Toward a Reference Architecture for Software Supply Chain Metadata Management

Nguyen Khoi Tran
CREST
The University of Adelaide
Adelaide, Australia
nguyen.tran@adelaide.edu.au

Samodha Pallewatta
CREST
The University of Adelaide
Adelaide, Australia
samodha.pallewatta@adelaide.edu.au

M. Ali Babar
CREST, The University of Adelaide
Cyber Security Cooperative Research Centre
Adelaide, Australia
ali.babar@adelaide.edu.a

ABSTRACT

This document is an appendix to the main paper "Toward a Reference Architecture for Software Supply Chain Metadata Management". It presents an overview of the SSC security frameworks and architectures utilised as the inputs to the RA construction process.

CCS CONCEPTS

• **Software and its engineering** → *Software design engineering*; • **Security and privacy** → **Software and application security**.

KEYWORDS

Software Supply Chain, SSC Metadata, SBOM, Software Provenance, Reference Architecture, SoK

ACM Reference Format:
Nguyen Khoi Tran, Samodha Pallewatta, and M. Ali Babar. 2024. Toward a Reference Architecture for Software Supply Chain Metadata Management . In *Proceedings of 46th International Conference on Software Engineering (ICSE 2024)*. ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 SSC SECURITY FRAMEWORKS AND ARCHITECTURES

This section presents an overview of the SSC security frameworks and architecture that serve as both the related work and the input of our proposed RA for SCM2.

SEI Supply Chain Risk Management (SCRM) Framework [1, 13, 14] presents a systematic approach to managing SSC risks. It identifies the interactions and activities among SSC actors and defines the responsibilities of suppliers and acquirers in analyzing and managing SSC risks throughout the software acquisition stage. While it does not provide a standardized way of generating and sharing machine readable SSC metadata, it discusses the use of various metadata (e.g., security test results, evidence of good coding practices, reports on potential security vulnerabilities, etc.) as evidence in assurance case for SSC risk.

In-toto Attestation Framework [4] addresses software supply chain security by generating and consuming verifiable claims (attestations) about software production. The core component of in-toto is the attestation specification, which defines a model and the corresponding syntax to writing attestations.

Governance Framework for Software Delivery [15] leverages software sensors and distributed ledger technology to establish a governance framework for software delivery. It models the provenance of software artefacts moving through the SSC using the PROV data model. The framework introduces software components for storing and analysing provenance, to be implemented using blockchain and smart contracts.

OWASP Software Component Verification Standard (SCVS) [12] is a community-driven framework for identifying activities, controls, and best practices for mitigating software supply chain risks. It defines security controls across six families, addressing various aspects of software component verification. While not explicitly providing an architectural model for the SSC metadata life cycle, it mandates the automatic creation of accurate SBOM in the build pipeline and outlines requirements for automated SBOM generation.

NTIA SBOM Generation Playbook [5, 6, 10, 11] provides a comprehensive overview of the SBOM generation stage introducing related terminology, participants and their roles, and elements of a SBOM including baseline and additional elements that facilitate various SBOM use cases. [10] defines a four-step workflow that encompasses the identification of software components in a software deliverable, the acquisition of data regarding these components, the incorporation of data into structured SBOM formats (i.e., SPDX, CycloneDX, and SWID) and SBOM format validation. NTIA reports offer comprehensive generation guidelines throughout the workflow and guide practitioners on effective utilization of tools for SBOM creation and validation.

SBOM Sharing and Exchange Playbook [8, 17] by NTIA [8] and CISA [17] detail the SBOM sharing stage, offering information on related terminology and technologies involved. NTIA introduces two main components of SBOM sharing, starting with the advertisement and discovery of the SBOMs, followed by the subsequent transfer of discovered SBOM to the consumer. CISA further refines these concepts and structures the SBOM sharing life cycle into discovery, access, and transport phases. Furthermore, CISA introduces SBOM enrichment as a form of SBOM consumption, which can instantiate another instance of sharing lifecycle with enriched

ICSE 2024, April 2024, Lisbon, Portugal

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

metadata. These reports also introduce various levels of sophistication related to the implementation of sharing life cycle stages and analyse the sophistication levels of existing sharing technologies.

SBOM Consumption Playbook [9] delineates SBOM consumption workflows, encompassing the acquisition, management, and utilization of SBOMs. The proposed workflow progresses from the SBOM sharing lifecycle stage to the subsequent stage of SBOM consumption, viewed from the consumer's perspective. This phase starts with the SBOM acquisition stage, where consumers access the SBOMs, facilitated by the advertisement and discovery of SBOMs. The workflow then proceeds to the management of SBOMs using storage solutions, adhering to related metadata retention policies and to the resolution and utilization through the extraction, transformation, and loading (ETL) of SBOMs into enterprise platforms.

SBOM Tool Taxonomy [7] classifies tools for generating and consuming SBOMs, providing a functional breakdown of SBOM production, consumption and transformation components.

Secure Software Factory (SSF) Reference Architecture [2] specifies components, interfaces, and control structures for constructing a secure software factory that can generate verifiable metadata documents during software artefact build. SSF focuses on creating and distributing attestations constructed from these metadata documents. The SSF components include Node Attestator, Workload Attestator, and Pipeline Observer, which handle the generation and signing of SSC metadata. An SSF also contains an Artefact Repository that handles the distribution of the artefacts generated by an SSF, including the signed metadata.

Supply Chain Integrity, Transparency and Trust (SCITT) Architecture [3], formerly known as Supply Chain Integrity Model (SCIM)[1], presents a scalable and decentralised architecture for generating, distributing, and consuming software attestations. The architecture introduces Transparency Service and Registry components for attestation storage and distribution. It also defines the necessary workflows for storing attestations, generating receipts and utilising these SSC metadata forms to verify the software products.

Supply-chain Levels for Software Artifacts (SLSA) Security Framework [16] specifies varying levels of control for enhancing supply chain security, particularly during the build process. It emphasises secure provenance and attestation generation. SLSA mandates package managers to define build processes and automatic provenance generation but lacks explicit detail on distribution and verification. However, it does not provide a functional decomposition of the mentioned SSC metadata life cycle.

Aforementioned frameworks and reference architectures address various crucial aspects related to secure SSC and SSC metadata management. While SCRM framework provides useful SSC concepts and contributes to providing modelling the SSC with its actors, their activities and interactions, it does not delve into discussions on machine readable SSC metadata management. Rest of the frameworks and architectures covers different types of machine readable SSC metadata and lifecycle stages. However, their scope is often limited to specific types of metadata and particular stages of the metadata lifecycle. We analyse this through a qualitative comparison based on metadata types, metadata standards and the level of coverage provided across different stages of the metadata life cycle (see Table 1). We establish four levels of coverage for each life cycle stage, ranging from lowest coverage to highest as follows: 0 if it does not offer any details about the stage, 1 if it provides a description of the life cycle stage, 2 if it describes workflow activities and actors, and finally, 3 if it provides functional decomposition and describes involved software components. As depicted in Table 1, while existing fremeworks and architectures collectively cover different metadata types and different stage of the metadata life cycle, none of them provide a comprehensive reference architecture for the management of the entire lifecycle of SBOMs, Provenance and Attestation with a focus on workflows from their generation to meaningful consumption.

Table 1: SCM2 Support of existing SSC Security Frameworks

Framework/ Architecture	Supported SSC Metadata Types	Supported SSC Metadata Standards	Metadata Life Cycle Coverage		
			Generation	Sharing	Consumption
In-toto Attestation Framework	Attestation Provenance (Partially)	in-toto attestation specification	3	1	3
Software Delivery Governance	Provenance	N/A	1	3	1
SCVS	SBOM	SPDX	1	0	1
SBOM Generation Playbook	SBOM	Any	2	1	1
SBOM Sharing Playbook	SBOM	Any	0	2	1
SBOM Consumption Playbook	SBOM	Any	0	1	2
SSF Architecture	Attestation	Any	3	3	0
SCITT	Attestation	SCITT Transparent Statement	1	3	2
SLSA	Provenance Attestation	SLSA Provenance Model, SLSA Software Attestation Model	1	1	1

While they provide a plethora of rich, but fragmented knowledge on concepts, architectures and tools, they lack common terminology and architectural vision, which hinders rapid SSC metadata adoption in practice. In light of this, we utilize these industry-driven architectures and frameworks to extract concepts to create a RA consisting of a coherent domain model and a technology-agnostic architectural blueprint, which can be readily embraced and customized by stakeholders to implement concrete solution architectures for SSC metadata management in practice.

REFERENCES

[1] C J Alberts, A J Dorofee, R Creel, R J Ellison, and C Woody. 2011. A Systemic Approach for Assessing Software Supply-Chain Risk. In *2011 44th Hawaii International Conference on System Sciences*. IEEE. https://doi.org/10.1109/hicss.2011.36

[2] CNCF. 2022. *The Secure Software Factory: A reference architecture to securing the software supply chain.* Technical Report.

[3] Henk Birkholz; Antoine Delignat-Lavaud; Cedric Fournet; Yogesh Deshpande. 2023. *An Architecture for Trustworthy and Transparent Digital Supply Chains.* Technical Report.

[4] in toto. 2017. *in-toto Specification v0.9.* Technical Report.

[5] NTIA. 2021. *Framing Software ComponentTransparency: Establishing a Common-Software Bill of Materials (SBOM).* Technical Report.

[6] NTIA. 2021. *How-To Guide for SBOM Generation.* Technical Report.

[7] NTIA. 2021. *SBOM Tool Classification Taxonomy.* Technical Report.

[8] NTIA. 2021. *Sharing and Exchanging SBOMs.* Technical Report.

[9] NTIA. 2021. *Software Consumers Playbook: SBOM Acquisition, Management, and Use.* Technical Report.

[10] NTIA. 2021. *Software Suppliers Playbook: SBOM Production and Provision.* Technical Report.

[11] NTIA. 2021. *Survey of Existing SBOM Formats and Standards.* Technical Report.

[12] OWASP. 2020. OWASP Software Component Verification Standard Version 1.0. https://owasp.org/www-project-software-component-verification-standard/.

[13] SEI. 2010. *Evaluating and Mitigating Software Supply Chain Security Risks.* Technical Report.

[14] SEI. 2010. *Software Supply Chain Risk Management: From Products to Systems of Systems.* Technical Report.

[15] Kapil Singi, Jagadeesh Chandra Bose R P, Sanjay Podder, and Adam P. Burden. 2019. Trusted Software Supply Chain. In *2019 34th IEEE/ACM International*

[1] https://github.com/microsoft/scim

Conference on Automated Software Engineering (ASE). IEEE. https://doi.org/10.1109/ase.2019.00141

[16] SLSA. 2023. SLSA Specification Version 1.0 RC. https://slsa.dev/spec/v1.0-rc1/onepage.

[17] Jeremiah Trent Stoddard, Michael Adam Cutshaw, Tyler Williams, Allan Friedman, and Justin Murphy. 2023. *Software Bill of Materials (SBOM) Sharing Lifecycle Report*. Technical Report.