National Telecommunications and Information Administration
Re: Software Bill of Materials Elements and Considerations

NTIA-2021-0001
SBOM_RFC@ntia.gov

June 16, 2021

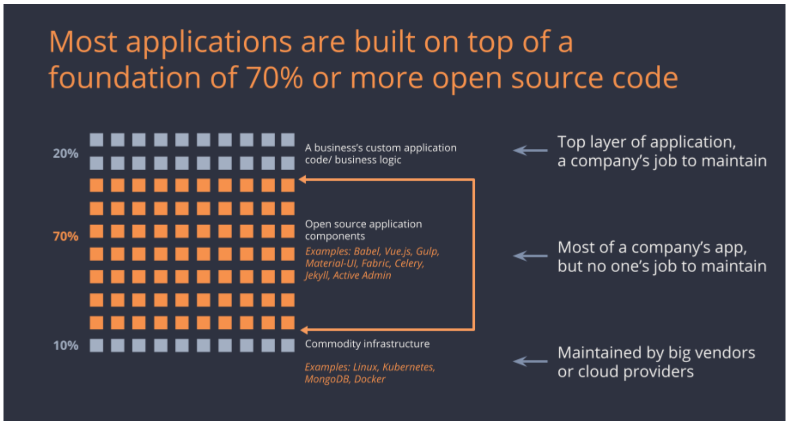
Executive Summary

Open source software forms the bulk of the software used in modern software development, and so any approach to SBOMs must work effectively for open source. Simultaneously, the maintainers of that open source software are the best positioned to create and maintain accurate SBOMs. As a result, any effort led by the US government to spur SBOM adoption is only likely to be successful if it actively works with—and compensates— “upstream” open source maintainers.

The scope of the SBOM problem

The scope of the modern software production supply chain is vast, with typical applications pulling in a thousand or more open source dependencies, and with many software-developing organizations having tens or hundreds of applications. To further complicate things, these dependencies are pulled from a universe of millions of open source packages, with most applications drawing at least some dependencies from a “long tail” of packages that are not widely used—such that it isn’t possible to address the scope of the problem by targeting only top packages.

The open source software industry refers to these packages as the “upstream”, where producers (ranging from the largest Fortune 500 companies to individuals working in their spare time) create components that are then consumed by “downstreams” who productize software based on this upstream software.


Most applications are built on top of a foundation of 70% or more open source code
20%
A business’s custom application code/ business logic
Top layer of application, a company’s job to maintain
70%
Open source application components
Examples: Babel, Vue.js, Gulp, Material-UI, Fabric, Celery, Jekyll, Active Admin
Most of a company’s app, but no one’s job to maintain
10%
Commodity infrastructure
Examples: Linux, Kubernetes, MongoDB, Docker
Maintained by big vendors or cloud providers

Current SBOM production is not efficient or effective

Faced with the challenge of representing information about tens of thousands of open source components, the current state of the art in SBOM production is neither efficient nor effective. It relies on brute-force software-based scanning to make a best guess about what is safe and healthy to use based on licensing and security metadata that is usually incomplete at best and inaccurate at worst.

> "In our analysis of [a market leading software composition analysis tool used to create SBOMs], more than 97% of reported problems were false positives."—Tidelift customer

This often-incorrect data is then supplemented by human labor, which is typically done by individuals who may have expertise in SBOMs but not in the actual software being described.

Given the poor state of the original metadata, these scanners and supplemental experts do heroic work, attempting to classify security and legal problems based on a mix of pattern-matching and historical databases. However, such a process is inherently limited—the automated tools cannot contact the human producers of the software, and so must attempt to divine their original intent based on source code. The subsequent human engagement can correct some of these errors, but given the vast scale of the problem, such engagement is necessarily limited. For example, the largest public database of such human improvements contains slightly under 30,000 human "curations"[1] affecting a database of nearly 13M pieces of data.[2]

Moving SBOM production upstream

Given the giant scope of the problem (millions of packages), and the unsolvable challenge of solving it through automated analysis and third-party experts (who are inherently limited by the incomplete and incorrect nature of the existing metadata), the only way to make SBOM processes for open source accurate and efficient is to involve the "upstream" providers—the open source maintainers who create the code and keep it up to date. Because these maintainers are typically those who

[1] 29,384 commits in https://github.com/clearlydefined/curated-data as of June 16, 2021.

[2] 12,821,002 definitions in https://clearlydefined.io/stats as of June 16, 2021.

have made critical security and legal decisions, they are best able to create the correct SBOM metadata for consumption and further curation by downstream users.

Such an “upstream” approach directly addresses many of the most important issues raised in the request for comment. Most notably, to address concerns about the frequency of updates and distribution, the best way for SBOMs for open source components to be distributed is as part of the upstream source code, contained and updated alongside the source code itself. Any other approach will inevitably be regularly out of sync each time the upstream open source software is released. It will also require complex mechanisms for distribution, which will at best duplicate existing infrastructure for software and metadata distribution, and at worst have much lower adoption than the actual software infrastructure that already exists.

Benefits of upstream span many metadata types

The request for public comment identifies a variety of fields that may be included within an SBOM, including supplier identity, cryptographic hashing of release information, and dependency relationships. For all of these, information is likely to be most accurate and up to date when maintained by the authors themselves. Those authors will have the best first-hand knowledge of the product, and will have the tightest possible timeline between making any changes and updating the SBOM metadata. Information management by third parties will, in contrast, reduce the quality and timeliness of the information, and require duplication of effort.

Challenges for upstream SBOM creation and curation

Despite the recent attention, attempts to solve the SBOM issue are not new. Motivated by legal concerns, the open source community has been attempting to solve this issue for over a decade in the form of SPDX. However, despite extensive efforts, SPDX has achieved only limited uptake in the open source development community. Studies of licensing at scale mention SPDX only in passing rather than relying on it to understand the ecosystem.[3]

[3] See, for example, “License Usage and Changes: A Large-Scale Study on GitHub”, https://mustang.cec.miamioh.edu/Resources/Publication/ICPC15-LicensingStudyGitHub.pdf, or “From One to Hundreds: Multi-Licensing in the JavaScript Ecosystem”, https://arxiv.org/abs/2012.05016, both of which are large-scale studies of licensing information. These studies relied on machine parsing of upstream license information rather than SPDX metadata, presumably because of the sparseness of proper SPDX data, despite a decade of effort to create SPDX metadata.

> “Although there are tools to aid developers in [validating license metadata] most of them are not applied accordingly on projects.”—*From One to Hundreds: Multi-Licensing in the Javascript Ecosystem*

This lack of adoption by open source developers should not be surprising. Studies of corporate adoption of SPDX find that “excessive complexity is getting in the way of adoption”, and most upstream developers have even less motivation to understand this complexity and adopt the standard. This is not the fault of SPDX—as the NTIA recognizes in this call for comments, these standards are necessarily complex. But it does strongly suggest that “upstream” SBOM creation will not occur without significant investment in education and motivation of developers.

Conclusion: upstream maintainers should be paid to maintain SBOMs

Given the intertwined challenges of accuracy, timeliness, complexity, and motivation, Tidelift strongly believes that in the open source era, SBOMs will only work at scale if upstream maintainers are directly engaged to produce accurate, up-to-date metadata that originates upstream and then is collated and handled by downstream software consumers. Only these maintainers have the accurate, up-to-date knowledge of the software and its problems that are necessary for any serious SBOM effort to succeed.

Given the poor adoption of SPDX, despite extensive efforts, Tidelift further believes that upstream maintainers will only create SBOMs if they are compensated to do so. If it does not involve compensation directly to maintainers for this important but dull work, any industry-wide or government-sponsored effort to encourage consistent, complete SBOM adoption throughout the industry will fail.

Sincerely,

Luis Villa, Co-Founder and General Counsel, Tidelift

luis@tidelift.com

About Tidelift

Tidelift helps organizations effectively manage the open source behind modern applications.

Through the Tidelift Subscription, the company delivers a comprehensive management solution, including the tools to create customizable catalogs of known-good, proactively maintained components backed by Tidelift and its open source maintainer partners.

Tidelift enables organizations to accelerate development and reduce risk when building applications with open source, so they can create even more incredible software, even faster.

Tidelift's founding team has decades of experience in open source across a variety of startups, non-profits, and publicly traded companies, including Red Hat, Mozilla, and Wikipedia.