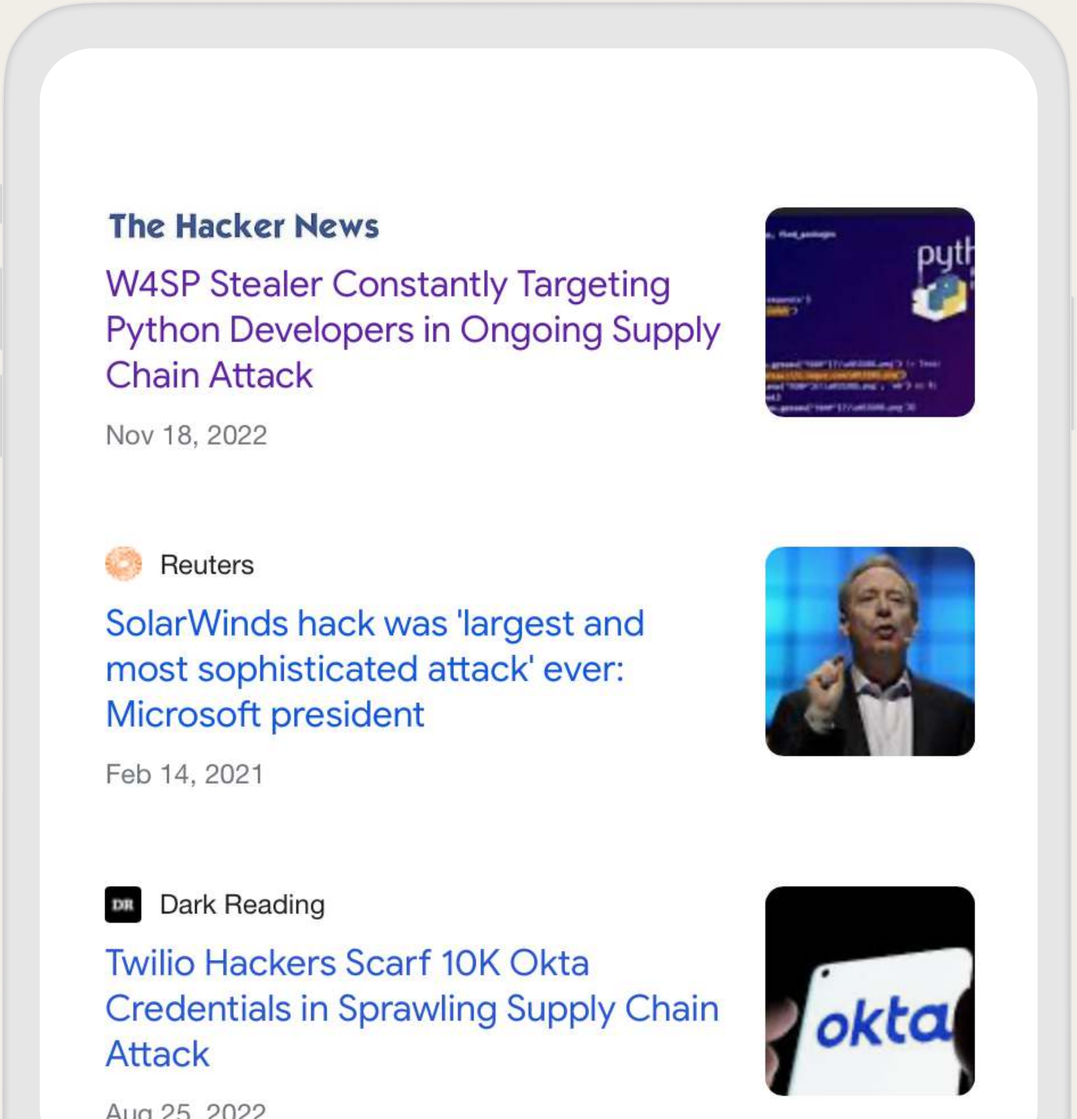
Kubecon EU 2023

Silly gooses,
let's make sense of the security supply chain,
together

Grace Nguyen

- Security issues introduced by the **third-party** components and technologies
- **700% increase** in attacks over 3 years
- Executive Order 14028



UNIVERSITY OF WATERLOO

Canada Revenue Agency

ERICSSON

Riskfuel

cisco Meraki

asana

Expo

- Kubernetes release team since v1.22
- Release Manager Associate v1.27
- v1.28 Release Lead

*not an expert

UNIVERSITY OF WATERLOO

Canada Revenue Agency

ERICSSON

Riskfuel

cisco Meraki

asana

Expo

- Kubernetes release team since v1.22
- Release Manager Associate v1.27
- v1.28 Release Lead

Agenda

1. Intro ✅
2. Frameworks
3. Landscape and tools
4. Misconceptions and discussion
5. What’s next?

Agenda

1. Intro ✅
2. Frameworks
3. Landscape and tools
4. Misconceptions and discussion
5. What's next?

goal: to establish foundation and familiarity to software supply chain security

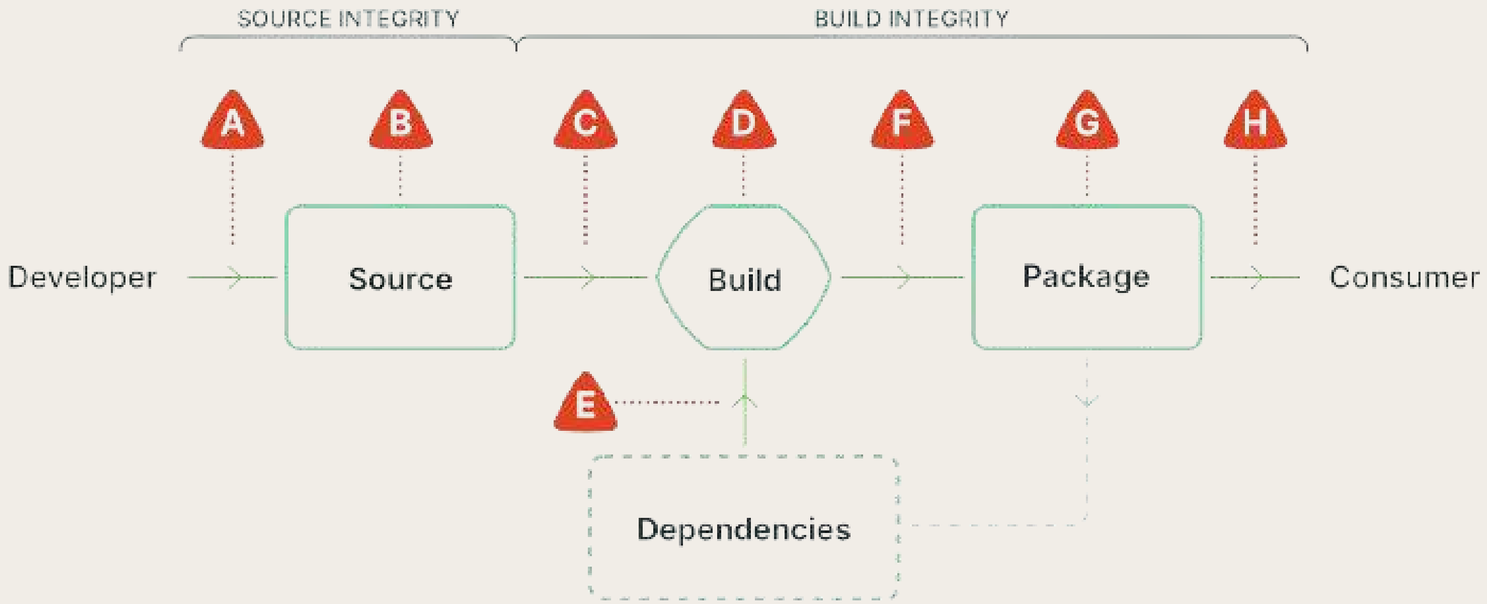
I excel at new, hard things

2. Framework

- ***Attestation***
- ***Provenance***
- ***SBOM***

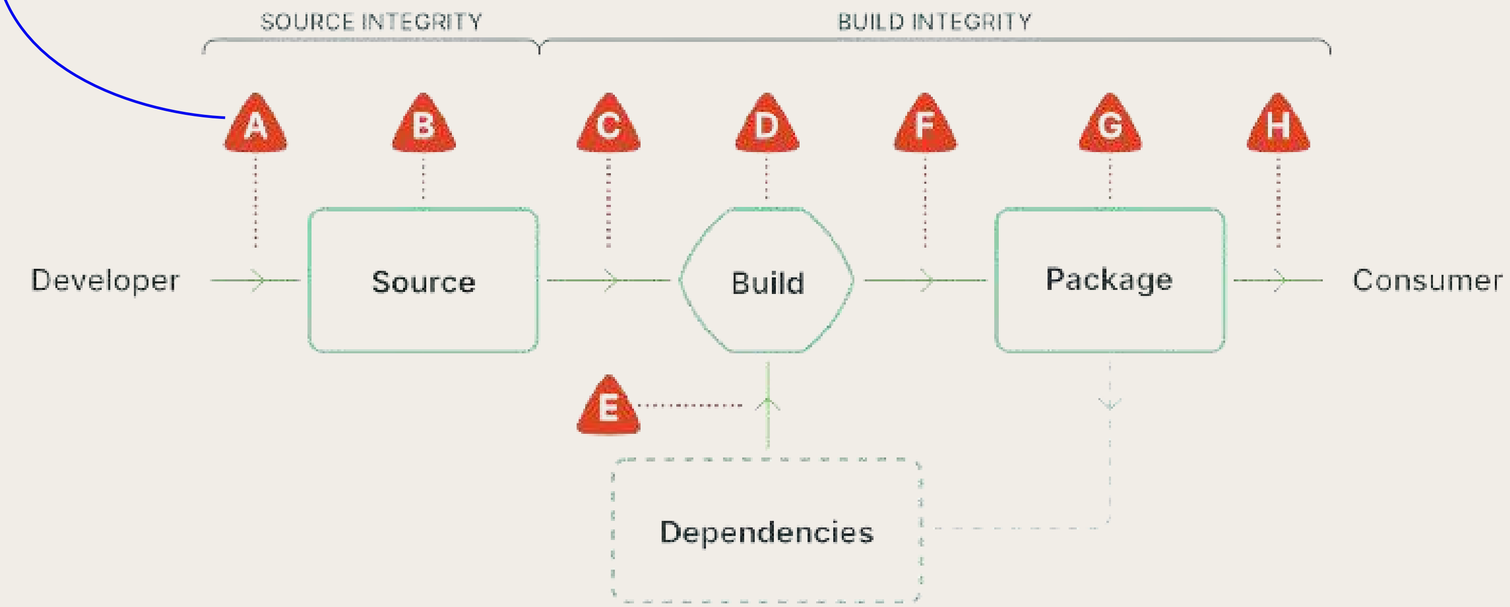
Level	Description	Example
1	Documentation of the build process	Unsigned provenance
2	Tamper resistance of the build service	Hosted source/build, signed provenance
3	Extra resistance to specific threats	Security controls on host, non-falsifiable provenance
4	Highest levels of confidence and trust	Two-party review + hermetic builds

from slsa.dev



from slsa.dev

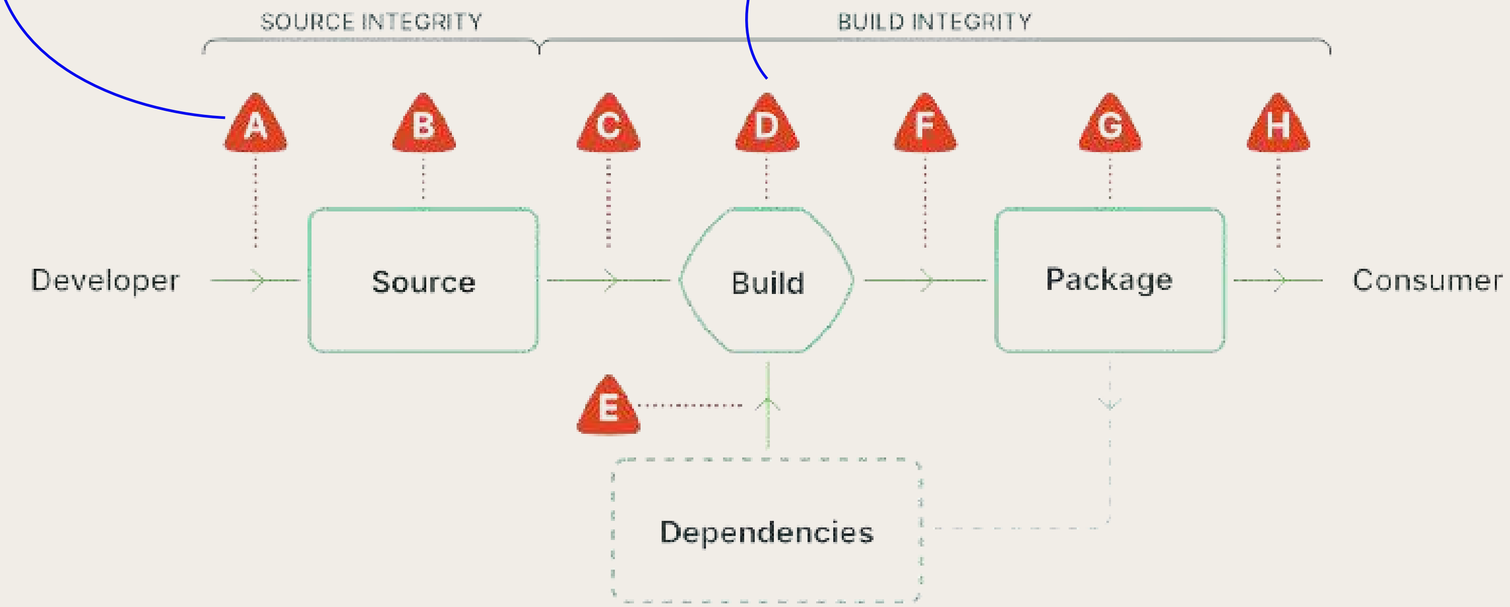
Linux hypocrite commits



from slsa.dev

Linux hypocrite commits

SolarWinds



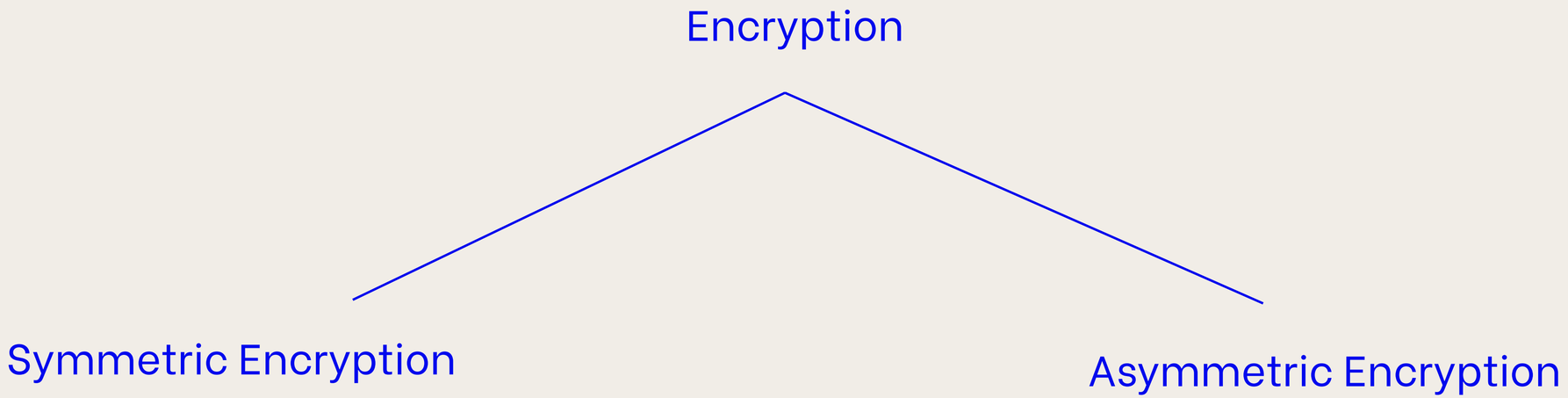
from slsa.dev

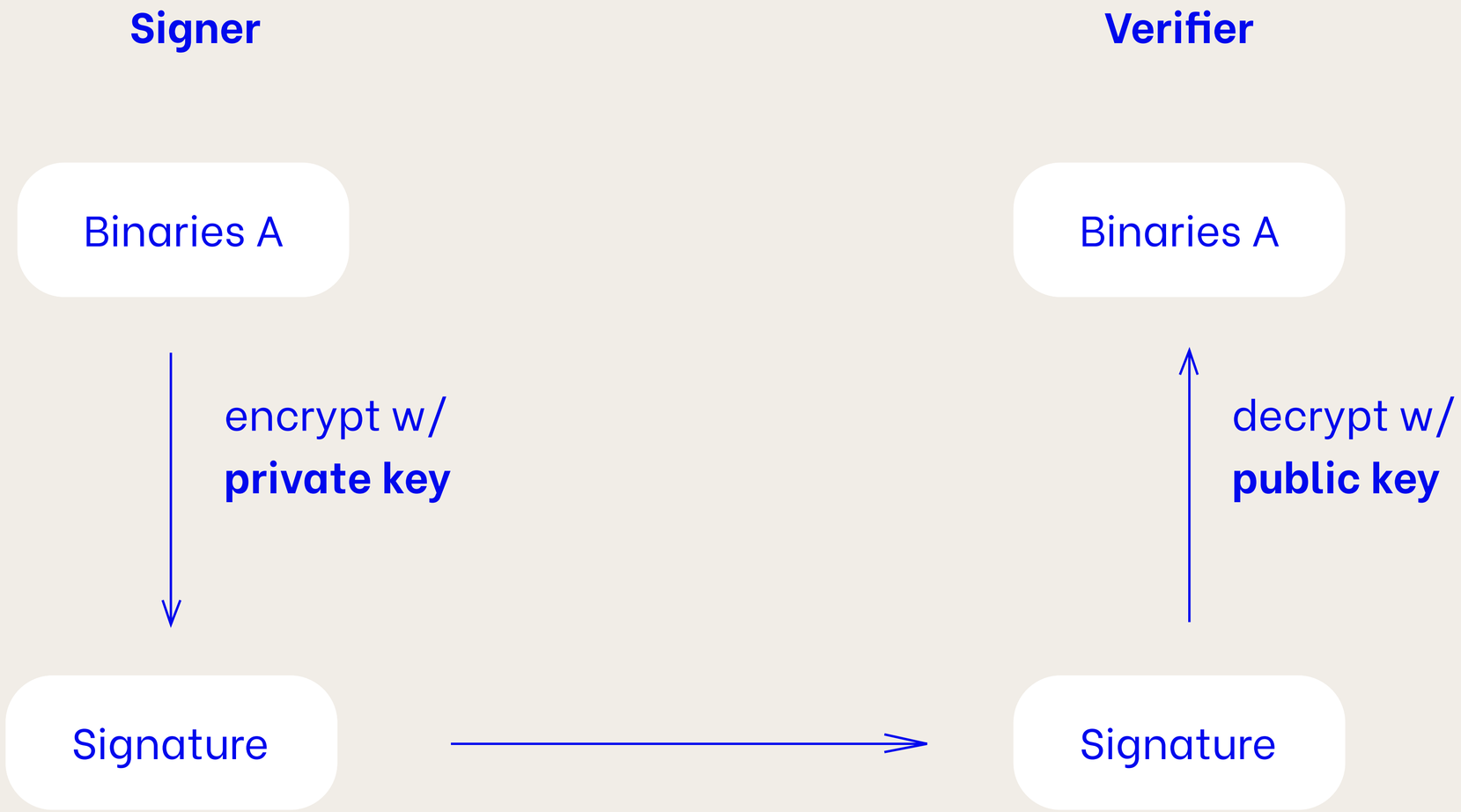
3. Landscape and tools

We can’t talk about securing the supply chain,
without understanding encryption

Assuming an understanding of encryption are leaving folks
out of the conversation

htttps://a-very-brief-attempt-at-teaching-encryption.pain


Encryption
Symmetric Encryption
Asymmetric Encryption


Signer
Verifier
Binaries A
Binaries A
encrypt w/
private key
decrypt w/
public key
Signature
Signature

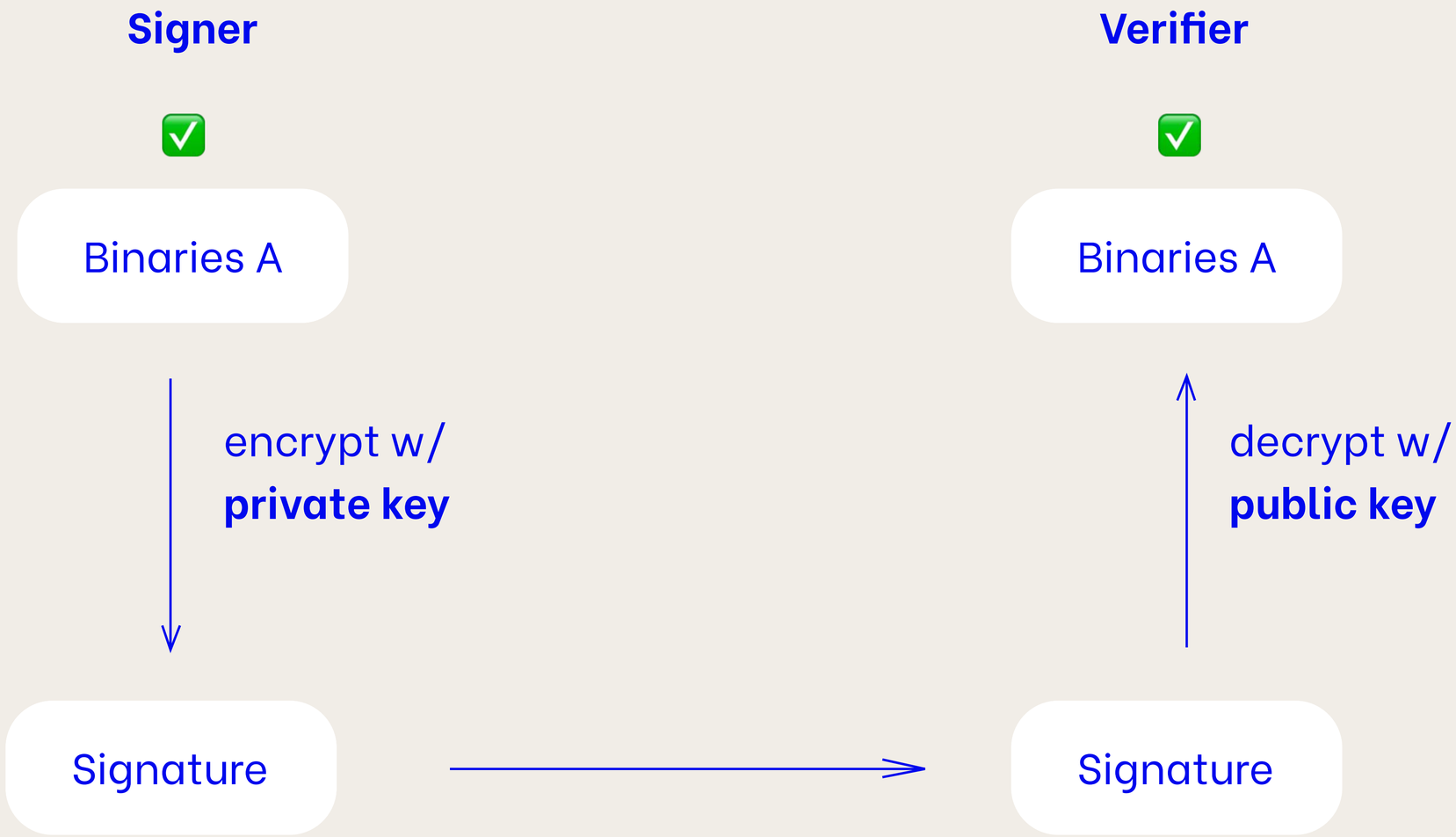
Certificate

Company

Public key

Signature

Binaries A

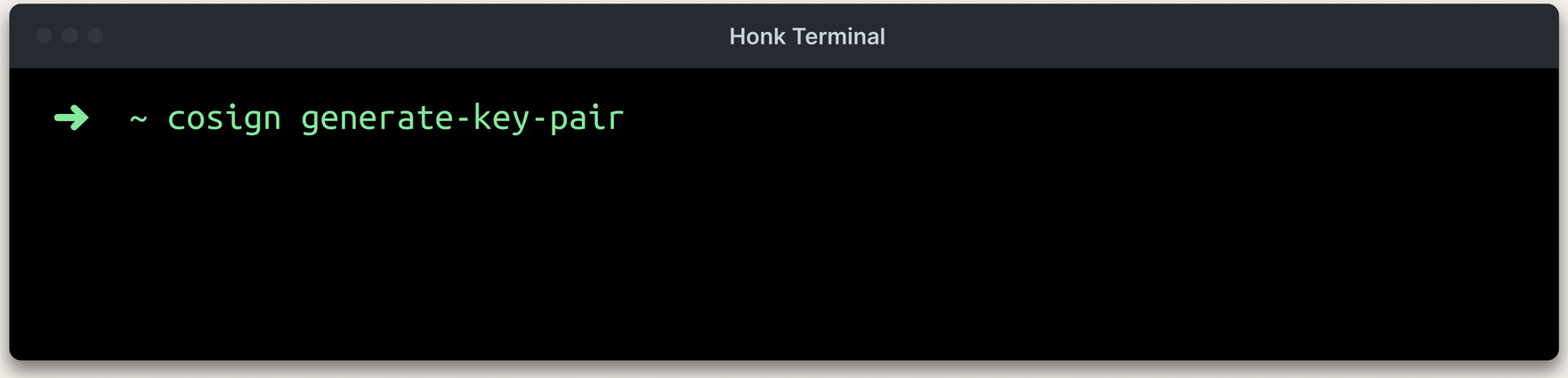

Signer
✅
Binaries A
encrypt w/
private key
Signature
Verifier
✅
Binaries A
decrypt w/
public key
Signature

Sigstore

- cosign: sign and verify software artifacts
- fulcio: root certificate authority
- rekor: transparency logs

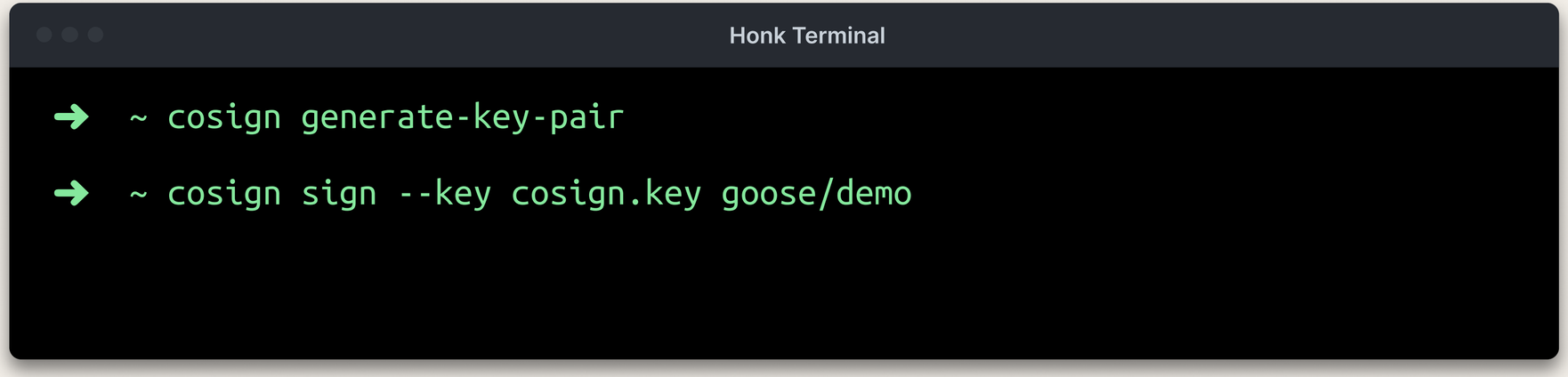
Cosign

- Sign and verify containers, standard files, blobs



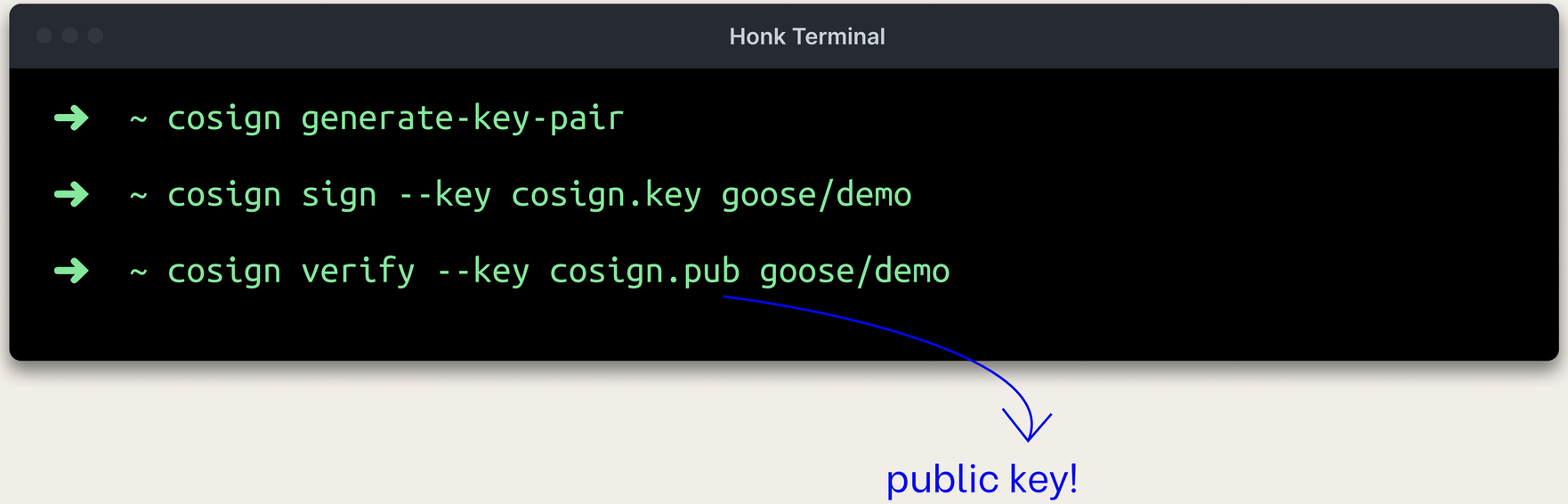
Cosign

- Sign and verify containers, standard files, blobs



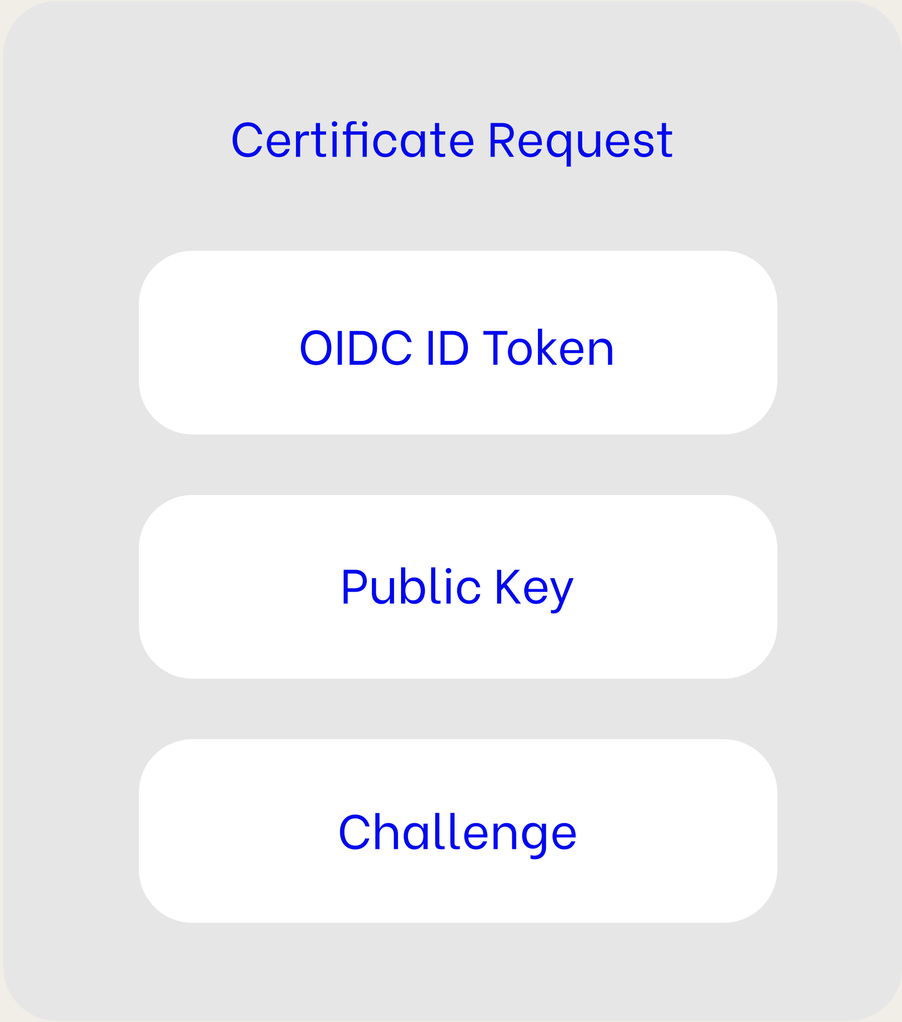
Cosign

- Sign and verify containers, standard files, blobs



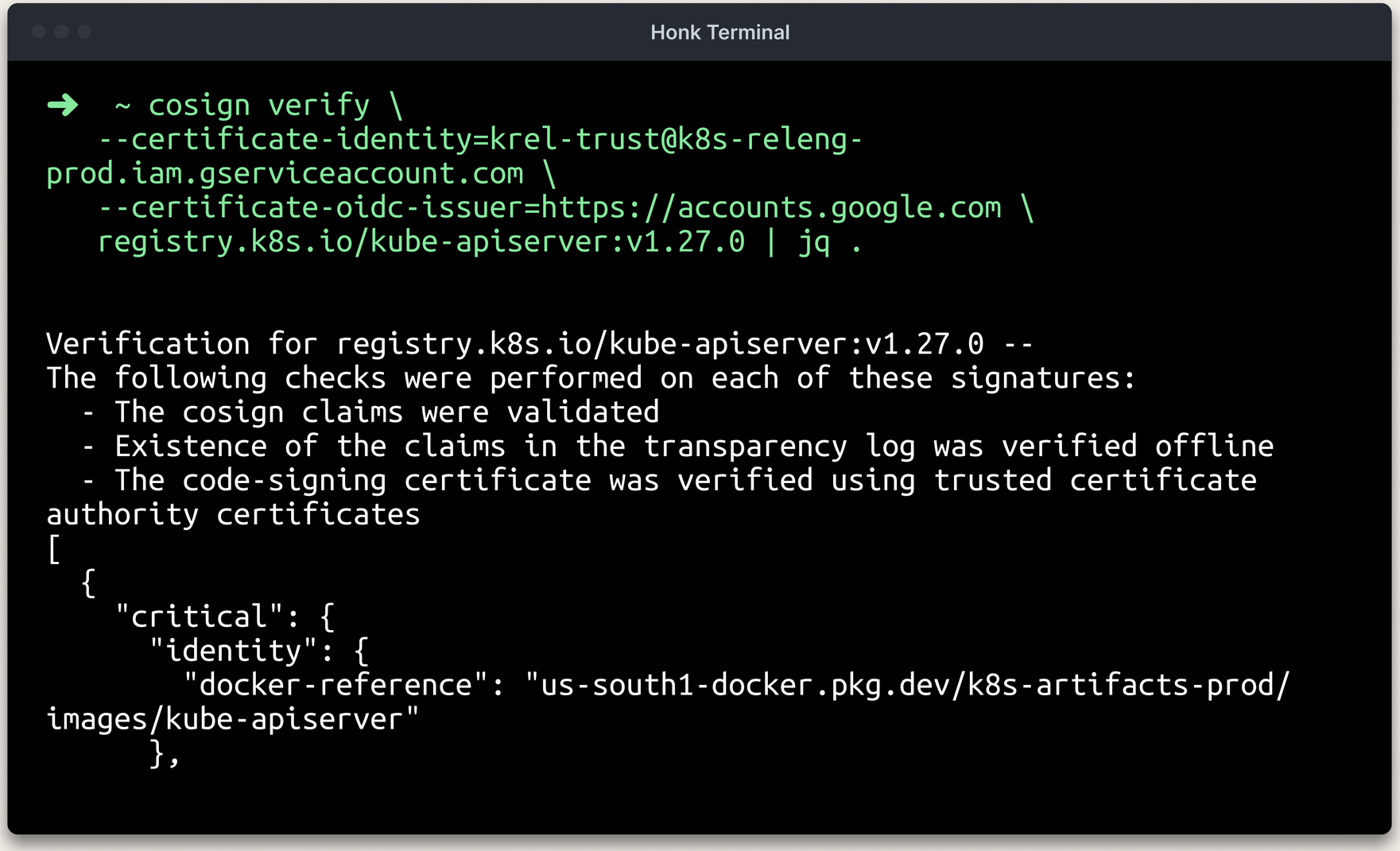
Fulcio

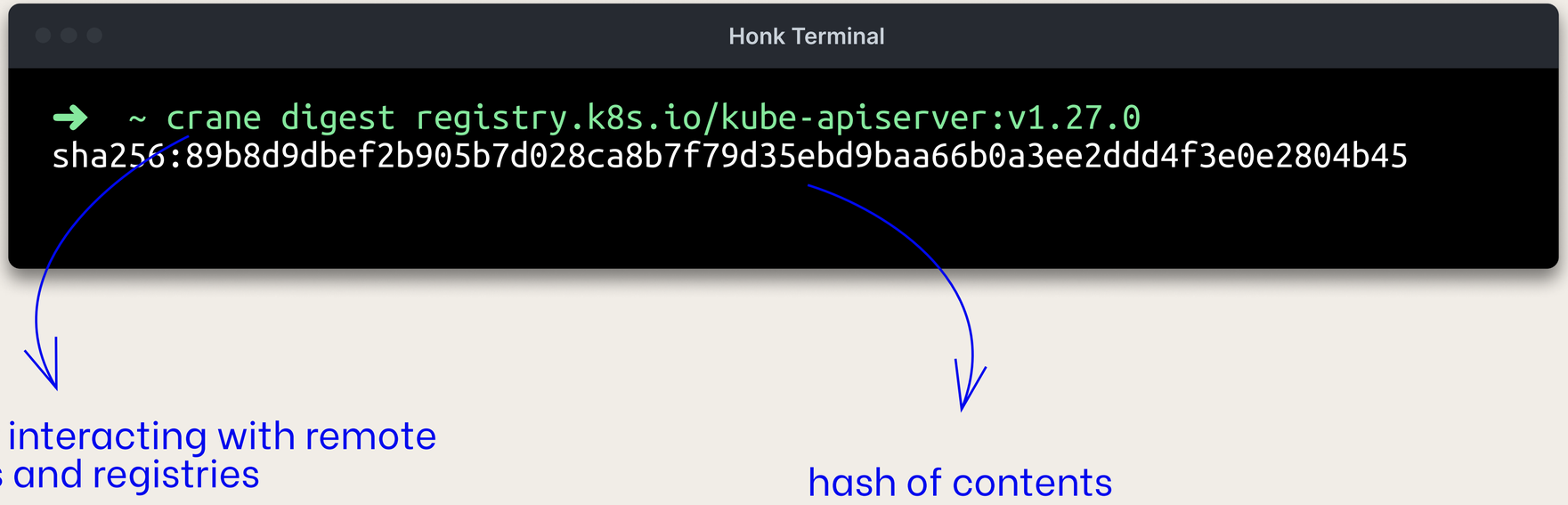
- Certificate Authority (CA)
- Short-lived and based on email address

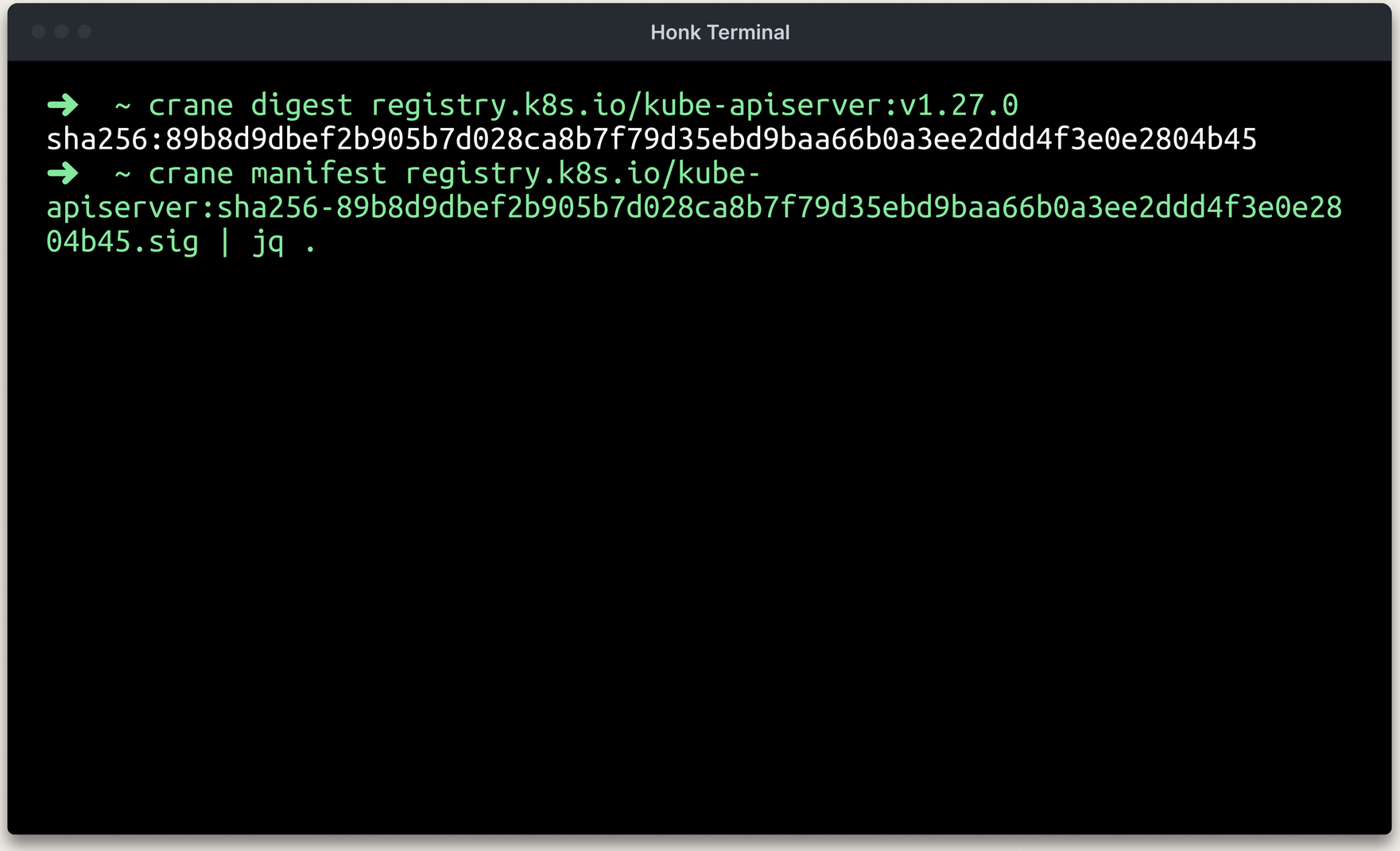


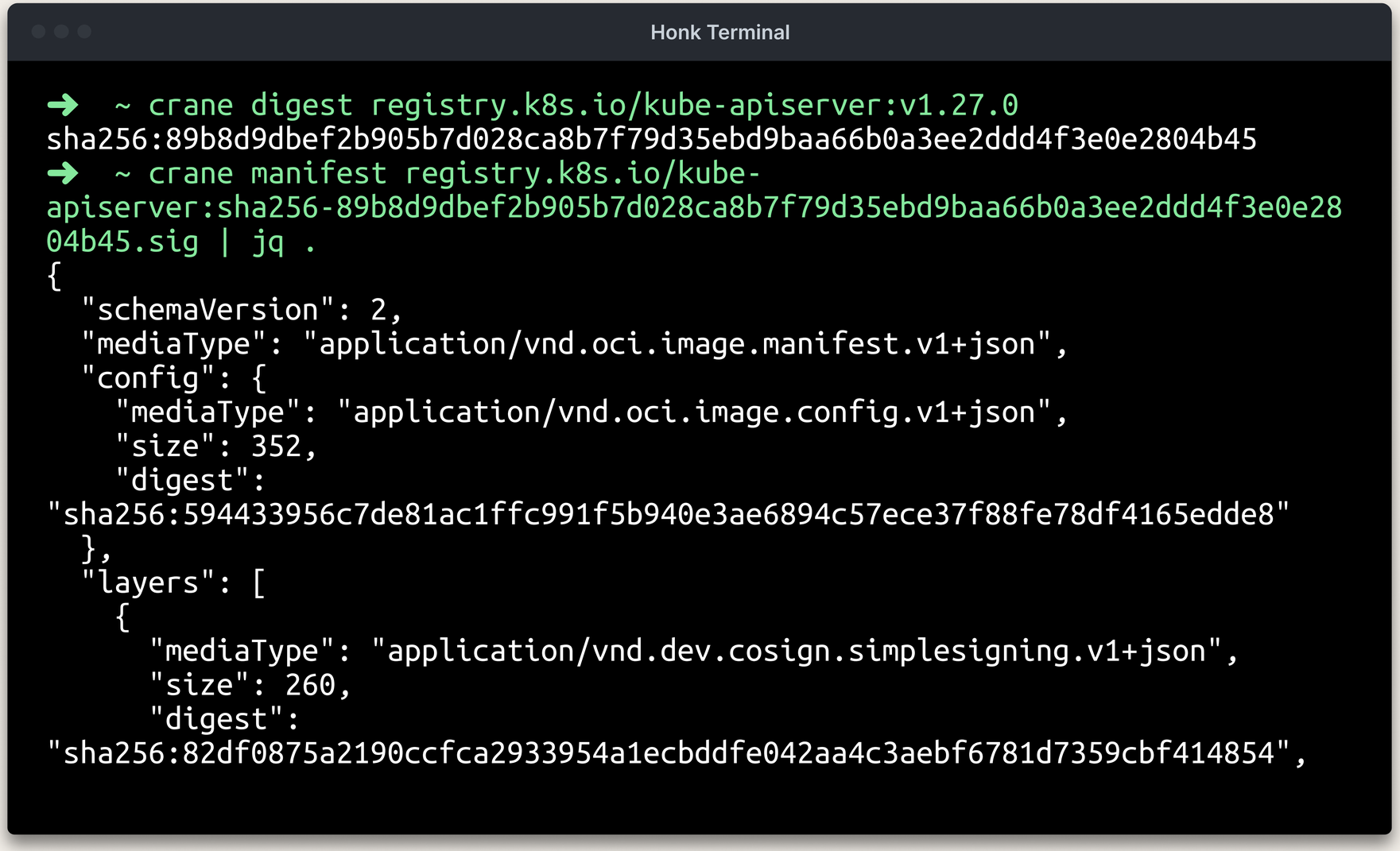
Rekor

- Tamper-proof log of metadata from the software supply chain
- Blockchain?

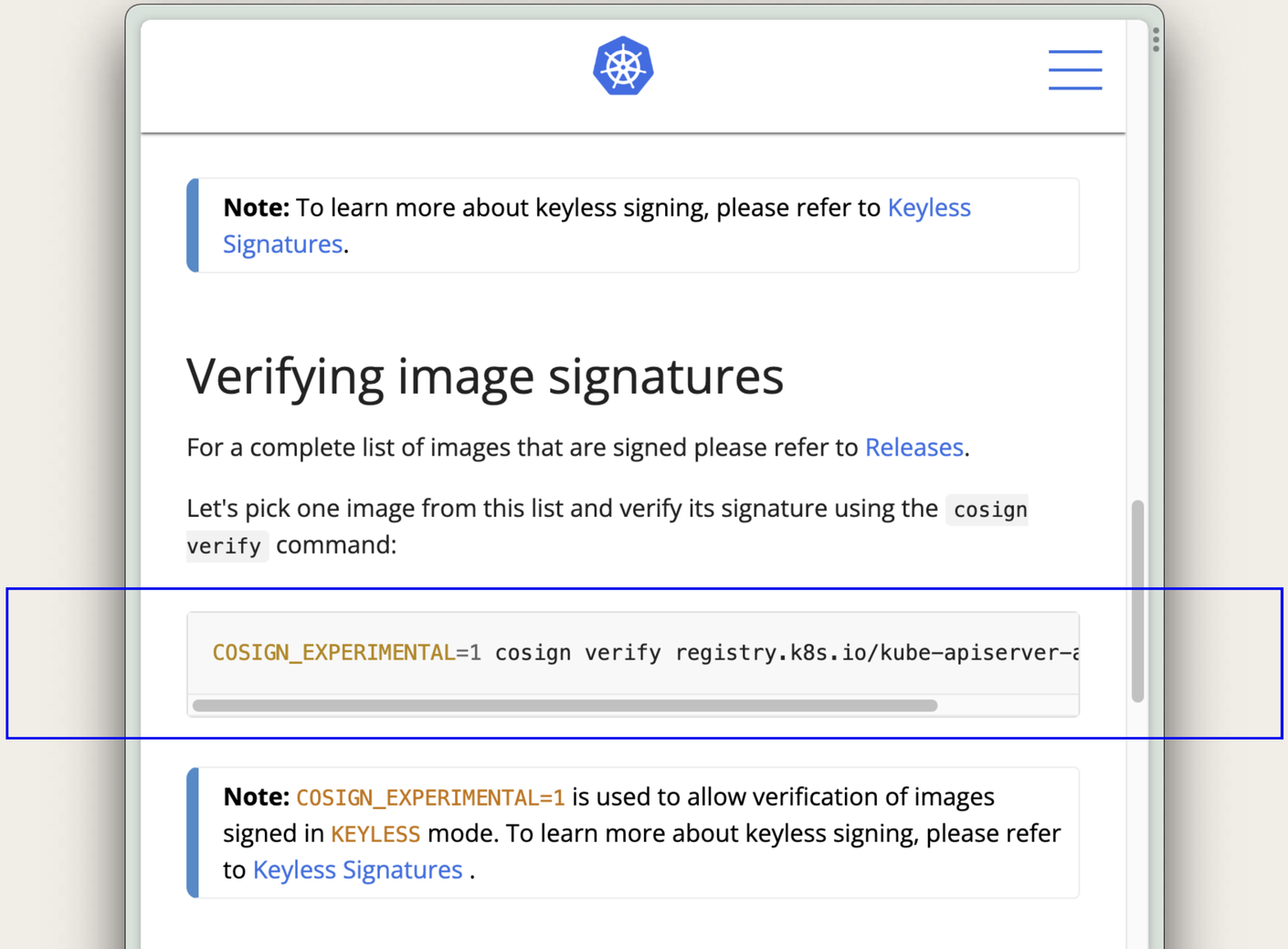

Honk Terminal
➜ ~ cosign verify \
--certificate-identity=krel-trust@k8s-releng-
prod.iam.gserviceaccount.com \
--certificate-oidc-issuer=https://accounts.google.com \
registry.k8s.io/kube-apiserver:v1.27.0 | jq .
Verification for registry.k8s.io/kube-apiserver:v1.27.0 --
The following checks were performed on each of these signatures:
- The cosign claims were validated
- Existence of the claims in the transparency log was verified offline
- The code-signing certificate was verified using trusted certificate
authority certificates
[
{
"critical": {
"identity": {
"docker-reference": "us-south1-docker.pkg.dev/k8s-artifacts-prod/
images/kube-apiserver"
},


Honk Terminal
➜ ~ crane digest registry.k8s.io/kube-apiserver:v1.27.0
sha256:89b8d9dbef2b905b7d028ca8b7f79d35ebd9baa66b0a3ee2ddd4f3e0e2804b45
tool for interacting with remote images and registries
hash of contents


Honk Terminal
➜ ~ crane digest registry.k8s.io/kube-apiserver:v1.27.0
sha256:89b8d9dbef2b905b7d028ca8b7f79d35ebd9baa66b0a3ee2ddd4f3e0e2804b45
➜ ~ crane manifest registry.k8s.io/kube-
apiserver:sha256-89b8d9dbef2b905b7d028ca8b7f79d35ebd9baa66b0a3ee2ddd4f3e0e28
04b45.sig | jq .


Honk Terminal
➜ ~ crane digest registry.k8s.io/kube-apiserver:v1.27.0
sha256:89b8d9dbef2b905b7d028ca8b7f79d35ebd9baa66b0a3ee2ddd4f3e0e2804b45
➜ ~ crane manifest registry.k8s.io/kube-apiserver:sha256-89b8d9dbef2b905b7d028ca8b7f79d35ebd9baa66b0a3ee2ddd4f3e0e2804b45.sig | jq .
{
"schemaVersion": 2,
"mediaType": "application/vnd.oci.image.manifest.v1+json",
"config": {
"mediaType": "application/vnd.oci.image.config.v1+json",
"size": 352,
"digest":
"sha256:594433956c7de81ac1ffc991f5b940e3ae6894c57ece37f88fe78df4165edde8"
},
"layers": [
{
"mediaType": "application/vnd.dev.cosign.simplesigning.v1+json",
"size": 260,
"digest":
"sha256:82df0875a2190ccfca2933954a1ecbddfe042aa4c3aebf6781d7359cbf414854",

(side notes uh oops docs are outdated for cosign 2.0, we're getting to it lol thanks)



Note: To learn more about keyless signing, please refer to Keyless Signatures.

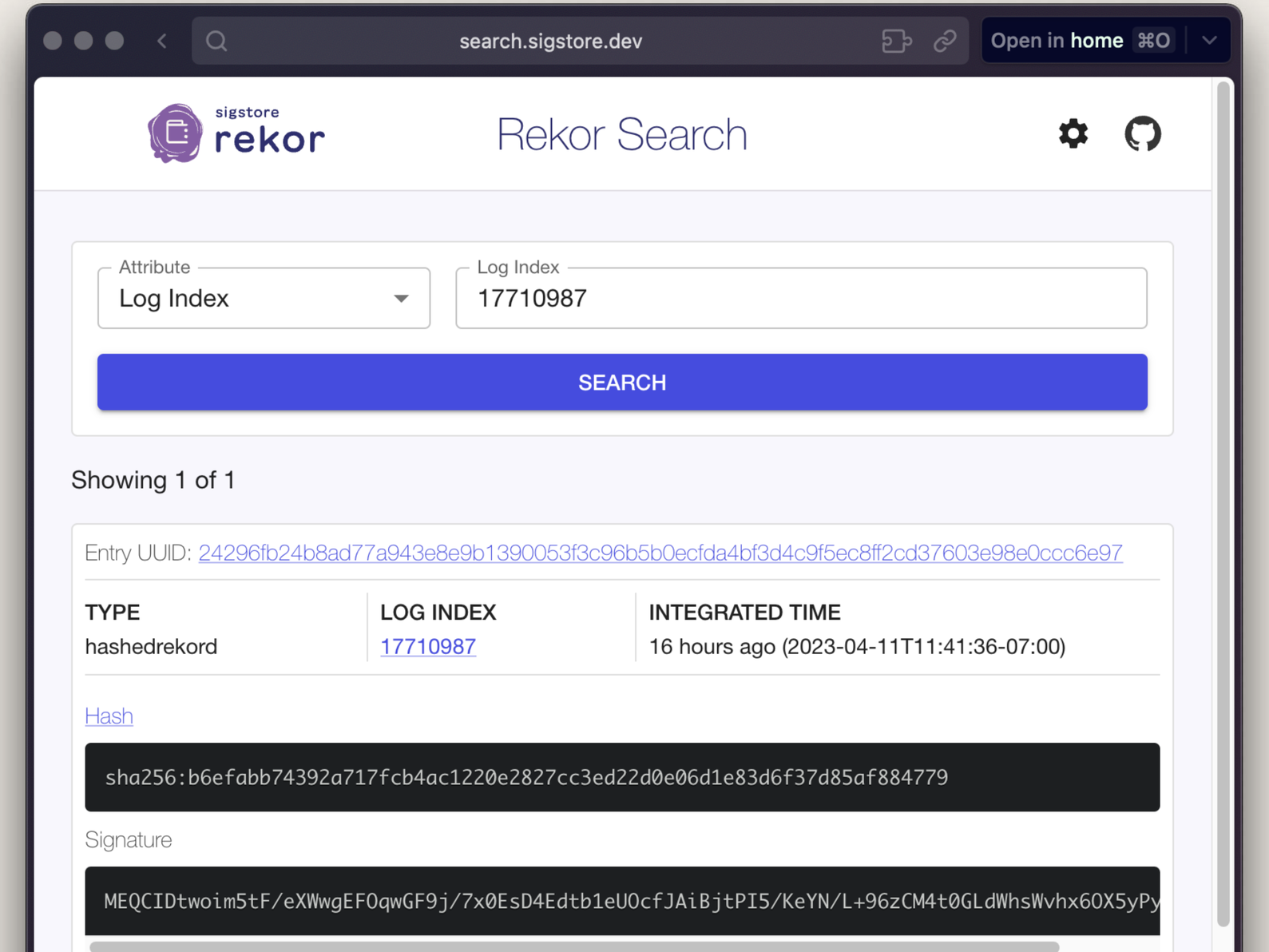
Verifying image signatures

For a complete list of images that are signed please refer to Releases.

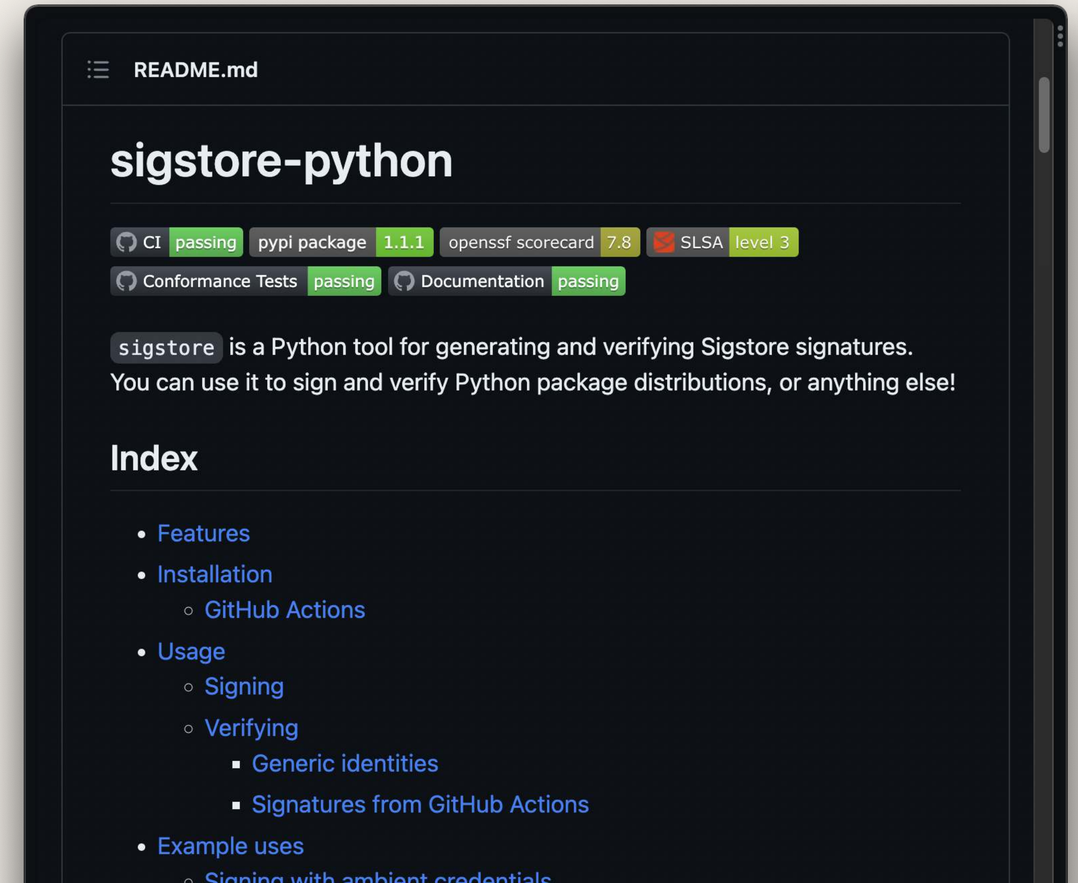
Let's pick one image from this list and verify its signature using the `cosign verify` command:

```
COSIGN_EXPERIMENTAL=1 cosign verify registry.k8s.io/kube-apiserver-a
```

Note: `COSIGN_EXPERIMENTAL=1` is used to allow verification of images signed in `KEYLESS` mode. To learn more about keyless signing, please refer to Keyless Signatures .


search.sigstore.dev
Open in home ⌘O
sigstore
rekor
Rekor Search
Attribute
Log Index
Log Index
17710987
SEARCH
Showing 1 of 1
Entry UUID: 24296fb24b8ad77a943e8e9b1390053f3c96b5b0ecfda4bf3d4c9f5ec8ff2cd37603e98e0ccc6e97
TYPE
hashedrekord
LOG INDEX
17710987
INTEGRATED TIME
16 hours ago (2023-04-11T11:41:36-07:00)
Hash
sha256:b6efabb74392a717fcb4ac1220e2827cc3ed22d0e06d1e83d6f37d85af884779
Signature
MEQCIDtwoim5tF/eXWwgEFOqwGF9j/7x0EsD4Edtb1eUOcfJAiBjtPI5/KeYN/L+96zCM4t0GLdWhsWvhx6OX5yPy

Sigstore Adoption



README.md

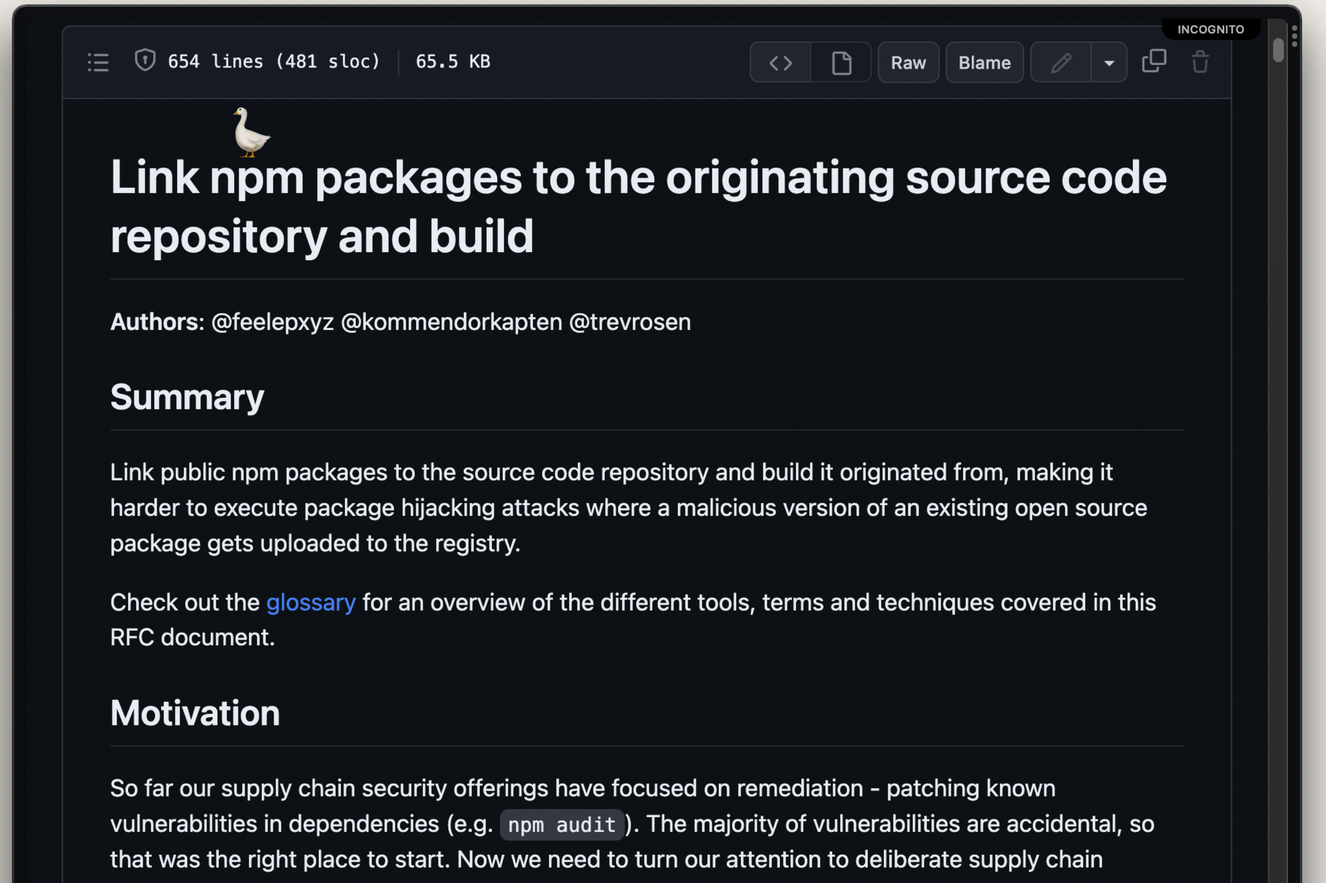
sigstore-python

CI passing | pypi package 1.1.1 | openssf scorecard 7.8 | SLSA level 3
Conformance Tests passing | Documentation passing

`sigstore` is a Python tool for generating and verifying Sigstore signatures.
You can use it to sign and verify Python package distributions, or anything else!

Index

- Features
- Installation
  - GitHub Actions
- Usage
  - Signing
  - Verifying
    - Generic identities
    - Signatures from GitHub Actions
- Example uses
  - Signing with ambient credentials



654 lines (481 sloc) | 65.5 KB | Raw | Blame | INCOGNITO

Link npm packages to the originating source code repository and build

Authors: @feelepxyz @kommendorkapten @trevrosen

Summary

Link public npm packages to the source code repository and build it originated from, making it harder to execute package hijacking attacks where a malicious version of an existing open source package gets uploaded to the registry.

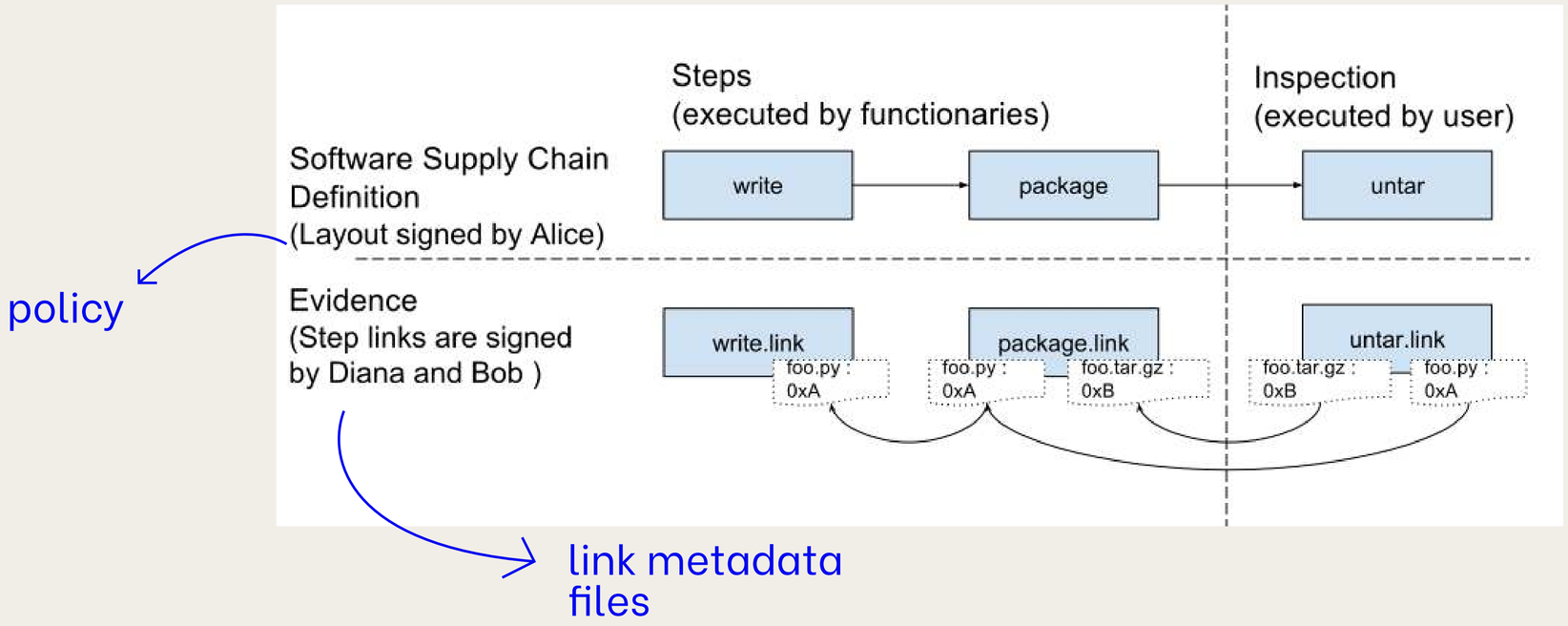
Check out the glossary for an overview of the different tools, terms and techniques covered in this RFC document.

Motivation

So far our supply chain security offerings have focused on remediation - patching known vulnerabilities in dependencies (e.g. `npm audit`). The majority of vulnerabilities are accidental, so that was the right place to start. Now we need to turn our attention to deliberate supply chain

In-toto

- Record what steps were performed, by whom and in what order
- Verify steps and its creator permissions



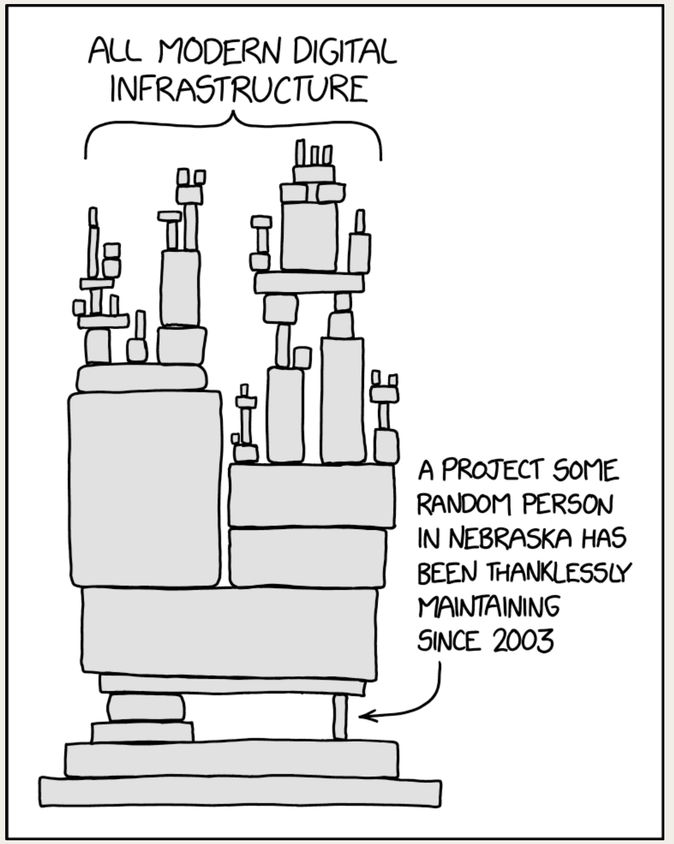
TUF

- Last mile – package distribution
- **Roles**:
  - root: specify other roles
  - targets: indicate packages validity
  - snapshot: provide package versions
  - timestamp: when package last updated
- Revocation + security compared to sigstore

4. Misconceptions & Discussion

1. It’s just about the code, right?
2. We’re probably not a good target
3. We can fully remove supply chain risks in our software

It’s a shared responsibility!



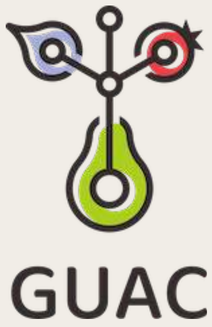
https://xkcd.com/2347/

What else? What's next?

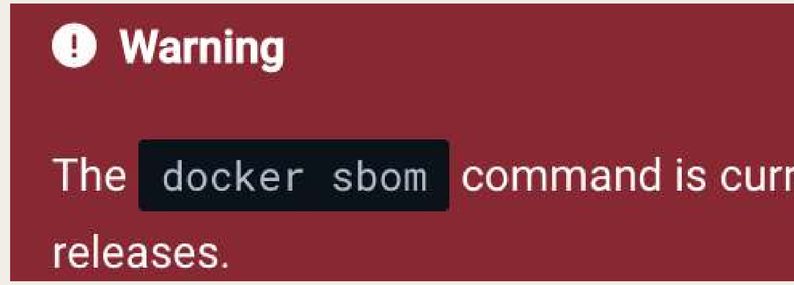
- Progress and new standards are being made *right now*
- Other components: vuln scanning, securing build enviroments...

What else? What's next?

- Progress and new standards are being made *right now*
- Other components: vuln scanning, securing build enviroments...



TACOS framework



Supply chain @ Kubecon

- Wed @ 15:25: From SBOMs to IBOMs - Know What's Happening in Your Clusters
- Thurs @ 15:25: In-Toto: Attestations and More for Software Supply Chain Security
- Thurs @ 15:25: Checking the Chains at the Gate: Building Supply Chain Policies with Gatekeeper and Ratify
- Thurs @ 16:30: Secure Your Project with the SIG Release Supply Chain Kit
- Fri @ 14:55: Building SLSA 3 Conforment Attestors for Artifacts Generated on GitHub

That's a wrap

Twitter @ gracenng

Hire me for security new-grad roles
@ May 2024 lol pls

https://www.youtube.com/watch?v=wwtL9iIM_rg&t=327s&ab_channel=CNCF%5BCloudNativeComputingFoundation%5D

https://blog.sigstore.dev/sigstore-blockchain-vs-transparency-logs-d673ea41a9be/

https://docs.google.com/spreadsheets/d/1CzvnInT7QOmTOz20W5TiX8tJiG9XZvdqYA3TivLx-PI/edit#gid=0

asdfg

https://www.sonatype.com/state-of-the-software-supply-chain/introduction

https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

https://docs.google.com/presentation/d/1MFNUVBas_PXT9AbrLIUOtqWPrtB-AJSI6EMncvjdtoE/edit#slide=id.g1251d519bb9_0_13