More than a “SBOM button” for compliance: SBOM quality matters!

AboutCode

Agenda

- About me, AboutCode and nexB
- What is a SBOM (Software Bill of Materials)?
  - Minimum requirements, widely used standards, applicable regulations
- SBOM quality: things to look out for
  - Support for package manifests, ecosystems
  - When is SBOM generated: source, build, analyzed, deployed, runtime
  - Hidden items: binaries, vendored/copied code, AI generated code
  - The FOSS community approach matters
  - Misc: Automation, Open Data, Benchmarks, other BOMs
- Questions?

About Ayan

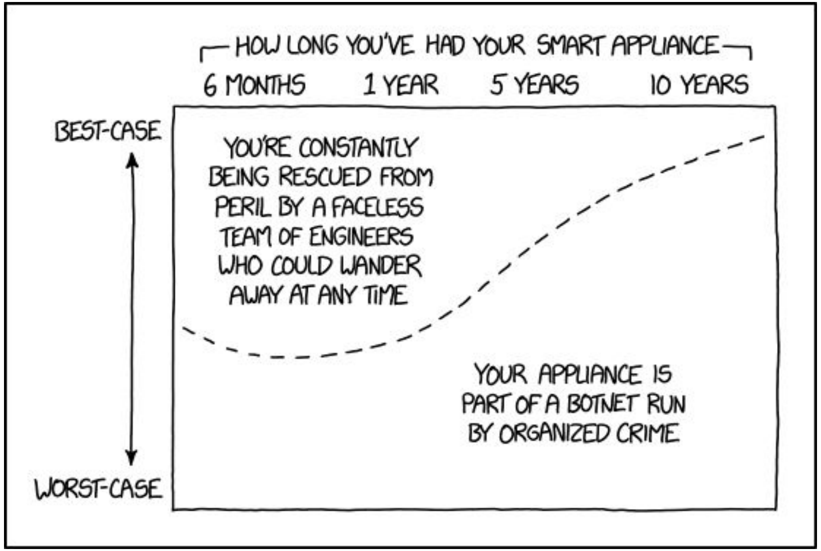
- Core maintainer of ScanCode (scancode-toolkit and scancode.io)
  - also contributes to and helps maintain other AboutCode tools: license-expression licenseDB scancode-workbench PURLdb
- Working on license detection, package identification, binary scanning, SBOMs and data summarization
- Google Summer of Code Mentor at AboutCode (2021-2025)
  - participant in GSoC2020 and GSoD2019 (Season of Docs)
- Links:
  - asmahapatra@aboutcode.org
  - GitHub: https://github.com/AyanSinhaMahapatra/
  - LinkedIn: https://www.linkedin.com/in/ayansinhaju/

AboutCode and nexB

- AboutCode's FOSS-first mission: FOSS for FOSS
    - Open source tools and open knowledge base (AboutCode stack)
    - Simple and practical standards (Package-URL)
    - Applications for Legal/Security Business users (DejaCode, also FOSS)
- Trusted experts on Software Composition Analysis (SCA) since 2007
        - Creator of Package-URL: https://github.com/package-url
        - Co-founders of SPDX: https://spdx.org
        - Contributors to CycloneDX: https://cyclonedx.org
        - Co-founders of ClearlyDefined: https://clearlydefined.io
- nexB: professional services for SCA
    - 800+ SCA projects completed to-date
    - Sponsored development for AboutCode projects
    - Technical support and advisory for SCA process, and deployments

Software Bill Of Materials: why?

- We need to:
  - know what is in our software
  - reuse and consume FOSS safely
- SBOM is a build audit, not a parts list
- SPDX and CycloneDx, packageURL
- minimum: a list of packageURLs
- Regulations:
  - CRA in EU, US executive order
  - SEBI CSCRF
  - CERT-In SBOM Guidelines


HOW LONG YOU'VE HAD YOUR SMART APPLIANCE
6 MONTHS
1 YEAR
5 YEARS
10 YEARS
BEST-CASE
WORST-CASE
YOU'RE CONSTANTLY BEING RESCUED FROM PERIL BY A FACELESS TEAM OF ENGINEERS WHO COULD WANDER AWAY AT ANY TIME
YOUR APPLIANCE IS PART OF A BOTNET RUN BY ORGANIZED CRIME

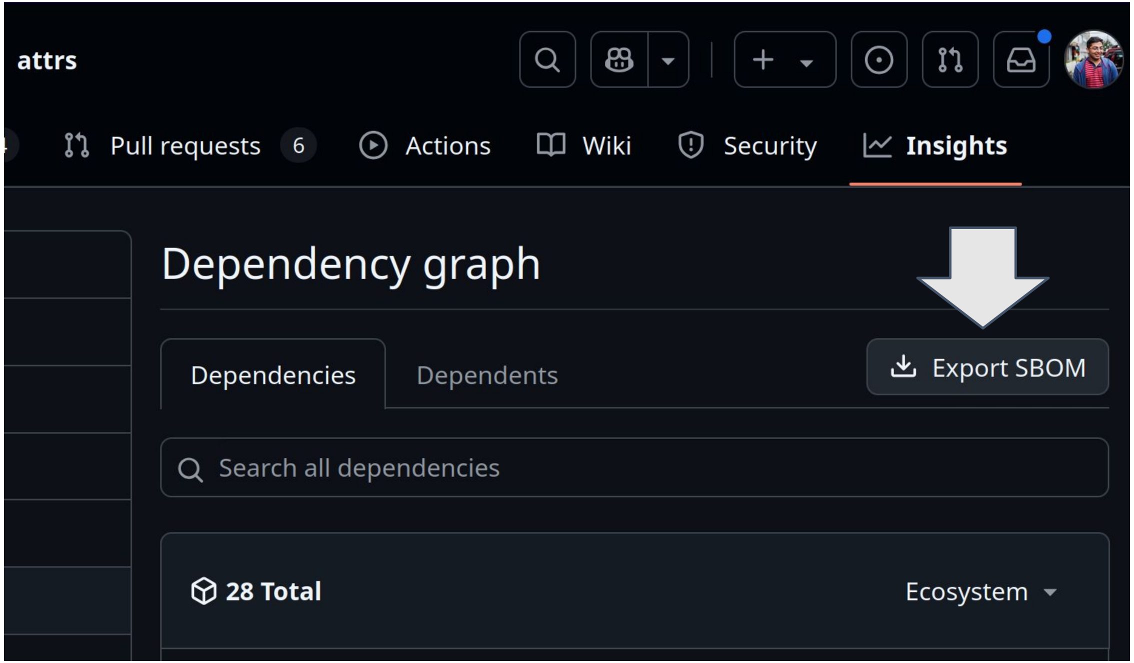
https://xkcd.com/1966/ : Smart Home Appliances

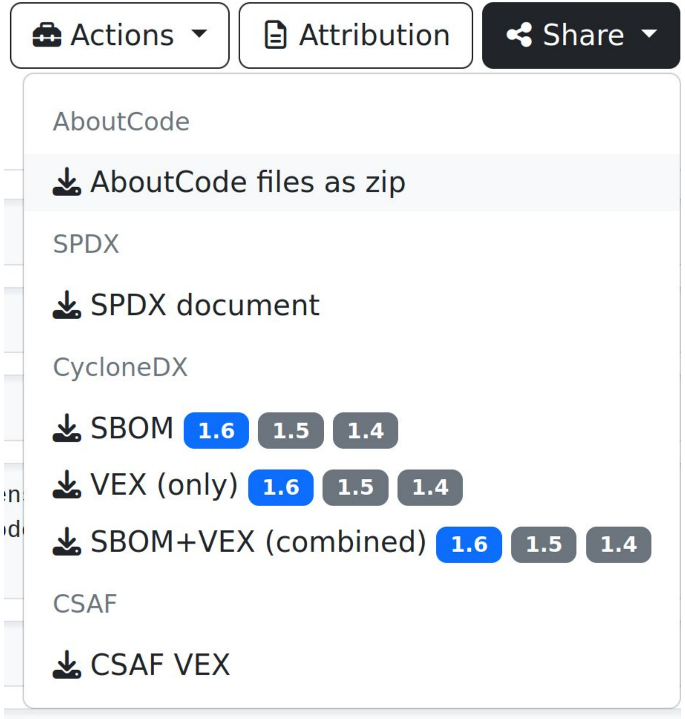
PackageURL

Started in ScanCode to uniquely identify packages.

- pkg:type/namespace/name@version?qualifiers#subpath
  - Specification: https://github.com/package-url/purl-spec
- PURL examples:
  - pkg:deb/debian/curl@7.50.3-1?arch=i386&distro=jessie
  - pkg:github/aboutcode-org/scancode-toolkit@32.4.1
  - pkg:pypi/django@1.11.1
  - pkg:rpm/fedora/curl@7.50.3-1.fc25
  - pkg:golang/google.golang.org/genproto#googleapis/api/annotations
- Vers: https://github.com/aboutcode-org/univers/

Just click the SBOM button!





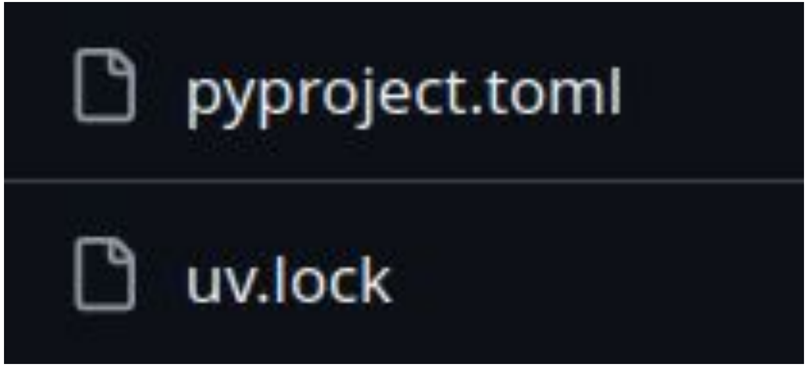
AboutCode

0 packages = 0 vulnerabilities

- always a new kid in town!
  - pypi.org/project/poetry: 2018
  - pypi.org/project/uv: 2024

- package ecosystems: 32
- types of package manifests: ~135

- SPDX licenses: 779
- scancode-licensedb: 2579

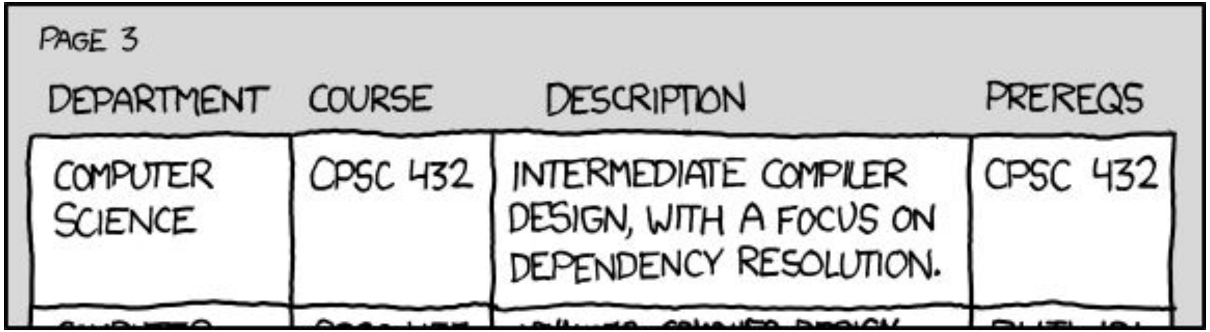

pyproject.toml
uv.lock

github.com/package-url/purl-spec/blob/main/PURL-TYPES.rst
scancode-toolkit.rtfd.io/reference/available_package_parsers.html
spdx.org/licenses/
scancode-licensedb.aboutcode.org/

May the source be with you!

- github-actions have (vulnerable?) dependencies ?!
- not everything is deployed
- 1 repo -> 20 packages (with mono repos)
- dependencies? unresolved
- last commit? 10 years ago
  - end-of-life.date
  - OpenSSF Scorecard

Source SBOM!



PAGE 3

DEPARTMENT	COURSE	DESCRIPTION	PREREQS
COMPUTER SCIENCE	CPSC 432	INTERMEDIATE COMPILER DESIGN, WITH A FOCUS ON DEPENDENCY RESOLUTION.	CPSC 432

https://xkcd.com/754/ : Dependencies

🏗 building ⟶ deployed binary 🚀

- binary: bare minimum to run code
- massive # of packages in containers/apps
- build system has access to origin and results
- trust but verify: back2source
- reproducible builds
- immutable releases
- CI: [aboutcode-org/scancode-action]

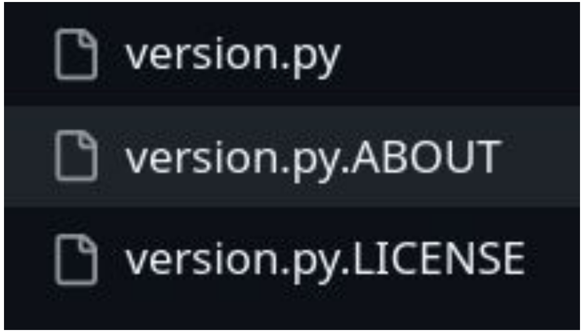
Build SBOM &
Deployed SBOM

Not everything is declared in manifests

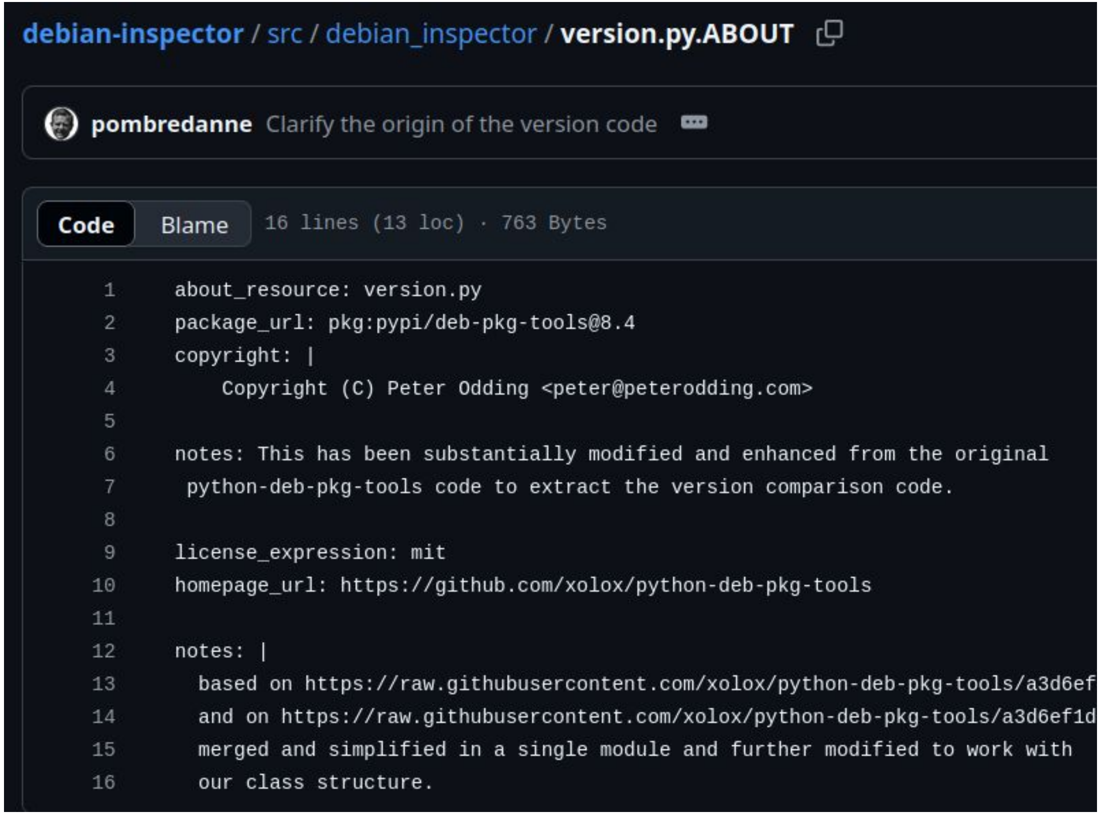
- comes in all sizes: binaries, files, snippets
- Convenience: lets just include everything!
- Can we get to the source?
  - which part of the source is deployed
- Is this modified?
  - Index and match by checksums
- scan once, then always match
  - by archives
  - by directories
  - by files
  - snippets

Copying code is okay! But...

- Declare
- Update periodically
- release?
- otherwise have to match



version.py
version.py.ABOUT
version.py.LICENSE



debian-inspector / src / debian_inspector / **version.py.ABOUT**

pombredanne Clarify the origin of the version code

```
about_resource: version.py
package_url: pkg:pypi/deb-pkg-tools@8.4
copyright: |
    Copyright (C) Peter Odding <peter@peterodding.com>

notes: This has been substantially modified and enhanced from the original
 python-deb-pkg-tools code to extract the version comparison code.

license_expression: mit
homepage_url: https://github.com/xolox/python-deb-pkg-tools

notes: |
  based on https://raw.githubusercontent.com/xolox/python-deb-pkg-tools/a3d6ef
  and on https://raw.githubusercontent.com/xolox/python-deb-pkg-tools/a3d6ef1d
  merged and simplified in a single module and further modified to work with
  our class structure.
```

AI generated code? 🤖

- Generated code can be very similar to FOSS code
  - Finding public code that matches GitHub Copilot suggestions
- Approximate snippet matching algorithm
  - working prototype on popular npm code
  - need to scale!
- LLMs trained on permissively licensed code!
  - https://huggingface.co/blog/starcoder2
  - uses scancode to detect licenses

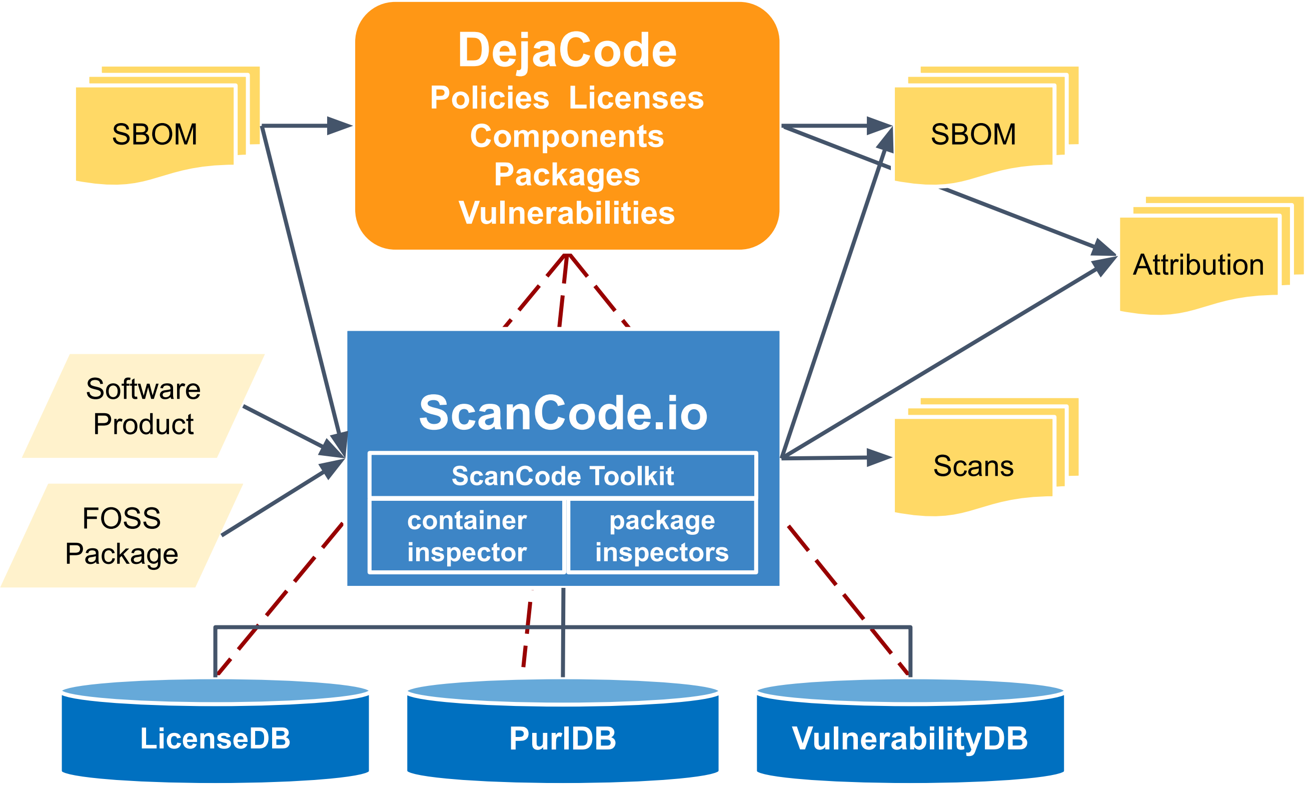
Fix it at the source!

- reducing complexity >> modeling complexity
- include data on origin/dependencies:
  - cargo-auditable : auditable production Rust binaries
  - PEP 725 : Specifying external dependencies
- license fixes:
  - SPDX License identifiers in Linux kernel
  - PEP 639 : License Clarity with Better Package Metadata
- ecosystem wide scans:
  - fix licensing issues
  - detect undeclared/vendored code or binaries
  - open data

What AboutCode is doing differently

- Non-profit, fully open source, open data, public instances
- coming up soon: AboutCode Foundation!
- options: CLI tool, Github action, web app, scans: containers, source/binary
- supports and working with package ecosystems
  - to build better metadata, more transparency
  - solve ecosystem wide problems at once
- Open data (soon: federated data)
  - Curated, open data on Licensing, Vulnerabilities, Packages
- Large community
  - working with FOSS orgs to improve standards, data and transparency
  - OSPOs, Security, Lawyers, Specifications, Developers

The AboutCode stack:



AboutCode: Who is using it?

(based on public data)

Most FOSS Orgs, many commercial and open source SCA providers use our libraries or standards

- Most FOSS Foundations
- Five of the top big tech companies
- A leading database company, a leading Linux company
- 2 leading code hosting companies
- European and US government agencies
- All major European car manufacturers and most of their vendors
- Major US chip and microprocessor providers
- All SBOM and VEX standards

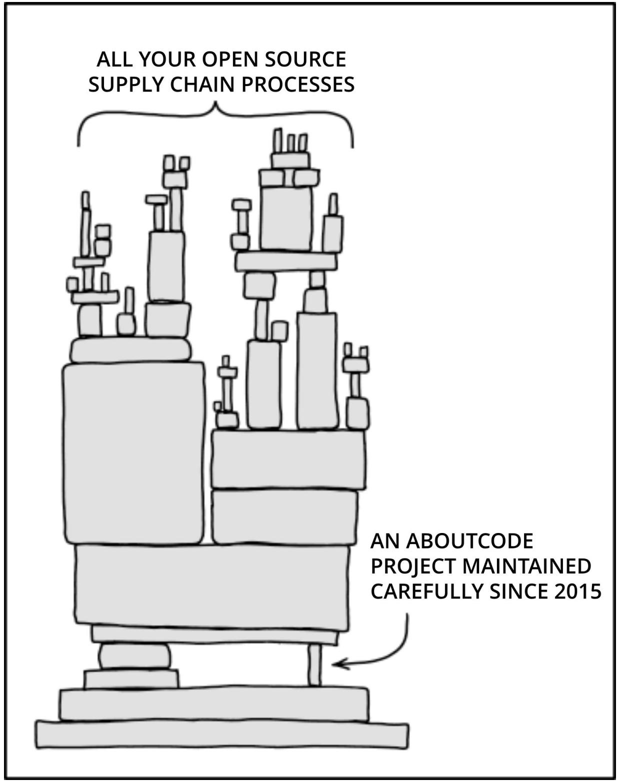
Other FOSS SCA tools and projects

- ORT: OSS Review Toolkit (Uses ScanCode)
- FOSSology (Uses ScanCode)
- SW360
- TERN (Uses ScanCode)
- ClearlyDefined (Uses scancode)
- OSSelot
- OWASP DependencyTrack
- OWASP DepScan
- AppThreat projects: atom, chen, vdb
- CycloneDx cdxgen
- Anchore: syft, grype
- Aquasec trivy

AboutCode also needs your help!

AboutCode

- Contribute to an AboutCode project with code, documentation, use cases, bug reports
  - https://github.com/aboutcode-org
- Sponsor AboutCode project maintainers, accelerate development of new features https://github.com/sponsors/aboutcode-org
- Buy support, implementation, and advisory services from nexB to pay the maintainers
- Join the community:
  - https://www.aboutcode.org/
  - https://matrix.to/aboutcode-org_discuss


ALL YOUR OPEN SOURCE SUPPLY CHAIN PROCESSES
AN ABOUTCODE PROJECT MAINTAINED CAREFULLY SINCE 2015

"Dependency" by xkcd, Modified text from original

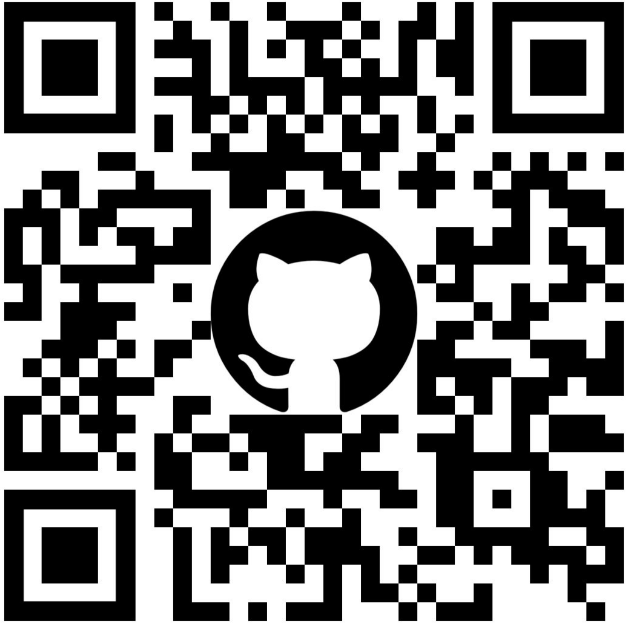
Questions?

AboutCode

Another talk (with live demo):
From Policy to Pipeline: How OSPOs Can Power Regulatory Readiness and Upstream Impact
with Arun Azhakesan
OSPOCon, room: G.01 + G.02

AboutCode

Link to slides



github/aboutcode-org

Credits

Special thanks to all the people who made and released these excellent free resources:

- ▷ All the open source software authors that make AboutCode possible
- ▷ Emojis are from https://openmoji.org/ under cc-by-sa-4.0
- ▷ xkcd comics under cc-by-nc-2.5
- ▷ Presentation template by SlidesCarnival