From Chaos to Clarity: Modern SBOM Practices That Actually Work

sbomify

IANAL

sbomify

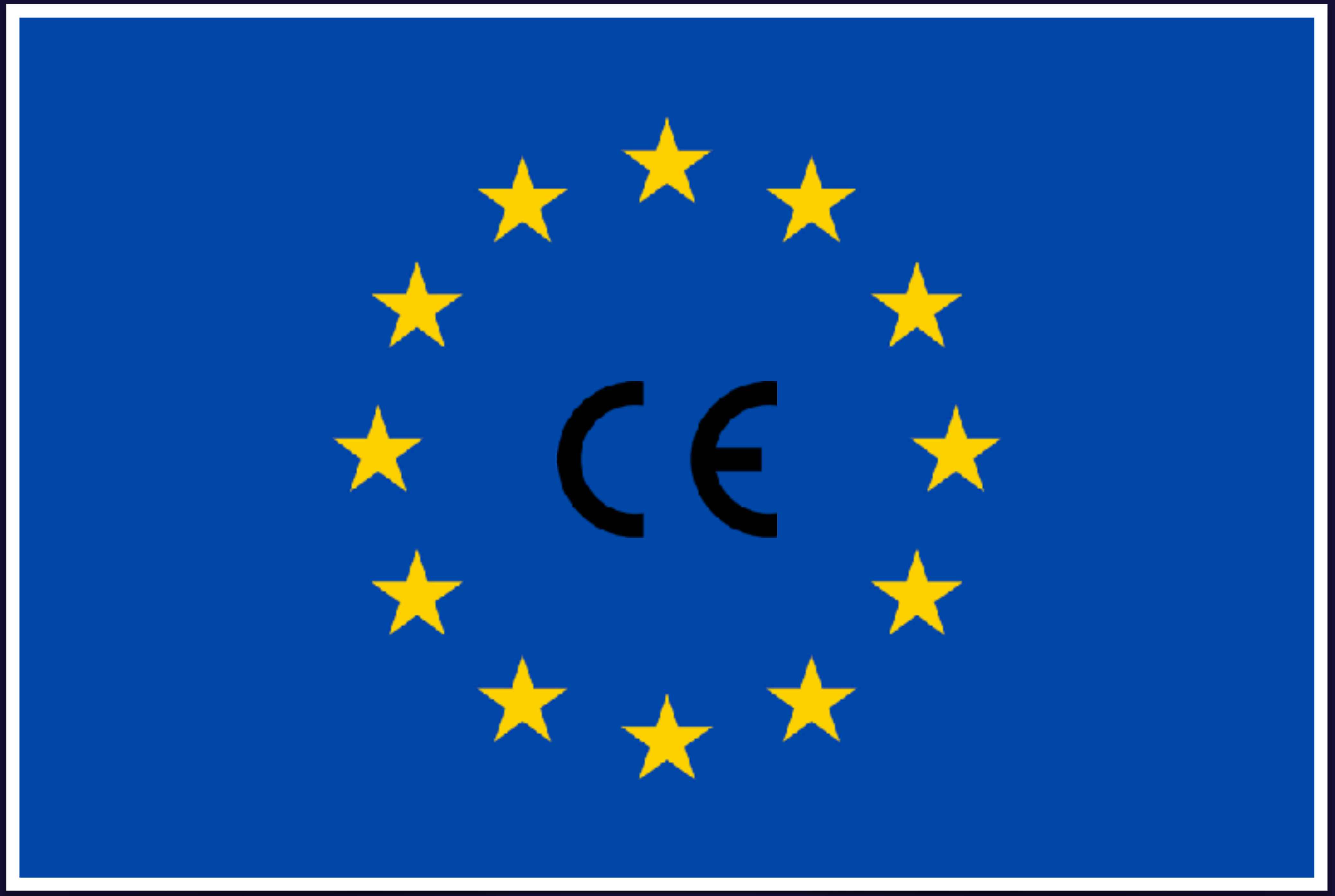
$ whoami

Pre-cap

1. Why you need to care now
2. How to generate high quality SBOMs
3. SBOM life cycle management
4. How to share SBOMs at scale

Why now?

sbomify



EU Cyber Resilience Act (CRA)

Obligations start **September 11, 2026** with full force on **December 11, 2027**

sbomify

This is just the start.

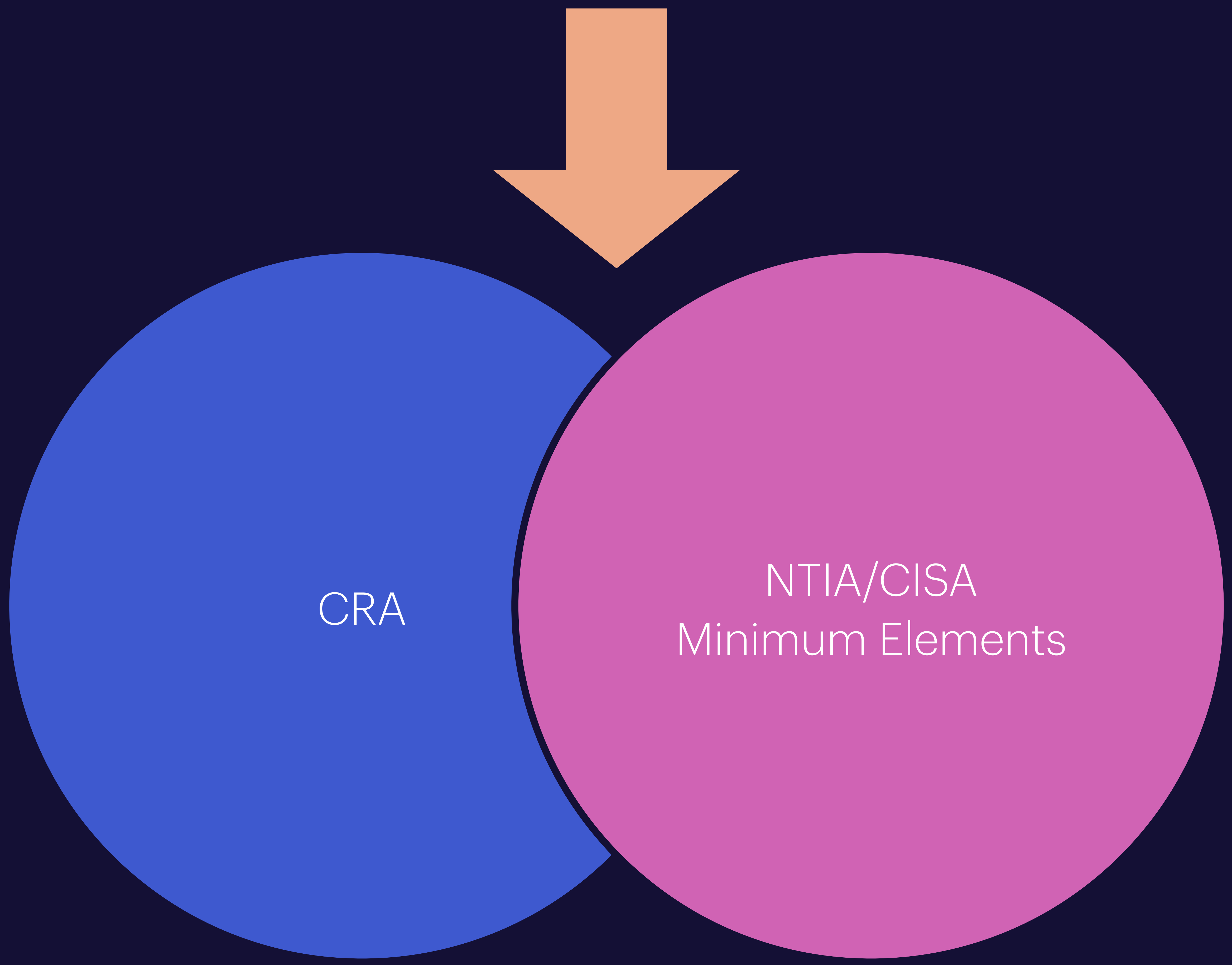
sbomify

FDA U.S. FOOD & DRUG ADMINISTRATION

PCI DSS COMPLIANT

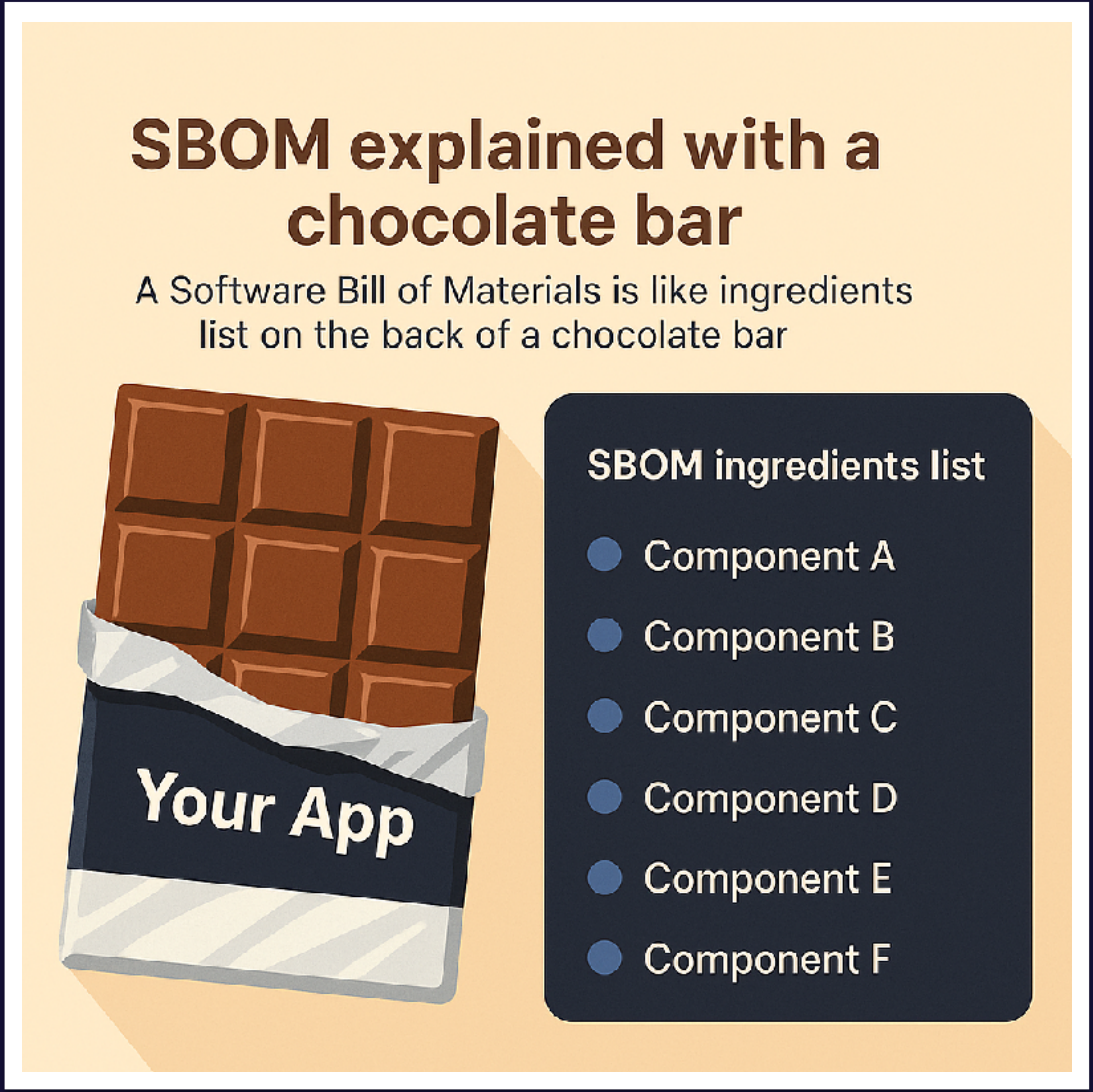
AICPA SOC 2 aicpa.org/soc4so SOC for Service Organizations™

ISO 27001



sbomify

But what **is** an SBOM?


SBOM explained with a chocolate bar
A Software Bill of Materials is like ingredients list on the back of a chocolate bar
Your App
SBOM ingredients list
Component A
Component B
Component C
Component D
Component E
Component F

sbomify

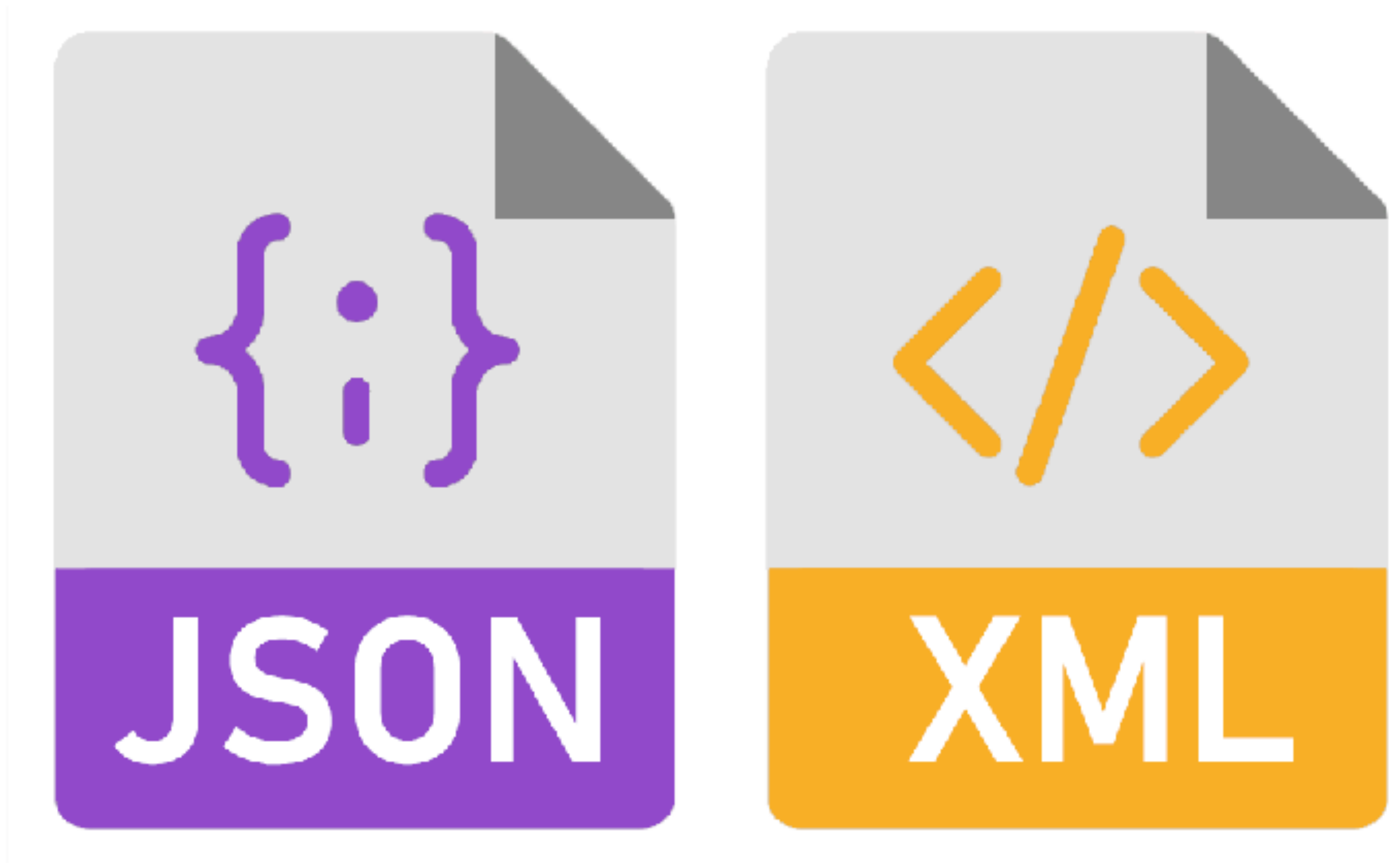
What are SBOMs used for?

1. Vulnerability management
2. License compliance

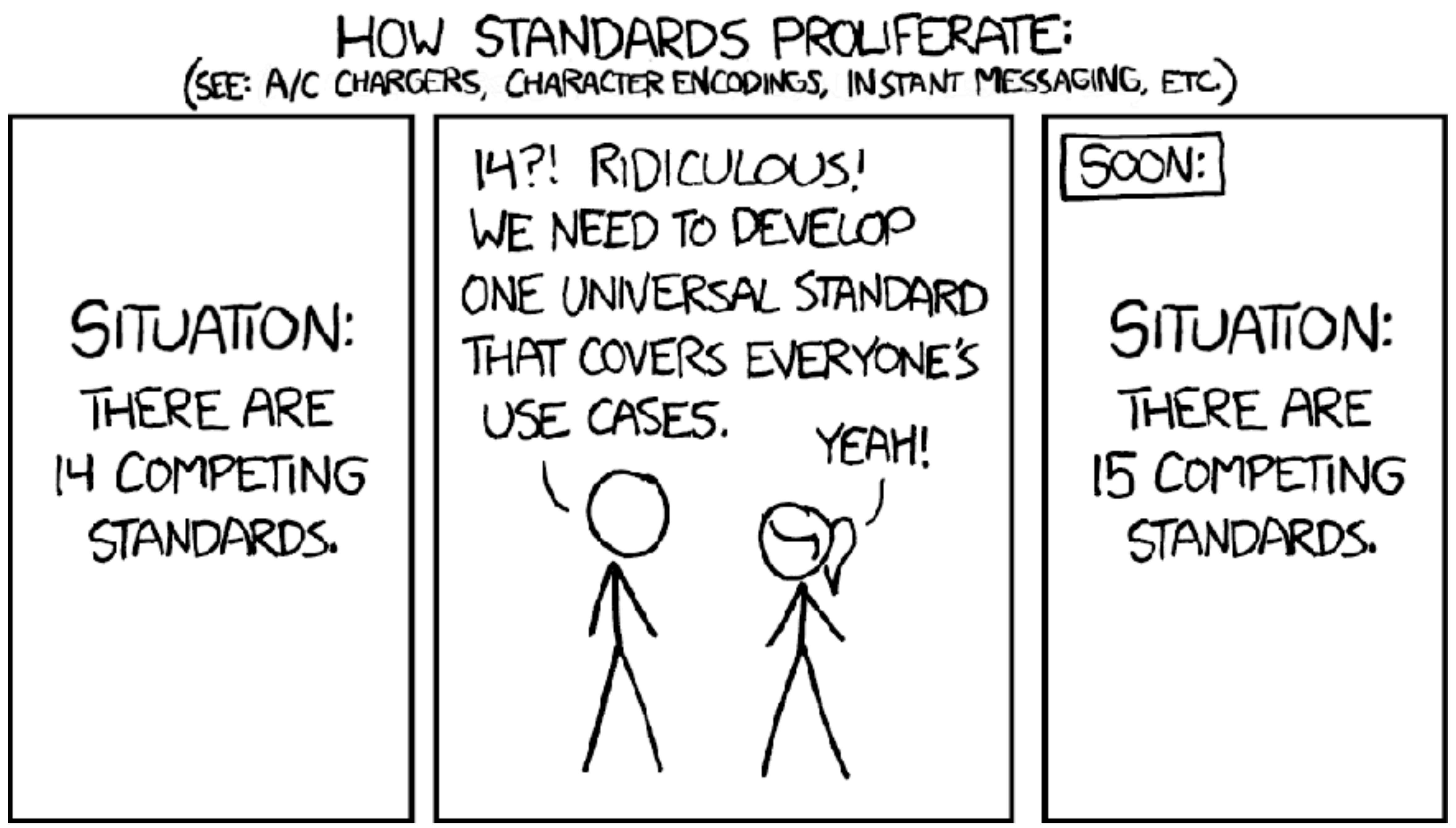
SPDX

sbomify

CycloneDX



sbomify


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

Dictionary

1. Known Exploited Vulnerabilities (KEV)
2. Vulnerability Exploitability eXchange (VEX)
3. Vulnerability Disclosure Report (VDR)

How did I get here?

sbomify

Screenly

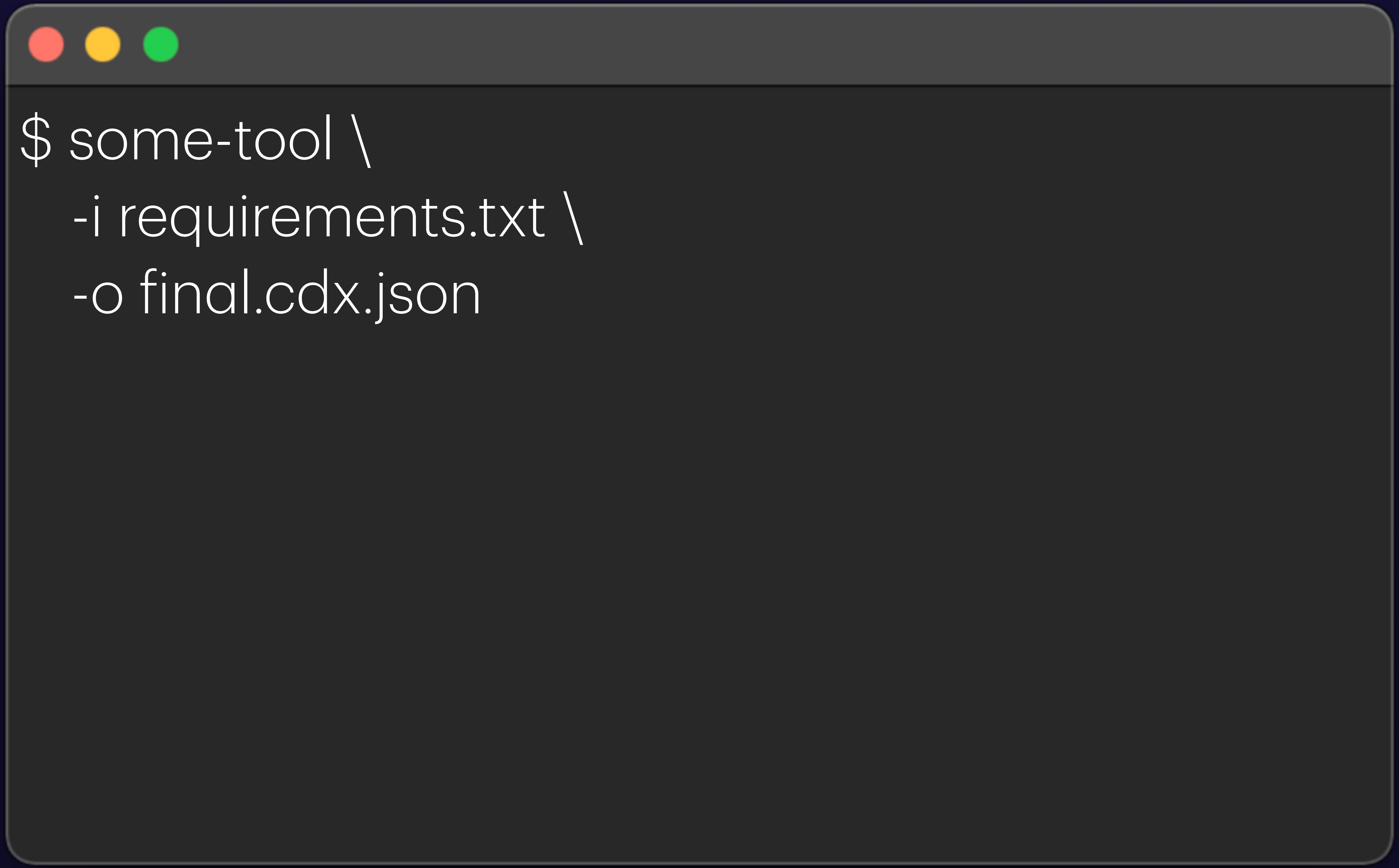
sbomify

Expectation

- Run one of the many tools
- Get a valid SBOM
- Move on

or

- Download SBOM from GitHub



```
$ some-tool \
   -i requirements.txt \
   -o final.cdx.json
```

sbomify

sbomify

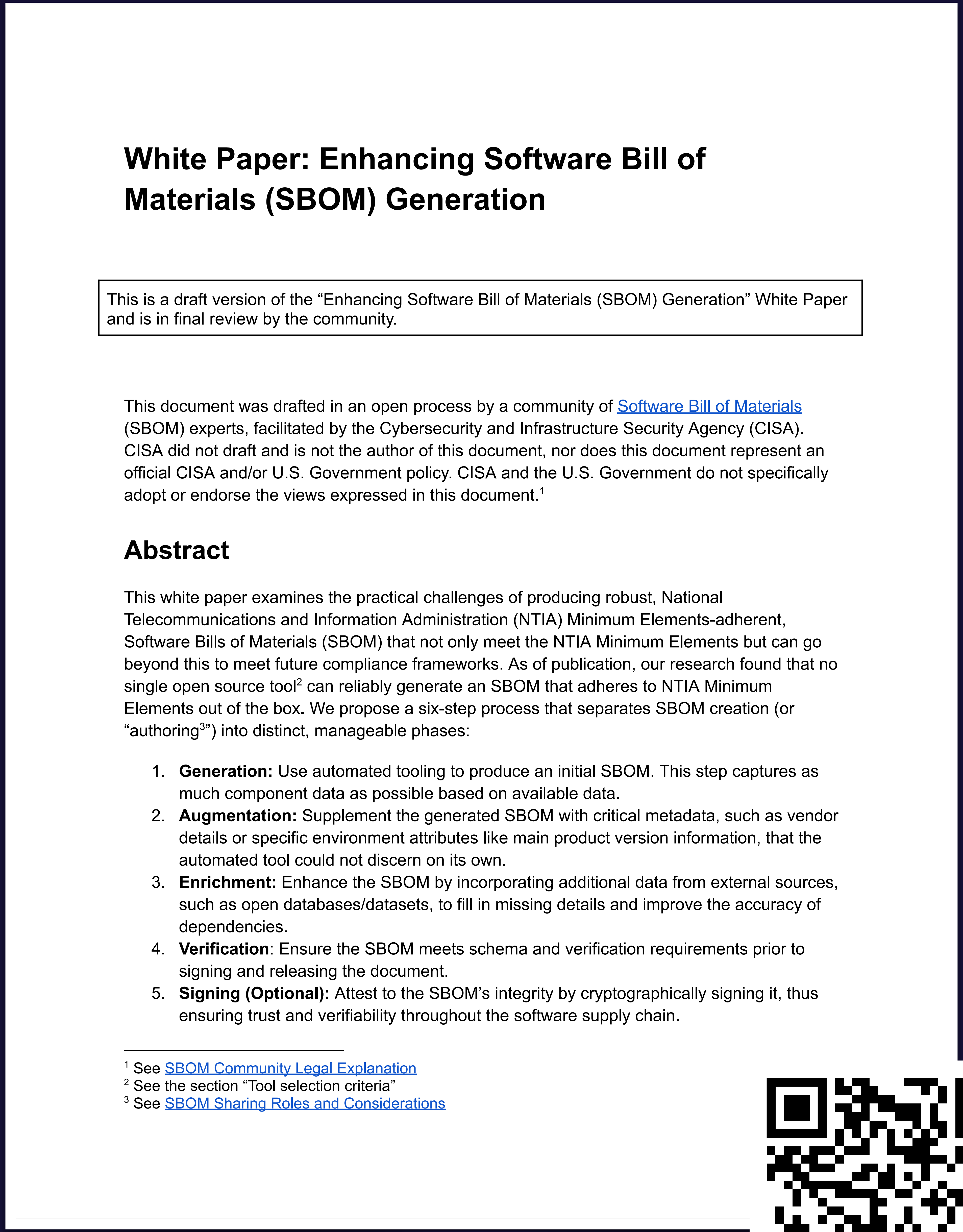
Dr. Allan Friedman
"Father of SBOMs"

sbomify

Formed a Working Group

sbomify

Wrote a white paper



White Paper: Enhancing Software Bill of Materials (SBOM) Generation

This is a draft version of the "Enhancing Software Bill of Materials (SBOM) Generation" White Paper and is in final review by the community.

This document was drafted in an open process by a community of Software Bill of Materials (SBOM) experts, facilitated by the Cybersecurity and Infrastructure Security Agency (CISA). CISA did not draft and is not the author of this document, nor does this document represent an official CISA and/or U.S. Government policy. CISA and the U.S. Government do not specifically adopt or endorse the views expressed in this document.[1]

Abstract

This white paper examines the practical challenges of producing robust, National Telecommunications and Information Administration (NTIA) Minimum Elements-adherent, Software Bills of Materials (SBOM) that not only meet the NTIA Minimum Elements but can go beyond this to meet future compliance frameworks. As of publication, our research found that no single open source tool[2] can reliably generate an SBOM that adheres to NTIA Minimum Elements out of the box**.** We propose a six-step process that separates SBOM creation (or "authoring[3]") into distinct, manageable phases:

1. **Generation:** Use automated tooling to produce an initial SBOM. This step captures as much component data as possible based on available data.
2. **Augmentation:** Supplement the generated SBOM with critical metadata, such as vendor details or specific environment attributes like main product version information, that the automated tool could not discern on its own.
3. **Enrichment:** Enhance the SBOM by incorporating additional data from external sources, such as open databases/datasets, to fill in missing details and improve the accuracy of dependencies.
4. **Verification**: Ensure the SBOM meets schema and verification requirements prior to signing and releasing the document.
5. **Signing (Optional):** Attest to the SBOM's integrity by cryptographically signing it, thus ensuring trust and verifiability throughout the software supply chain.

[1] See SBOM Community Legal Explanation
[2] See the section "Tool selection criteria"
[3] See SBOM Sharing Roles and Considerations

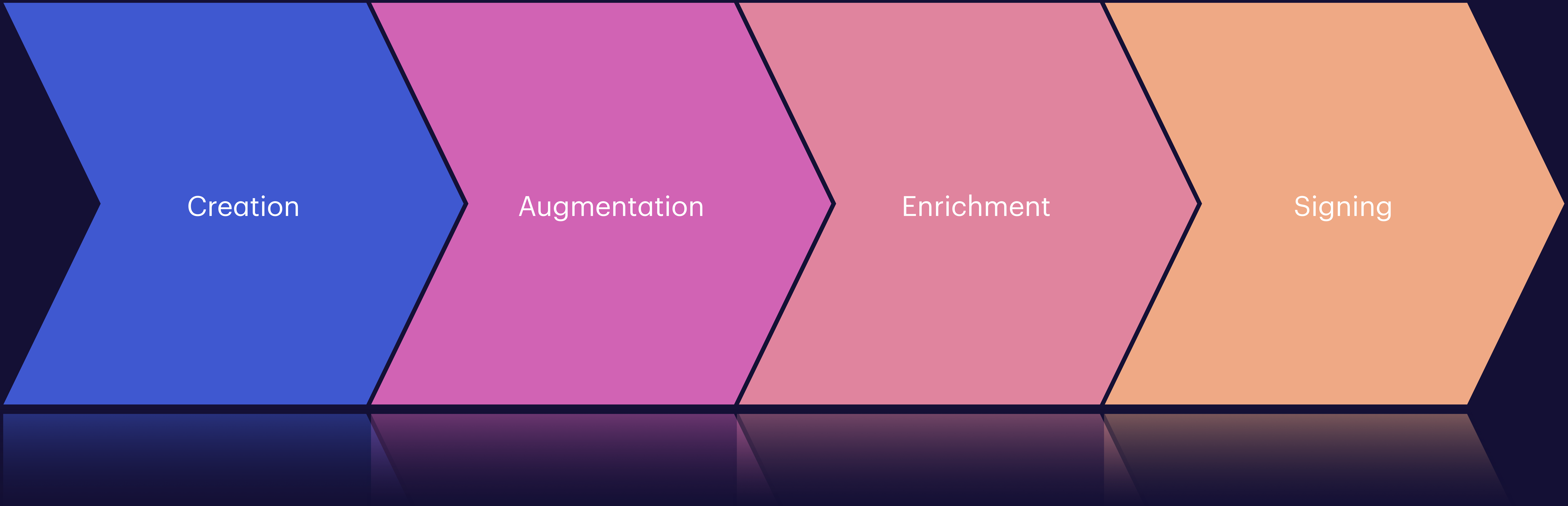
sbomify

What is NTIA/CISA Minimum Elements?

Data Field	Description	Status
➡ **Supplier Name**	The name of the entity that creates, defines, and identifies	Minimum element
Component Name	Designation assigned to a unit of software defined by the	Minimum element
Version	Identifier used by the supplier to specify a change in software	Minimum element
Other Unique Identifiers	Other identifiers used to identify a component or serve as a	Minimum element
Dependency Relationship	Characterizing the relationship that an upstream component	Minimum element
➡ **Author of SBOM Data**	The name of the entity that creates the SBOM data	Minimum element
Timestamp	Record of the date and time of the SBOM data assembly	Minimum element

sbomify

SBOM Generation Steps



sbomify

Generation Tools

SPDX

CycloneDX

syft

aqua trivy

sbomify ACTION

sbomify

Built-in Generation

yocto PROJECT

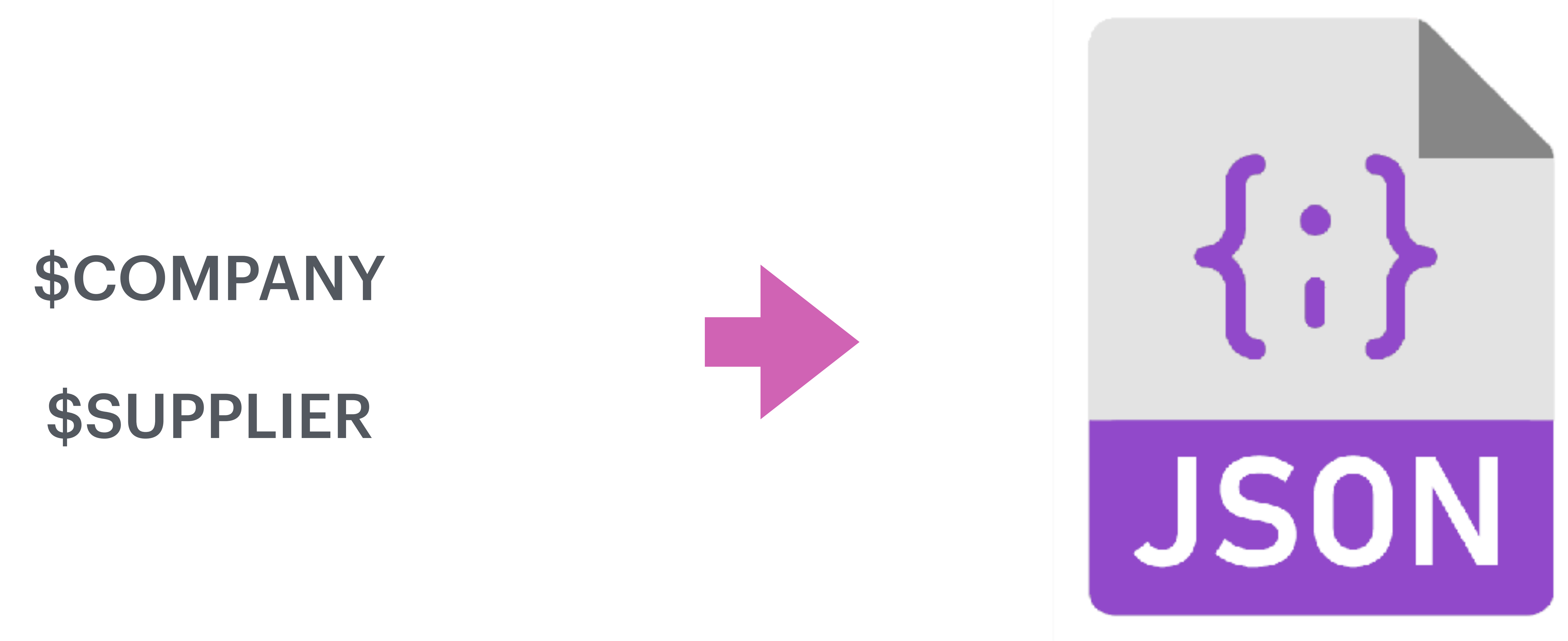
Zephyr®

sbomify

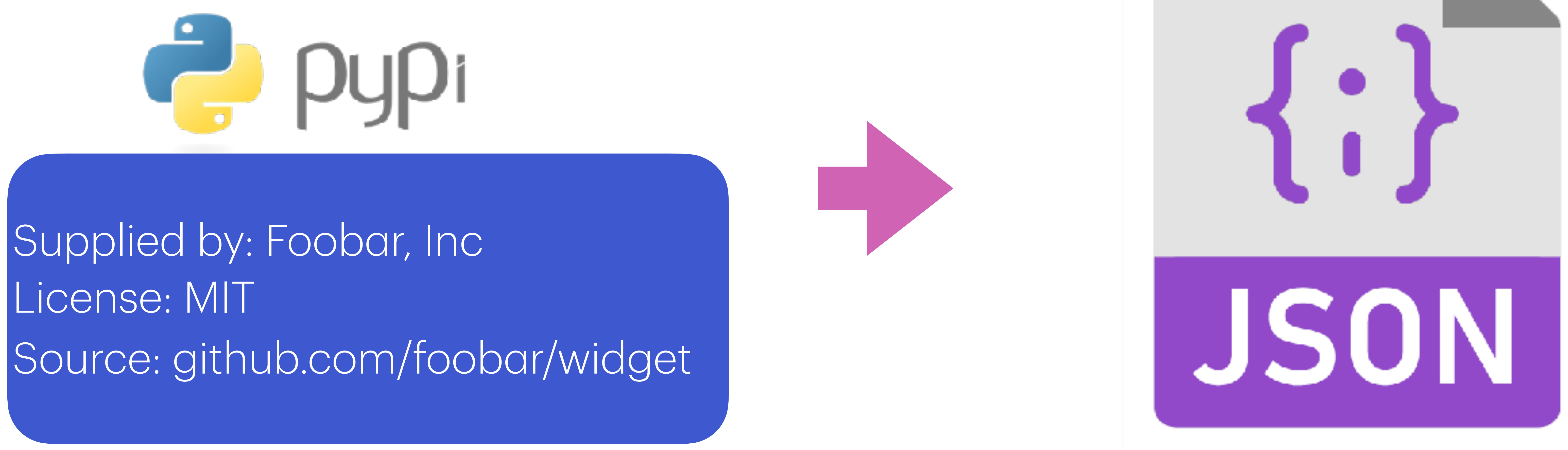
A word of warning!

sbomify

Augmentation

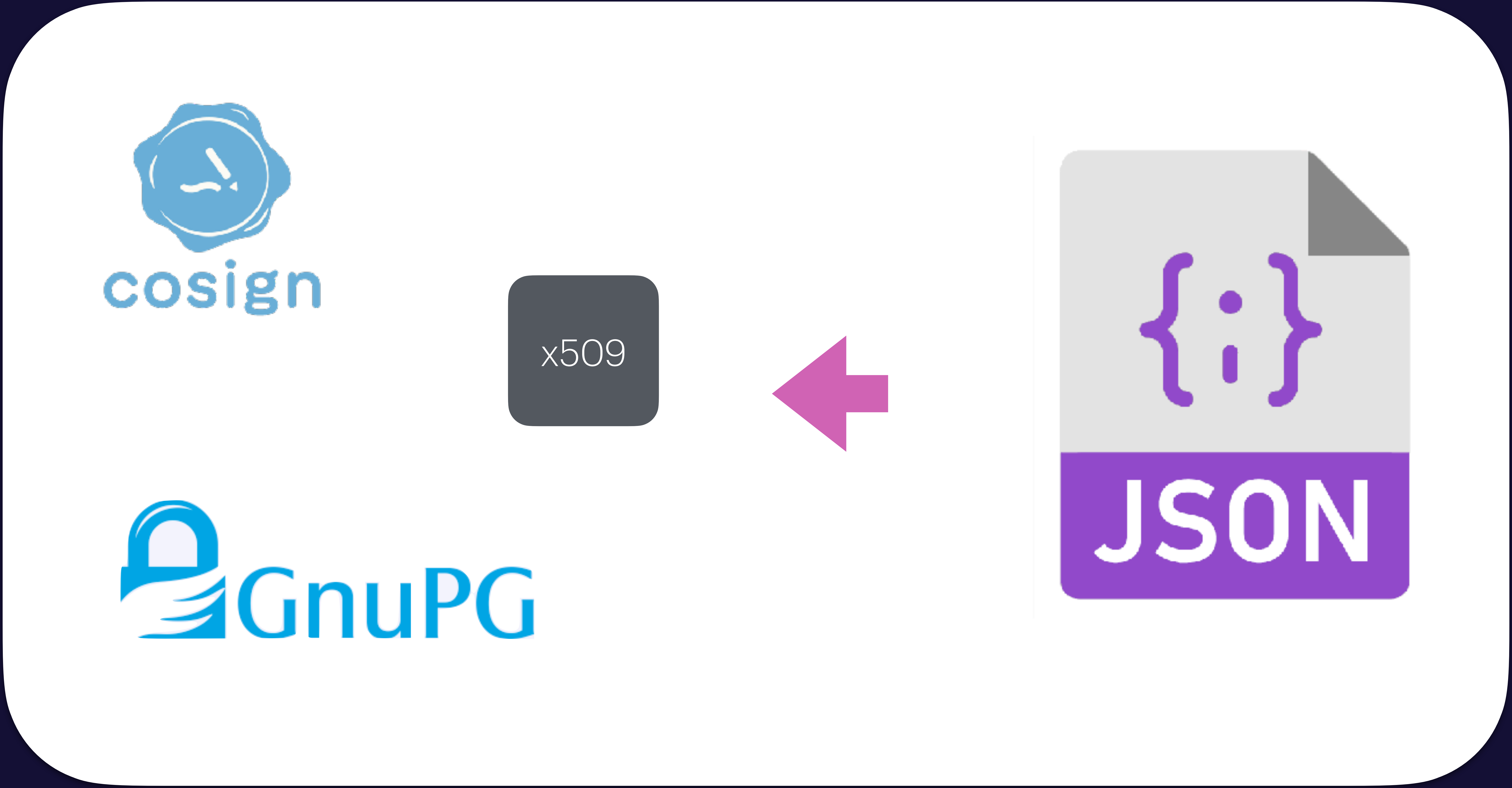


Enrichment



Supplied by: Foobar, Inc
License: MIT
Source: github.com/foobar/widget

Signing



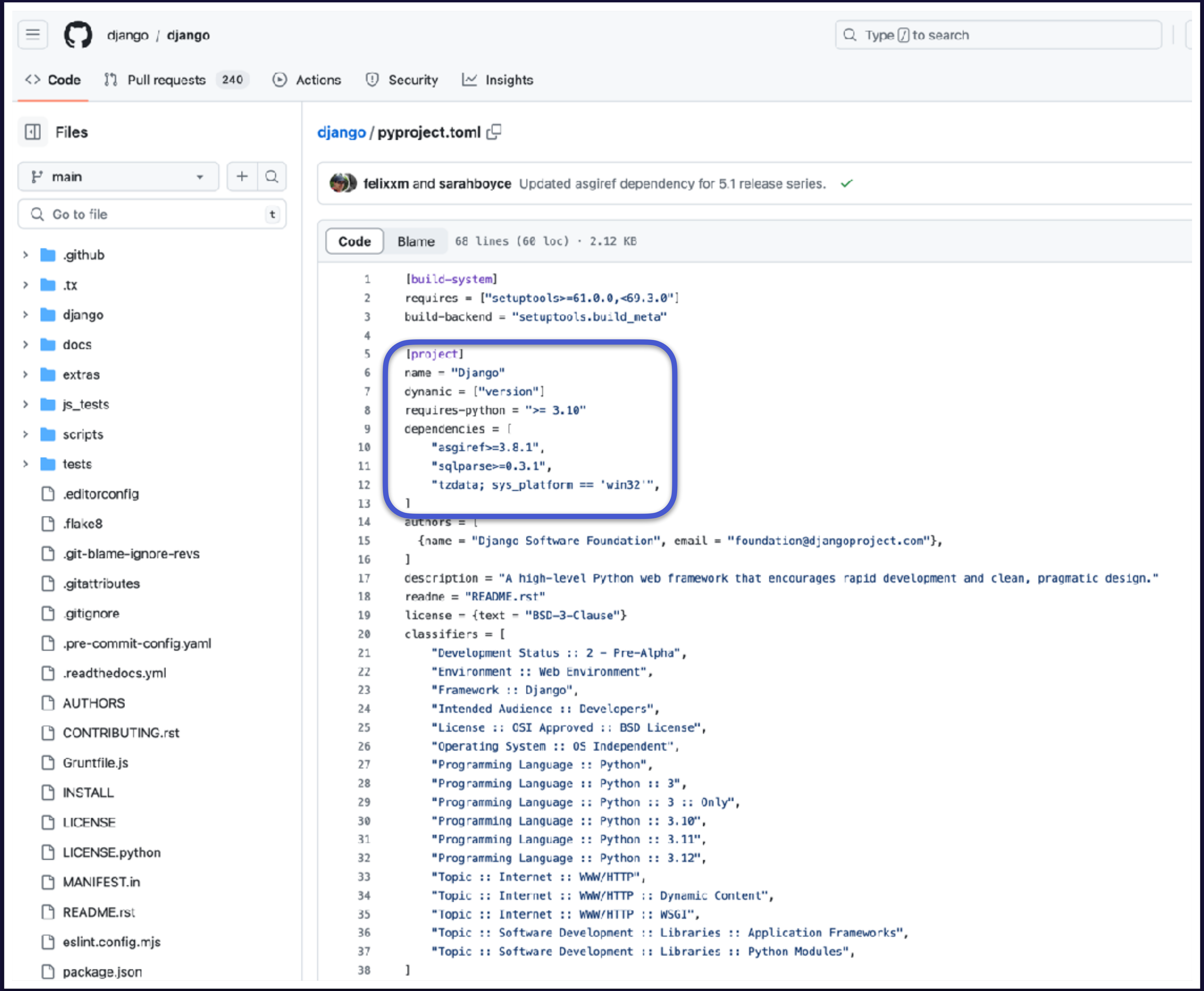
Types of SBOMs

1. Design
2. **Pre-build / Source**
3. **Build**
4. Post-build / Runtime
5. Operations
6. Discovery
7. Decomission

sbomify

Transitive vs. primary dependencies

sbomify



felixxm and sarahboyce Updated asgiref dependency for 5.1 release series. ✓

68 lines (60 loc) · 2.12 KB

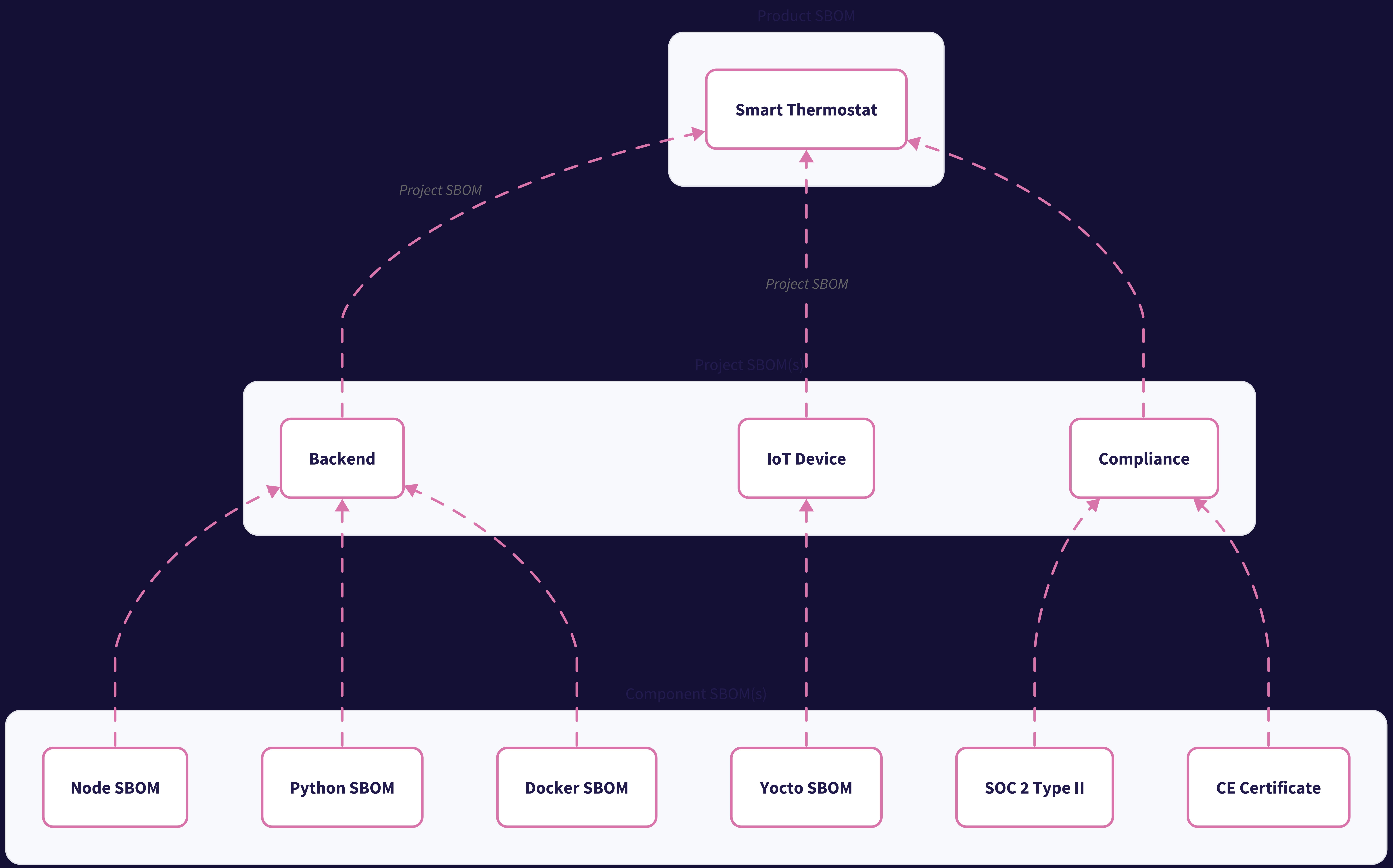
```
[build-system]
requires = ["setuptools>=61.0.0,<69.3.0"]
build-backend = "setuptools.build_meta"

[project]
name = "Django"
dynamic = ["version"]
requires-python = ">= 3.10"
dependencies = [
    "asgiref>=3.8.1",
    "sqlparse>=0.3.1",
    "tzdata; sys_platform == 'win32'",
]
authors = [
  {name = "Django Software Foundation", email = "foundation@djangoproject.com"},
]
description = "A high-level Python web framework that encourages rapid development and clean, pragmatic design."
readme = "README.rst"
license = {text = "BSD-3-Clause"}
classifiers = [
    "Development Status :: 2 - Pre-Alpha",
    "Environment :: Web Environment",
    "Framework :: Django",
    "Intended Audience :: Developers",
    "License :: OSI Approved :: BSD License",
    "Operating System :: OS Independent",
    "Programming Language :: Python",
    "Programming Language :: Python :: 3",
    "Programming Language :: Python :: 3 :: Only",
    "Programming Language :: Python :: 3.10",
    "Programming Language :: Python :: 3.11",
    "Programming Language :: Python :: 3.12",
    "Topic :: Internet :: WWW/HTTP",
    "Topic :: Internet :: WWW/HTTP :: Dynamic Content",
    "Topic :: Internet :: WWW/HTTP :: WSGI",
    "Topic :: Software Development :: Libraries :: Application Frameworks",
    "Topic :: Software Development :: Libraries :: Python Modules",
]
```

sbomify

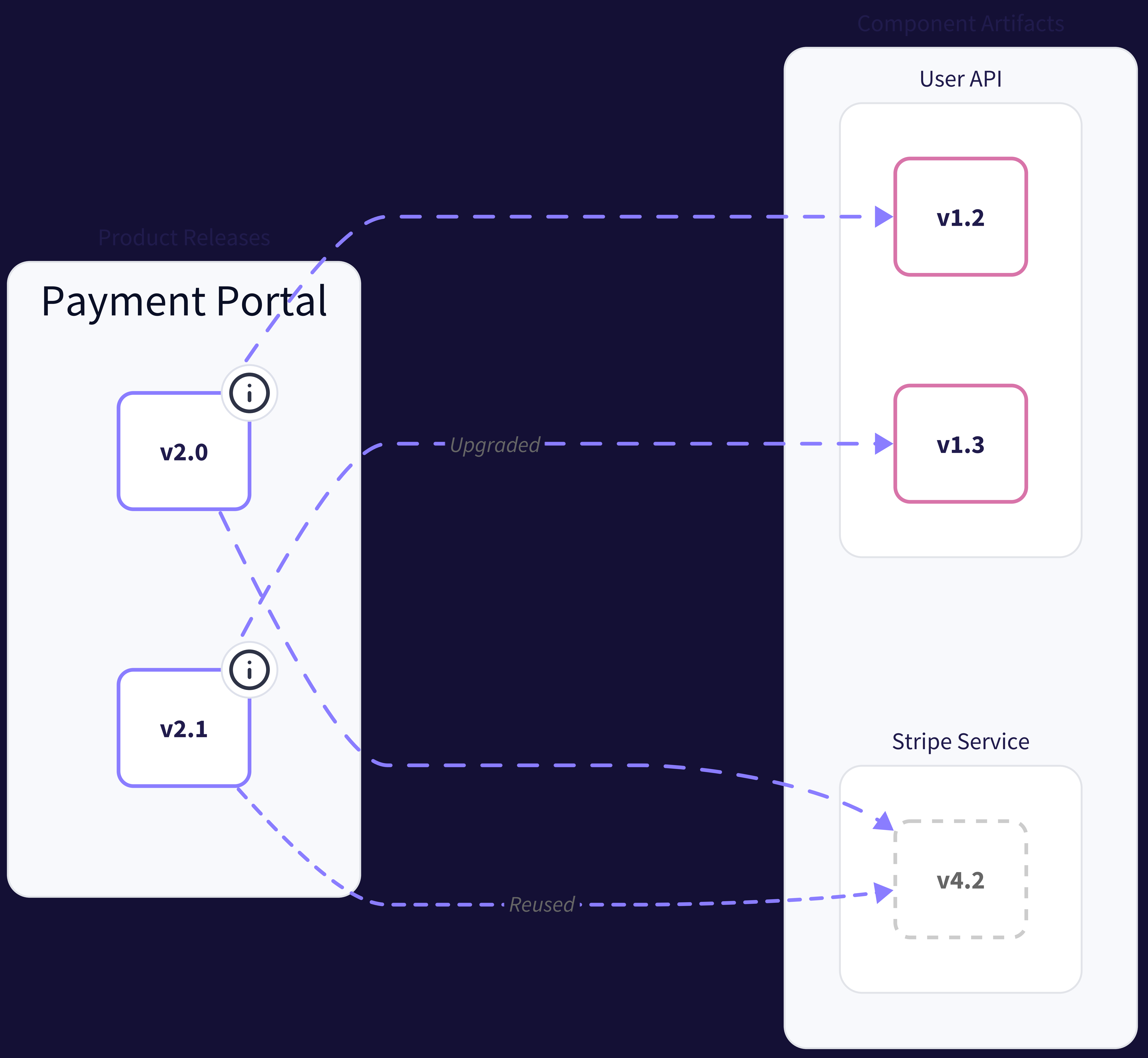
Generation: ☑️

What now?

sbomify


Product SBOM
Smart Thermostat
Project SBOM
Project SBOM
Project SBOM(s)
Backend
IoT Device
Compliance
Component SBOM(s)
Node SBOM
Python SBOM
Docker SBOM
Yocto SBOM
SOC 2 Type II
CE Certificate

sbomify


Component Artifacts
User API
v1.2
v1.3
Stripe Service
v4.2
Product Releases
Payment Portal
v2.0
v2.1
Upgraded
Reused

sbomify

Generation: ☑️

Automation: ☑️

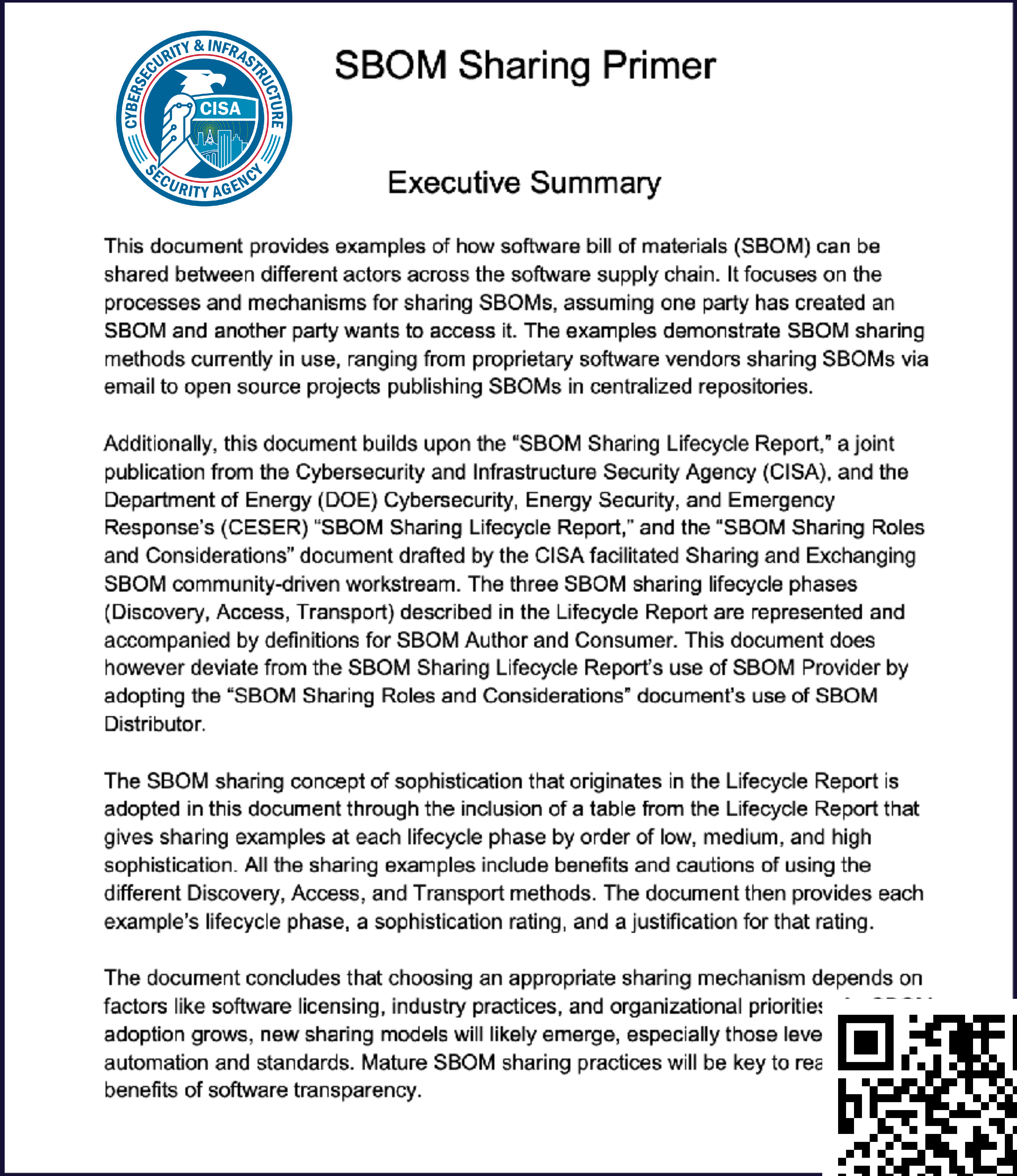
How do I share them?

Handling SBOMs today feels like managing source code in the 90s, with patches sent over email.

sbomify

Status

🥺



SBOM Sharing Primer

Executive Summary

This document provides examples of how software bill of materials (SBOM) can be shared between different actors across the software supply chain. It focuses on the processes and mechanisms for sharing SBOMs, assuming one party has created an SBOM and another party wants to access it. The examples demonstrate SBOM sharing methods currently in use, ranging from proprietary software vendors sharing SBOMs via email to open source projects publishing SBOMs in centralized repositories.

Additionally, this document builds upon the "SBOM Sharing Lifecycle Report," a joint publication from the Cybersecurity and Infrastructure Security Agency (CISA), and the Department of Energy (DOE) Cybersecurity, Energy Security, and Emergency Response's (CESER) "SBOM Sharing Lifecycle Report," and the "SBOM Sharing Roles and Considerations" document drafted by the CISA facilitated Sharing and Exchanging SBOM community-driven workstream. The three SBOM sharing lifecycle phases (Discovery, Access, Transport) described in the Lifecycle Report are represented and accompanied by definitions for SBOM Author and Consumer. This document does however deviate from the SBOM Sharing Lifecycle Report's use of SBOM Provider by adopting the "SBOM Sharing Roles and Considerations" document's use of SBOM Distributor.

The SBOM sharing concept of sophistication that originates in the Lifecycle Report is adopted in this document through the inclusion of a table from the Lifecycle Report that gives sharing examples at each lifecycle phase by order of low, medium, and high sophistication. All the sharing examples include benefits and cautions of using the different Discovery, Access, and Transport methods. The document then provides each example's lifecycle phase, a sophistication rating, and a justification for that rating.

The document concludes that choosing an appropriate sharing mechanism depends on factors like software licensing, industry practices, and organizational priorities
adoption grows, new sharing models will likely emerge, especially those leve
automation and standards. Mature SBOM sharing practices will be key to rea
benefits of software transparency.

sbomify

How to share SBOMs

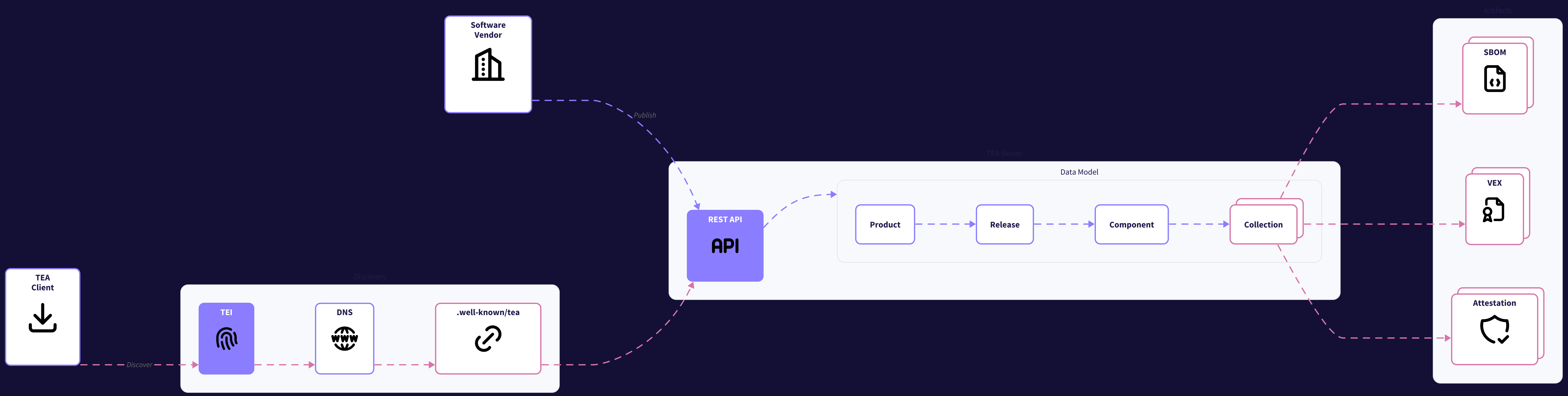
1. ~~Email~~
2. ~~Sharepoint~~
3. Your Trust Center
4. Transparency Exchange API (TEA)
5. GitHub Releases (Open Source)
6. Ecosystem specific methods (e.g. PEP770)

sbomify

TEA

OWASP TRANSPARENCY EXCHANGE API

sbomify


Software Vendor
Publish
TEA Client
Discover
Discovery
TEI
DNS
.well-known/tea
TEA Server
REST API
API
Data Model
Product
Release
Component
Collection
Artifacts
SBOM
VEX
Attestation

sbomify

Common Pitfalls

- SBOMs generated at someone computer, not CI
- SBOMs not being signed properly
- Merging multiple SBOMs into one

Recap

1. Why now: ✅
2. How to generate high quality SBOMs: ✅
3. SBOM life cycle management: ✅
4. How to share SBOMs at scale: ✅

And remember, CRA comes into effect **this** year!

sbomify

Thank you!

Questions?

Please grab me afterwards if you want to talk SBOMs.

Socials and contact:
301.vpetersson.com

sbomify