CNCF PUBLIC SECTOR WORKING GROUP

PUBLIC SECTOR SOFTWARE SUPPLY CHAIN

Authors

Daniel Moch (Lockheed Martin)
William Crum (Spectro Cloud)
Ihor Dvoretskyi (Cloud Native Computing Foundation)
Ian Dunbar-Hall
Hari Kunduru (Applied Research Associates)
Sean Bentley (Boeing)

Reviewers

Christopher Robinson (OpenSSF Chief Architect)
Joel Krooswyk (GitLab Federal CTO)
Jordan Kasper
Zach Steindler (OpenSSF TAC, GitHub)

CLOUD NATIVE COMPUTING FOUNDATION

PUBLISHED
NOVEMBER 9, 2025

EXECUTIVE SUMMARY

The software supply chain is a vital component of successful software development organizations. However, incidents such as the 2020 Solarwinds attack, the 2021 Log4J vulnerability (Log4Shell), the 2024 xz backdoor (CVE-2024-3094), and the 2025 "tj-actions/changed-files" supply chain attack (CVE-2025-30066) have demonstrated how easily backdoors and breaches in the software supply chain can be exploited, impacting organizations and individuals globally. The Log4J vulnerability, in particular, is notable due to its widespread deployment and the staggering 10 million exploitation attempts per hour reported just one month after its discovery. Just one month after being discovered, the Wall Street Journal had identified a staggering 10 million exploitation attempts per hour.[1]

Organizations that implement robust secure software supply chain tools and practices are able to respond faster to such incidents, thanks to increased visibility and transparency. But a rising tide lifts all boats and a secure software supply chain would significantly mitigate the risk of such attacks by ensuring the integrity and authenticity of software dependencies from development to deployment, preventing malicious code or unauthorized modifications.

The Cloud Native Public Sector User Group[2] was formed in 2023 to serve as a hub for discussing and advancing cloud computing within the public sector. Alongside enumerating current best practices, we are dedicated to improving public sector workflows and supply chain security by advocating for the development and implementation of secure and resilient cloud-native software found within the public sector.

In this whitepaper, we aim to clearly address the current and future challenges of securing the public sector software supply chain, and propose long-term, sustainable solutions for using open source technologies to meet the needs of government systems, whilst ensuring cost-effective solutions exist for the software supply chain.

TABLE OF CONTENTS

Problem Statement

The CNCF Public Sector User Group would like to leverage secure software supply chain (S3C) tooling, developed in the broader software industry, to secure the software development lifecycle.

We believe a practical solution will allow for leveraging the S3C work of conscientious open source maintainers, be easy to adopt by governments, small business, and large entities alike, and enable trust and transparency between partners within the public sector. Furthermore, it should aid compliance with NIST standards and other international standards and guidelines (see "References" in the Appendix for a non-exhaustive list of relevant documents).

Inside the Attack Vector

Understanding how these attacks occur is crucial for developing effective defenses. Here are some common attack vectors in the software supply chain:

- **Dependency Confusion:** Attackers exploit the use of internal package names by publishing malicious packages with the same names to public repositories. When developers inadvertently download these malicious packages, the attackers gain access to the system.
- **Typosquatting:** Attackers create packages with names similar to popular libraries, hoping that developers will mistype the name and download the malicious package instead.
- **Compromised Maintainers:** Attackers target maintainers of popular open source projects, gaining access to their accounts and injecting malicious code into legitimate packages.

Timeline of the XZ Backdoor Hack into Linux (CVE-2024-3094)

By understanding these attack vectors, the CNCF Public Sector User Group can better design and implement secure software supply chain practices to mitigate these risks and protect the integrity of the software development lifecycle.

- **Build Server Compromise:** Attackers infiltrate the build servers used to compile and package software, inserting malicious code during the build process.
- **Code Injection in CI/CD Pipelines:** Attackers exploit vulnerabilities in Continuous Integration/ Continuous Deployment (CI/CD) pipelines to inject malicious code into the software during the development process.
- **Backdoored Dependencies:** Attackers contribute seemingly benign code to open source projects, which is later revealed to contain backdoors or other malicious functionality.

Audiences and Desired Outcomes

The Public Sector User Group (part of the Cloud Native Computing Foundation) has drafted this white paper with three goals. Correspondingly, we address the concerns listed below. The remainder of this section will summarize these concerns in more detail.

1. Hosting and discoverability of S3C artifacts
2. Public sector hosting of S3C infrastructure
3. A design for an end-to-end S3C solution useful to the public sector

First, we address software maintainers and registry operators to advocate for the inclusion alongside software releases of artifacts that would aid in securing software supply chains. Such artifacts include software bill-of-materials (SBOM), vulnerability exploitability exchange (VEX), and Supply-Chain Levels for Software Artifacts (SLSA) provenance.

Second, we address government organizations with an interest in securing software supply chains. Such organizations can partner with their contractors in this effort by providing infrastructure that parallels the "public good" infrastructure available to the open source community. In this paper we advocate for the government to provide such infrastructure to enable secure software deliveries between contractors and from contractors to the government.

Finally, we see a need for a reference architecture public sector organizations can use to apply these new artifacts toward securing their software supply chains. This paper lays out such an architecture in detail, recommending specific technologies that can help address the challenges facing the public sector. Before we do that, though, we need to discuss what those challenges are.

Challenges in the Public Sector

Organizations serving the public sector face an evolving set of hurdles that render supply chain security particularly challenging. Not only are these organizations consistently navigating strict procurement requirements and tight regulatory frameworks, they are often operating in a fast-evolving geopolitical landscape where they have little control over external pressures and demands.

In addition, public sector organizations are rapidly preparing for the introduction of mandates outlined in the requirements of Strategies for the Integration of Software Supply Chain Security in DevSecOps CI/CD Pipelines (NIST SP 800-204D[3]), based on Executive Order 14028, which will soon become a standard practice.

Presently there are several challenges to software supply chain security that hinders standardization of practices. There is no uniform standard provided by security groups about what makes its way into an isolated network, and therefore no easy way to automate the process. There is no standard non-public root of trust. There is no guidance on integrating open source attestations with private attestations, nor is there a framework for how to leverage shared, non-public infrastructure.

Against this backdrop:

- How can the public sector ensure that their consumers receive attestations that attest to the creator, source, and integrity of software products between entities (companies, networks, and customers)?
- What specifically is needed for public partners to trust the deliverables they have received?
- Is there a minimum level of assurance required to trust integrity, for example, non-public PKIs and Auditing ledgers shared between public sector organizations?

The SSDF answers this question at a high level and how will this level of assurance evolve to meet future customer requirements for transparency across entities. The architecture laid out below is intended to satisfy this minimum level and beyond.

Lastly, how can we ensure that the public sector remains competitive by proposing solutions that do not unduly burden smaller suppliers, who must work with the same stringent customer requirements but significantly less resources than larger suppliers?

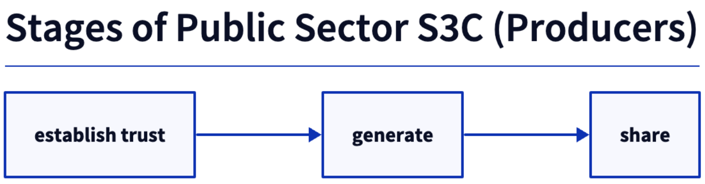
Software Registries

Critical to securing the software supply chain is getting security-related artifacts from producers (i.e., open source maintainers, private industry, and elsewhere within the public sector) to their users and customers. Software Registries, such as Maven, PyPI, currently play a central role in distributing software artifacts. We believe they should embrace this by additionally offering to host and distribute supply-chain related artifacts, including SBOM, SLSA and VEX artifacts.

Registries make it easy for developers to publish their software, making it available for broad use by others with access to the registry. Once published, software becomes both highly discoverable and easy to download and make use of. We would like to see supply-chain artifacts included in this process. Doing so will require that registry operators make it possible for developers to upload these artifacts. Search and discovery interfaces should also be enhanced so potential users know when a published package includes these artifacts. Client-side registry tooling might also be enhanced by including the ability for users to verify attestation signatures.

Registry operators can also enhance the publishing process itself by supporting digital signing. "Trusted Publishers" is one example of this that PyPI has done. This has the effect of securing the publishing process, allowing trusted publishers such as GitHub to publish artifacts (including supply-chain artifacts) from within GitHub Actions Workflows.

To sum up, registry operators should work to make supply chain artifacts like SBOMs and SLSA Build Provenances available alongside software packages.

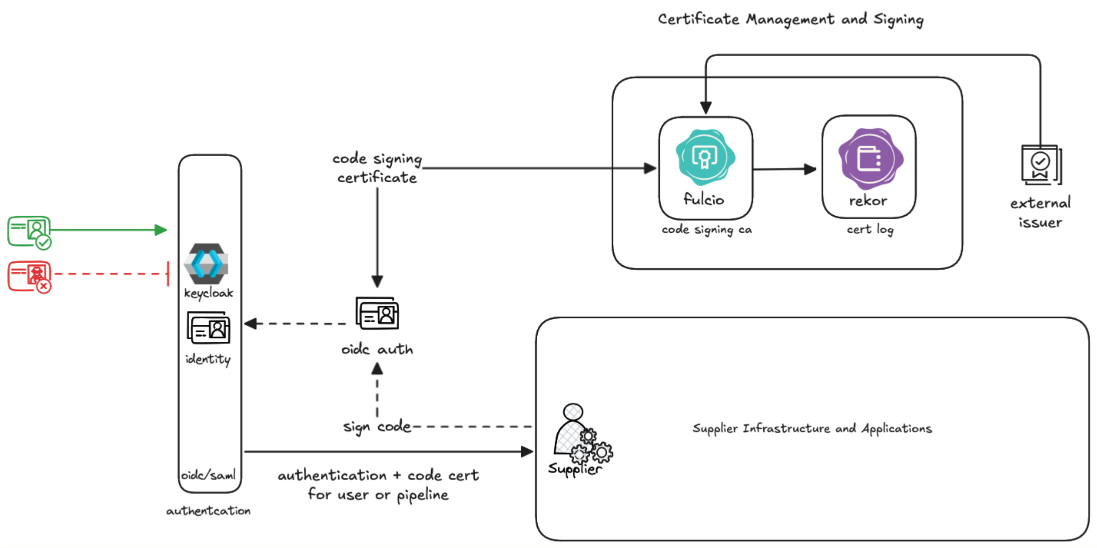


Supply Chain Reference Architecture

The architecture described in the following subsections addresses the common need for public sector contractors and other entities to gain insight into their software supply chains. This insight enables a further goal: enforcing policies on software components as they are shepherded through the supply chain.

We approach the following architecture functionally, conceptualizing the supply chain in three stages. The first stage, Establish Trust, considers how public sector contractors and entities can establish trust between one another for purposes of sharing software products. As such, it does not address the relationship between the public sector and the broader open source community. That relationship remains mediated primarily by registry operators.

The next stage, Generate, outlines the tooling and processes needed for public sector contractors and entities to author supply chain artifacts including signed attestations, vulnerability exchange disclosures,



and software bill-of-materials. The goal is for signatures and transparency ledgers to be non-public, ruling out the use of public-good instances of, e.g., Sigstore, while keeping them accessible to public sector entities that need them.

The final stage, Share, conceptualizes how to safely share software artifacts between mutually-trusted entities. This process allows for software products to be securely shared at any level of security provided there is network connectivity.

ESTABLISHING TRUST

Trust

Establishing trust is the core of S3C, and trustworthy signatures are the end product of a comprehensive plan to establish trust in an individual's digital identity.

NIST Special Publication 800-63, Digital Identity Guidelines[4], outlines the US Government guidelines for digital identity management at varying levels of assurance across identity proofing, authentication, and federation. As of September 2024, version Revision 3 is in force, but NIST is currently soliciting feedback for the fourth revision of this publication.

Identity Proofing

SP 800-63A describes the requirements applied to federal agencies seeking to implement digital identity services, and is not normative for other purposes. It is, however, a useful resource for organizations looking to implement a robust process to establish user identity.

The purpose of identity proofing is to reasonably ensure that public sector employees are who they say they are, allowing, in turn, for public sector organizations to make trustworthy claims regarding the security of their supply chains. Personally identifiable information collected in the course of employee hiring and onboarding is probably sufficient to this purpose.

The goal is not in general for employees to act as individual agents in the supply chain. It is organizations and not individuals that formally attest to the provenance of software builds.

Public sector organizations should ensure that the PII collected during employee hiring and onboarding is sufficient to establish identity to at least IAL2 per SP 800-63A.

Authentication

SP 800-63B supplements SP 800-63A with guidance on best practices for *authentication* – to provide a high degree of confidence that a user possesses two single-factor authenticators, or a multi-factor authenticator bound to the user's account.

Authentication is often achieved today via protocols that support single sign-on (SSO) like Security Assertion Markup Language (SAML) or OpenID Connect (OIDC). Second- or multi- factor authentication is usually achieved with time-based one-time codes (TOTC) or with biometric authentication available on many devices today.

Public sector organizations should ensure that authenticator practices achieve *at least* AAL2 per SP 800-63B.

Federation

SP 800-63C describes federation as it pertains to federated authentication, which is not relevant to our use case. However, federation is relevant insofar as we would like for attestations made by public sector actors in the course of S3C to be automatically trusted.

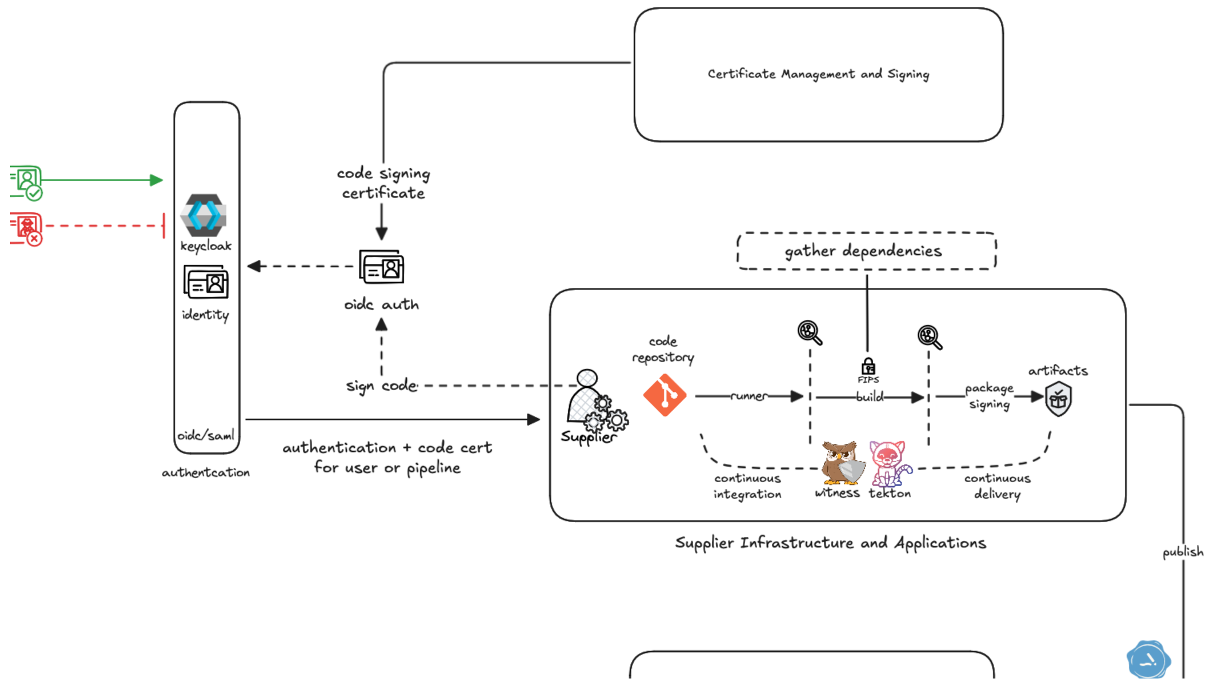
Nonetheless, SP 800-63 is foundational to the Federal Public Key Infrastructure guidance governing how cryptographic trust is established between the US federal government and non-government entities.

Of particular interest for our purposes is guidance regarding the Federal Bridge Certification Authorities (CA) mechanism, which provides for trust to be established across organizations. This means that any shared Sigstore instance used for public sector work will probably need to use a CA that is cross-trusted by the Federal Bridge CA with at least a medium level of assurance.

Finally, to enable trust across entities, public sector organizations should focus on workload identities instead of developer identities. Keycloak or other well-known OIDC providers should be leveraged alongside tools like Sigstore to create short-lived keys that minimize the impact of private keys leaking and provide an audit log of issued keys. Organizations should ensure that the PII collected during employee hiring and onboarding is sufficient to establish identity to at least IAL2 per SP 800-63A, and should ensure that authenticator practices achieve at least AAL2 per SP 800-63B.

***Tools:** Keycloak, Dex*

GENERATING S3C DATA



Generate

Gather and Analyze

Before any product can be built, its dependencies must be gathered and analyzed. As part of this process, data required to generate complete SLSA Build Attestations and Verification Summary Attestations should also be collected, including:

- Data required for SLSA Resource Descriptor
- SLSA build level of the dependency (Level 0 means no SLSA compliance)

In this phase, organizations are essentially consumers and should analyze dependencies to ensure there are no open, critical vulnerabilities (i.e., CVE's) against the versions in use. Dependency information should also be fed into a continuous monitoring service like Dependency Track so that you will be proactively notified of any new vulnerability disclosures. Finally, may opt to apply policy enforcement to dependencies to ensure baseline security guarantees are met.

This process is not without issue. Package managers and repositories have become a critical component of modern language ecosystems, making it easy to find and download software packages. However, with few exceptions, these package managers do not yet support the inclusion of Build Provenance, SBOM, or other artifacts useful to secure supply chain management. Package registries should work to make supply chain artifacts like SBOM's and SLSA Build Provenances available alongside software packages (see Software Registries above).

Tools: *Dependency Track, Sonatype Nexus, JFrog Artifactory, Harbor, Zot, Distribution*

Build and Attest

Whichever build system organizations choose to leverage, it must be capable of generating a signed SLSA Build Provenance to satisfy SLSA Level 2 compliance. Ideally organizations should be working toward SLSA Level 3 compliance. This will require controls separating developers from build infrastructure, which will need to be managed by a different organization. An example would be GitHub actions, which supports SLSA Level 3. Alternatively, GitLab runners support up to SLSA Level 2 and they are working toward Level 3 support.

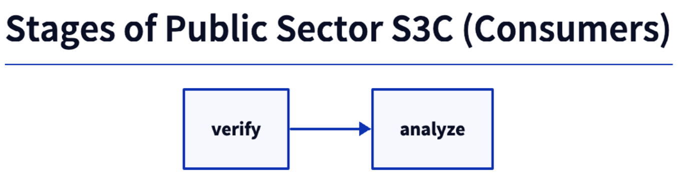
Build Provenance must be non-forgeable (achieving SLSA Level 2 or higher), and signed with keys issued by a shared, non-public Sigstore instance that includes Rekor to allow for transparency and possible audit. Where sufficient trust exists between parties, but where conveyance of a full SLSA Build Attestation is not advisable, a signed SLSA Verification Summary Attestation may be used, with the Build Attestation retained for possible out-of-band conveyance.

Tools: *Tekton, Witness, GitHub Actions, GitLab Runners*

SHARING S3C DATA

Share

Sharing artifacts involves having artifacts like packages, images, helm charts, etc. being stored in a repository that can be pushed/published to by suppliers and pulled/retrieved by consumers. The shared repository should at a minimum be OCI compliant, and ideally store at least helm charts. Projects like Harbor, Zot, and Distribution are a great option for securely storing and managing these types of artifacts. If possible, a shared repository could also support other popular package managers, e.g. cargo, npm, pypi, debian, rpm, maven, etc.



Tools: *Harbor, Zot, Distribution*

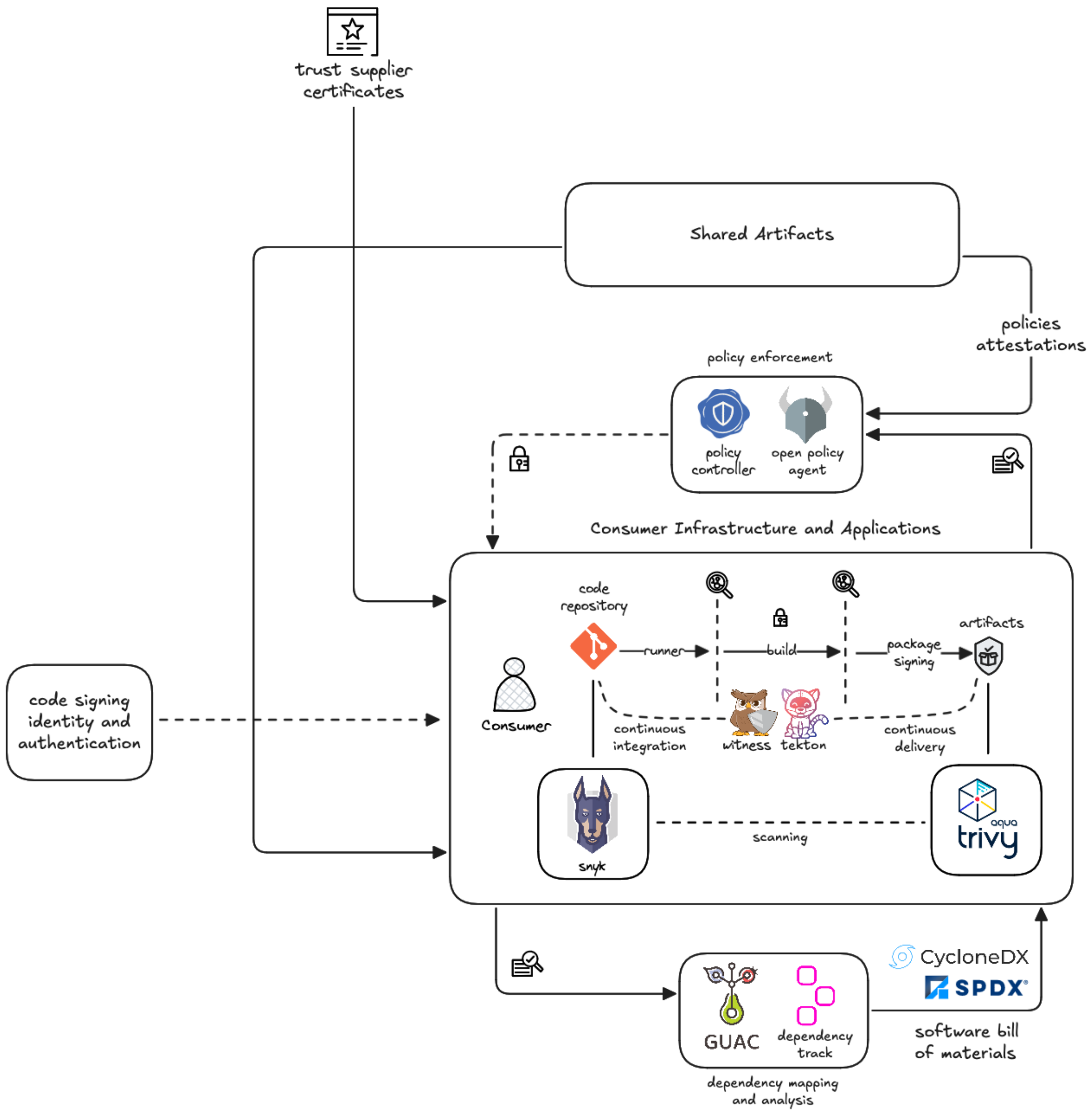
Attestations should be published in a shared storage medium, such as Archivista. When publishing to shared infrastructure is not advisable, attestations may be distributed party-to-party. For instance, an attestation may be included as part of a product delivery package.

Tools: *Archivista*

The Update Framework (TUF) is a software framework designed to protect mechanisms that automatically identify and download updates to software. TUF uses a series of roles and keys to provide a means to retain security, even when some keys or servers are compromised. Repository Service for TUF (RSTUF) simplifies the adoption of TUF by removing the need to design a repository integration. RSTUF encapsulates that design.[5]

Tools: *RSTUF*

VERIFYING & ANALYZING S3C DATA



Consumers

Upon receiving components, consumers should take a two-phased approach to ensuring security thresholds are met.

Verify

Firstly, downloaded dependency artifacts are verified against a supplied build provenance or Verification Summary Attestation (VSA). Organizations should apply policy enforcement to ensure baseline security standards.

Upon receipt of components, organizations should leverage VSA attestations to summarize attestations with traceability to build provenance. Governments should begin to accept these machine-readable attestations as evidence of compliance.

Tools: *cosign*

Analyze

Secondly, dependency information is analyzed using a combination of automated and manual tooling to uncover potential weak points in the supply chain. For manual tooling, Guac can visualize dependencies as a graph.

Taking an automated approach, DependencyTrack can continuously monitor dependencies for new security vulnerabilities, meanwhile CI/CD Pipelines ensure minimum security thresholds are met at build time (e.g. no critical vulnerabilities in any dependencies).

Tools: *Guac, DependencyTrack, CI/CD Pipelines (e.g., Jenkins, GitLab CI/CD), Snyk, Trivy*

Policy Enforcement

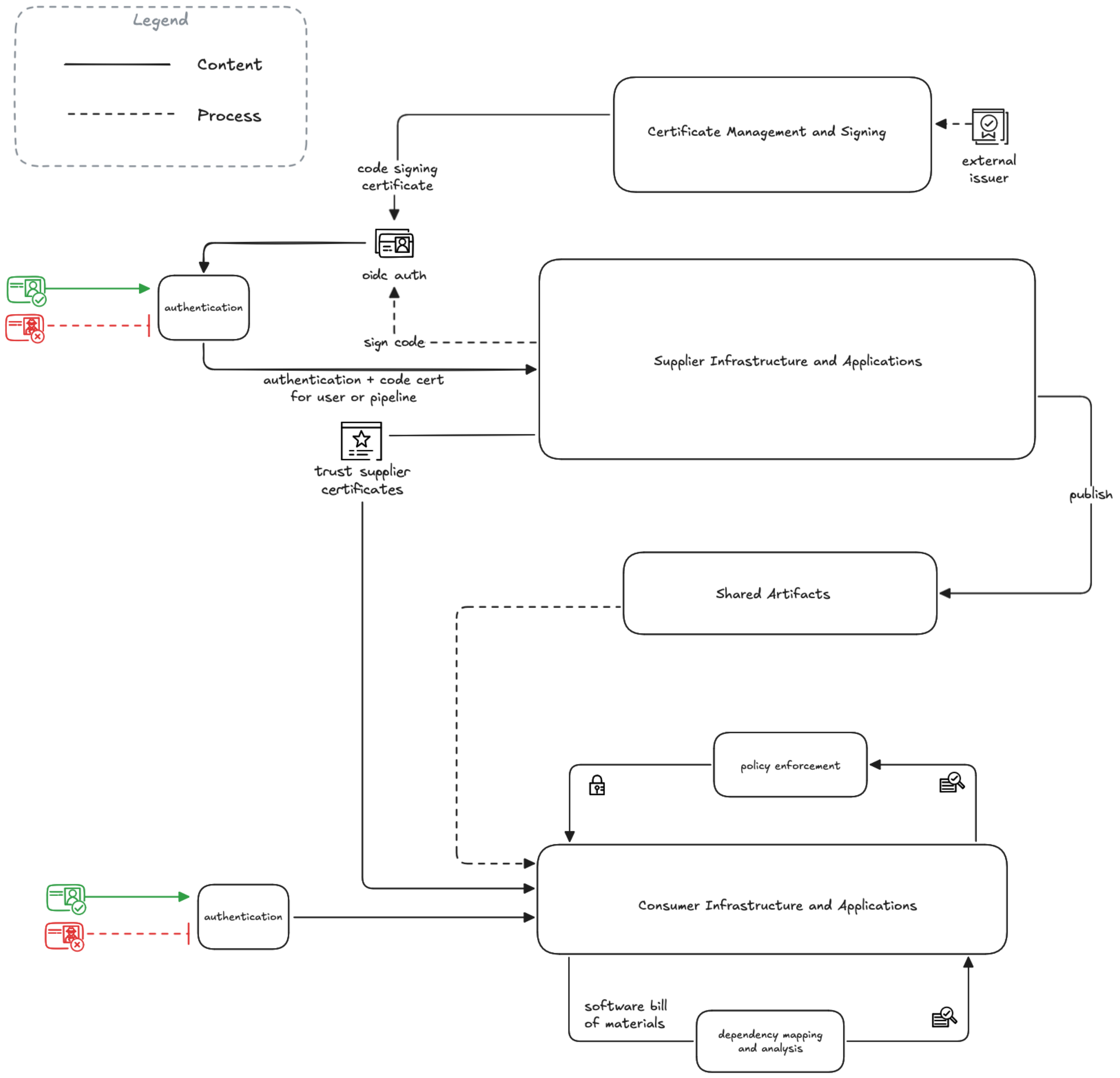
Policies play a critical role in defining the rules and standards that safeguard the software supply chain, establishing the expectations for behavior, procedures, and guidelines within an organization. Moreover, policies ensure that producers meet the security thresholds demanded by consumers, who, in turn, can trust the components they are receiving.

The Open Policy Agent (OPA), and its policy language Rego, and the Open Security Controls Assessment Language (OSCAL) can significantly enhance supply chain security and management of attestations. The in-toto Policies Working Group is also working to standardize policy enforcement with respect to attestations. Policies and automated enforcement enabled by, for example, Rego or OSCAL increase the trustworthiness of Verification Summary Attestations – ensuring components meet required consumer thresholds before admitting them into production environments. Furthermore, integrating additional tools such as Sigstore, Witness, Dependency-Track, and CI/CD providers bolsters security measures, providing comprehensive protection throughout the software development lifecycle.

Tools: *OPA, OSCAL, Sigstore, Witness, Dependency-Track*

TECHNICAL IMPLEMENTATION

Having established the functional architecture for a secure public sector software supply chain, we now turn to its implementation. Figure [Number] presents a detailed technical implementation diagram, outlining specific tools and technologies recommended for each stage of the process. This diagram moves beyond the conceptual framework and provides actionable guidance, identifying concrete solutions that public sector organizations can adopt to secure their software supply chains. Notably, the diagram highlights supplier and consumer infrastructure and applications, such as build pipelines based on Tekton and Witness, policy enforcement with OPA and policy controller, and dependency mapping and analysis with GUAC and dependency track. The diagram also includes shared resources such as registries/repositories for artifacts as well as parent certificate authorities for certificate distribution. By mapping these tools, the diagram provides a concrete pathway for organizations seeking to operationalize secure software supply chain practices.governments are in the best position to host this infrastructure.


Legend
Content
Process
Certificate Management and Signing
external issuer
code signing certificate
oidc auth
authentication
sign code
Supplier Infrastructure and Applications
authentication + code cert for user or pipeline
trust supplier certificates
publish
Shared Artifacts
policy enforcement
Consumer Infrastructure and Applications
authentication
software bill of materials
dependency mapping and analysis

ADOPTION ROADMAP

Given the foregoing discussion, this section lays out a notional adoption roadmap with a view to providing the tools necessary for government contractors to effectively secure their supply chains.

First, contractors should pursue adoption of the Consumer side of the above architecture internally. Not only can this be worked independently of other contractors and government provision of infrastructure, but it also provides for a critical view into the supply chain of open source software already in use. Next (and ideally in parallel), the government should stand up necessary infrastructure to allow for contractors to share S3C artifacts with one another without potentially sensitive information leaking into public view (which would happen if existing, public good infrastructure were used). Finally, contractors should use government-provided infrastructure to share S3C information between one another, and with the government.

CONCLUSION

The implementation diagram presented demonstrates the flexibility necessary to address evolving transparency demands while maintaining security and operational integrity. Future enhancements in cryptographic standards, regulatory frameworks, and inter-organizational cooperation can be accommodated through utilizing new projects or updated components.

- **Implementation Diagram:** Public sector organizations can leverage the proposed architecture to secure their supply chains using standardized S3C artifacts
- **Government-Provided Infrastructure:** Governments host common infrastructure for sharing S3C artifacts across public sector entities
- **Registry Support:** Software registries must begin providing native support for publishing and discovering required S3C artifacts alongside software packages

In this whitepaper, the CNCF Public Sector User Group has proposed enhancements that registry operators can make to increase the accessibility of S3C artifacts. Namely, we are asking registry operators to provide a means for maintainers to publish these artifacts alongside software packages, and for users to easily find and download S3C artifacts associated with them.

As a minimal level of assurance, Public Sector organizations should adopt standards and inclusive and federated infrastructure to ensure trust in identity, federation, and cryptographic requirements. For example, the United States includes NIST SP 800-53, NIST SP 800-63, and FIPS 140 which together cover security controls, digital identity assurance, and cryptographic module requirements. Other nations or international frameworks (such as ISO/IEC 27001, ISO/IEC 15408, ETSI EN 319 411) may serve as equivalent standards.

For public sector organizations, we have proposed an architecture composed of technologies that make it possible to use the same artifacts to secure their supply chains. Appropriately using this architecture will require new ways of working involving common infrastructure used across the public sector. We believe governments are in the best position to host this infrastructure.

APPENDIX

Endnotes

1 https://www.wsj.com/articles/what-is-the-log4j-vulnerability-11639446180

2 https://github.com/cncf/public-sector-user-group

3 Chandramouli R, Kautz F, Torres-Arias S (2024) Strategies for the Integration of Software Supply Chain Security in DevSecOps CI/CD Pipelines. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-204D. https://doi.org/10.6028/NIST.SP.800-204D

4 https://www.nist.gov/identity-access-management/nist-special-publication-800-63-digital-identity-guideline

5 https://repository-service-tuf.readthedocs.io/en/stable/

References

NIST SP 800-204D	Strategies for the Integration of Software Supply Chain Security in DevSecOps CI/CD Pipelines
Executive Order 14028	Improving the Nation's Cybersecurity
NIST SP 800-63	Digital Identity Guidelines
NIST SP 800-63A	Digital Identity Guidelines: Enrollment and Identity Proofing
NIST SP 800-63B	Digital Identity Guidelines: Authentication and Lifecycle Management
NIST SP 800-63C	Digital Identity Guidelines: Federation and Assertions

Glossary

Supplier: An organization or entity that provides, develops, or packages software for use by a Consumer. This is software vendors developing proprietary or open-source solutions for Public Sector use. Public Sector contractors producing software deliverables that must meet security and compliance standard (e.g., NIST SP 800 series). An open-source dependency (i.e. library) may or may not be a Supplier.

Consumer: An entity that receives and uses software from Suppliers while ensuring that the software meets security, compliance and operation requirements. These are Public Sector agencies that obtain software from private or open-source distribution channels or Public Sector Suppliers that rely on external software components in their own development process.

From: *https://slsa.dev/*

Attestation: An authenticated statement (metadata) about a software artifact or collection of software artifacts.

Provenance: Verifiable information about software artifacts describing where, when, and how something was produced.

Acronyms

AAL	Authenticator Assurance Level
CA	Certificate Authority
CI/CD	Continuous Integration/Continuous Deployment
CNCF	Cloud Native Computing Foundation
CRADA	Cooperative Research and Development Agreement
DIB	Defense Industrial Base
EKU	Extended Key Use
IaC	Infrastructure as Code
IAL	Identity Assurance Level
NIST	National Institute of Standards and Technology
OIDC	OpenID Connect
OPA	Open Policy Agent
OpenSSF	Open Source Security Foundation
OSCAL	Open Security Controls Assessment Language
PII	Personally Identifiable Information
S3C	Secure Software Supply Chain (sometimes referred to as SSC or SSSC across the ecosystem)
SAML	Security Assertion Markup Language
SBOM	Software Bill of Materials
SCA	Software Composition Analysis
SSO	Single Sign-On
SSP	System Security Plan
VEX	Vulnerability Exploitability Exchange
VSA	Verification Summary Attestation

THANK YOU!

CLOUD NATIVE
COMPUTING FOUNDATION