London DevOps

#85

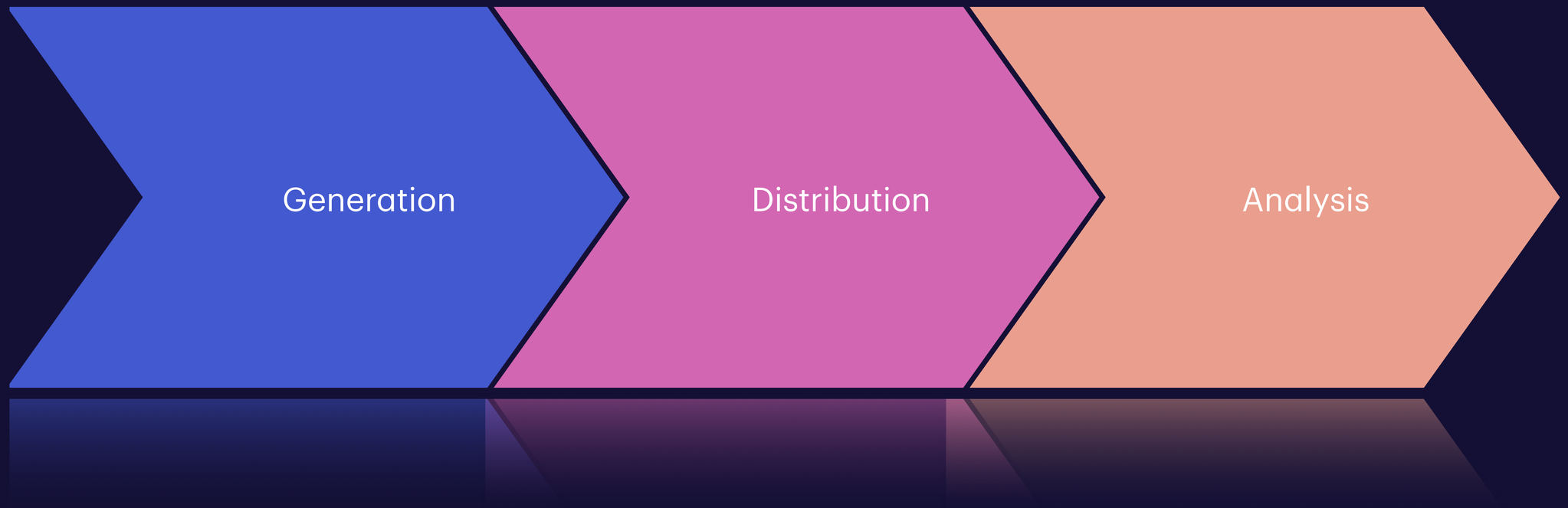
Viktor Petersson
vpetersson.com

$ whoami

What are SBOMs?

Why now?

What are SBOMs used for?



sbomify

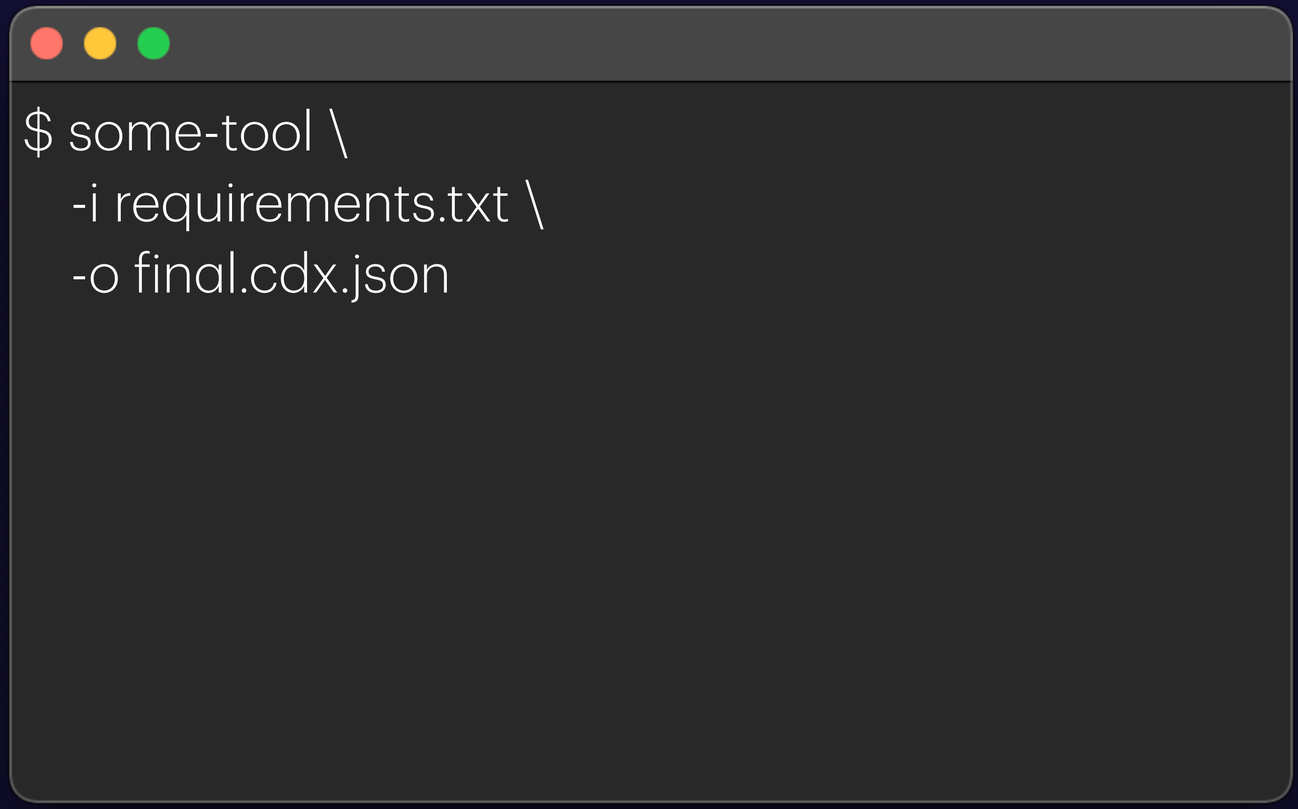
Generation

Expectation

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

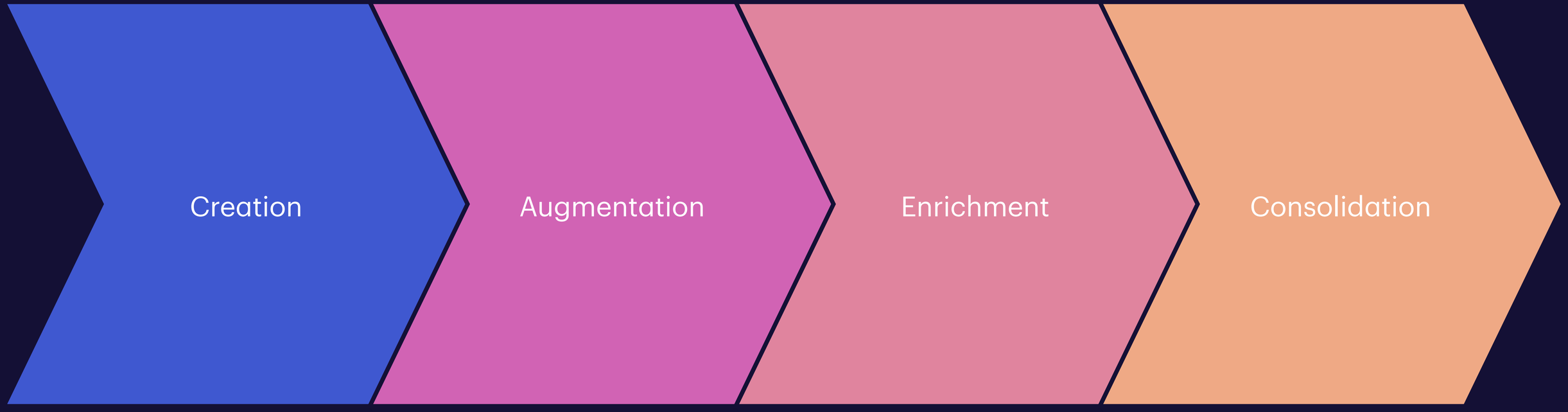
or

- Download SBOM from GitHub



sbomify

SBOM Generation Steps



sbomify

The format war

SPDX

from

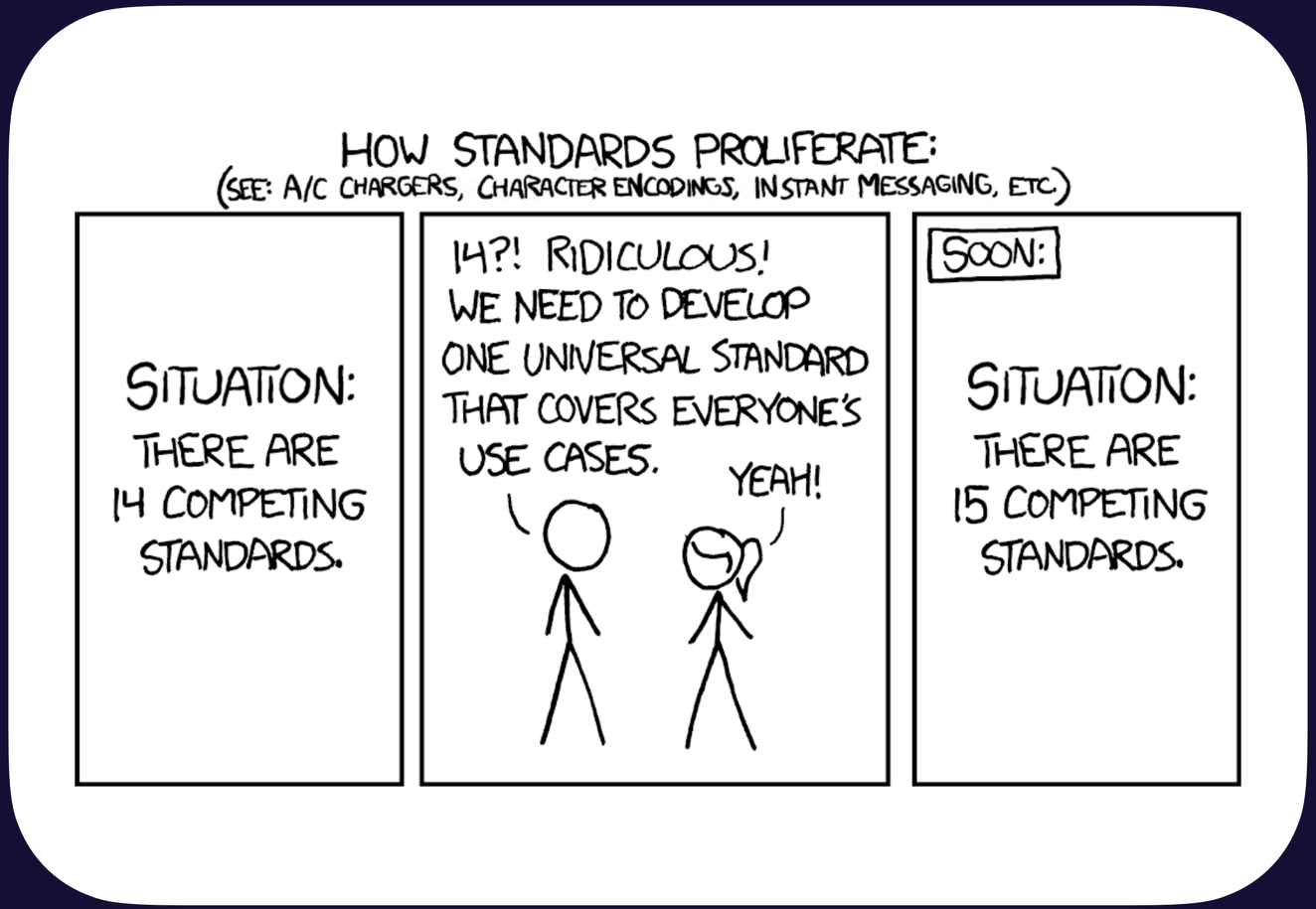
THE LINUX FOUNDATION

CycloneDX

from

OWASP

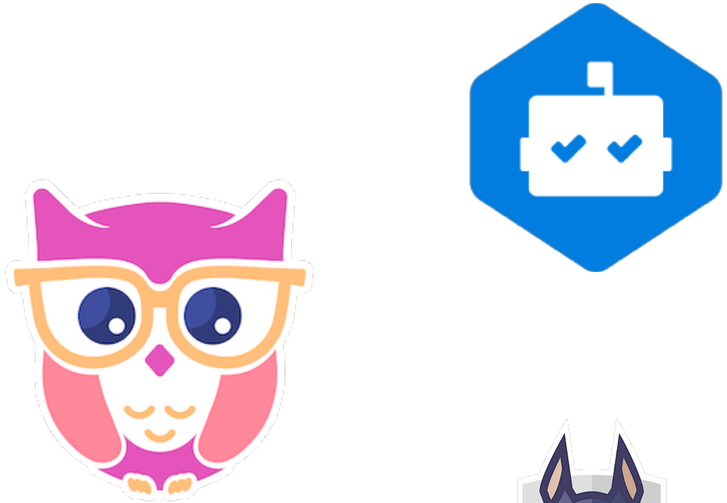
Open Web Application
Security Project


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.

sbomify

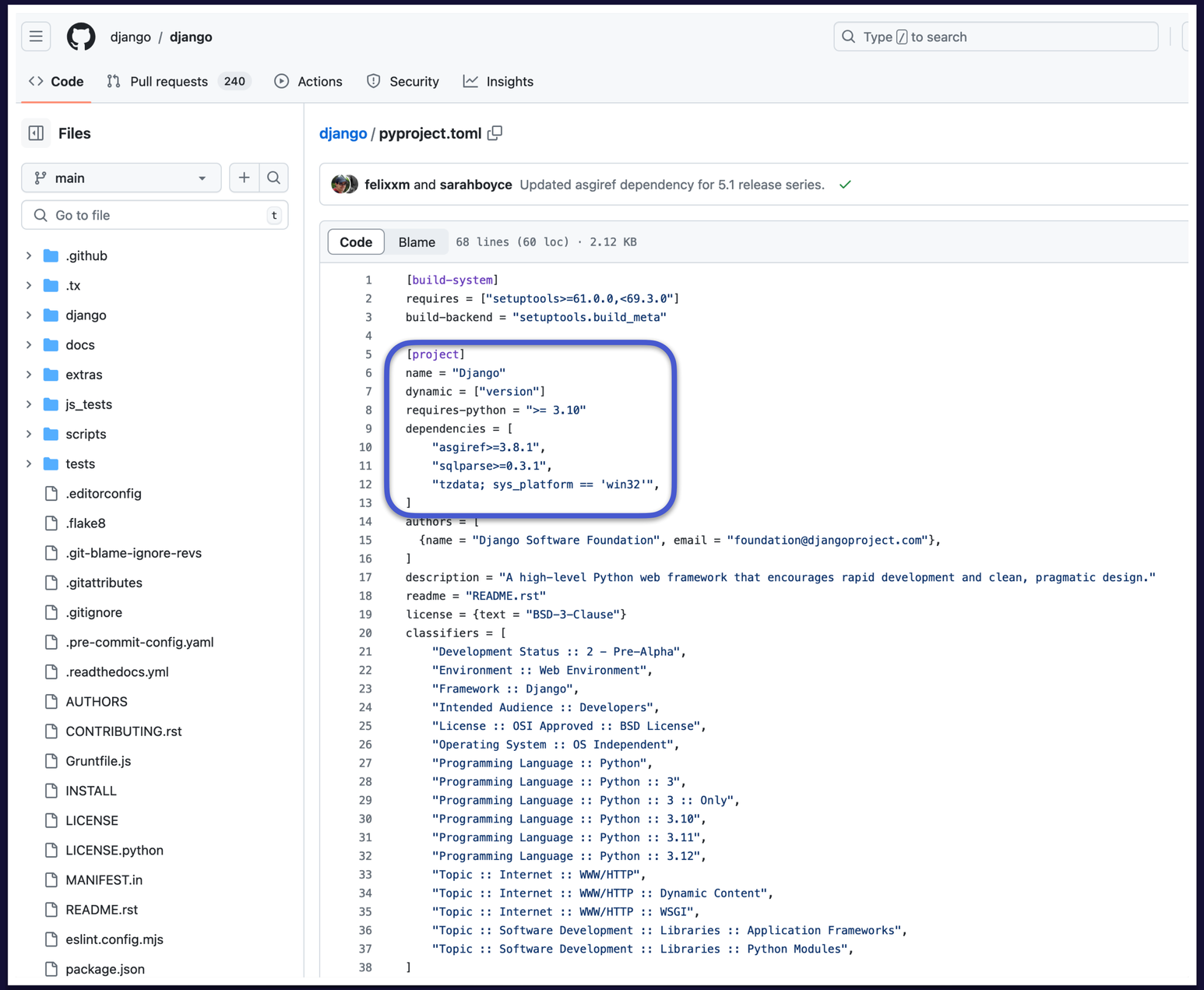
The tools

Generic



Domain specific tools

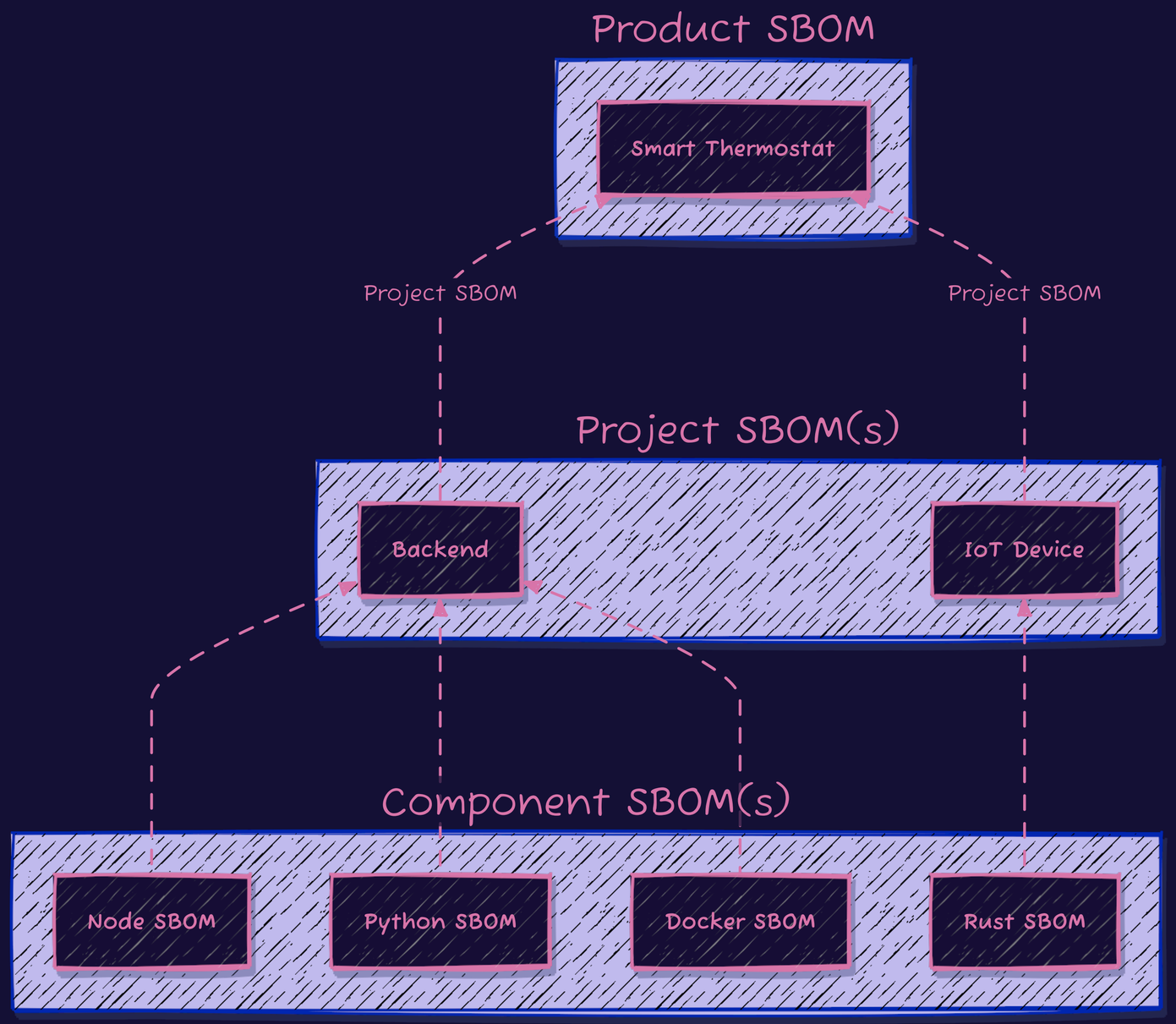
Transitive vs. primary dependencies


django / django
Type / to search
Code
Pull requests 240
Actions
Security
Insights
Files
main
Go to file t
.github
.tx
django
docs
extras
js_tests
scripts
tests
.editorconfig
.flake8
.git-blame-ignore-revs
.gitattributes
.gitignore
.pre-commit-config.yaml
.readthedocs.yml
AUTHORS
CONTRIBUTING.rst
Gruntfile.js
INSTALL
LICENSE
LICENSE.python
MANIFEST.in
README.rst
eslint.config.mjs
package.json
django / pyproject.toml
felixxm and sarahboyce Updated asgiref dependency for 5.1 release series.
Code
Blame
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

Source vs. Build SBOMs

But wait...there's more

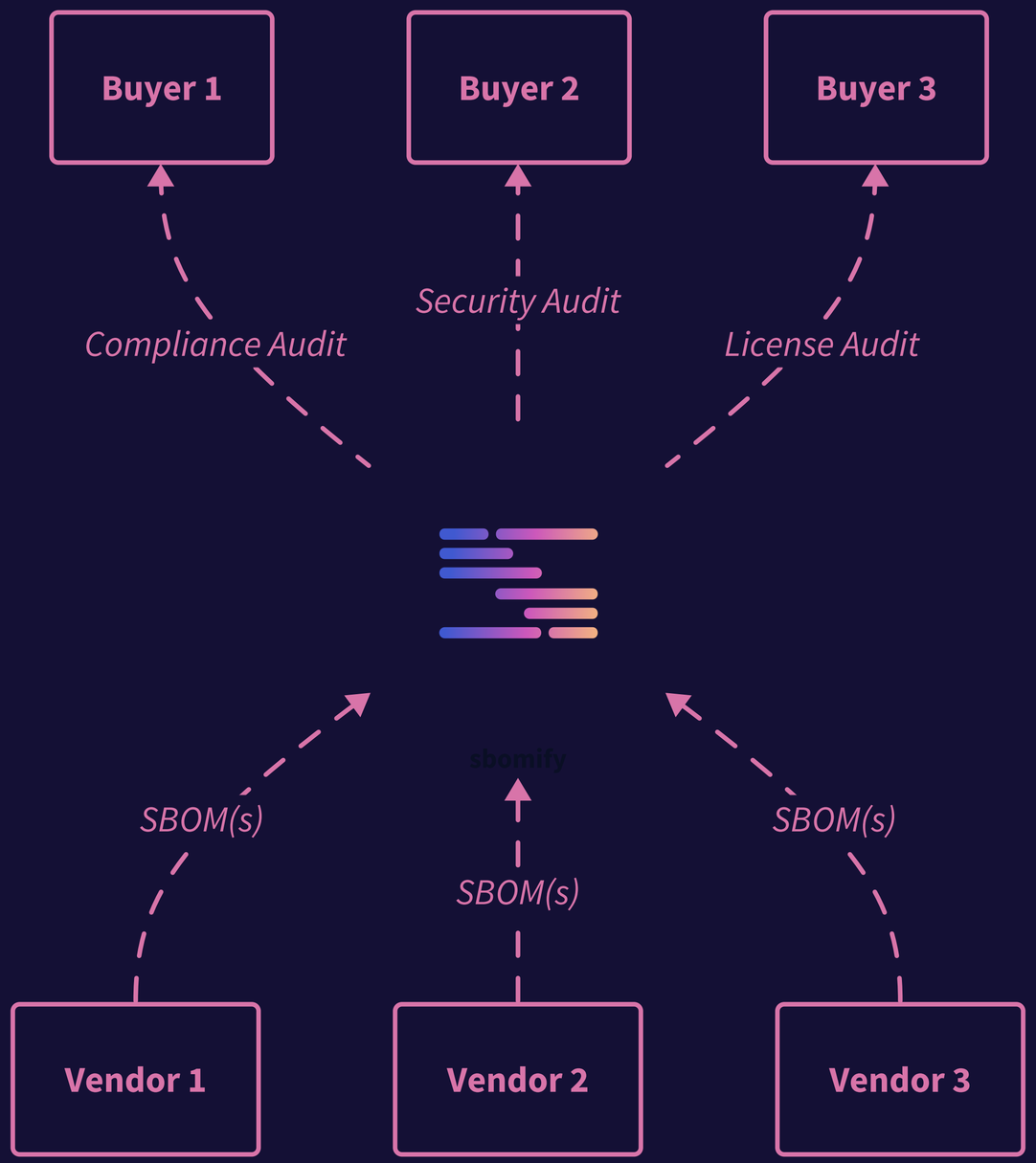

Product SBOM
Smart Thermostat
Project SBOM
Project SBOM
Project SBOM(s)
Backend
IoT Device
Component SBOM(s)
Node SBOM
Python SBOM
Docker SBOM
Rust SBOM

sbomify

But what about security?

Distribution

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

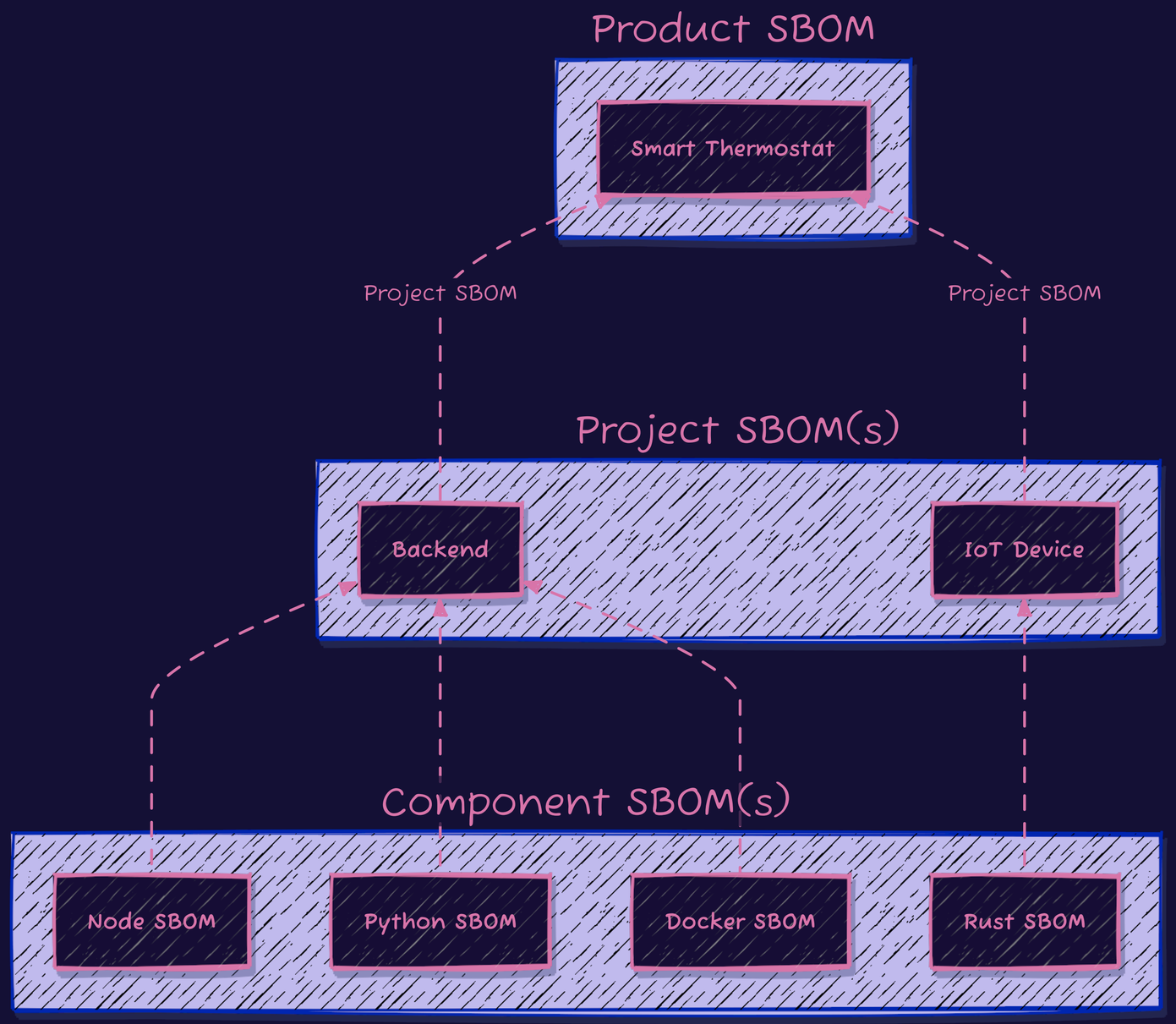

Buyer 1
Buyer 2
Buyer 3
Compliance Audit
Security Audit
License Audit
sbomify
SBOM(s)
SBOM(s)
SBOM(s)
Vendor 1
Vendor 2
Vendor 3

sbomify

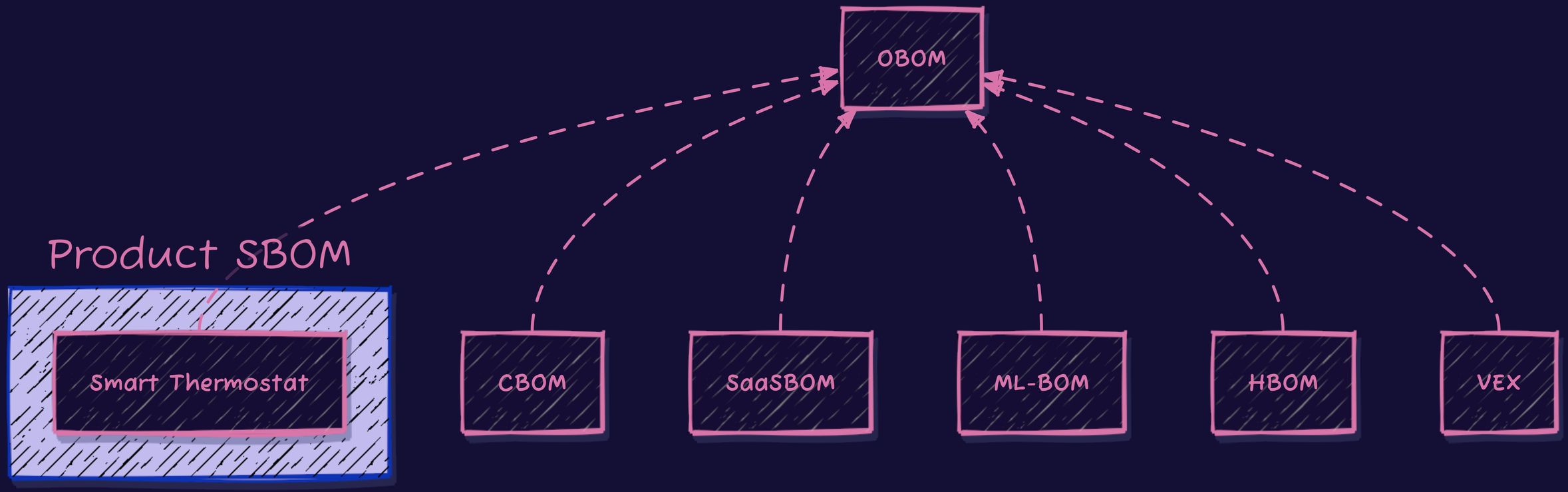
Analysis

dependency track

Big Picture


Product SBOM
Smart Thermostat
Project SBOM
Project SBOM
Project SBOM(s)
Backend
IoT Device
Component SBOM(s)
Node SBOM
Python SBOM
Docker SBOM
Rust SBOM

sbomify


OBOM
Product SBOM
Smart Thermostat
CBOM
SaaSBOM
ML-BOM
HBOM
VEX

sbomify

State of SBOMs

Q & A


sbomify

More reading

- NTIA Minimum Elements
- Framing Software Component Transparency: Establishing a Common Software Bill of Materials (SBOM) (2nd edition)
  - 3nd edition is released shortly
- SBOM Resources
- CISA Working Group: SBOM Generation
- Shameless self plug: sbomify
- Slides will be available on vpetersson.com/about

Scan for deep dive!

sbomify