NORDIC SOFTWARE SECURITY SUMMIT

NSSS.SE

THE SOFTWARE SECURITY REGULATION REVOLUTION

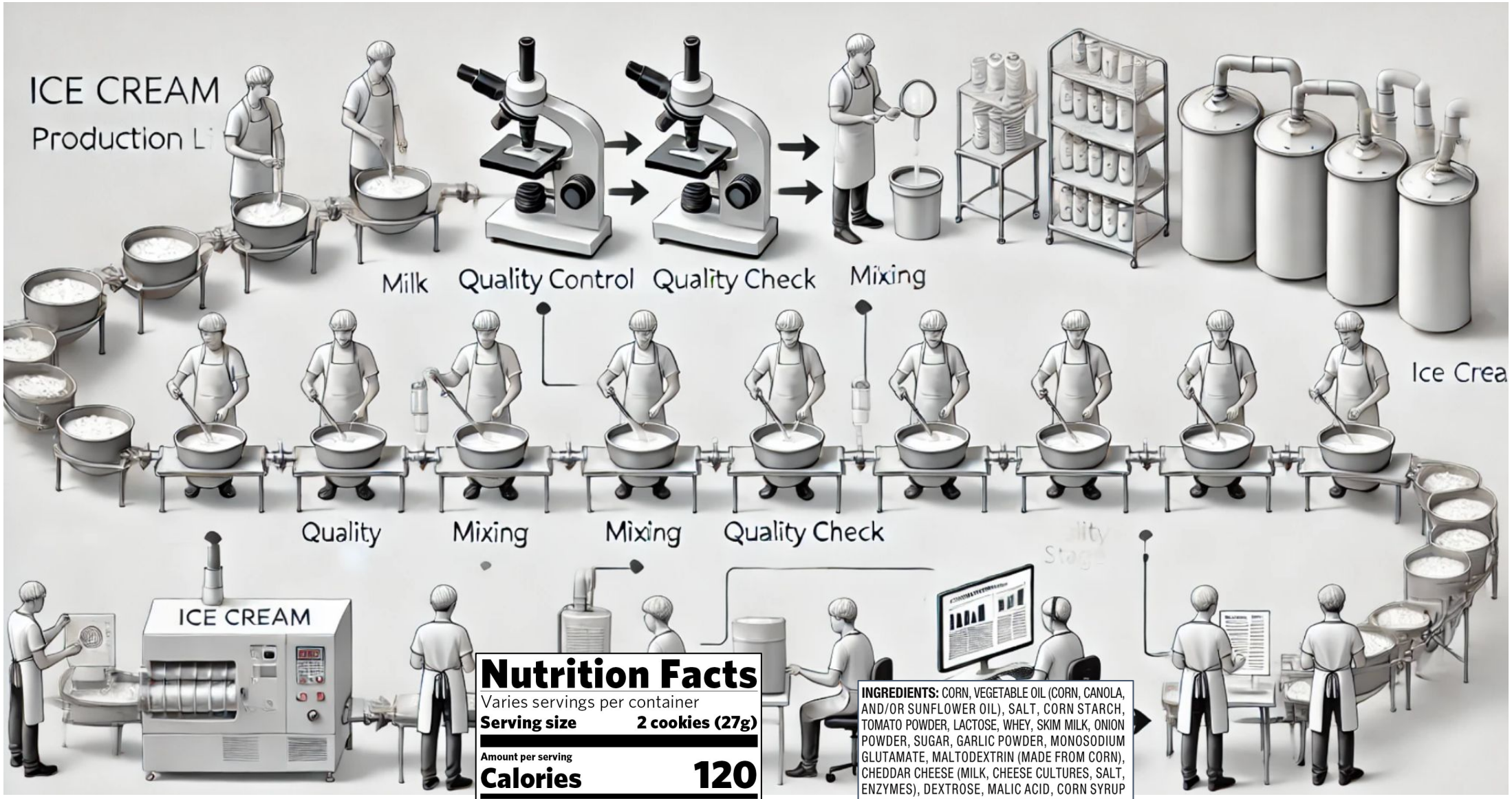
SBOMs that you can trust

the good, the bad, and the ugly

Daniel Liszka & Miguel Martinez, Chainloop

Tuesday September 24, 2024


ICE CREAM
Production L
Milk
Quality Control
Quality Check
Mixing
Ice Crea
Quality
Mixing
Mixing
Quality Check
ICE CREAM



Nutrition Facts

Varies servings per container

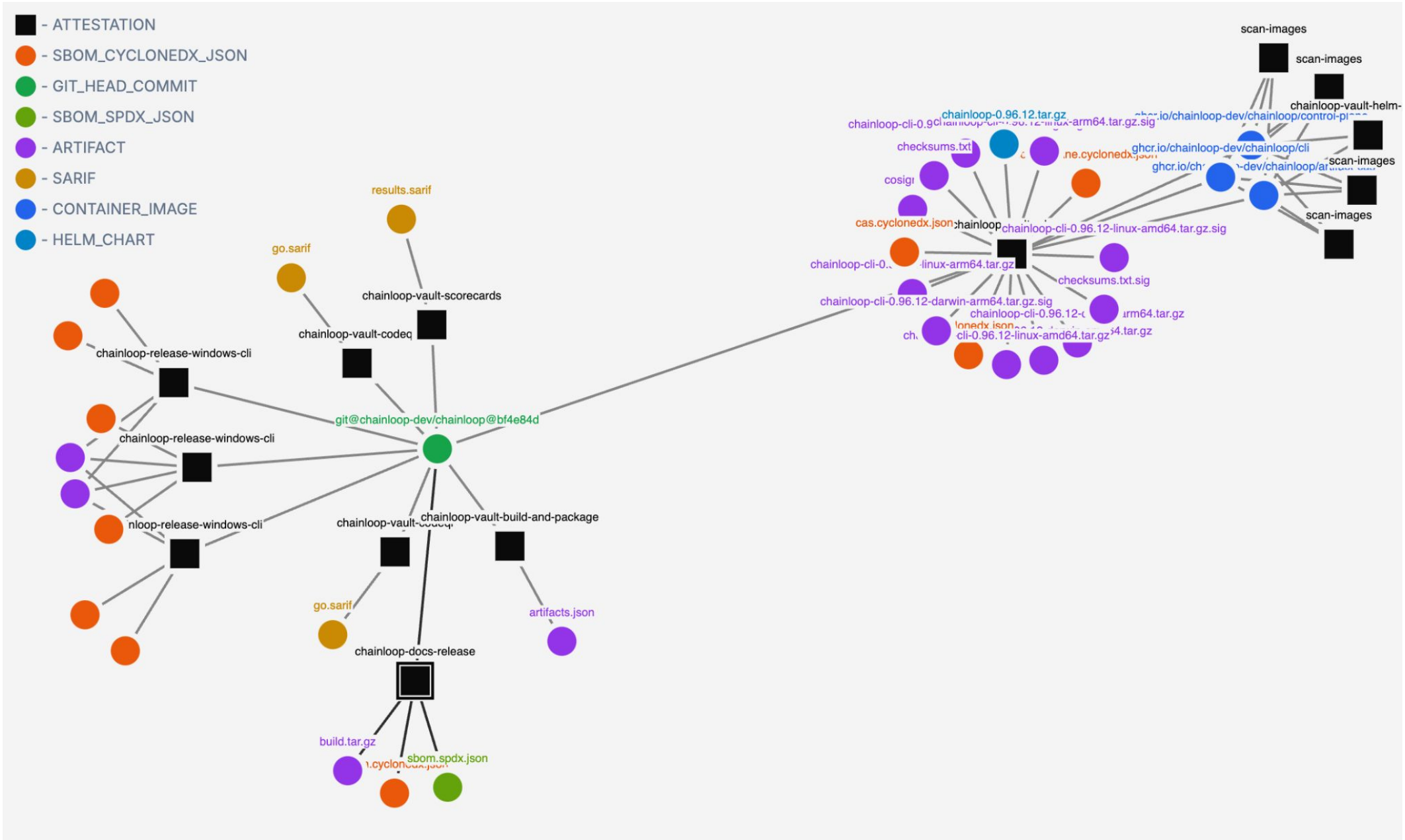
Serving size **2 cookies (27g)**

Amount per serving

Calories **120**

	% Daily Value*
Total Fat 7g	**9%**
Saturated Fat 4.5g	**23%**
Trans Fat 0g	
Cholesterol 20mg	**7%**
Sodium 75mg	**3%**

INGREDIENTS: CORN, VEGETABLE OIL (CORN, CANOLA, AND/OR SUNFLOWER OIL), SALT, CORN STARCH, TOMATO POWDER, LACTOSE, WHEY, SKIM MILK, ONION POWDER, SUGAR, GARLIC POWDER, MONOSODIUM GLUTAMATE, MALTODEXTRIN (MADE FROM CORN), CHEDDAR CHEESE (MILK, CHEESE CULTURES, SALT, ENZYMES), DEXTROSE, MALIC ACID, CORN SYRUP SOLIDS, BUTTERMILK, NATURAL AND ARTIFICIAL FLAVORS, SODIUM ACETATE, ARTIFICIAL COLOR (RED 40, BLUE 1, YELLOW 5), SPICE, CITRIC ACID, DISODIUM INOSINATE, AND DISODIUM GUANYLATE.
CONTAINS MILK INGREDIENTS.


- ATTESTATION
- SBOM_CYCLONEDX_JSON
- GIT_HEAD_COMMIT
- SBOM_SPDX_JSON
- ARTIFACT
- SARIF
- CONTAINER_IMAGE
- HELM_CHART
scan-images
scan-images
chainloop-vault-helm-
scan-images
scan-images
chainloop-0.96.12.tar.gz
checksums.txt
cosig
cas.cyclonedx.json
chainloop
chainloop-cli-0.96.12-linux-amd64.tar.gz.sig
chainloop-cli-0.96.12-darwin-arm64.tar.gz.sig
checksums.txt.sig
ghcr.io/chainloop-dev/chainloop/cli
results.sarif
go.sarif
chainloop-vault-scorecards
chainloop-vault-codeq
chainloop-release-windows-cli
chainloop-release-windows-cli
git@chainloop-dev/chainloop@bf4e84d
chainloop-vault-build-and-package
go.sarif
artifacts.json
chainloop-docs-release
build.tar.gz
sbom.spdx.json

Bitnami Application Catalog

Find your favorite application in our catalog and launch it.
Learn more about the benefits of the Bitnami Application Catalog.

Home › Applications

These software listings are packaged by Bitnami. The respective trademarks mentioned in the offerings are owned by the respective companies, and use of them does not imply any affiliation or endorsement. The software is licensed to you subject to one or more open source licenses and VMware provides the software on an AS-IS basis.

If you are looking to use our images in production environments, VMware recommends using VMware Tanzu Application Catalog, the commercial edition of the Bitnami catalog.

Search applications

chainloop

Find your solution

Single-Tier | Multi-Tier | Docker | Kubernetes | Virtual Machines

Chainloop

Application Security

Chainloop Artifact CAS

Application Security

Chainloop Control Plane migrations

Application Security

Rich library of 250+ trusted building blocks

bitnami

Continuous Monitoring of upstream source code changes triggers rebuild, test and update

Language Runtimes





App Components






Supporting Apps







bitnami

1M+

Deployments Per Month

Every

Major Cloud

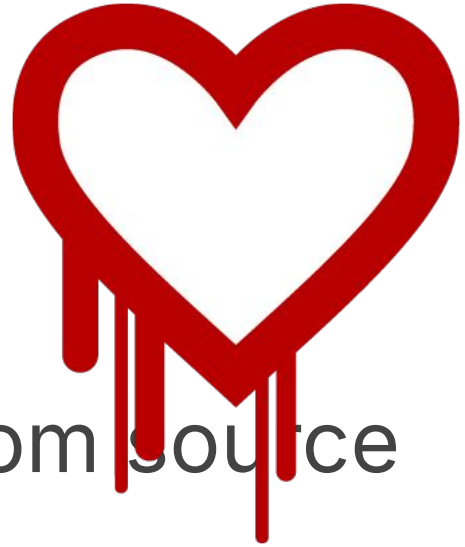
260+

Open Source Applications

Any

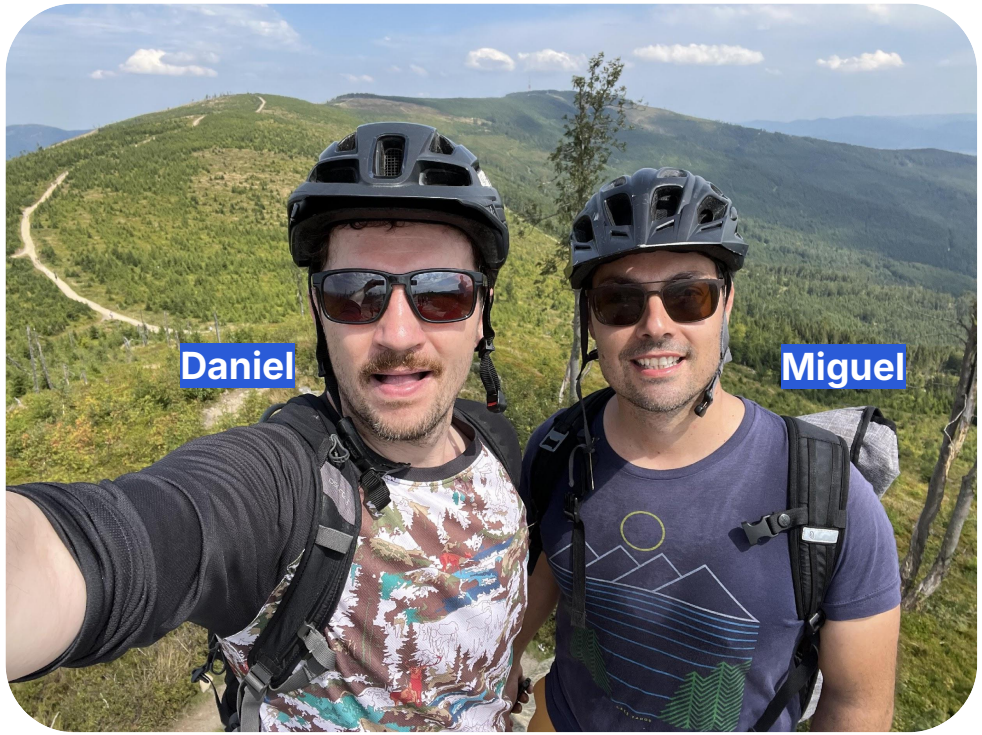
Deployment Format

(2014) OpenSSL Heartbleed



- ➔ Hundreds of **Bitnami** Apps updated from source and delivered to major cloud marketplaces
- ➔ **In less than 10 hours**
- ➔ Everything tested: unit, functional, security
- ➔ Across many different platforms
- ➔ Open Source License Compliance

Hi 👋
We Breathe Automation



Miguel Martinez

co-founder at Chainloop. 10+ years designing, implementing and operating Software Supply Chain automation at Bitnami/VMware. **The IT-crowd fan**

Daniel Liszka

co-founder and Chainloop maintainer, previously Engineering at Bitnami and Product at VMware. Dad, previously traveller, biker, and skier ;)

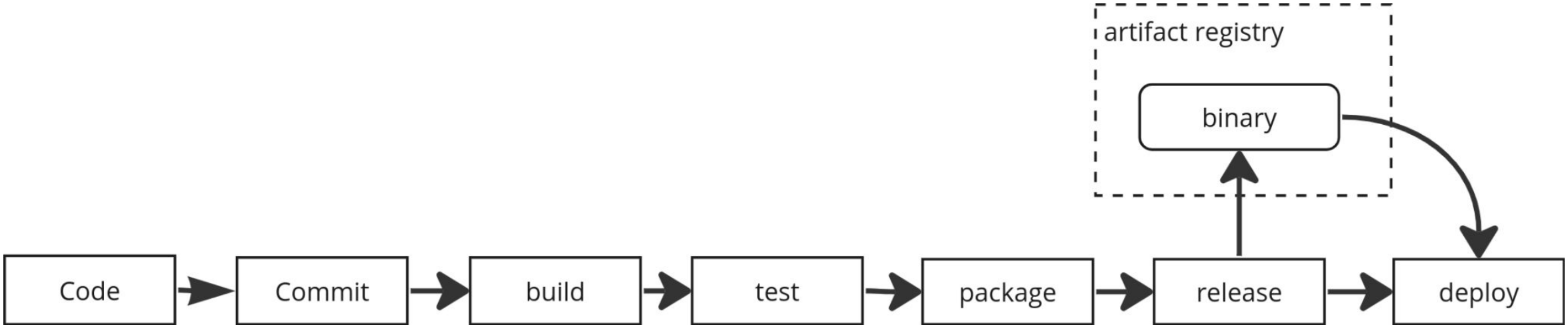
☆ if you like what we do, give our GitHub **chainloop-dev/chainloop** a star :) ☆

[bit.ly/addoc8]

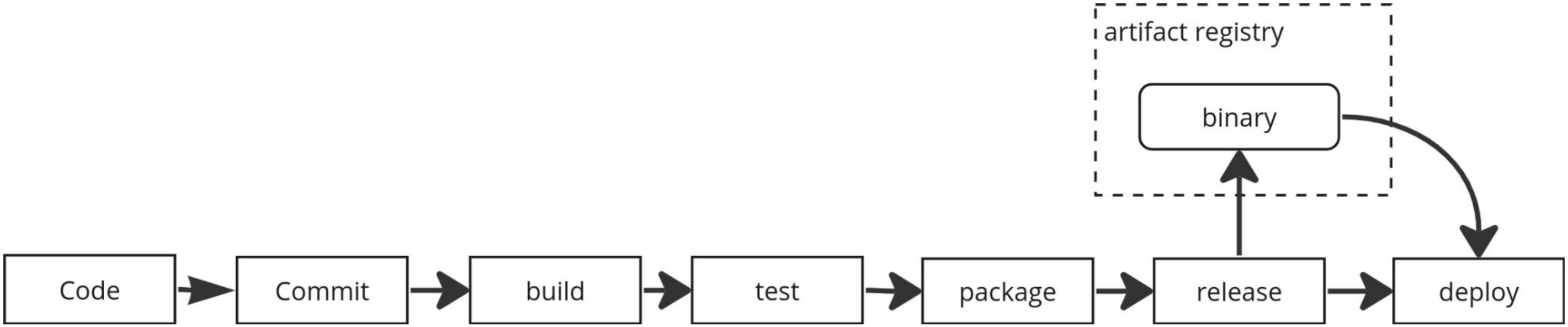
Agenda: Trustworthy SBOM

- ➔ What does it mean?
- ➔ How can we achieve it?
- ➔ Demos: SBOM collection, signing, storage and analysis; SBOM Quality Control Gates, SBOM sharing, Vulnerability Management
- ➔ Where can you go from here?

Yet another CI/CD pipeline



Security/Compliance hello world



Security team

➔ Vulnerabilities management

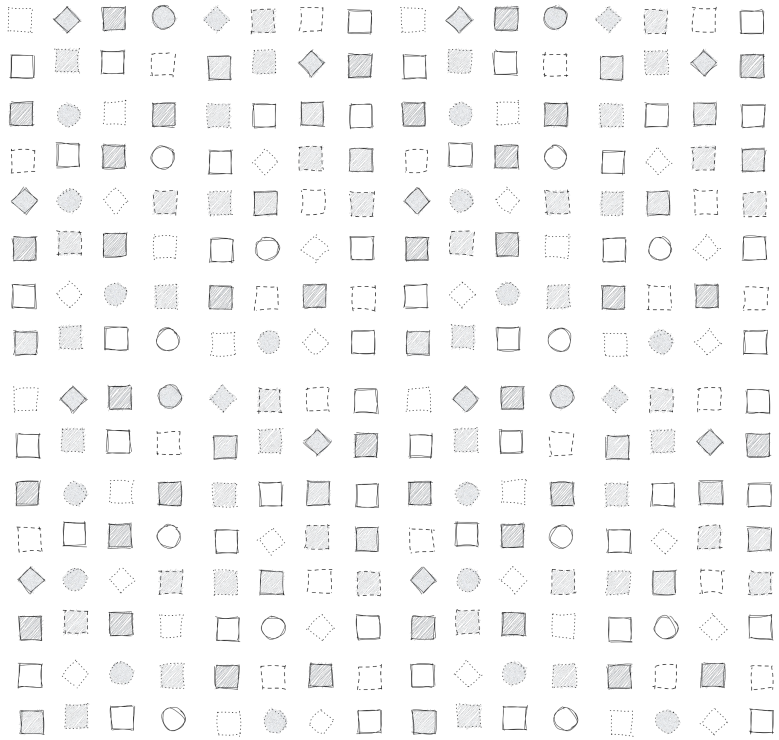
Compliance team

➔ Approved OSS licenses

Platform/SRE team

➔ Visibility / Alerting

Software Supply Chain Metadata is the bedrock



Any piece of context of what and how software is built in your organization

SBOMs, QA tests/reports, CVE scans, VEX, legal/security/architecture reviews, etc.

Software Bill Of Materials is a canonical example

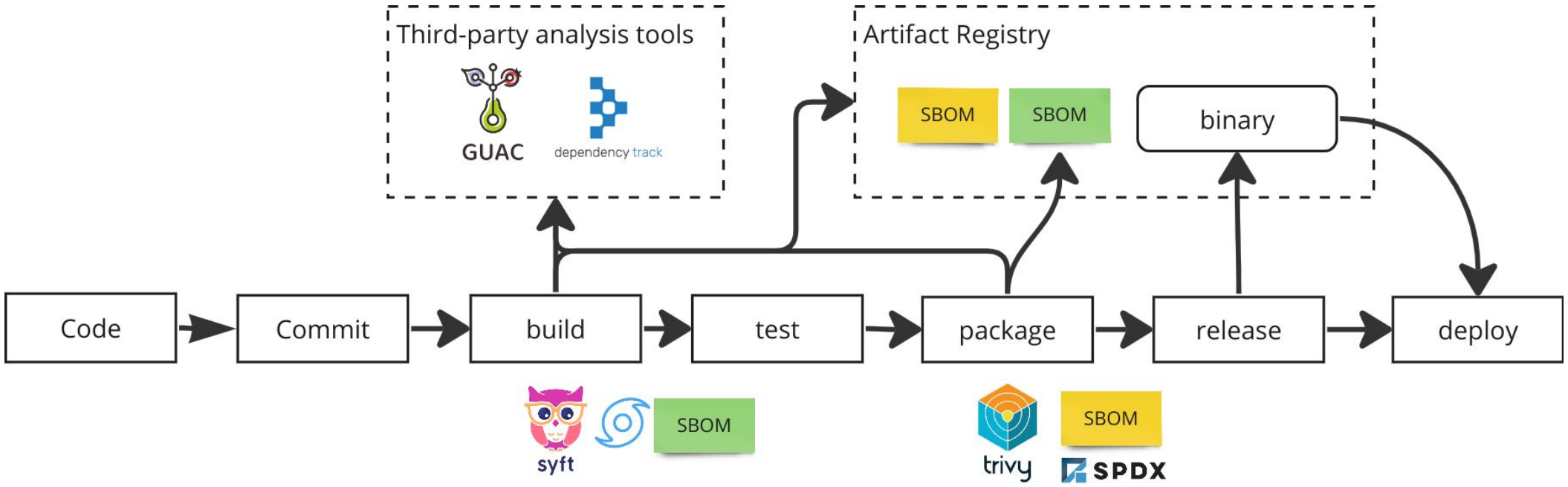
Sample label for Macaroni & Cheese

Source: Wikipedia

Standardised, machine-readable, list of ingredients* for your software

* packages, licenses etc.

DevSecOps to the rescue

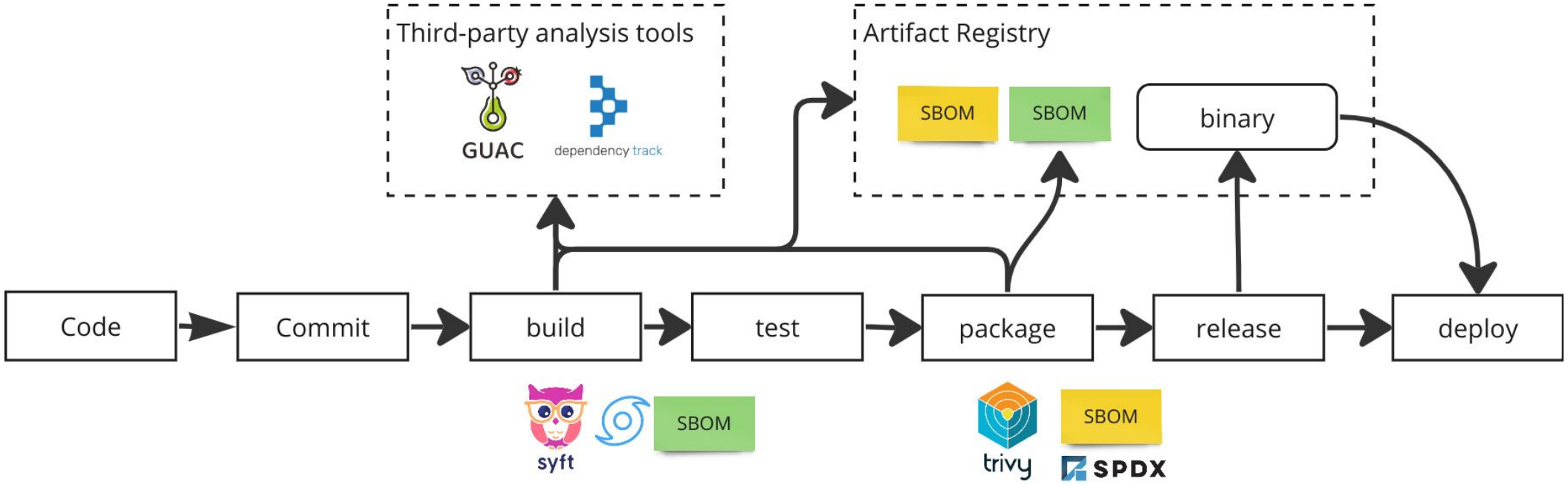


Developers

➔ Take care of choosing, implementing and maintaining tools and services

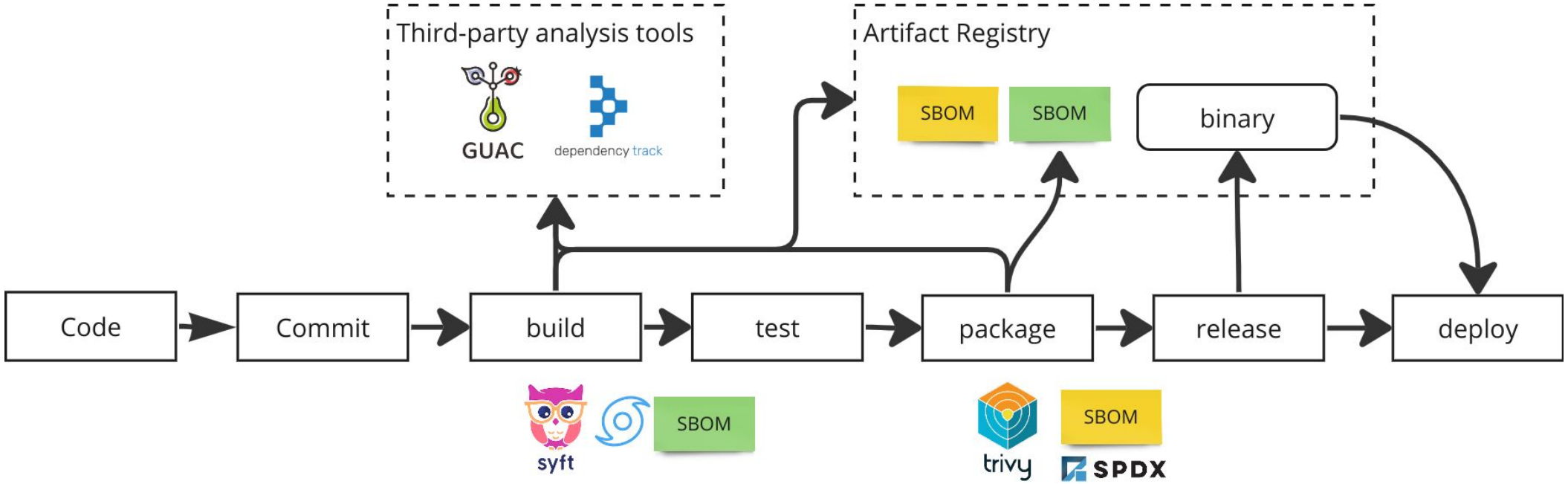
➔ Good starting point but what's next?

What's trust?



SBOM Trust

What's trust?



- ➔ Can I uniquely identify an **SBOM**?
- ➔ Will it be **available** when I need it?
- ➔ Can I trust that the content has not been tampered with?
- ➔ How was it built, from whom or where does it come from?
- ➔ Is it **complete** and **consistent**?
- ➔ Does it even exist?

An **SBOM** **that you can't trust** is useless

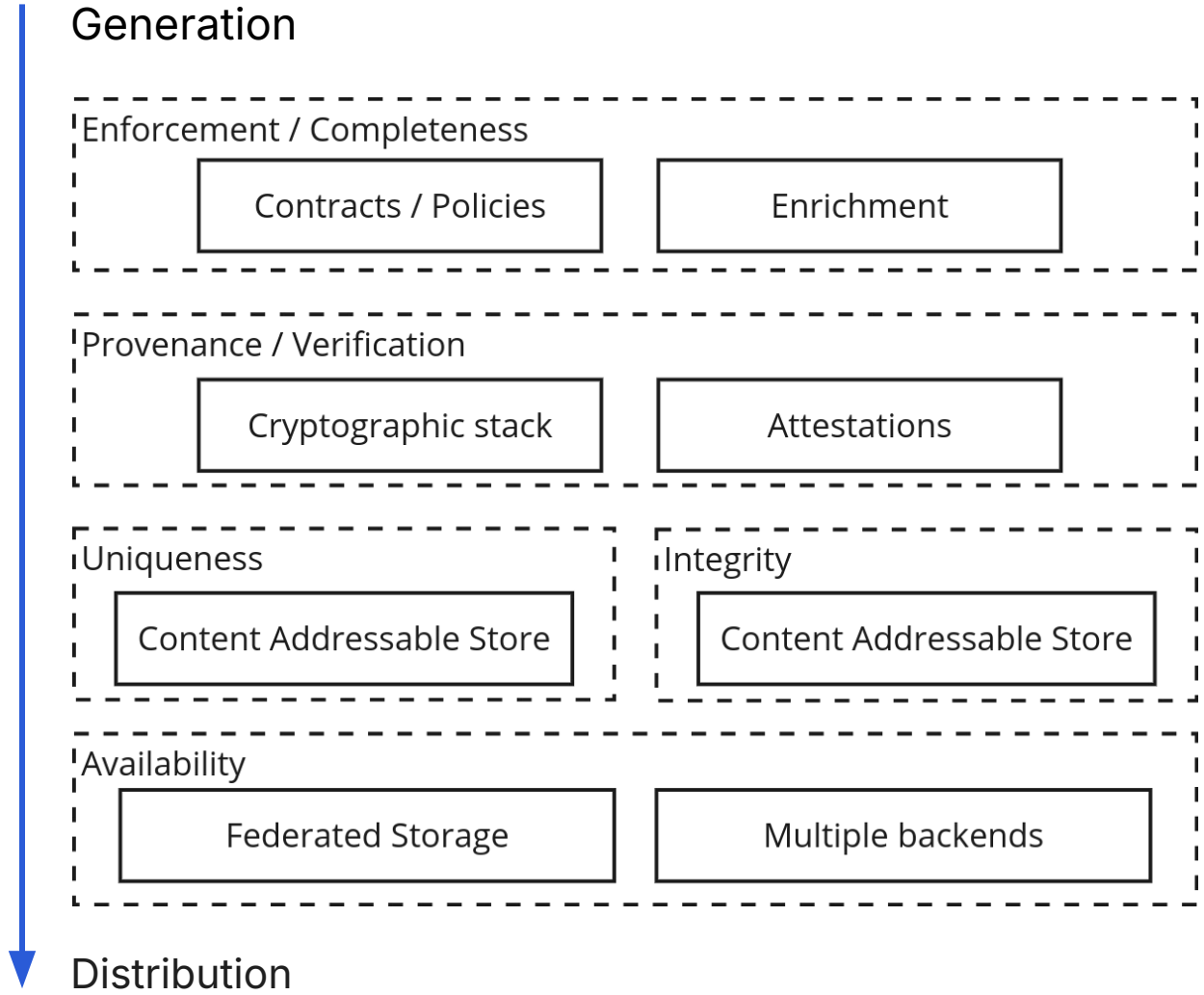
and in fact **dangerous** 🚨...

...we need our SBOMs to be uniquely

identifiable, **unforgeable**, **complete** and

available

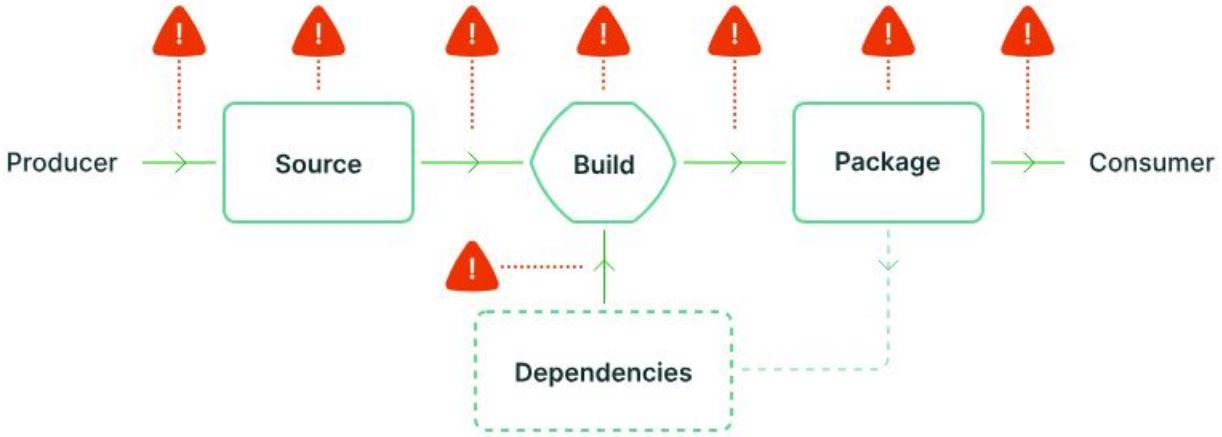
Building the Trust Layer - Pattern



Core components

- ➔ Decentralized storage
- ➔ Content Addressable Storage
- ➔ Attestations
- ➔ Contracts / Policies

Where to start?



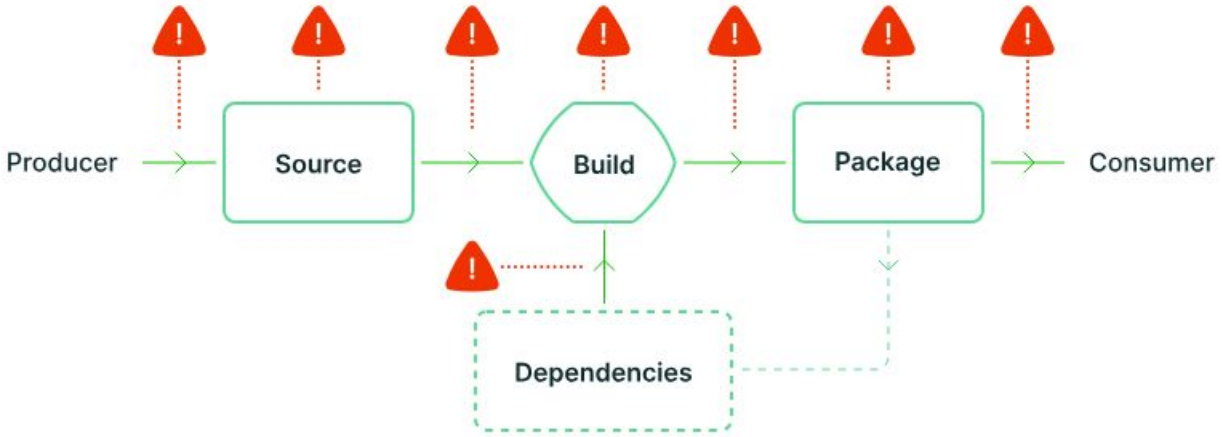
An SBOMs are yet another artifact assembled in your SSC

➔ They **must meet the highest security posture**.

➔ **They can get compromised too**.

Source: https://slsa.dev

Where to start?

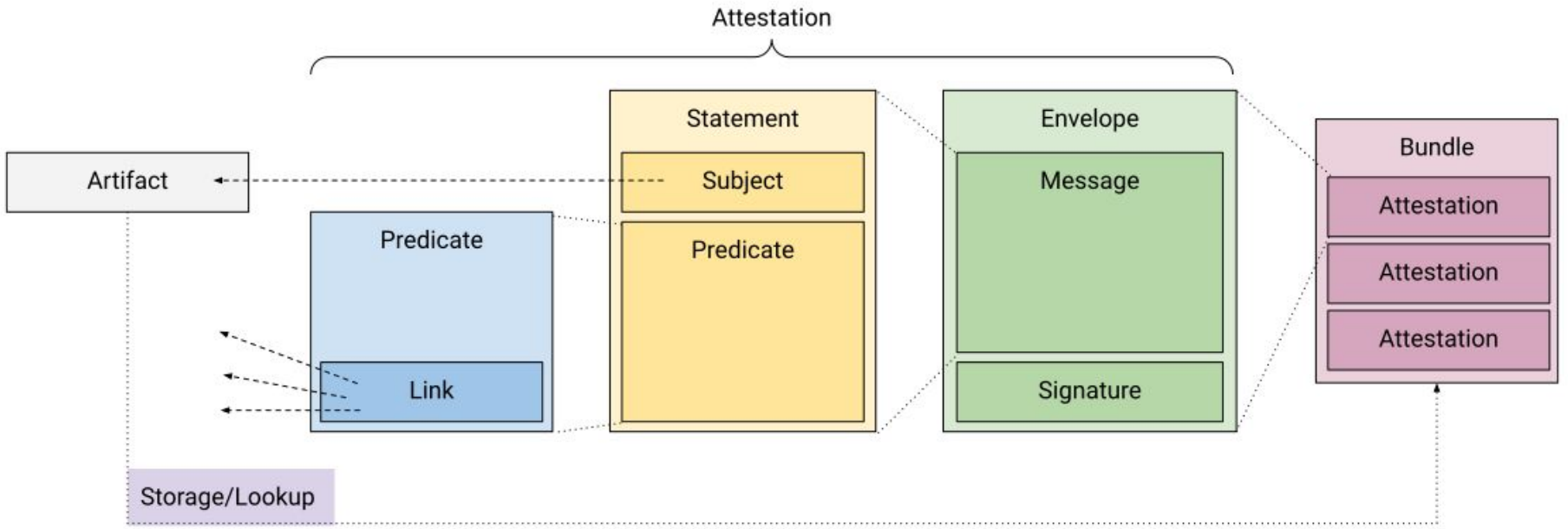


There is a framework for this - slsa.dev

"Any software can introduce vulnerabilities into a supply chain[...] **it's critical to** already **have checks** and best practices in place **to guarantee artifact integrity**, **that the source code you're relying on is the code you're actually using**[...]"

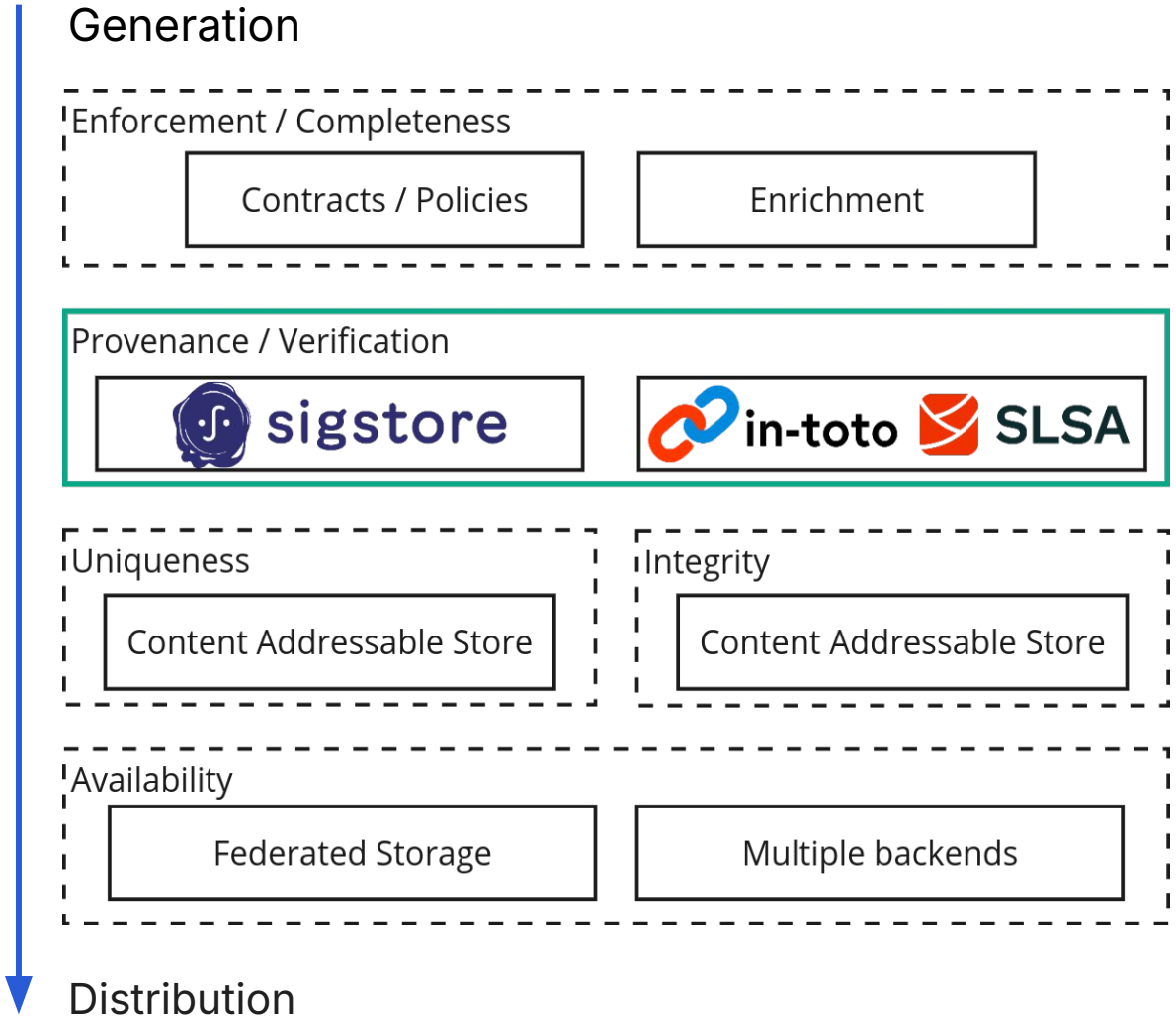
Source: https://slsa.dev

Building the Trust Layer - Attestations



"A software attestation is an **authenticated** statement (metadata) about a software artifact or collection of software artifacts ... **a generalization of raw artifact/code signing**" ~ *slsa.dev*

Building the Trust Layer - Attestations

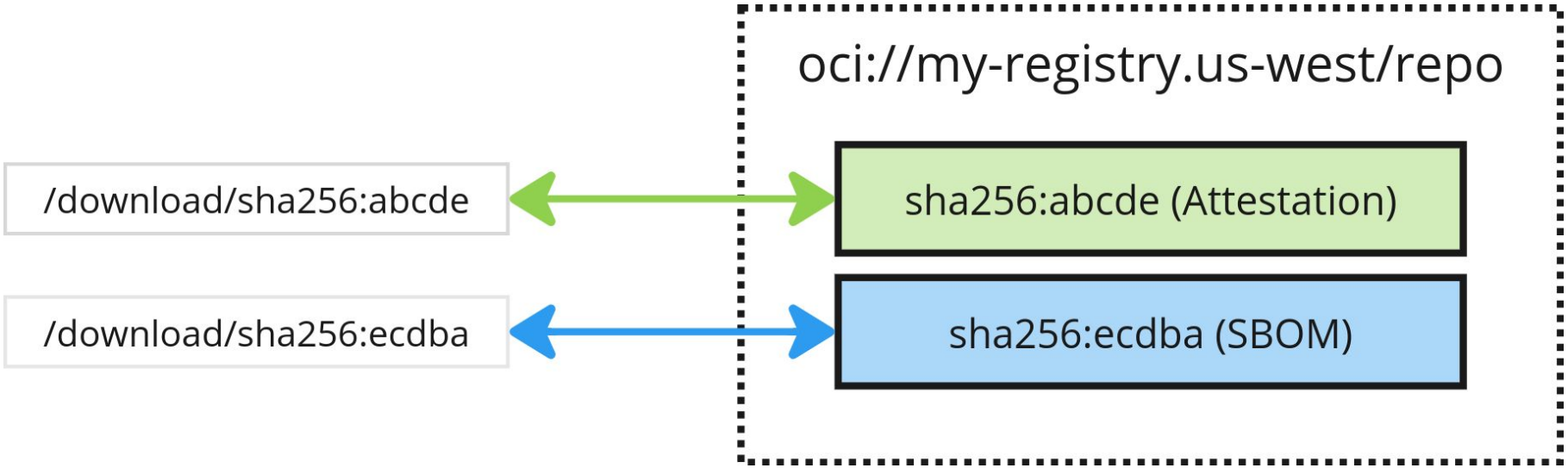


Attestations will wrap SBOMs with additional information and a signature to **enable integrity and provenance verifications**.

in-toto sigstore

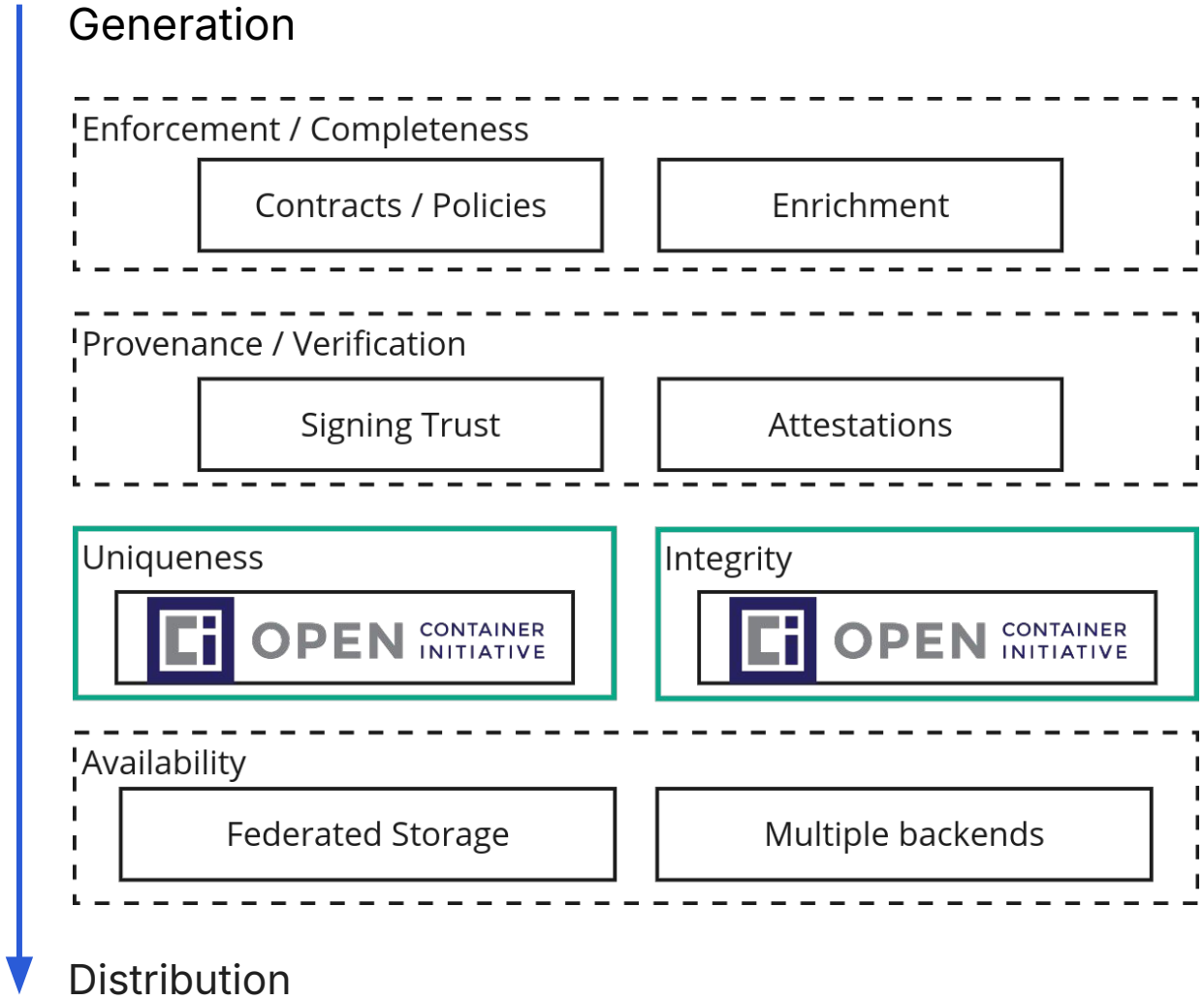
SBOMit

Building the Trust Layer - CAS



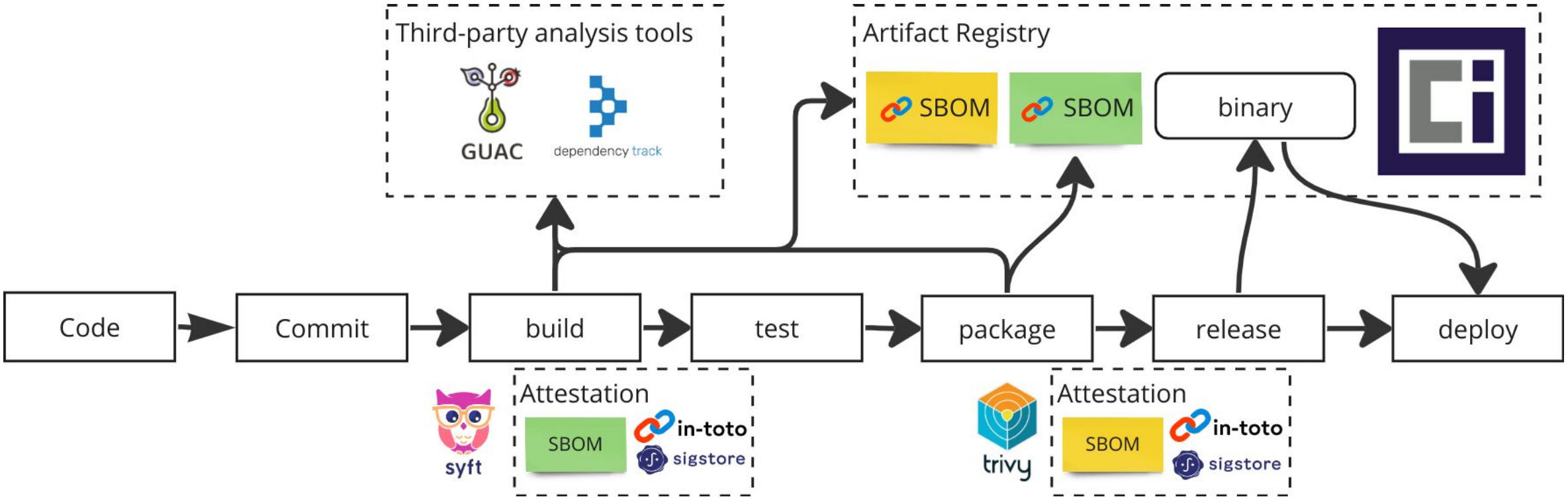
Content-Addressable Storage (CAS) is a system that organizes and **retrieves** data **based on the data's content**, **rather than** its location or **name**, ensuring data integrity and immutability

Building the Trust Layer - CAS



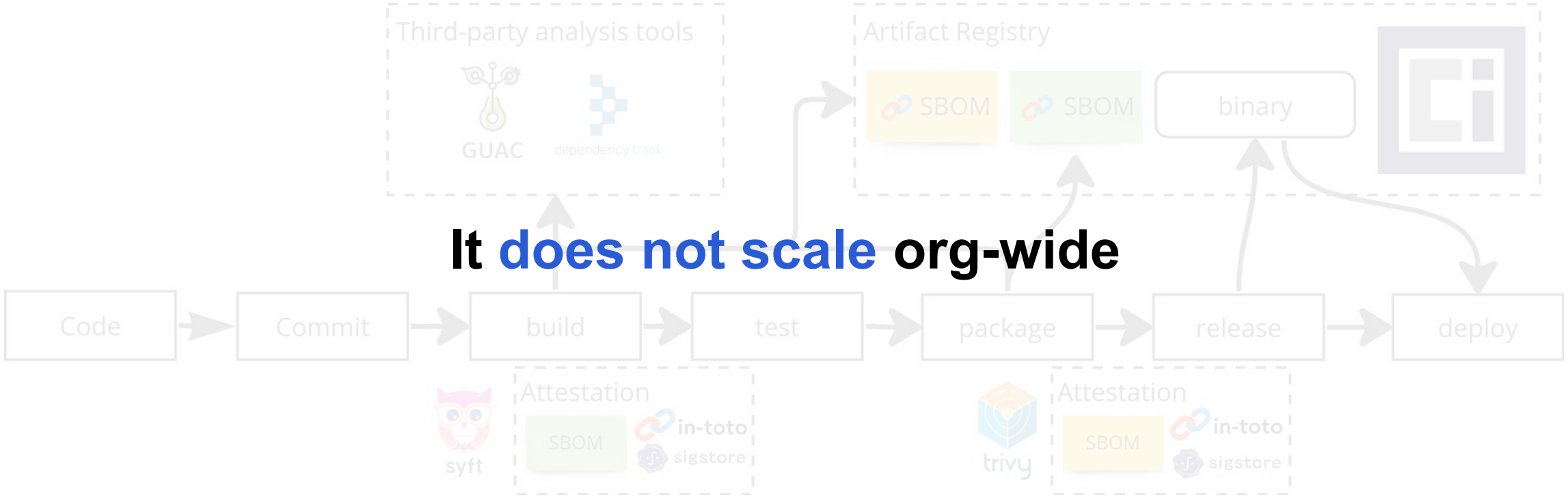
Stored SBOMs will be **unique**, **identifiable** and **integrity verifiable**

Implementation



SBOMs that you can trust in identity, integrity and origin

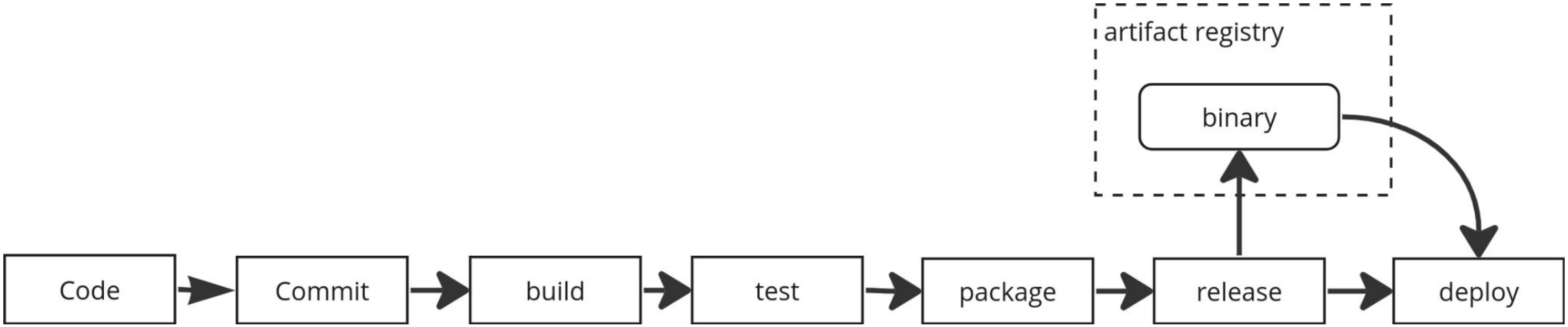
Implementation



It does not scale org-wide

Your Software Supply Chain **Metadata is scattered across silos, inconsistent, optional, and not enforced**

Security/Compliance hello world



Security team

➔ Vulnerabilities management

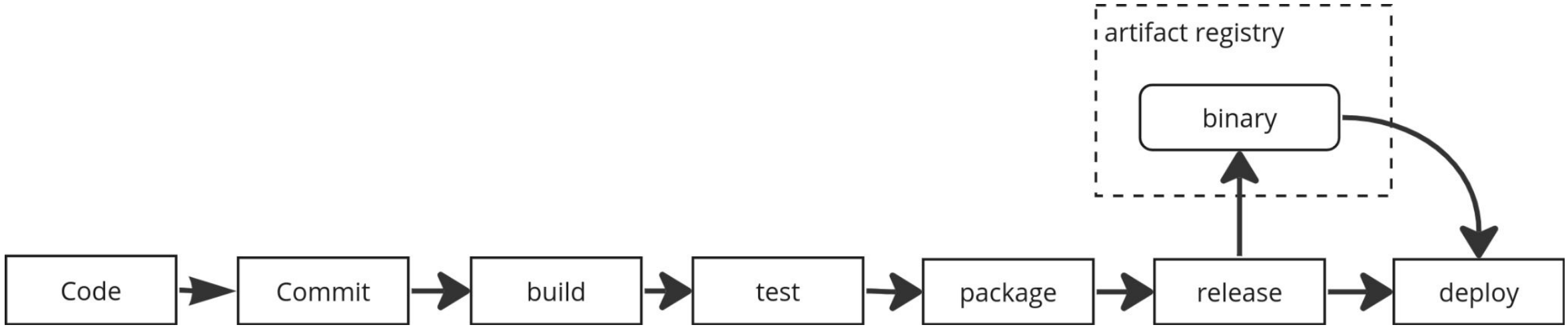
Compliance team

➔ Approved OSS licenses

Platform/SRE team

➔ Visibility / Alerting

Security/Compliance posture ++



Security team

- ➔ Vulnerabilities management
- ➔ Provenance information (build system)
- ➔ SLSA 3 (integrity)
- ➔ Operationalize SBOMs

Compliance team

- ➔ Approved OSS licenses
- ➔ SOC 2, ISO 27001, ...

Platform/SRE team

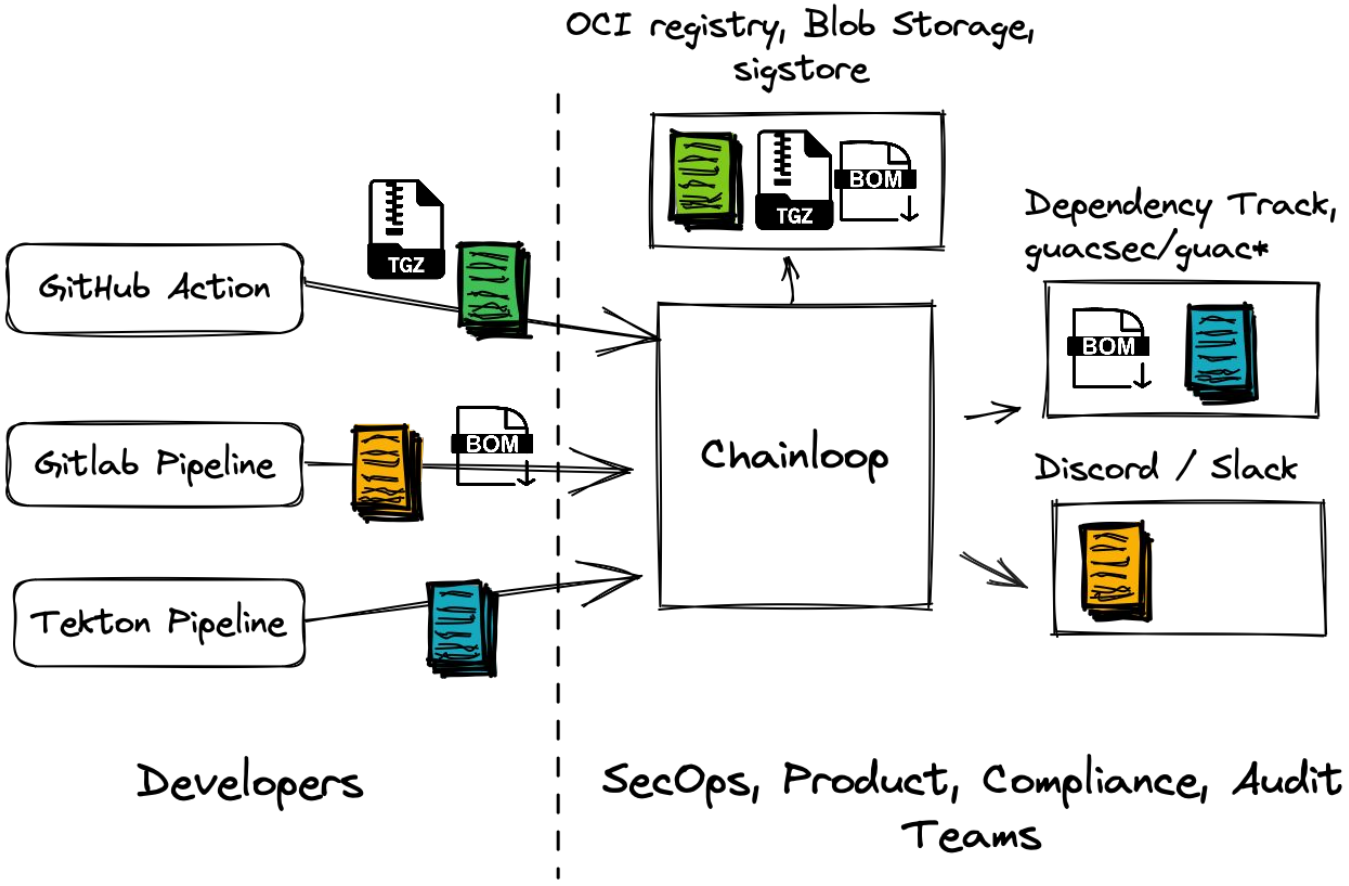
- ➔ Visibility / Alerting

Trusted Supply Chain Metadata

Chainloop

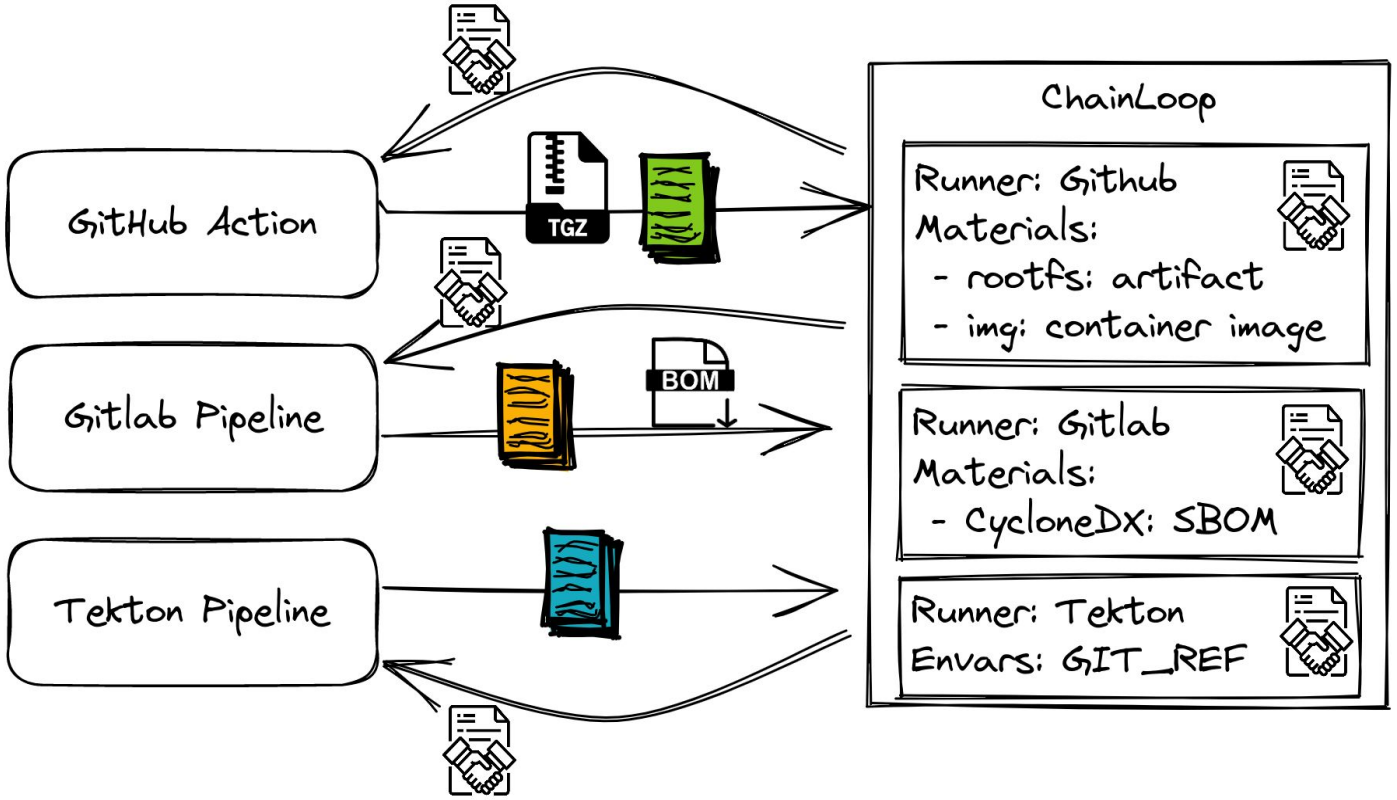
Chainloop is an Open Source Evidence Store for your Software Supply Chain metadata, **SBOM**s, **VEX**, **SARIF** files and more

github.com/chainloop-dev/chainloop



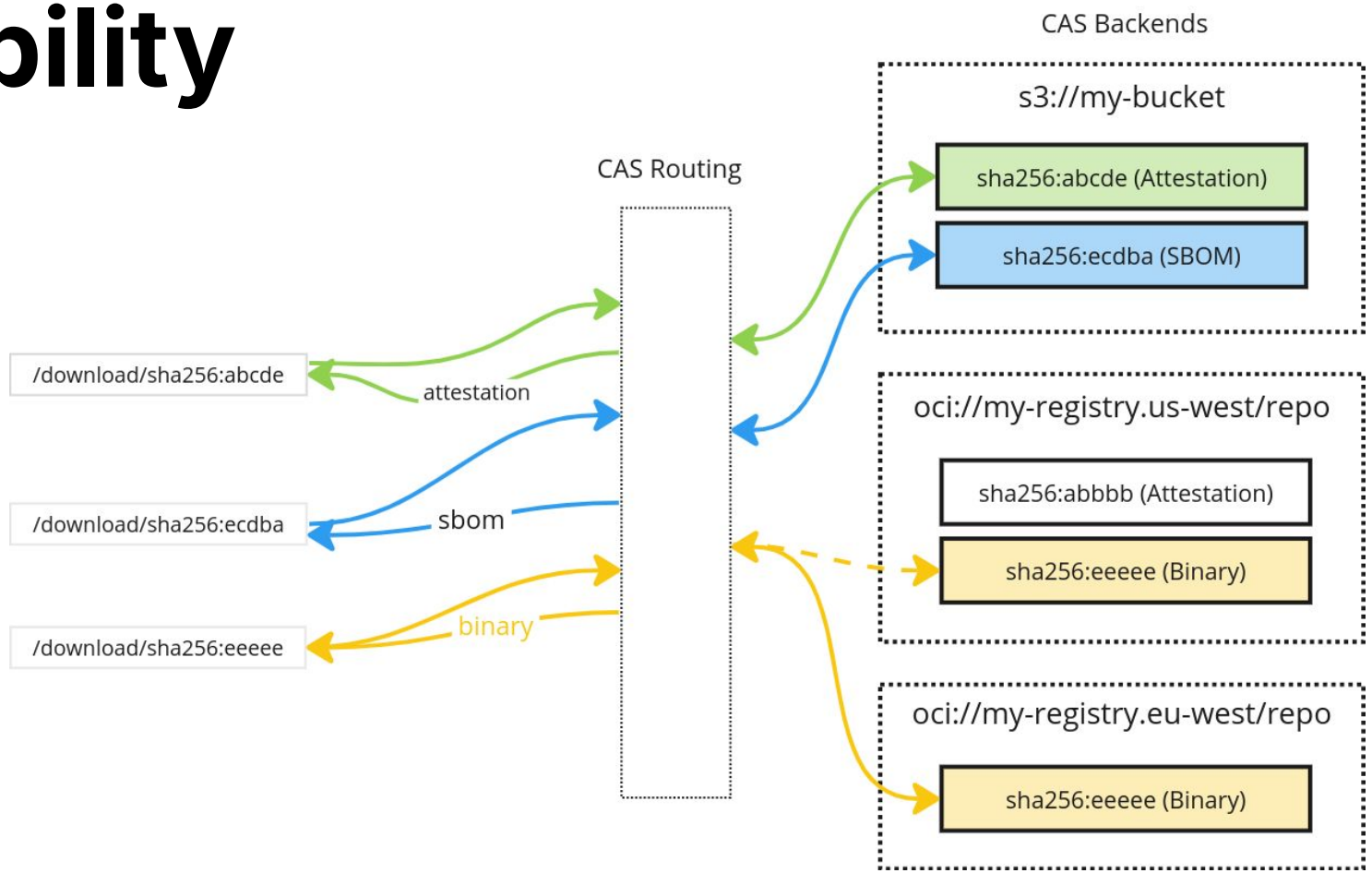
Enforcement and Validation

Contracts are declarative requirements of the pieces of evidence a development team needs to provide




```
schemaVersion: v1

materials:

  - type: ARTIFACT

    name: binary

    output: true

  - type: SBOM_CYCLONEDX_JSON

    name: sbom

runner:

  type: "GITHUB_ACTION"
```

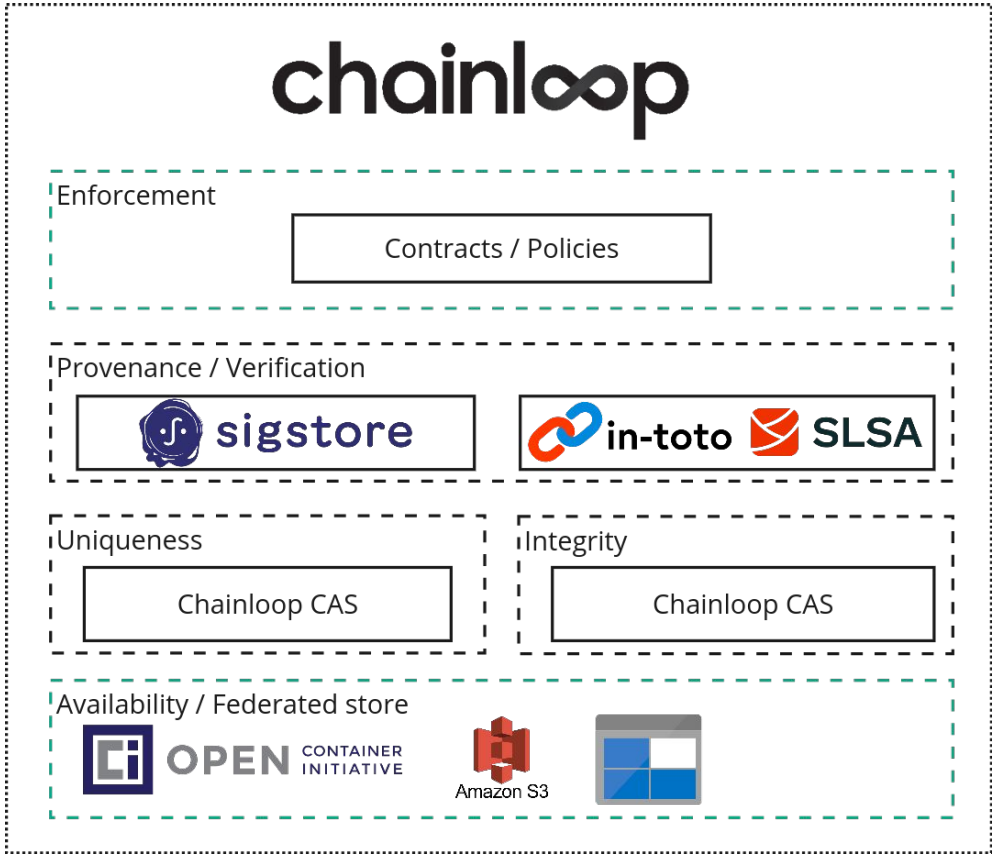
Availability



Federated Content-Addressable Storage (CAS) works across backends enabling advanced routing for replication, geolocation, retention rules, ...

Trusted Supply Chain Metadata - Chainloop

Generation



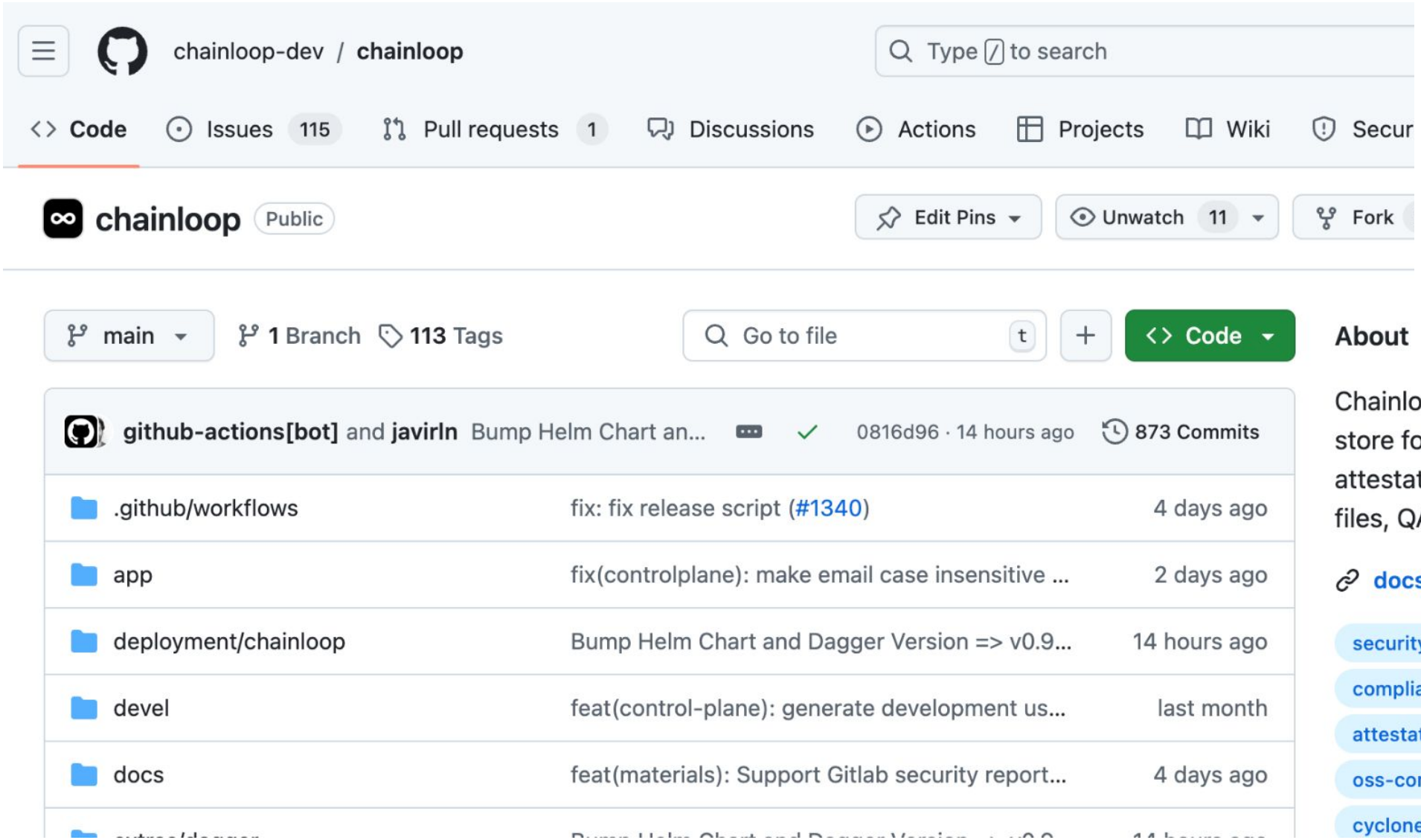
Distribution

SBOMs that you can trust, on identity, integrity and origin. Also **storage compliant** and **enforced**

github.com/chainloop-dev/chainloop

Use Cases Examples

You can try this at home

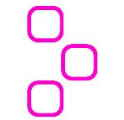


Chainloop Evidence Store
SBOM Use-Cases

- ➔ SBOM collection, signing, storage and analysis
- ➔ SBOM Quality Gates
- ➔ Control Gates
- ➔ Vulnerability Management
- ➔ SBOM Sharing

Centralised, enforced, collection, signing, storage and analysis of SBOMS

Centralized SBOM management



GUAC

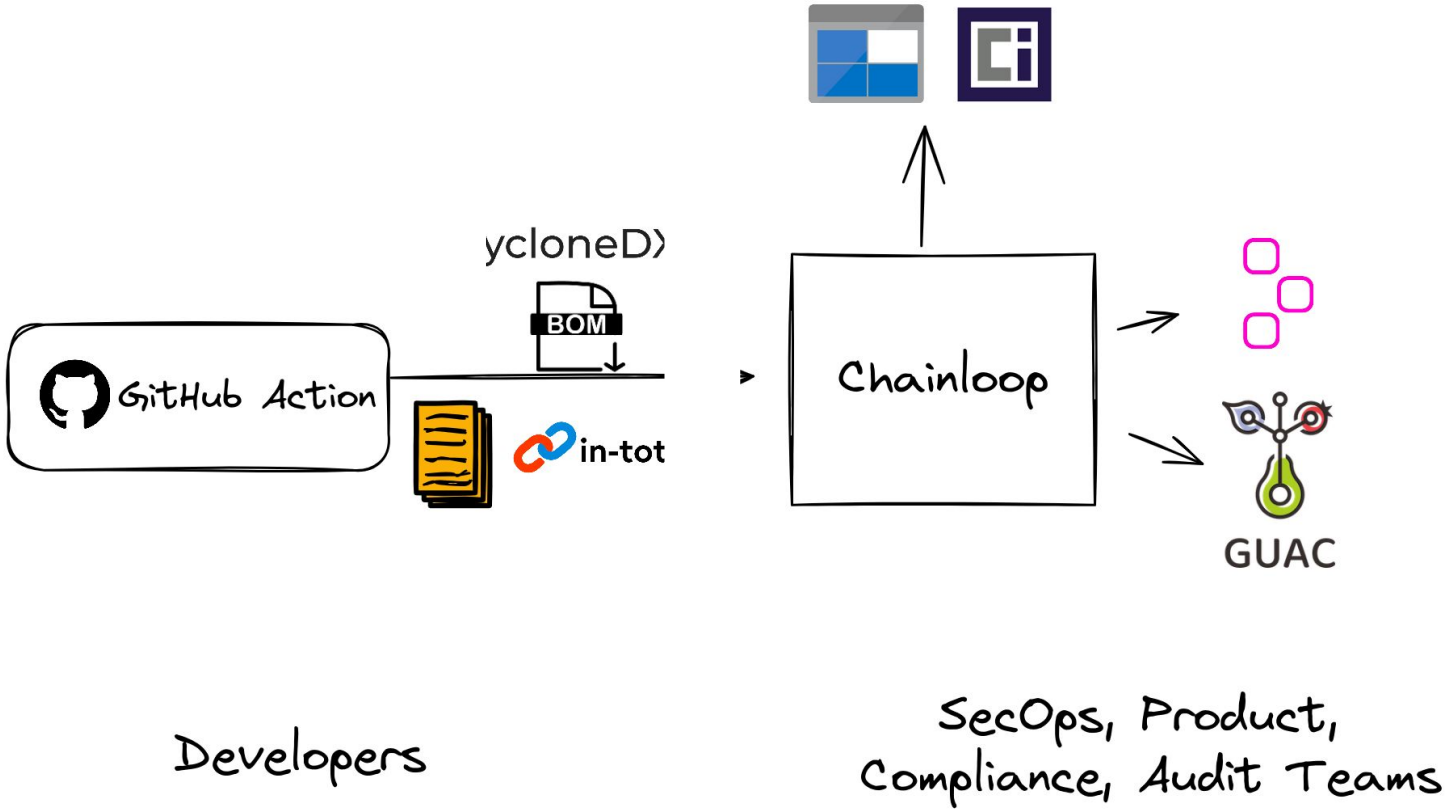
Developers

SecOps, Product,
Compliance, Audit Teams

Setup, collection and storage

1. Collect CycloneDX SBOM from GitHub
2. Wrap it in in-toto attestation
3. Store it in Azure Blob Storage and OCI registry
4. Send it to
   a. Dependency-Track
   b. guacsec/guac

Centralized SBOM management

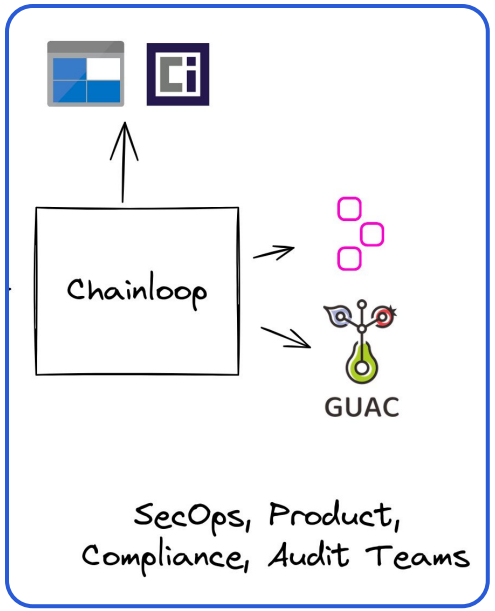


With Chainloop

- Enforced but with minimal friction to developers
- Clear separation of concerns
  - SecOps Tooling
  - Dev practices
- Get visibility on progress
- BYO tools/CI/PKI/Storage

https://github.com/chainloop-dev/chainloop

SecOps 1 - Storage and Analysis



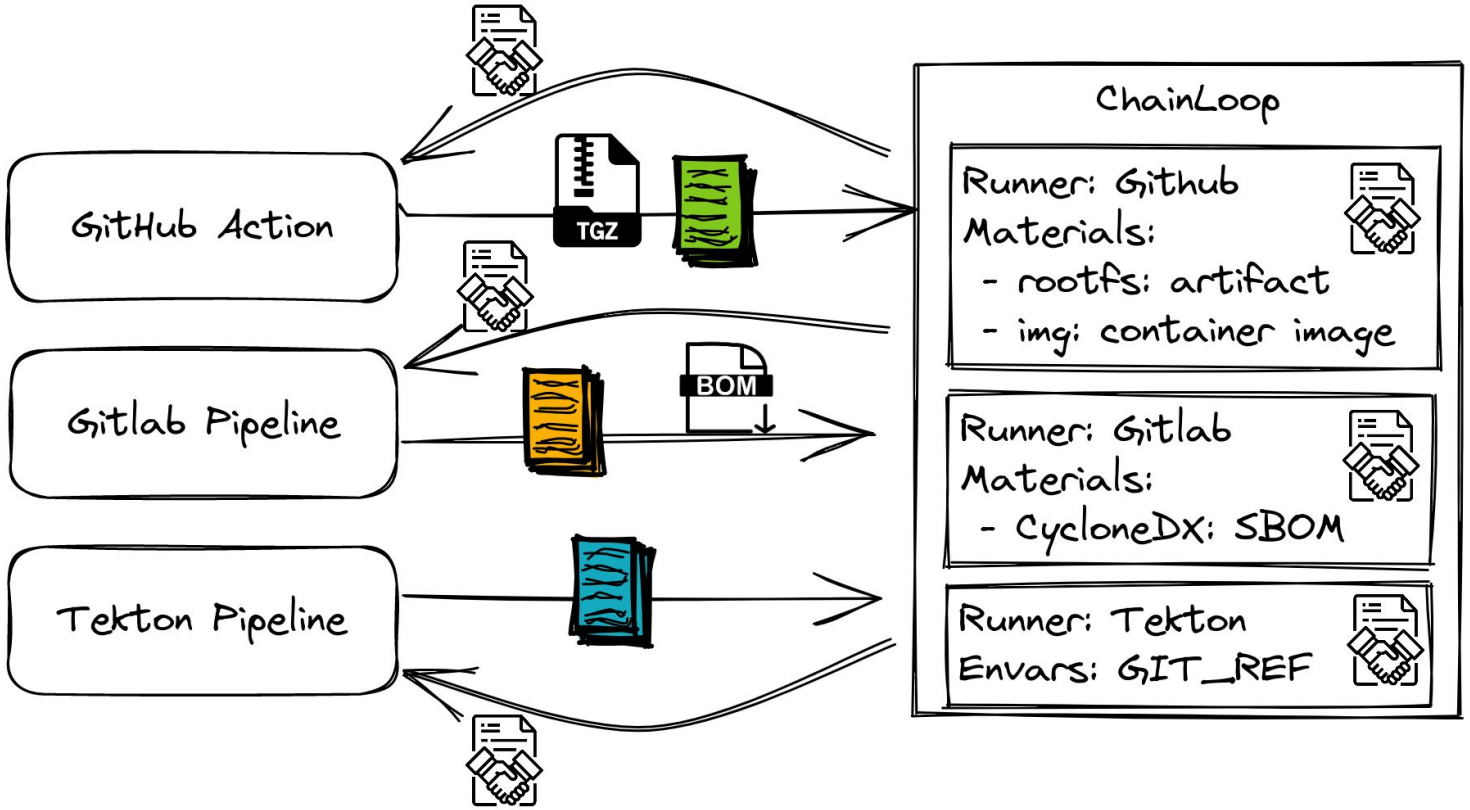

```
→  ~ chainloop cas-backend ls
```

NAME	LOCATION	PROVIDER	DESCRIPTION	LIMITS	DEFAULT
cloud-flare-test	https://35c24d45e1cd7bc36279b8a4d130c7fe.r2.cloudflarestorage.com/chainloop	AWS-S3		MaxSize: 100M	true
default-inline		INLINE	Embed artifacts content in the attestation (fallback)	MaxSize: 500K	false

```
→  ~ chainloop integration available ls
```

NAME	VERSION	MATERIAL REQUIREMENT	DESCRIPTION
dependency-track	1.4	SBOM_CYCLONEDX_JSON	Send CycloneDX SBOMs to your Dependency-Track instance
discord-webhook	1.1		Send attestations to Discord
guac	1.0	SBOM_CYCLONEDX_JSON, SBOM_SPDX_JSON	Export Attestation and SBOMs metadata to a blob storage backend so guacsec/guac can consume it
slack-webhook	1.0		Send attestations to Slack
smtp	1.0		Send emails with information about a received attestation

SecOps 2 - Enforcement and Validation




```
schemaVersion: v1

materials:

  - type: ARTIFACT

    name: binary

    output: true

  - type: SBOM_CYCLONEDX_JSON

    name: sbom

runner:

  type: "GITHUB_ACTION"
```

Contracts are declarative requirements of the pieces of evidence a development team needs to provide

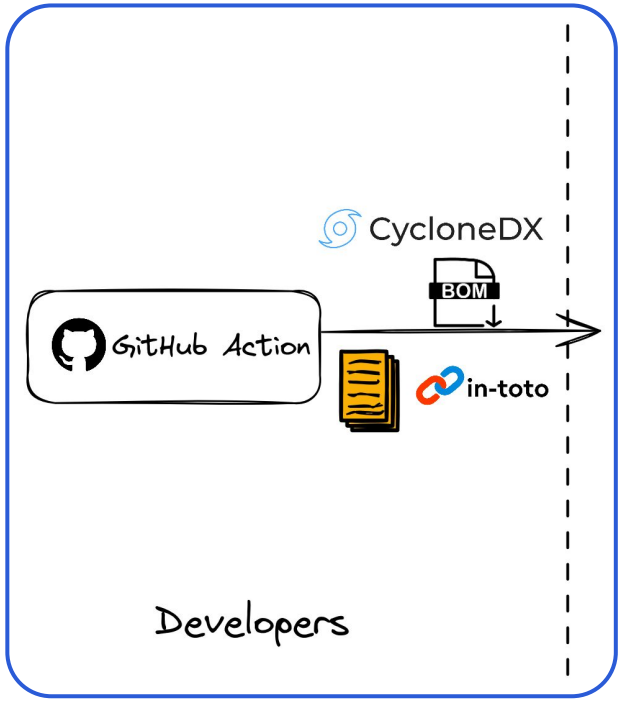
Developers

An **attestation** is the process of collecting pieces of evidence (metadata), securing it, and sending it to Chainloop control plane

```
export CHAINLOOP_TOKEN=asd...

$ chainloop attestation init

$ chainloop attestation add --name
binary --value
spring-petclinic-3.2.0-SNAPSHOT.jar

$ chainloop attestation add --name
sbom --value sbom-cyclonedx.json

$ chainloop attestation push
```

https://docs.chainloop.dev/getting-started/attestation-crafting

Developers



6 ■■■■■■ .chainloop.yml

Viewed

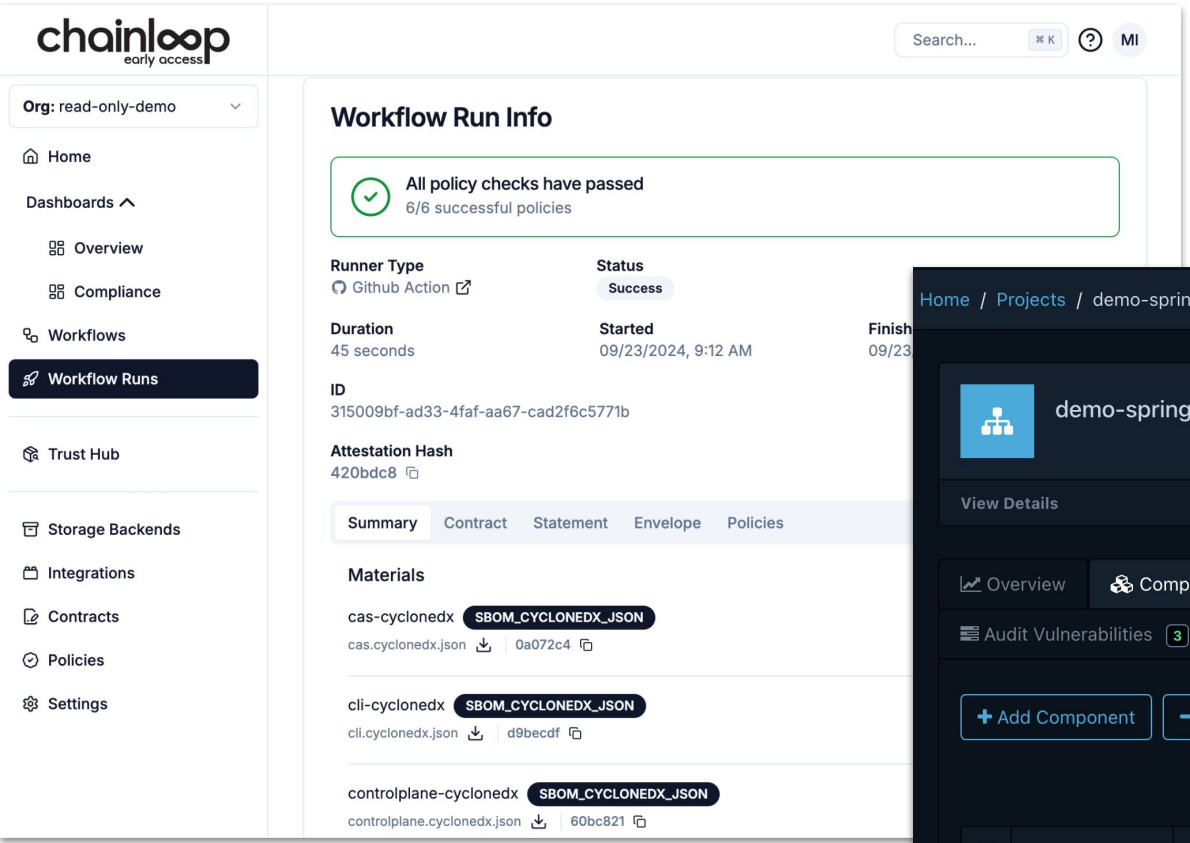
```
+ # defines files to be added to the attesta
+ attestation:
+   - name: binary
+     path: artifacts/spring-petclinic-*.jar
+   - name: sbom
+     path: metadata/sbom.cyclonedx.json
```

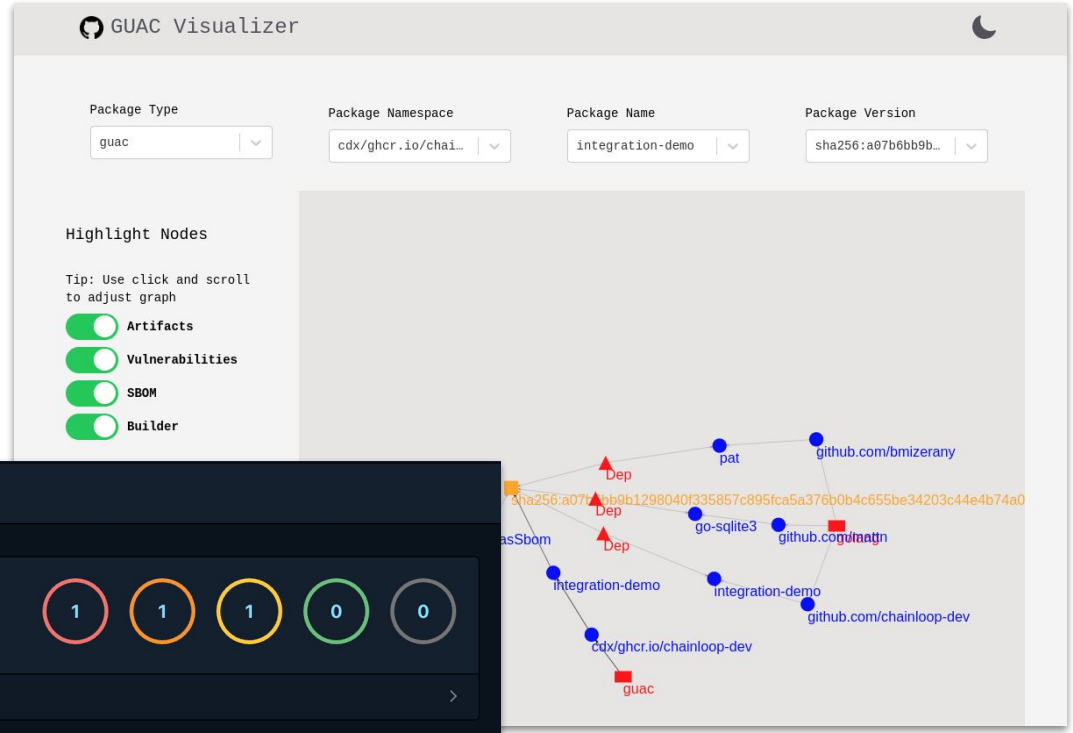
12 ■■■■■■ .github/workflows/maven-build.yml

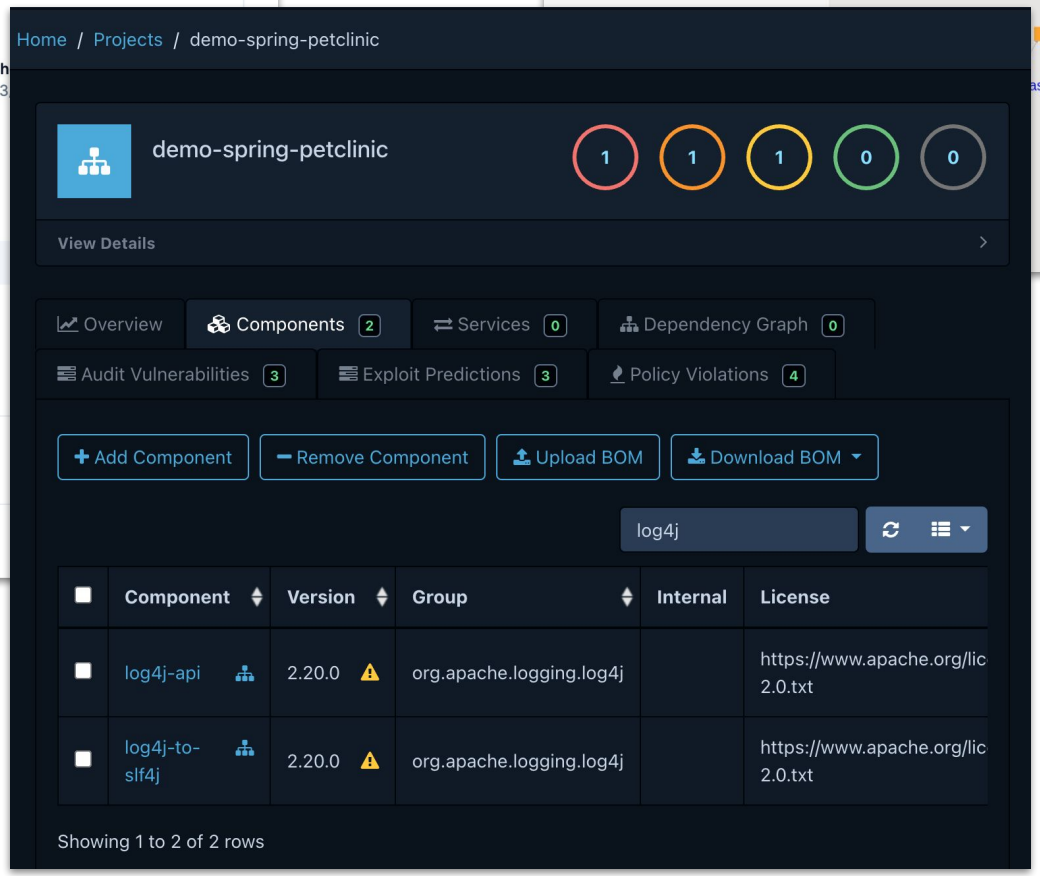
Viewed

```
@@ -57,3 +57,15 @@ jobs:
          with:
            name: metadata
            path: metadata/*
+
+   # Send metadata to Chainloop
+   chainloop:
+     name: Chainloop
+     uses: chainloop-dev/labs/.github/workflows/chainloop.yml@3ac622143863b6cb4f1f4e30f5982bbe
+     needs: collect-metadata
+     with:
+       contract_revision: 2
+     secrets:
+       api_token: ${{ secrets.CHAINLOOP_ROBOT_ACCOUNT }}
+       signing_key: ${{ secrets.PRIVATE_KEY
+       signing_key_password: ${{ secrets.PRIVATE_KEY_PASSWORD }}
```

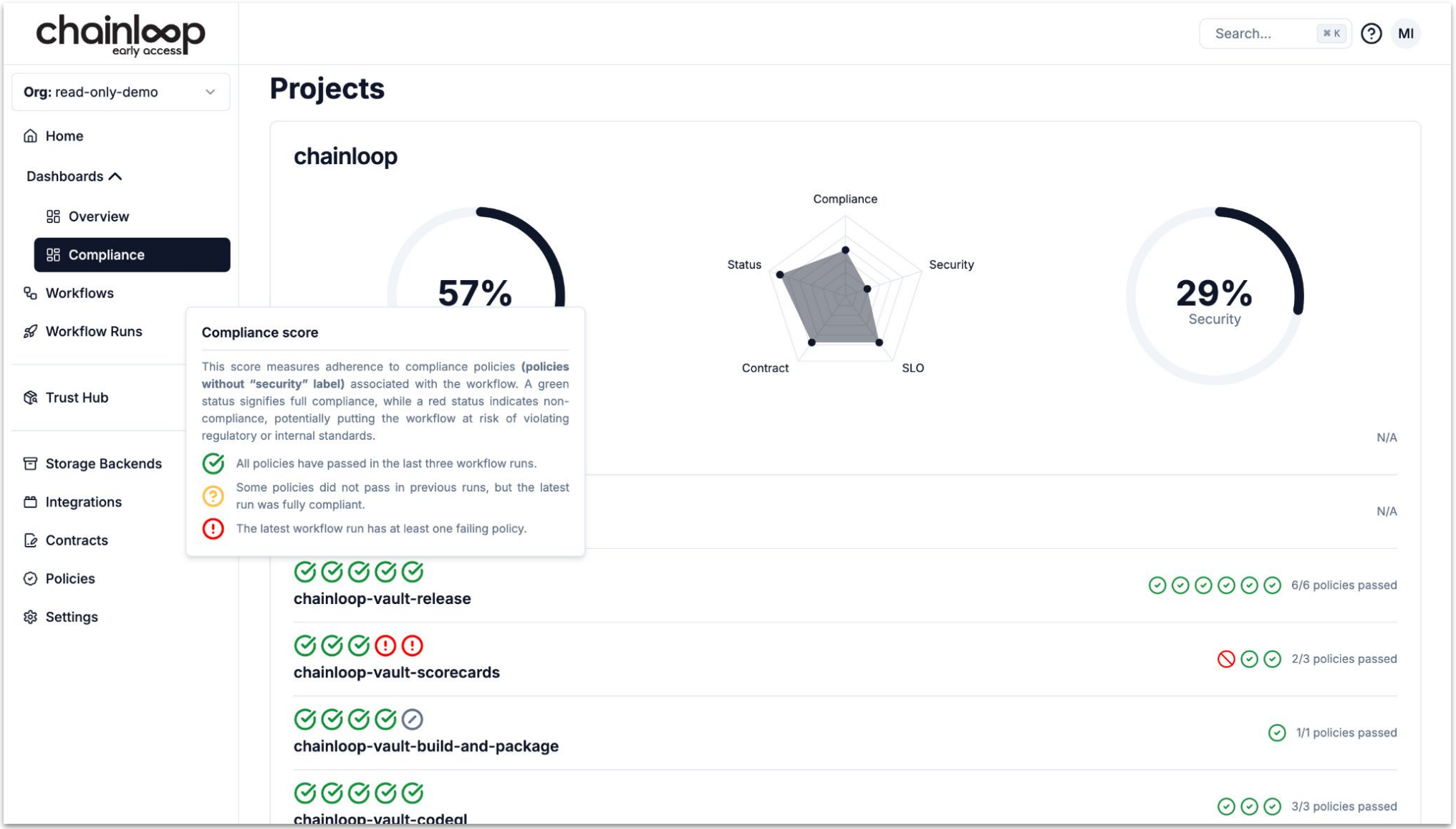
Result stored and analyzed







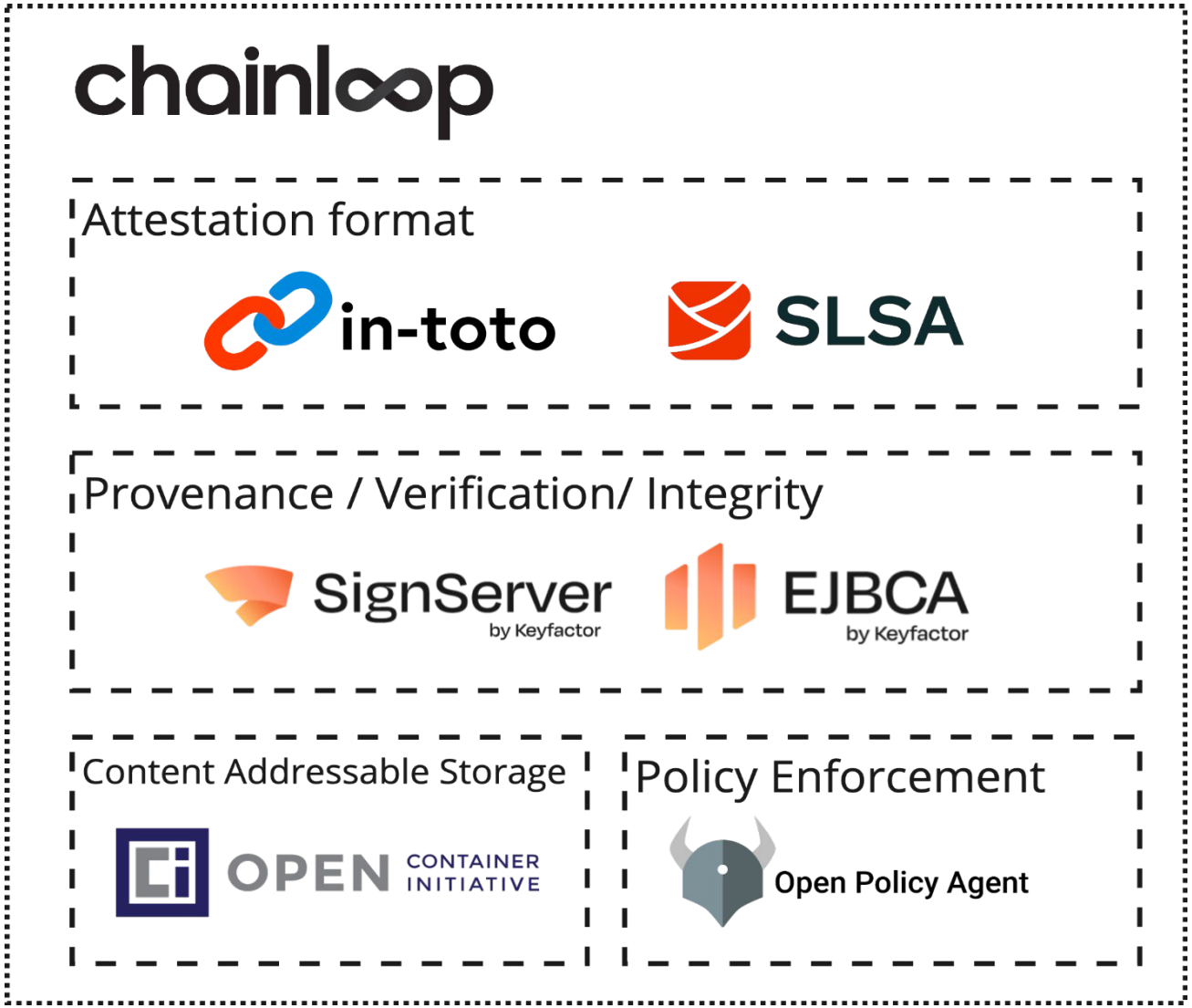
Visibility on adoption



Pluggable

Bring your own tools

- Software Attestations
- PKI solutions
- Content Addressable storage
- Policy Engine



https://docs.chainloop.dev/guides/ejbca/
https://docs.chainloop.dev/guides/signserver/

SBOM Quality Control Gates

Problem: SBOM quality is immeasurable, inconsistent or non existing

SBOM Quality properties

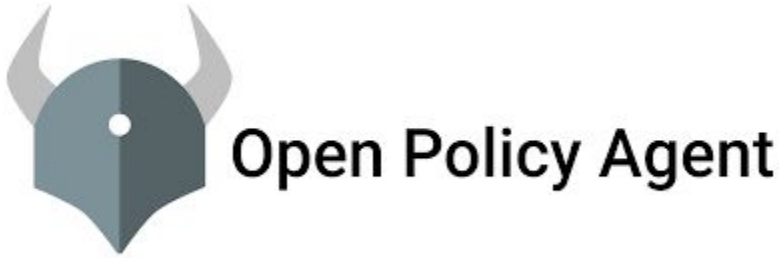
- NTIA / TR-03183
- Freshness
- Generator check
- Spec version
- Signed SBOM
- ...

SBOM Minimum Elements

NTIA	TR-03183
Supplier Name	Component Creator
Component Name	Component Name
Version of the Component	Component Version
Other Unique Identifiers	Other Unique Identifiers
Dependency Relationship	Dependencies on Other Components
Author of SBOM Data	Creator of the SBOM
Timestamp	Timestamp
	Licence
	Hash value of the executable component
	SBOM URI
	Source code URI
	URI of the executable form of the component
	Hash value of the source code of the component

SBOM Quality Gate

Chainloop policies written in Rego



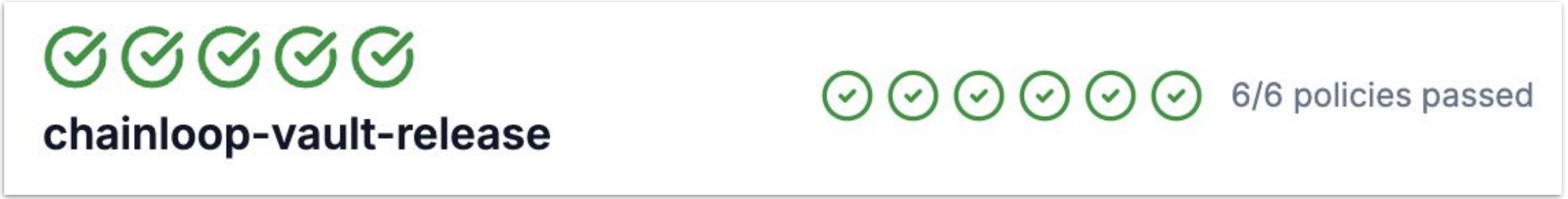

```yaml
schemaVersion: v1
policies:
  materials:
    # NTIA components check
    - ref: cyclonedx-ntia
    # Up to date SBOM
    - ref: cyclonedx-freshness
      with:
        limit: 5 # < 5 days old
    - ref: cyclonedx-version
      with:
        version: 1.5
    - ref: cyclonedx-generation
      with:
        tool: syft
```

https://docs.chainloop.dev/reference/policies

https://www.openpolicyagent.org/docs/latest/policy-language

SBOM Quality Gate

Materials	
Name	cas-cyclonedx
Type	SBOM_CYCLONEDX_JSON
Filename	cas.cyclonedx.json
Value	
Digest	sha256:0a072c4323ef6bc6f3c6577915ad68fe3b0e3abf49a2015b8cd28323d4ddaaa3
Policies	
Policy Violations	cyclonedx-licenses None
Policy Violations	cyclonedx-freshness None
Policy Violations	cyclonedx-banned-licenses None
Policy Violations	cyclonedx-banned-components None



Vulnerability Management

Problem: How to contextualize SBOMs and Vex Files

Vulnerability mgmt: VEX + SBOM files

SPDX

CycloneDX

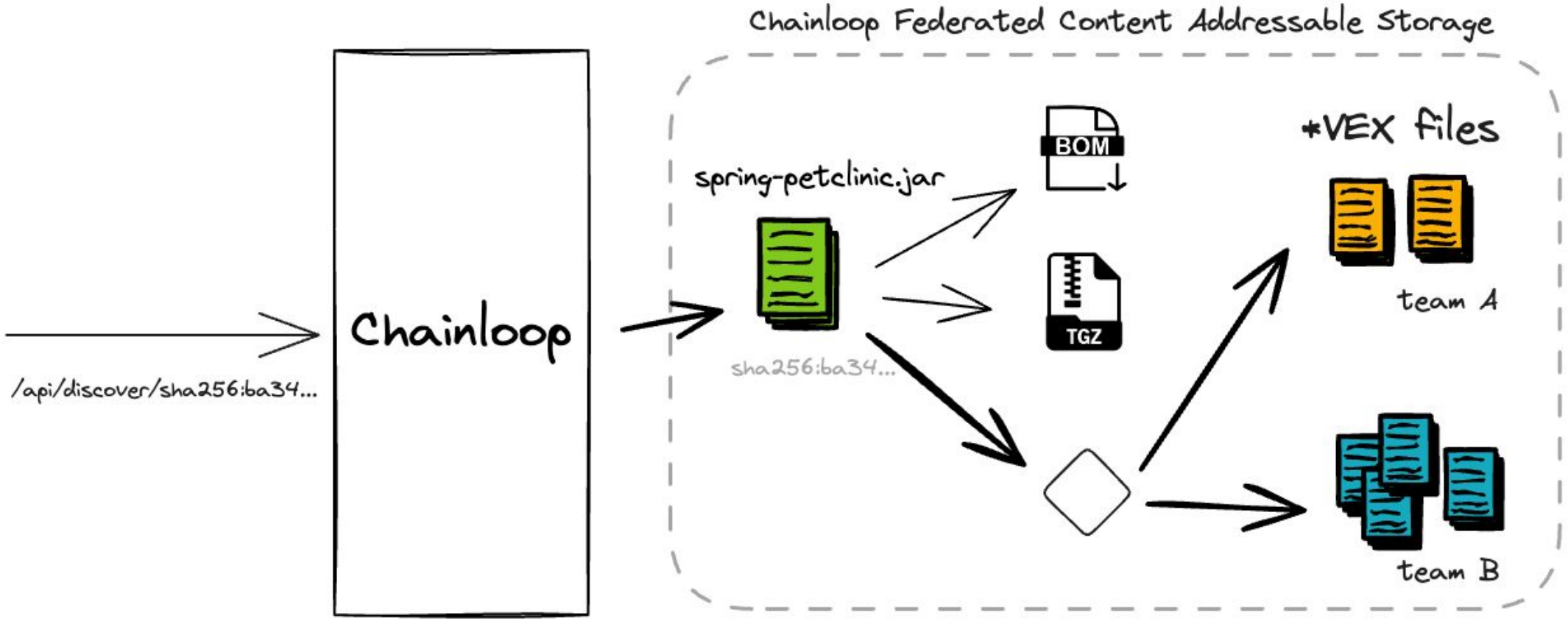
+

VEX

aqua trivy

Dependency Track

Vulnerability mgmt: VEX + SBOM files



Vulnerability mgmt: VEX + SBOM files

```
❯ trivy image --severity=CRITICAL,HIGH,MEDIUM --vex platform.vex.json ghcr.io/chainloop-dev/platform/frontend:v0.87.16
2024-09-24T09:50:58+02:00       INFO    [vuln] Vulnerability scanning is enabled
2024-09-24T09:50:58+02:00       INFO    [secret] Secret scanning is enabled
2024-09-24T09:50:58+02:00       INFO    [secret] If your scanning is slow, please try '--scanners vuln' to disable secret scanning
2024-09-24T09:50:58+02:00       INFO    [secret] Please see also https://aquasecurity.github.io/trivy/v0.55/docs/scanner/secret#recommendation for faster secret d
etection
2024-09-24T09:50:58+02:00       INFO    Detected OS     family="wolfi" version="20230201"
2024-09-24T09:50:58+02:00       INFO    [wolfi] Detecting vulnerabilities...    pkg_num=26
2024-09-24T09:50:58+02:00       INFO    Number of language-specific files       num=2
2024-09-24T09:50:58+02:00       INFO    [gobinary] Detecting vulnerabilities...
2024-09-24T09:50:58+02:00       INFO    [node-pkg] Detecting vulnerabilities...
2024-09-24T09:50:58+02:00       WARN    Using severities from other vendors for some vulnerabilities. Read https://aquasecurity.github.io/trivy/v0.55/docs/scanner
/vulnerability#severity-selection for details.

ghcr.io/chainloop-dev/platform/frontend:v0.87.16 (wolfi 20230201)

Total: 0 (MEDIUM: 0, HIGH: 0, CRITICAL: 0)


app/node_modules/@esbuild/linux-x64/bin/esbuild (gobinary)

Total: 3 (MEDIUM: 2, HIGH: 1, CRITICAL: 0)
```

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
stdlib	CVE-2024-34156	HIGH	fixed	1.22.5	1.22.7, 1.23.1	encoding/gob: golang: Calling Decoder.Decode on a message which contains deeply nested structures... https://avd.aquasec.com/nvd/cve-2024-34156
	CVE-2024-34155	MEDIUM				go/parser: golang: Calling any of the Parse functions containing deeply nested literals... https://avd.aquasec.com/nvd/cve-2024-34155
	CVE-2024-34158					go/build/constraint: golang: Calling Parse on a "// +build" build tag line with... https://avd.aquasec.com/nvd/cve-2024-34158

Sharing SBOMs

Problem: Automated tamper resistant, authenticated, controlled SBOM sharing

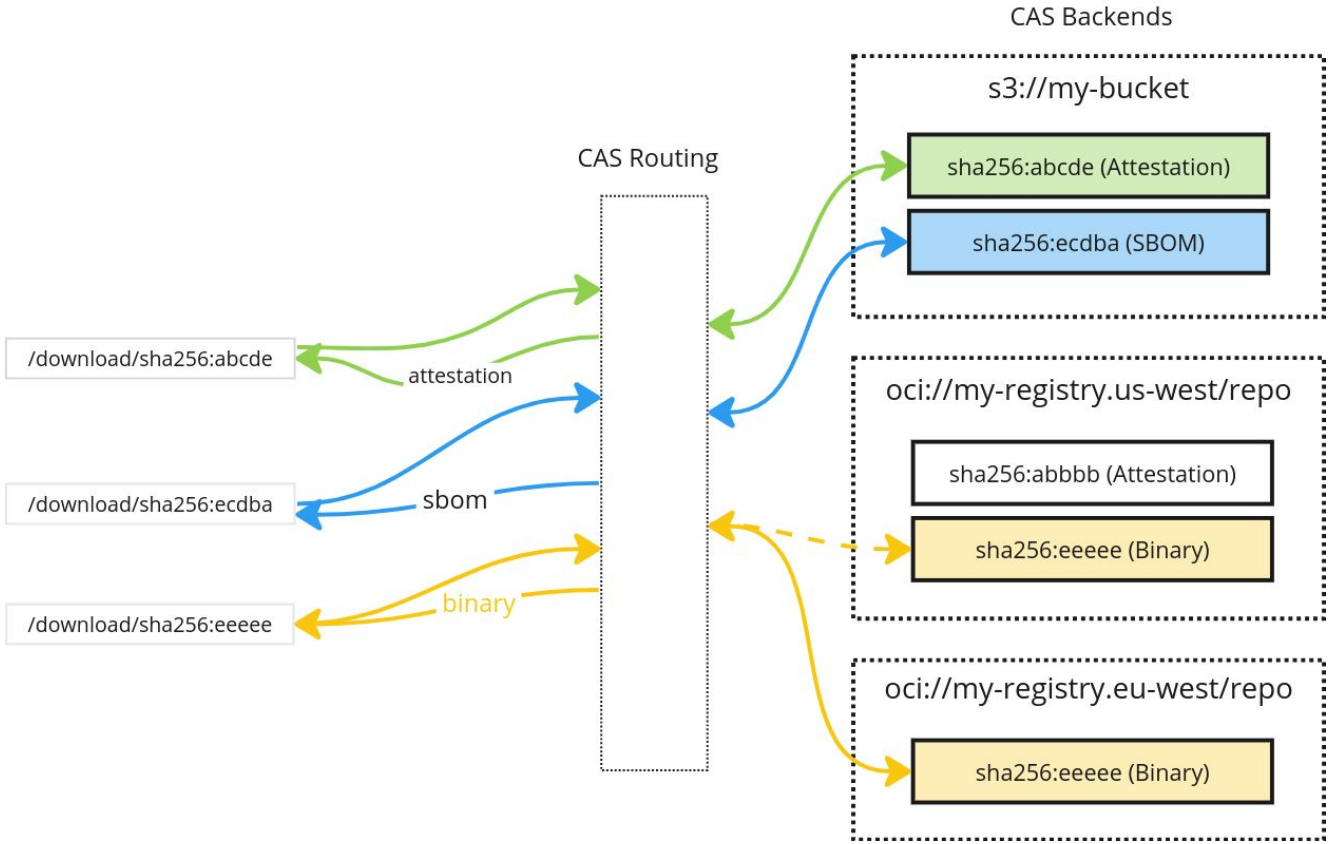
Sharing SBOMs

Manual, custom-made processes

servicenow

Sharing SBOMs

- **Integrity -** Signed/Attested SBOM
- **Availability -** Federated Content-Addressable Storage (CAS)
- **Authorization** - Artifact RBAC
- **Discoverability** (IPFs? TEA? [1])
- **Accessibility** (Sharepoint)



- https://docs.chainloop.dev/reference/operator/cas-backend
- [1] https://github.com/CycloneDX/transparency-exchange-api

Sharing SBOMs

Attestation information

Runner Type
Github Action

Status
Success

Duration
6 minutes 26 seconds

Started
03/13/2024, 1:38 PM

Finished
03/13/2024, 1:45 PM

Attestation Hash
sha256:a016051c4a2ca8ce307125b90cfdbcf5b96fad17cf9d2ab408a0375fec17abcc

Summary | Contract | Statement | Envelope

Materials

Name	Type	Value
artifact-cas-image	CONTAINER_IMAGE	ghcr.io/chainloop-dev/chainloop/artifact-cas@sha256:33eccf31862ebf7339acce9f0a29200b394
artifact-cas-linux-amd64	ARTIFACT	artifact-cas
chainloop-plugin-discord-webhook-linux-amd64	ARTIFACT	chainloop-plugin-discord-webhook
cli-linux-amd64	ARTIFACT	chainloop
control-plane-image	CONTAINER_IMAGE	ghcr.io/chainloop-dev/chainloop/control-plane@sha256:b9c97de01221658c9c89012d806e670992efe23954882c6dfa0a05e2d9f4b099
control-plane-linux-amd64	ARTIFACT	control-plane
sbom-artifact-cas	SBOM_CYCLONEDX_JSON	sbom.cas.cyclonedx.json

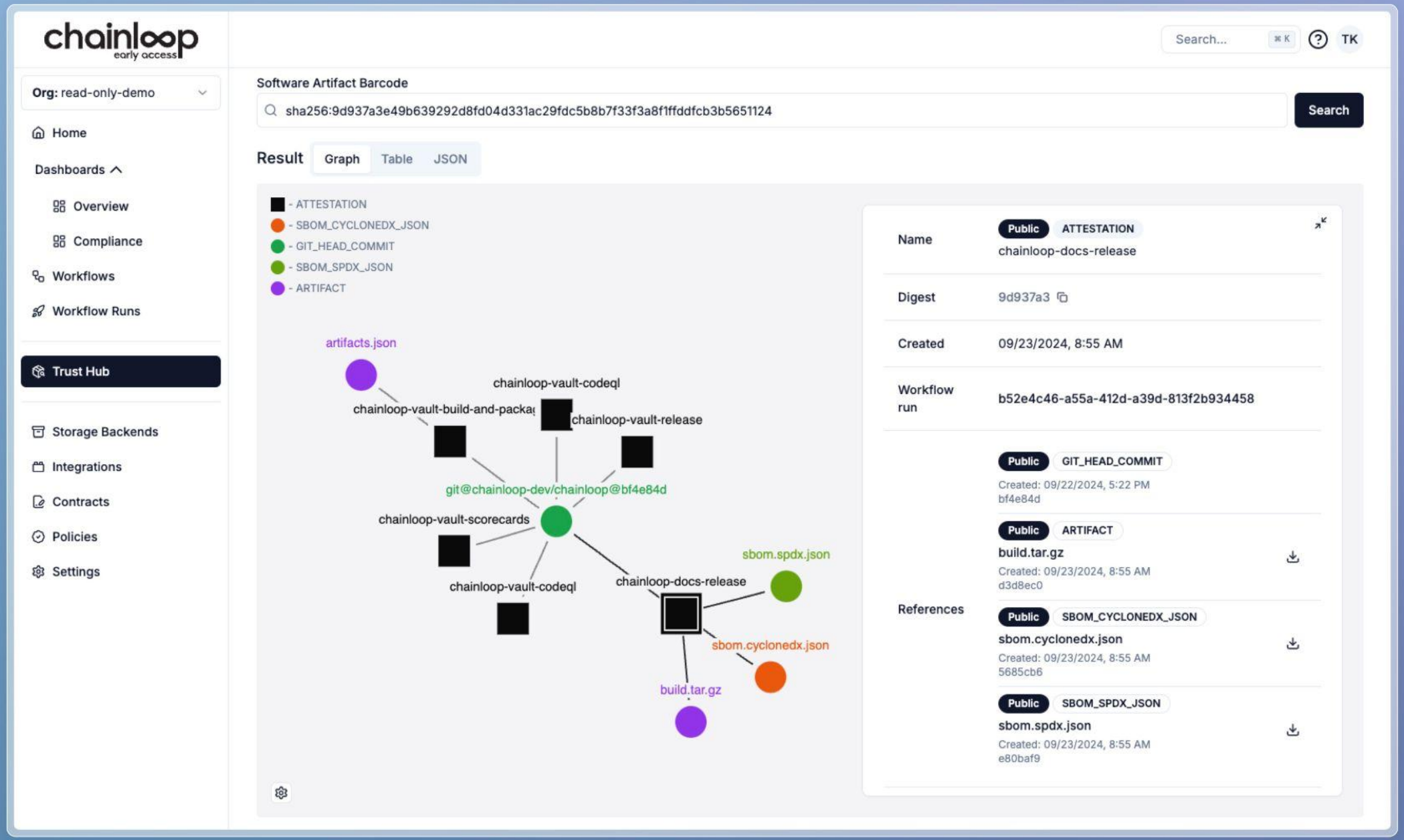
Control Gates

Problem: how to programmatically make decisions in different steps of SDLC

Control Gates

End Of Life (EOL)

Block With Confidence


chainloop
early access
Search...
⌘ K
TK
Org: read-only-demo
Home
Dashboards
Overview
Compliance
Workflows
Workflow Runs
Trust Hub
Storage Backends
Integrations
Contracts
Policies
Settings
Software Artifact Barcode
sha256:9d937a3e49b639292d8fd04d331ac29fdc5b8b7f33f3a8f1ffddfcb3b5651124
Search
Result
Graph
Table
JSON
- ATTESTATION
- SBOM_CYCLONEDX_JSON
- GIT_HEAD_COMMIT
- SBOM_SPDX_JSON
- ARTIFACT
artifacts.json
chainloop-vault-codeql
chainloop-vault-build-and-packa
chainloop-vault-release
git@chainloop-dev/chainloop@bf4e84d
chainloop-vault-scorecards
chainloop-vault-codeql
chainloop-docs-release
sbom.spdx.json
sbom.cyclonedx.json
build.tar.gz
Name
Public
ATTESTATION
chainloop-docs-release
Digest
9d937a3
Created
09/23/2024, 8:55 AM
Workflow run
b52e4c46-a55a-412d-a39d-813f2b934458
References
Public
GIT_HEAD_COMMIT
Created: 09/22/2024, 5:22 PM
bf4e84d
Public
ARTIFACT
build.tar.gz
Created: 09/23/2024, 8:55 AM
d3d8ec0
Public
SBOM_CYCLONEDX_JSON
sbom.cyclonedx.json
Created: 09/23/2024, 8:55 AM
5685cb6
Public
SBOM_SPDX_JSON
sbom.spdx.json
Created: 09/23/2024, 8:55 AM
e80baf9

And more....

SBOMS are just one example of metadata, see SARIF, CSAF, ...

Foundation for future use-cases

metadata gathering for SOC2, ISO27001, ...

Sharing SBOMs

Control Gates

Continuous Assurance

Metadata Trust

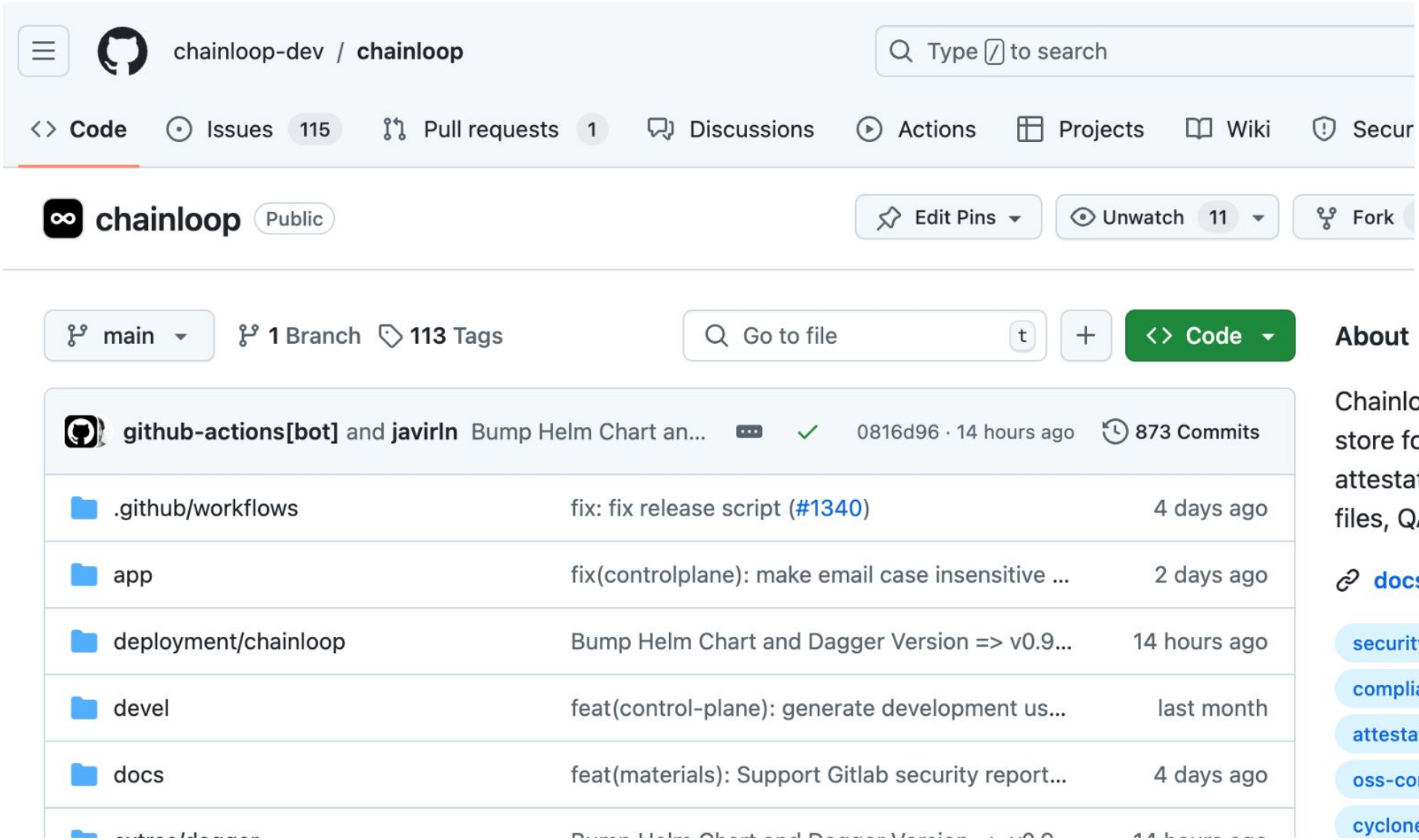
Continuous Compliance

Metadata

SBOM, *VEX discovery

Automated Trust

Remember - You can try this at home



Thank you

Find us in Slack

- **@migmartri, @danliska**

chainloop-dev/chainloop

☆ if you like what we do, give our GitHub **chainloop-dev/chainloop** a star :) ☆