global software supply chains, the CRA sets the de facto enforcement floor that EO 14028 did not.

China does not currently have a dedicated SBOM mandate equivalent to the NTIA guidance or the EU CRA. However, its Cybersecurity Law, Critical Information Infrastructure Security Protection Regulations, and Network Data Security Management Regulations collectively impose traceability, supply chain security, and vulnerability management obligations on operators of critical information infrastructure. These obligations align with SBOM principles in practice and are administered and enforced by the Cyberspace Administration of China.

SBOM Minimum Elements

The NTIA, acting on the directive in EO 14028, defined seven minimum data fields in its 2021 guidance document required for a conforming SBOM:

1. **Supplier Name:** The name of an entity that creates, defines, and identifies components. This field establishes accountability and traceability back to the originating source of each component.
2. **Component Name:** The designation assigned to a unit of software defined by the original supplier. This is the primary identifier used to locate and reference the component across systems and databases.
3. **Version of the Component:** The identifier used by the supplier to specify a change in software from a previously identified version. Accurate version data is foundational to vulnerability correlation and dependency management.

4. **Other Unique Identifiers:** Other identifiers used to identify a component, or to serve as a look-up key for relevant databases. This includes identifiers such as package URLs (pURL) and Common Platform Enumeration (CPE) entries that enable automated tooling to cross-reference components against vulnerability and license databases.
5. **Dependency Relationships:** Characterizing the relationship that an upstream component X is included in software Y. This field maps how components relate to one another, enabling impact analysis when a vulnerability or license issue is identified in a dependency.
6. **Author of SBOM Data:** The name of the entity that created the SBOM data for the component. This field distinguishes between the supplier of the component and the party responsible for the accuracy of the SBOM record itself, which may differ in supplier-provided SBOMs.
7. **Timestamp:** The record of the date and time of the SBOM data assembly. Timestamps are critical for audit purposes and for determining whether an SBOM reflects the current state of a product or a prior release.

These seven fields represent the minimum viable data set for a machine-readable, interoperable SBOM. They are not a ceiling. In practice, organizations and tools typically capture additional fields beyond these minimums, including license identifiers, cryptographic hashes, and build environment metadata, to support more comprehensive security and compliance workflows. The EU CRA's requirements build on this foundation, adding obligations around vulnerability tracking and disclosure that extend well beyond what the NTIA minimum elements alone address. Together, the NTIA baseline, the CRA's enforcement framework, and the obligations arising from China's Cybersecurity Law collectively shape the scope of what a production-grade SBOM program must deliver.

With the regulatory baseline established, the next question is operational: How does an organization manage an SBOM systematically from the moment it is created or received through to its long-term retention?

SBOM Lifecycle Overview

An SBOM is a regulated artifact that moves through a clear lifecycle from creation to long-term retention. Each phase now carries specific expectations from US, EU, and Chinese regulatory frameworks, the latter derived from China's Cybersecurity Law, Critical Information Infrastructure Security Protection Regulations, and Network Data Security Management Regulations. The following subsections examine each lifecycle stage in turn.

ENGINEERING PHASE • STAGES 1-4

01
CREATION / INTAKE

- → Build time generation
- → Supplier provided SBOM

02
VERIFICATION

- → Format validation
- → Data completeness
- → Component normalization

03
SECURITY REVIEW

- → CVE correlation
- → Exploitability check
- → Regulatory mapping

04
LICENSE COMPLIANCE

- → License identification
- → Obligation analysis
- → Notice prep

SECURITY PHASE • STAGES 5-6

05
RISK REVIEW

- → Security risks
- → License risks
- → Legal sign-off
- → Release approval

06
RISK SIGN-OFF

- → Customer / partners
- → Regulators
- → Internal teams

LEGAL PHASE • STAGES 7-8

07
UPDATE & MAINTENANCE

- → New SBOM per patch
- → Continuous monitoring
- → Multi-year obligations

08
ARCHIVAL & RETENTION

- → Long-term storage
- → Audit access
- → Supply chain trace

SBOM Creation or Intake

The SBOM intake can take one of two paths: an organization can generate the SBOM internally during build, or they can receive it from a supplier. A comprehensive SBOM captures the seven NTIA minimum data fields, and in practice, typically extends to additional attributes such as license identifiers and cryptographic hashes, the latter captured under the Other Unique Identifiers field.

There are several regulations globally that take effect here, including:

- **US:** Guidance from NTIA points to the minimum data fields required in an SBOM.
- **EU:** The CRA requires manufacturers to generate a machine-readable SBOM covering at a minimum the top-level dependencies of the product, as part of the technical documentation required under Annex I.

- **China:** The Cybersecurity Law and CII Security Protection Regulations require traceability and verifiability for critical information infrastructure operators.

SBOM Verification and Normalization

Once the organization has an SBOM, they need to check that it's usable. That process includes the following steps:

- Validating the format of the SBOM. The two most dominant formats are SPDX and CycloneDX, and they both handle most needs.
- Normalizing names of components to avoid false positives in downstream scans.
- Confirming that the versions listed in the SBOM match what was actually shipped in the build or product, using one or a combination of three complementary techniques: build system integration, hash and checksum verification, and binary analysis. Organizations typically combine all three: build integration to prevent drift, hash verification as a fast automated check, and binary analysis as the definitive audit method.

This step is where legal exposure starts. If the SBOM is inaccurate and an organization relies on it, they may breach due-care requirements in the US and the CRA compliance duties in the EU. SBOM accuracy is the prerequisite on which every downstream compliance and security decision depends.

Security Review

The next step in the lifecycle is vulnerability correlation. An organization would map components listed in the SBOM to known CVEs and decide which issues are critical and exploitable.

There are several regulations that kick in during this step:

- **US:** This security review supports CISA and NIST guidance issued under EO 14028, which requires federal suppliers to maintain vulnerability management practices tied to SBOM transparency.
- **EU:** The CRA requires manufacturers to track and address vulnerabilities throughout the supported life of a product. From September 11, 2026, manufacturers must report both actively exploited vulnerabilities and severe security incidents to authorities. For both tracks, an early warning is due within 24 hours of becoming aware, and a full notification within 72 hours. The final report deadlines differ: no later than 14 days after a corrective measure is available for actively exploited vulnerabilities, and within one month from submission of the 72-hour incident notification for severe incidents.

- **China:** The Cybersecurity Law and associated vulnerability management regulations require operators to maintain cybersecurity assurance and report vulnerabilities in critical network products to authorities.

License Compliance Analysis

Next, or even in parallel to the security review, an organization can run a license scan on the SBOM content. The goal is to ensure that the licenses of the components in the SBOM are compatible with their distribution model and to identify obligations like attribution, notices, source code availability, or copyleft triggers.

With respect to the EU and US regulations, the expectations on license compliance are similar: an organization must avoid IP violations and provide correct license disclosures. However, China's regulatory focus is less about open source licensing and more about software provenance and security risk.

Risk Review and Sign-off

Once the scans are done, a dedicated compliance owner (compliance officer, head of open source, or a similar role) would sign off. The final review and signing off include filing the SBOM with internal systems, attaching mitigations, and confirming that the organization is able to fulfill the obligations.

Beyond internal controls, regulators across all three jurisdictions increasingly expect organizations to demonstrate a documented, repeatable sign-off process:

- **US:** Agencies expect suppliers to demonstrate a functional SBOM review workflow.
- **EU:** The CRA mandates appropriate cybersecurity controls and documentation where the SBOM is part of that evidence.
- **China:** The Cybersecurity Law and CII Security Protection Regulations expect documented evidence of supply chain integrity for critical information infrastructure operators.

Distribution and Sharing

Finally, depending on the market, the organization may need to share the SBOM with partners, customers, or regulators.

The requirements for distribution and sharing differ by market:

- **US**: The US applies a tiered model for SBOM disclosure. Federal contractors and suppliers operating under EO 14028 face more prescriptive obligations, including providing SBOMs as a condition of procurement and making them available to federal agencies upon request. For the broader private sector and critical infrastructure operators, the expectation currently leans toward sharing on request, though sector-specific guidance from agencies such as CISA continues to raise the baseline for what constitutes adequate disclosure.

- **EU:** The CRA requires manufacturers to provide SBOMs to market surveillance authorities upon request. Manufacturers are not required to make SBOMs publicly available or share them with customers, though they must include the SBOM in their product technical documentation.

- **China:** China's Cybersecurity Law and CII Security Protection Regulations do not require public SBOMs, but authorities may request supply chain documentation from operators of critical information infrastructure, making it essential to have SBOMs ready to provide upon request.

Control Access to SBOMs

A dimension of SBOM distribution that is rarely discussed but critically important is confidentiality. An SBOM is a detailed inventory of every component in a product, including components with known vulnerabilities that have not yet been patched. When an organization shares its SBOM without controls, a potential attacker will receive a precise map of exploitable weaknesses. Therefore, an organization must treat the distribution of SBOMs as a controlled access operation. In practice, this means organizations should share SBOMs only with parties who have a legitimate need, under formal agreements that restrict further distribution, and for defined time periods after which access should be reviewed or revoked. Sharing mechanisms should provide access controls, audit trails, and expiry management rather than relying on email attachments or public repositories. When regulators or customers require SBOM access, the organization should scope and log that access to the minimum necessary. The goal is not to withhold transparency but to ensure that transparency does not become a liability.

With distribution controls in place, the next obligation is keeping the SBOM accurate as the product evolves.

Update and Maintenance

The usually unspoken aspect of SBOM is that every software update should generate a new SBOM, whether it's a single update to a component or a full software release including dozens or hundreds of changes. This practice ensures vulnerability management stays aligned with what's deployed in products.

The requirements for update and maintenance of SBOMs differ by market:

- **US:** The guidance emphasizes maintaining updated SBOMs throughout the product lifetime.
- **EU:** The CRA introduces multi-year obligations to track and address vulnerabilities post-market. That means an organization must maintain SBOM accuracy long-term. Vulnerability reporting obligations under Article 14 apply from September 11, 2026, to all products with digital elements made available on the EU market, including those placed on the market before December 11, 2027. SBOM and other essential cybersecurity requirements become enforceable from December 11, 2027, for products placed on the market from that date, and for products placed on the market before that date if they undergo a substantial modification from December 11, 2027, onward.
- **China:** The CII Security Protection Regulations and Network Data Security Management Regulations focus on lifecycle security management for critical network products, making updated SBOMs a practical tool for demonstrating compliance.

Archiving and Retention

Long-term retention is the final stage of the SBOM lifecycle and the one most directly tied to an organization's ability to respond to audits, litigation, and regulatory investigations years after a product ships.

The retention obligations differ in duration across jurisdictions, but the direction is consistent:

- **US:** The US federal procurement guidance under EO 14028 and NIST frameworks expects organizations to maintain SBOM records aligned with the product support lifecycle, though no single statutory retention period has been defined.
- **EU:** The CRA requires manufacturers to retain technical documentation, including SBOMs, for at least ten years after the product is placed on the market or for the support period, whichever is longer.
- **China:** The Cybersecurity Law and Network Data Security Management Regulations require records to be retained for audit and verification purposes, though no specific SBOM retention period is defined.

Summary

Most organizations treat SBOMs like static documents. SBOMs are delivered from a vendor or are exported from a build system, someone files them away, and they sit in digital storage until an auditor or a security incident forces a scramble. A real SBOM intake workflow fixes this scenario and gives organizations control from the moment an SBOM appears, and ensures the whole lifecycle can be automated, measured, and audited.

A modern SBOM is a living asset. It moves from creation to verification to risk review to sharing to long-term retention, and each step now intersects with regulatory expectations.

Organizations with a clear SBOM lifecycle in place reduce legal risk and make audits straightforward. If they don't, they expose themselves to vulnerability handling failures, licensing violations, and regulatory non-compliance.

Practical Recommendations

The following recommendations translate the SBOM lifecycle into concrete operational practice. They span eleven areas, from source validation and normalization through vulnerability correlation, license and IP screening, supplier risk assessment, change detection, audit logging, policy enforcement, supplier feedback loops, and pipeline integration. Together, these recommendations form a repeatable, auditable workflow that moves SBOM management from reactive document handling to active risk control. Organizations are not expected to implement all recommendations at once. Instead, organizations can adopt a phased approach, prioritizing validation, vulnerability correlation, and policy enforcement first, building the foundation on which the remaining capabilities can layer over time.

Source Validation

Before any SBOM enters an organization's workflow, it must be authenticated and structurally verified to confirm it is complete, trustworthy, and attributable to a known supplier.

Some of the best practices in this lifecycle phase include:

- Verifying the SBOM file is signed by the supplier.
- Checking schema validity (CycloneDX or SPDX) and that all required fields are present.
- Auto-flagging missing licenses, package URLs, or component metadata.
- Mapping supplier identity to your vendor master data.

Once an SBOM passes source validation, the next challenge is structural: SBOMs arrive in different formats and schemas, and making them usable across internal systems requires deliberate normalization.

Normalization and Conversion

Suppliers rarely deliver SBOMs in a consistent format, making normalization and conversion a prerequisite for any downstream analysis to function reliably across the organization's

toolchain. Since vendors send SBOMs in different formats or partial schemas, your organization's pipelines should standardize them.

Some of the best practices in this lifecycle phase include:

- Converting SPDX to CycloneDX or vice versa, depending on your internal standard.
- Normalizing naming using package URL and CPE mapping.
- Deduplicating components.
- Enriching with internal identifiers or product codes.

The goal is to create a uniform SBOM structure that downstream systems can use. With a normalized SBOM structure in place, the organization can now accurately map individual components to approved versions and detect gaps or risks within the dependency chain.

Dependency and Version Mapping

A normalized SBOM is only as useful as the accuracy of its component data, making it essential to verify that every dependency is mapped to a known, approved version and traceable to the correct product build.

Some of the best practices in this lifecycle phase include:

- Mapping components to approved package versions from your internal catalog.
- Flagging unapproved or deprecated dependencies.
- Detecting missing transitive dependencies.
- Linking SBOM data to the correct product release or build pipeline.

These actions let R&D and compliance work from the same source of truth. With dependencies confirmed and versions locked to specific releases, the organization has the component-level precision needed to run meaningful vulnerability correlation (next step) against known threat databases.

Vulnerability Correlation

With an accurate and version-confirmed component inventory in hand, the organization can now systematically cross-reference each component against known vulnerability databases to identify, prioritize, and track security risks across its product portfolio.

Some of the best practices in this lifecycle phase include:

- Auto cross-referencing components with NVD, OSV, CNVD for China, and JVN for Japan.
- Tracking risk levels by product and supplier.
- Flagging vulnerabilities that fall under CISA and NIST guidance issued in support of EO 14028.
- Supporting EU CRA requirements for incident reporting and software updates.
- Storing historical vulnerability states for audits.

Security risk is only one dimension of component-level exposure; each component also carries license obligations that, if unaddressed, can create equally serious legal and intellectual property liabilities.

License and Intellectual Property Screening

Every component in an SBOM carries a license that imposes specific obligations, and failing to identify conflicts, gaps, or restricted terms before distribution exposes the organization to legal liability and regulatory non-compliance.

Some of the best practices in this lifecycle phase include:

- Identifying all declared licenses and detecting conflicts with your policy.
- Flagging licenses missing from the SBOM.
- Identifying partially declared or vague identifiers like Dual or Unknown.
- Checking for use restrictions under EU CRA, US trade rules, and China's Cybersecurity Law and associated regulations.
- Routing violations to legal for approval or mitigation.

License and vulnerability data from individual SBOMs also reveal patterns at the supplier level, making it possible to assess and track the broader risk posture of the organizations providing those components.

Supplier Risk Assessment Integration

Individual SBOM reviews surface component-level risks, but aggregating that data across all supplier-provided SBOMs enables the organization to build a structured, evidence-based view

of each supplier's security and compliance posture over time.

Every SBOM should feed into how an organization rates its suppliers. Some of the best practices in this lifecycle phase include:

- Measuring SBOM completeness scores.
- Assessing the responsiveness of vendors to vulnerability remediation.
- Tracking changes to supplier security posture.
- Flagging suppliers that never provide SBOMs on time or provide low-quality ones.

Supplier risk ratings are only meaningful if they stay current, which requires continuous monitoring of how SBOMs change between releases and whether those changes align with what was actually shipped.

Change Detection and Drift Tracking

Software components change with every release, and without systematic detection of those changes, an organization cannot confirm that what is declared in an SBOM matches what is actually running in a shipped product.

Some of the best practices in this lifecycle phase include:

- Detecting major and minor changes between SBOM versions.
- Validating that supplier updates match shipped binaries.
- Triggering reviews when components appear or disappear unexpectedly.
- Comparing SBOM data to build pipeline SBOMs to detect mismatches or tampering.

Detecting drift and validating changes is only operationally useful if every action, decision, and finding is captured in a form that can be retrieved, reviewed, and presented as evidence during an audit or regulatory review.

Audit Logging and Evidence Packaging

Every validation, scan, sign-off, and remediation action taken across the SBOM lifecycle must be logged and packaged in a way that produces a complete, retrievable evidence trail for internal audits, customer requests, and regulatory obligations.

Some of the best practices in this lifecycle phase include:

- Archiving versions of every SBOM and every validation step.
- Storing signatures, fingerprints, and timestamps.
- Auto-generating audit bundles for internal audit, customers, or regulators.
- Capturing evidence needed for CRA reporting windows.

Audit logs document what happened, but preventing non-compliant SBOMs from advancing through the pipeline in the first place requires automated policy enforcement built directly into the workflow.

Policy Enforcement

Automated policy enforcement converts the organization's compliance requirements into hard gates that prevent non-conforming SBOMs, unresolved vulnerabilities, and license violations from advancing through the pipeline without explicit review and approval.

Some of the best practices in this lifecycle phase include:

- Auto-rejecting SBOMs missing critical metadata.
- Blocking product releases that have unresolved critical vulnerabilities.
- Enforcing minimum SBOM quality scores.
- Auto-notifying engineering when components violate license policies.

Blocking non-compliant SBOMs at the gate is necessary, but sustainable improvement requires closing the loop with suppliers so that quality issues are communicated, tracked, and resolved at the source.

Feedback and Supplier Loop

Policy enforcement identifies SBOM quality failures, but those failures only improve if suppliers receive structured, timely feedback that gives them the information and tools needed to meet the organization's requirements consistently.

Some of the best practices in this lifecycle phase include:

- Sending automated feedback to vendors on SBOM quality.

- Tracking improvements or regressions over time.
- Providing templates or tools to suppliers who struggle.
- Escalating for persistent non-compliance.

Improving supplier-provided SBOMs addresses one input to the workflow, but full lifecycle accuracy also depends on generating and validating SBOMs internally, directly within the organization's own development and build pipelines.

Integration With Secure Development Pipelines

Embedding SBOM generation and validation directly into the secure development pipeline ensures that component transparency is not a post-build activity but a continuous, automated practice that keeps pace with every code change and release.

Some of the best practices in this lifecycle phase include:

- Embedding SBOM generation and review into standard engineering workflows.
- Generating SBOMs at build time and comparing them with vendor-provided SBOMs.
- Requiring developers to confirm SBOM acceptance before promoting builds.
- Syncing SBOM data with SCA tools and CI/CD pipelines.
- Alerting teams when dependencies fall out of policy.

These practices ensure SBOMs remain accurate from code to customer. When SBOMs are generated, validated, and enforced across both internal pipelines and supplier workflows, the organization has the foundation needed to move from reactive compliance to a defensible, repeatable posture, which is the goal this report has worked toward from the outset.

Conclusion

The software supply chain is now a triple target: a primary attack surface, a legal liability domain, and a regulatory battleground with SBOMs sitting at the intersection of all three.

> ***Organizations that treat SBOMs as living assets, managed through a defined lifecycle with clear controls at every stage, are materially better positioned than those that do not.***

The regulatory direction is unambiguous. The EU CRA imposes enforceable obligations with real legal consequences on any manufacturer placing products with digital elements on the EU market, and China's Cybersecurity Law and associated regulations impose equivalent obligations on operators of critical information infrastructure. US federal procurement requirements continue to raise expectations for suppliers operating in government markets. An inaccurate SBOM, an untracked vulnerability, or an undisclosed license obligation is no longer a technical oversight. It is a compliance failure with potential financial, reputational, and legal exposure.

The recommendations in this report are actionable today. Source validation, normalization, vulnerability correlation, license screening, supplier feedback, and pipeline integration are achievable with current tooling, phased implementation, and clear ownership. The organizations that act first will reduce risks and build a durable advantage as customers, regulators, and partners increasingly demand proof of software supply chain integrity.

The question is no longer whether SBOM lifecycle management matters. We know it does, and we know what the lifecycle stages are. The question is whether your organization will build that capability deliberately, on your own terms, or reactively, under pressure and scrutiny.

Disclaimer

This report has been prepared to the best of the author's ability. Despite careful review, typographical errors or inaccuracies may remain. The author apologizes in advance for any such oversights and welcomes corrections. If you identify an error, please contact the author directly via LinkedIn so that a correction can be issued promptly.

Acknowledgements

The author wishes to express sincere gratitude to FossID for their support in publishing this report and their commitment to bringing it to a wider audience. By helping to promote and distribute this report, FossID is contributing directly to a broader industry conversation that organizations urgently need to engage with as SBOM requirements move from voluntary best practice to enforceable obligation. The author hopes that this collaboration will help organizations across industries better understand the SBOM lifecycle, build the operational practices needed to manage it effectively, and approach the compliance challenges ahead with clarity and confidence.

Additional Resources

The following resources, by the same author, provide additional depth on SBOM adoption, open source compliance, and software supply chain governance:

- SBOM Adoption: In Support of Better License Compliance and Software Security Practices, Dr. Ibrahim Haddad, October 2025.
- Strengthening License Compliance and Software Security with SBOM Adoption: A Definitive SBOM Guide for Enterprises, Dr. Ibrahim Haddad, Foreword by Melissa Evers, The Linux Foundation, August 2024.
- Recommended Open Source Compliance Practices for the Enterprise, Dr. Ibrahim Haddad, The Linux Foundation, April 2019.
- Open Source Compliance in the Enterprise, Dr. Ibrahim Haddad, The Linux Foundation, January 2019.

About the Author

Dr. Ibrahim Haddad is a senior technology executive and trusted advisor with over two decades of experience building and leading global engineering organizations, AI ecosystems, open source governance, and the organizational frameworks that enable technology to reach global scale. Currently serving as Head of Infotainment Engineering at Volvo Cars, Dr. Haddad leads an engineering organization responsible for the next-generation Android-based in-vehicle infotainment platform. Prior to Volvo, Dr. Haddad served as Vice President of AI Strategic Programs at the Linux Foundation, where he scaled the LF AI & Data Foundation into a globally recognized hub for open source AI, growing its project portfolio from 3 to 70 initiatives and its developer community to 100,000+ across 3,000 organizations. He subsequently led the PyTorch Foundation as its inaugural Executive Director, uniting AMD, AWS, Google, Meta, Microsoft, and Nvidia under a neutral governance model, doubling foundation revenues within a year, and driving a 36% increase in code contributions in the first six months.

Prior to the Linux Foundation, Dr. Haddad served as Vice President of R&D and Distinguished Engineer at Samsung Research, where he built and scaled Samsung's open source engineering organization. He has held senior leadership roles at Ericsson Research, Motorola, Palm, and the Open Source Development Labs, accumulating deep experience in distributed team leadership, platform strategy, and technical governance.

Dr. Haddad holds a Ph.D. with honors in Computer Science from Concordia University. He is the author/co-author of 7 books and over 150 technical reports and is a frequent keynote speaker at global technology forums. He advises organizations across North America, Europe, and Asia on open source governance, AI strategy, and collaborative partnerships.

If your organization is navigating open source strategy, AI governance, or software supply chain compliance and would benefit from senior advisory support, Dr. Haddad is available for short or long-term engagements. Reach him via LinkedIn or his personal website.

To reference the work, please cite as follows: Ibrahim Haddad, “Beyond the Static SBOM: Operationalizing Software Bill of Materials Across the Full Lifecycle”, FossID, March 2026.