Arm, Inc.
Comments to the **U.S. National Telecommunications and Information Administration Software Bill of Materials Elements and Considerations** (Docket No. 210527-0117[1])

On behalf of Arm, we are pleased to have the opportunity to comment in this notice on Software Bill of Materials Elements and Considerations. Arm has participated in the National Telecommunications and Information Administration's ongoing multistakeholder project on software bill of materials (SBOM) and commend the progress being made there. We welcome the Administration's heightened focus on cybersecurity through White House Executive Order 14028, among other important actions and work, and aim to provide input to assist the government in its efforts to improve the overall state of cybersecurity in the public and private sectors.

Arm is the leading global supplier of intellectual property (IP) to the semiconductor sector, licensing processor designs, security IP, system IP, software, and development tools, along with many other technologies essential to modern computing.[2] Arm believe secure computing starts with security in our technologies. As such, Arm puts the utmost importance on developing and providing secure products. In 2017, Arm released a call to action to the industry to do more to address security by releasing its first Security Manifesto, also discussing what more Arm itself could do to enhance security; in 2018 Arm released and updated version that built on that vision and call to action.[3]

Arm agrees SBOMs can play an important role in improving transparency of product components and overall cybersecurity, in addition to serving many other purposes. The purpose of an SBOM for a supplier should be to convey minimum information relating to the content of the software in a standard format providing visibility of all components of the software. This enables the entity receiving the software to assess any associated risks within the software as well as monitoring any future vulnerabilities that may arise.

Arm is a founding member of the OpenChain project, which has been recognized by the International Standards Organization as ISO 5230.[4] OpenChain defines the requirements for achieving the potential elements for an SBOM as described in the notice but should be noted it is focused on license compliance. OpenChain is aimed at building trust across the supply chain by defining a process an organization must follow to convey key information relating to components incorporated within the software including applicable licenses, name of component, version of components, etc.

---

[1] See Federal Register :: Software Bill of Materials Elements and Considerations
[2] For more about Arm, see www.arm.com
[3] See IoT Security Manifesto 2018/2019 – Arm
[4] See www.openchainproject.org

Arm encourages the use of OpenChain as a framework for establish SBOM requirements. In addition to being able to deliver many, if not all the components discussed in the notice, there is already work within OpenChain to add a security specific extension. The extension will address any additional requirements which do not exist in the license compliance part of the OpenChain specification. This security component is planned to be added as an extension rather than an expansion of the existing specification directly.

Lastly, NTIA should provide opportunity to use other data standards to convey the data fields beyond SPDX, CycloneDX, and SWID tags. Other standard formats which are machine readable could be utilized. A supplier should clearly indicate which format they are using and should not prevent the supply chain from converting into other formats given the underlying data fields are the same (for example name of component, version of component, etc.).

Arm inputs on the "Request for Comment" section and questions:

1. Are the elements described above, including data fields, operational considerations, and support for automation, sufficient? What other elements should be considered and why?

   Arm Comments: License information should also be included here given close alignment between security related fields and license compliance fields. For an entity to understand and analyze any security risks an SBOM should typically provide details of about the software components incorporated within the software such as name of component, version of component, current known security vulnerability, any dependencies, and other key details. Incidentally, this information is also normally required for license compliance (there is additional elements needed such as applicable license) which is where OpenChain is crucial as it defines what is required when software is distributed across the supply chain. It defines a clear set of requirements for an organization to follow when conveying software to allow the receiving entity to comply with any applicable licenses. The information supplied in relation to the SBOM will allow the receiving entity to assess the security risks at the point of delivery but more importantly it will also enable it to monitor any vulnerability which arise in the future.

2. Are there additional use cases that can further inform the elements of SBOM?

   Arm Comments: There needs to be a distinction between dependencies distributed as part of the SBOM and those which the receiving entity must obtain from elsewhere for the software to function.
   A clearer definition of a dependency should be provided. Dependencies can be difficult to manage given software can rely on grabbing new versions as it is compiled, and a supplier will only have knowledge of the versions that were used

at the point of distribution but an entity receiving the software can build or compile the software itself resulting in news versions of the dependencies being used. In addition, the definition of a dependency should be limited to essential components required for the software to compile. For example, if a software can run on a Windows machine, then it will not be reasonable to expect the supplier to address vulnerabilities in the operating system itself; this is the responsibility of the entity running the software.
One approach could be to draw a distinction between a dependency which is distributed by the vendor and a dependency where the entity receiving the software must have access to directly. In the latter case, it should be the responsibility of the entity receiving the software to assess the security vulnerability risks.
This is a complex issue and additional technical work will be needed to work through this.

3. SBOM creation and use touches on a number of related areas in IT management, cybersecurity, and public policy. We seek comment on how these issues described below should be considered in defining SBOM elements today and in the future.

   Arm Comments: Specific comments on SBOM elements are provided below but Arm recommends license compliance management should be an additional element.

   a. Software Identity: There is no single namespace to easily identify and name every software component. The challenge is not the lack of standards, but multiple standards and practices in different communities.

      Arm Comments: Legacy software used today will be extremely difficult to adjust to retrofit any new standard or even existing standard, so an SBOM must allow for incorporation of these into the supplied software

   b. Software-as-a-Service and online services: While current, cloud-based software has the advantage of more modern tool chains, the use cases for SBOM may be different for software that is not running on customer premises or maintained by the customer.

      Arm Comments: Arm agrees SaaS provides a different set of complexities that requires additional work to better understand the nature of security vulnerabilities for an SaaS solution. For example, SaaS can continually update and does not necessarily have a concept of versions. Depending on the SaaS, the SBOM could update on a daily basis. The frequency of SBOM changes and dissemination would have to be considered. More work is needed, and likely more flexibility given the variety of SaaS offerings.

c. Legacy and binary-only software: Older software often has greater risks, especially if it is not maintained. In some cases, the source may not even be obtainable, with only the object code available for SBOM generation.

   Arm Comments: Arm agrees with this. An SBOM must enable for such old software to be used. One way to address this would be to add a new field to convey whether certain components are legacy or unmaintained.

d. Integrity and authenticity: An SBOM consumer may be concerned about verifying the source of the SBOM data and confirming that it was not tampered with. Some existing measures for integrity and authenticity of both software and metadata can be leveraged.

   Arm Comments: This may be able to be addressed through ongoing work with a consortium such as OpenChain. As OpenChain is an ISO standard, many organizations would be incentivized to take this up.

e. Threat model: While many anticipated use cases may rely on the SBOM as an authoritative reference when evaluating external information (such as vulnerability reports), other use cases may rely on the SBOM as a foundation in detecting more sophisticated supply chain attacks. These attacks could include compromising the integrity of not only the systems used to build the software component, but also the systems used to create the SBOM or even the SBOM itself. How can SBOM position itself to support the detection of internal compromise? How can these more advanced data collection and management efforts best be integrated into the basic SBOM structure? What further costs and complexities would this impose?

   Arm Comments: Having an SBOM should enable an entity receiving the software to monitor to address this risk. It is more likely that making SBOMs available will facilitate reporting of potential threats.
   It is also crucial to note that for license compliance there is a legal obligation within such licenses to disclose certain information, whereas with an SBOM there is no legal obligations within licenses to create and supply SBOMs.

f. High assurance use cases: Some SBOM use cases require additional data about aspects of the software development and build environment, including those aspects that are enumerated in Executive Order 14028. How can SBOM data be integrated with this additional data in a modular fashion?

   Arm Comments: Extensions to SBOMs could be considered, for example for Trade Compliance related information, or license compliance which would help ensure that information is modular.

g. Delivery. As noted above, multiple mechanisms exist to aid in SBOM discovery, as well as to enable access to SBOMs. Further mechanisms and standards may be needed, yet too many options may impose higher costs on either SBOM producers or consumers.

   Arm Comments: The supplier should choose the delivery mechanism but the order should put an obligation on the supplier to provide such information giving the entity receiving the software the ability to make such requests as and when they require access.

h. Depth. As noted above, while ideal SBOMs have the complete graph of the assembled software, not every software producer will be able or ready to share the entire graph.

   Arm Comments: It is incredibly difficult to do this in practice given the varied software development processes. Dependencies related information could be encouraged but made optional to start with until tools are in place to automate this process.
   That said, encouraging delivery of a minimum, or less than complete SBOM will help steer the community into building improvements within these areas. Having a complete dependency graph is critical for maintaining security within a software system. While a complete graph may not be possible today, starting the community focusing on this will eventually lead to delivery of more complete information.

i. Vulnerabilities. Many of the use cases around SBOMs focus on known vulnerabilities. Some build on this by including vulnerability data in the SBOM itself. Others note that the existence and status of vulnerabilities can change over time, and there is no general guarantee or signal about whether the SBOM data is up-to-date relative to all relevant and applicable vulnerability data sources.

   Arm Comments: Monitoring SBOM vulnerabilities should be up to the entity receiving the software, and that should be made clear to the receiver. Including a static list of security vulnerabilities in an SBOM could lead some receivers into a false sense of security and not monitor for newly discovered and reported vulnerabilities. However, a supplier should update the vulnerability of the SBOM when and if the software is updated by the supplier even when underlying components have not been updated. Monitoring for vulnerabilities can be costly on an ongoing basis but a supplier could be contracted to provide such service as part of a contractual arrangement.

j. Risk Management. Not all vulnerabilities in software code put operators or users at real risk from software built using those vulnerable components, as the risk could be mitigated elsewhere or deemed to be negligible. One approach to managing this might be to communicate that software is "not affected" by a specific vulnerability through a Vulnerability Exploitability eXchange (or "VEX"), but other solutions may exist.

Arm Comments: This will often depend on how and where the software is used, and the burden should be put on the last supplier in the supply chain.
For most use cases it is not possible for upstream users to communicate that a vulnerability is not an issue as the use case is not known. Rating a vulnerability will help the receiver to determine whether the vulnerability is a threat to their solution, but even then, without knowing the end use it is difficult to rank aspects like exposure.

4. Flexibility of implementation and potential requirements. If there are legitimate reasons why the above elements might be difficult to adopt or use for certain technologies, industries, or communities, how might the goals and use cases described above be fulfilled through alternate means? What accommodations and alternate approaches can deliver benefits while allowing for flexibility?

Arm Comments: From the OpenChain experience, the requirements should be made flexible initially to enable automation of the SBOM creation to happen before such requirements are made compulsory.

Again, Arm appreciates the opportunity to provide inputs to this consultation. Arm agrees with the underlying aim of this work and agrees that to effectively monitor security vulnerabilities the entities receiving software need to fully understand what is included in the software in terms of third-party components (including open source) at the point the software is delivered. This will enable the entity to review the risks and put mechanisms to monitor for any future security vulnerabilities that arise. Arm stands ready to work with NTIA and the U.S. government more broadly to effectively develop this guidance. Please do not hesitate to reach out if we can be of assistance or provide additional clarity on our responses.

Respectfully Submitted,
Mr. Vince Jesaitis
Director, Government Affairs

Mr. Sami Atabani
Director, Third Party IP Licensing

Dr. Lyndon Fawcett
Principal Software Security Architect