Google

A Retrospective on Google's SBOM Implementation

Brandon Lum (@lumjjb), Google
Marco Deicas (@mdeicas), Google

Part 1 recording:
https://drive.google.com/file/d/1wjuNEo7GwwZTehpcgVJaMI-8TdXkG2KF/view?usp=drivesdk

Feb 2025

1. How do we SBOM?
2. Lessons learnt from SBOM'ing

1. How do we SBOM?
2. Lessons learnt from SBOM'ing

Generate

Store

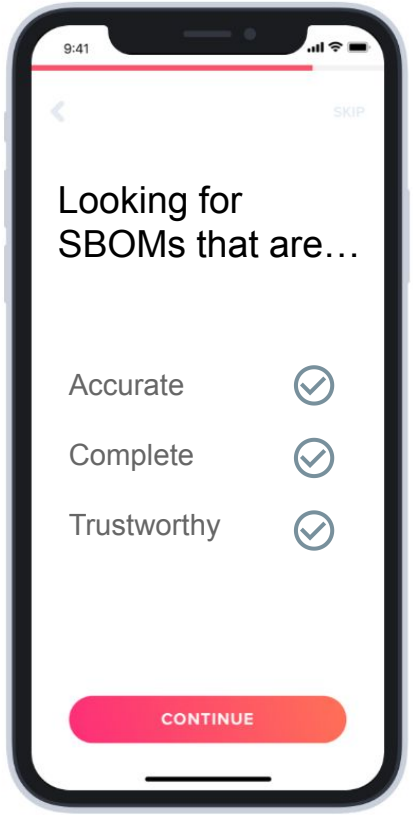
Retrieve

Applications

01

SBOM Design Principles

Where do we start? SBOMs?



What are properties of the SBOMs we want to strive for?

- Design doc with properties of SBOMs and best practices to achieve them
  - Properties
    - Accurate and complete
    - Trustworthy (Integrity & Provenance)

- Best practices: more throughout the talk!

Link to SBOM generation principles doc

Where do we start? YES

Opinionated or not? YES

- Problem scope is HUGE, many moving parts.

- Less is MORE
  - 1 standard: SPDX SBOMs
  - 1 storage and retrieval process
  - n builders << m products
  - etc.

SPDX

02

SBOM Generation

Generate

Source SBOMs??

Build SBOMs??

Analysis SBOMs?

Generate

Source SBOMs??

Build SBOMs??

Analysis SBOMs?

Source

Build

Artifact

- Includes tests and plugins
- Ambiguous dependency resolution

- Builds are lossy
- Loses context

Generate

Attaining good quality (accuracy and completeness):

- **BUILD-time whenever possible!! (extra credit for build tools)**

Source

Build

Artifact

Too much information (Inaccurate)

Too little information (Incomplete)

Generate: What we did

1. **Only build processes/builders can generate SBOMs**

2. **SBOM Generation tooling**
   a. Where possible, use build-tooling to generate SBOMs
      i. Android: dev/donation of SPDX Gradle plugin
      ii. Google3 (monorepo): Tooling leverages google3 metadata, annotations, and blaze
   b. Otherwise, use generic composition tooling (Syft, and internal version of osv-scalibr)

02

Store SBOM

Store

DATABASE !!
BLOB STORE!! WEB SCALE

Store

How do we create a SBOM database that we trust?

⇔

If we are using SBOMs to make security decisions, we need to trust them.

Store

Supply Chain Integrity Log
(SCILo)

For those familiar with
GUAC, it is similar
(& under the same team)

Store

I built an artifact, here's the build provenance to show it was securely built + build info

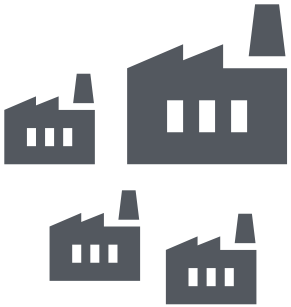
Signed by builder key

Great! This is a securely built artifact ("abcd..") by a trusted builder.

Builders

SCILo

Store



Builders

fe34..

in-toto

predicateType:
“ReferenceAttestation”

subject: {
...// software artifact hash
“sha256”:”abcd..”
}

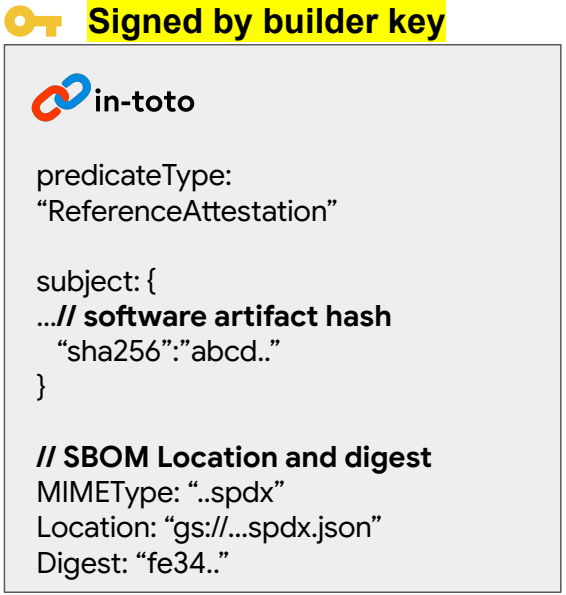
// SBOM Location and digest
MIMEType: “..spdx”
Location: “gs://...spdx.json”
Digest: “fe34..”

Great SBOM “fe34…” is for software “abcd..”.

SCILo

Link: [Intoto Reference Attestation]

Store



Great! I trust this builder’s SBOM generation process! Your SBOM is good!

Builders

SCILo

- SBOMs should be signed to ensure integrity
- The provenance of the SBOM should be accounted for

Store

SCILo

Artifact URI	Artifact Hash	SBOM
container_image://gcr.io/k8s	sha256:fefe..	/path/to/sbom-blob.spdx1.json
file://networkstore/somedir/binary	sha256:1234..	/path/to/sbom-blob.spdx2.json
container_image://staging.gcr.io/gke/abc	sha256:abcd..	/path/to/sbom-blob.spdx3.json
…	…	…

Great... This should be easy...

03

Retrieve SBOM

Retrieve: Ideal

<u>Want:</u>

Lookup
(“container_image://gcr.io/gke/abc”) =

Retrieve: Ideal

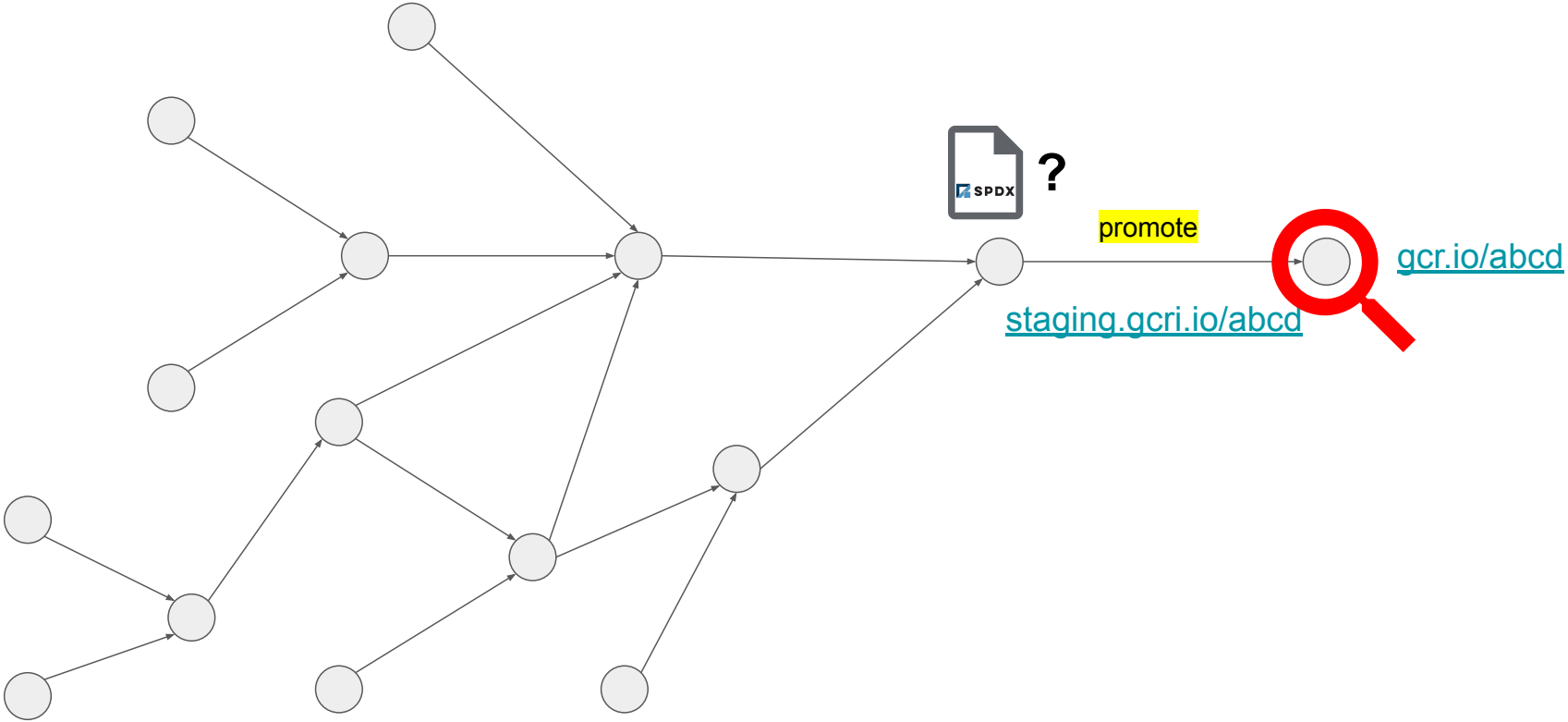
<u>Want:</u>

Lookup
(“container_image://gcr.io/gke/abc”) =

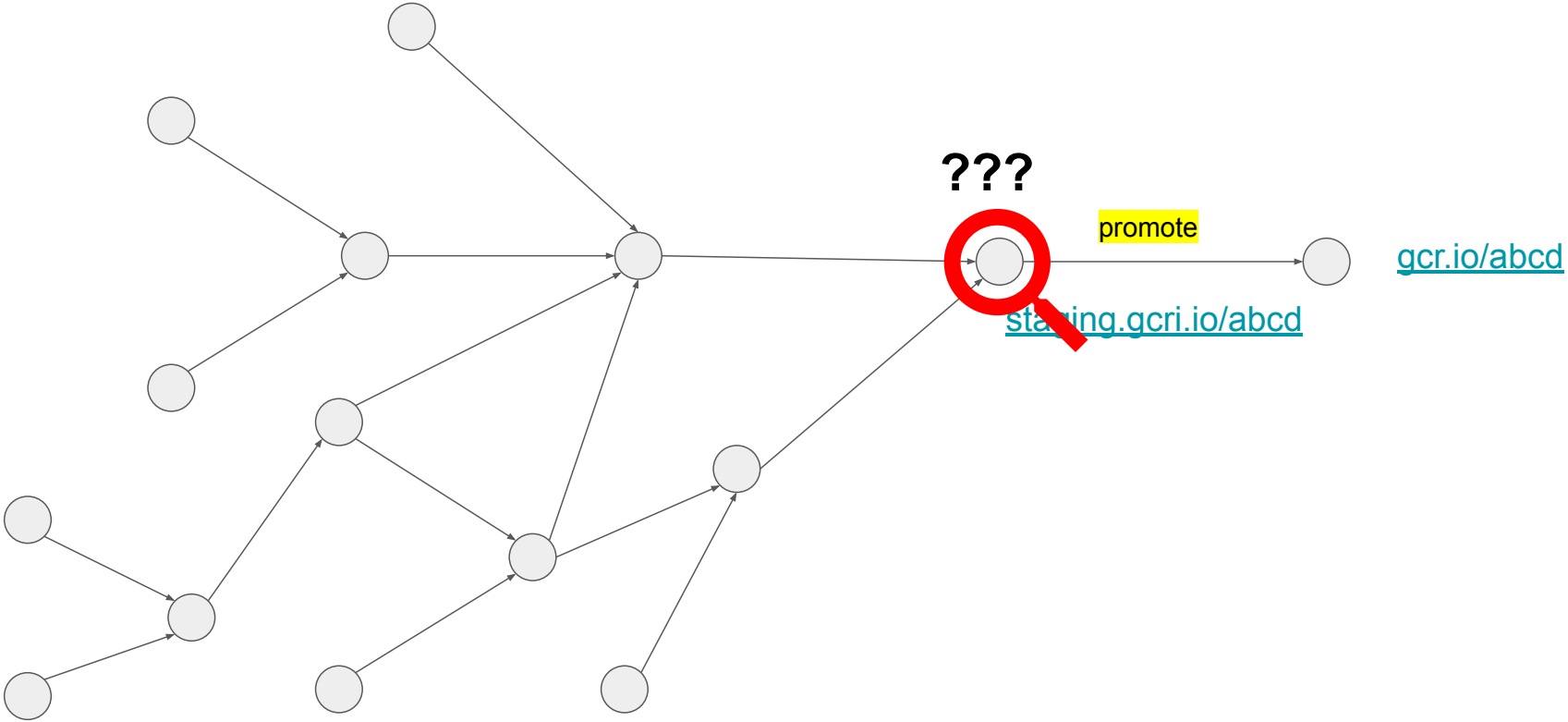
<u>Got:</u>

Lookup
(“container_image://gcr.io/gke/abc”) = NULL??

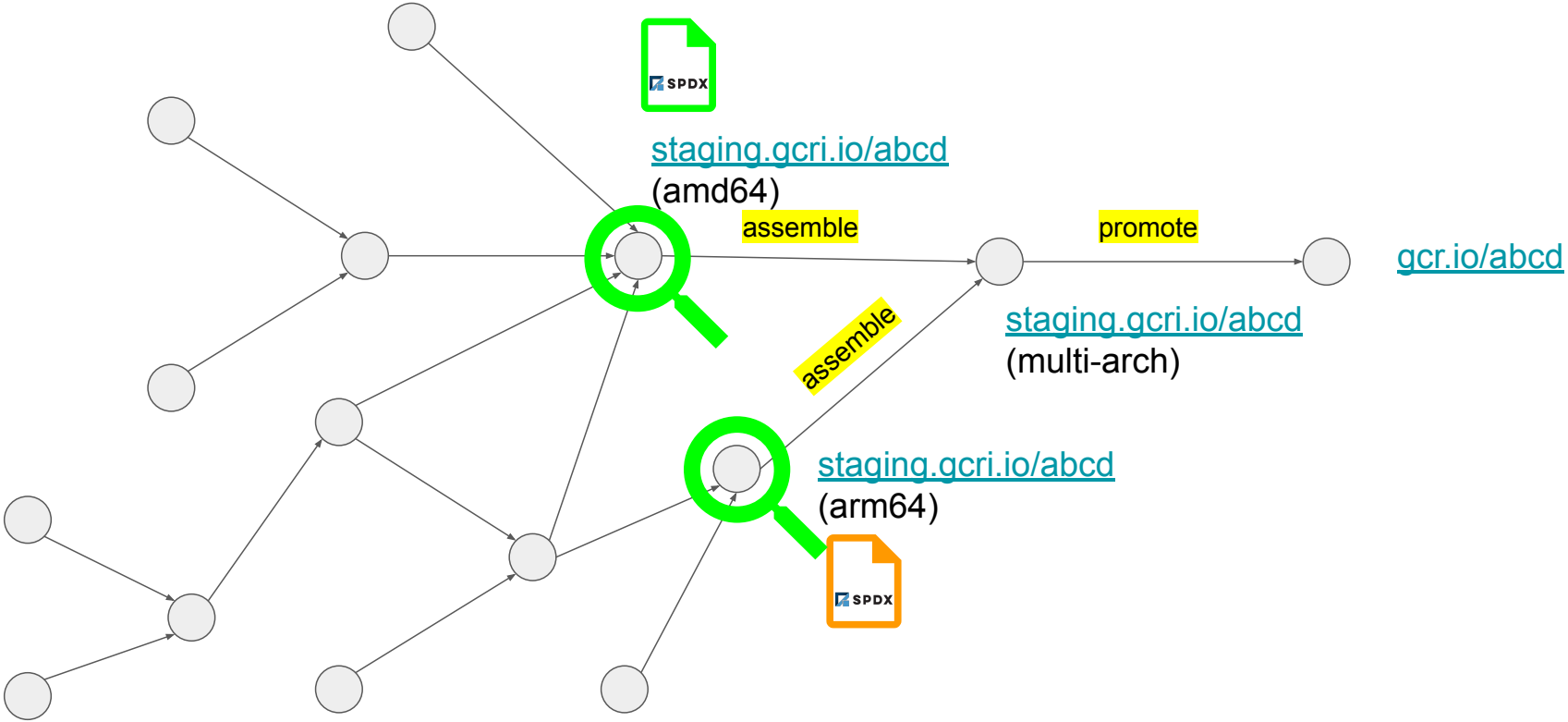
It’s a supply CHAIN....



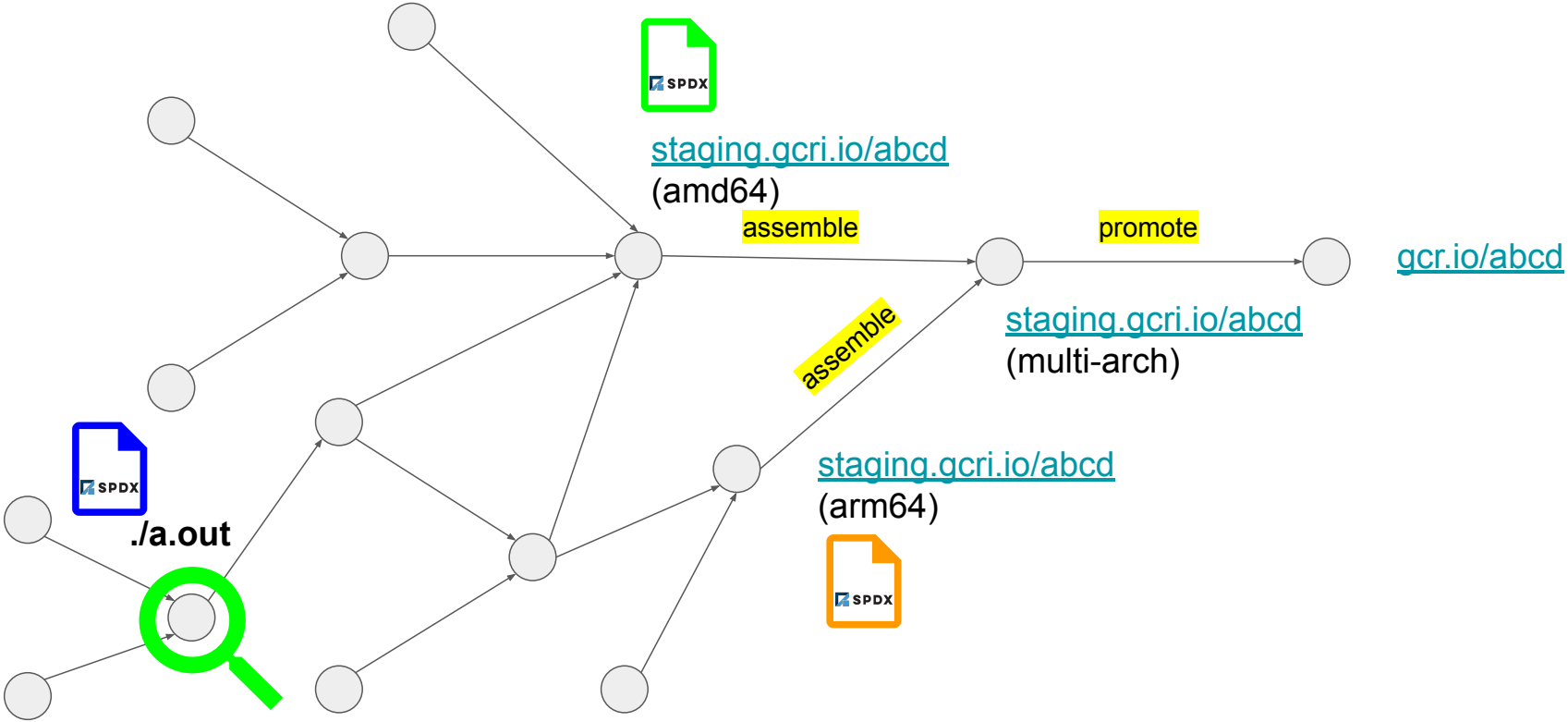
It’s a supply CHAIN....



It’s a supply CHAIN....



It’s a supply CHAIN....



Retrieve: Ideal

<u>Want:</u>

Lookup
(“container_image://gcr.io/gke/abc”) =

<u>Got:</u>

Lookup
(“container_image://gcr.io/gke/abc”) =

Retrieve: Edge cases

<u>“Edge” Cases</u>

- Container images manifest or config change (drift in hashes)
- Promoting images from staging to prod (change in reference)
- CI stages which result in change of hash (e.g. signing APKs)
- Inclusion of binaries which do not provide additional info
- Inclusion of binaries in packages that SCA tools can’t scan (e.g. installables)

Attaining good quality (accuracy and completeness):

- **Compose SBOMs to obtain a more complete SBOM**

WAIT!! What graph?

?????

gcr.io/abcd

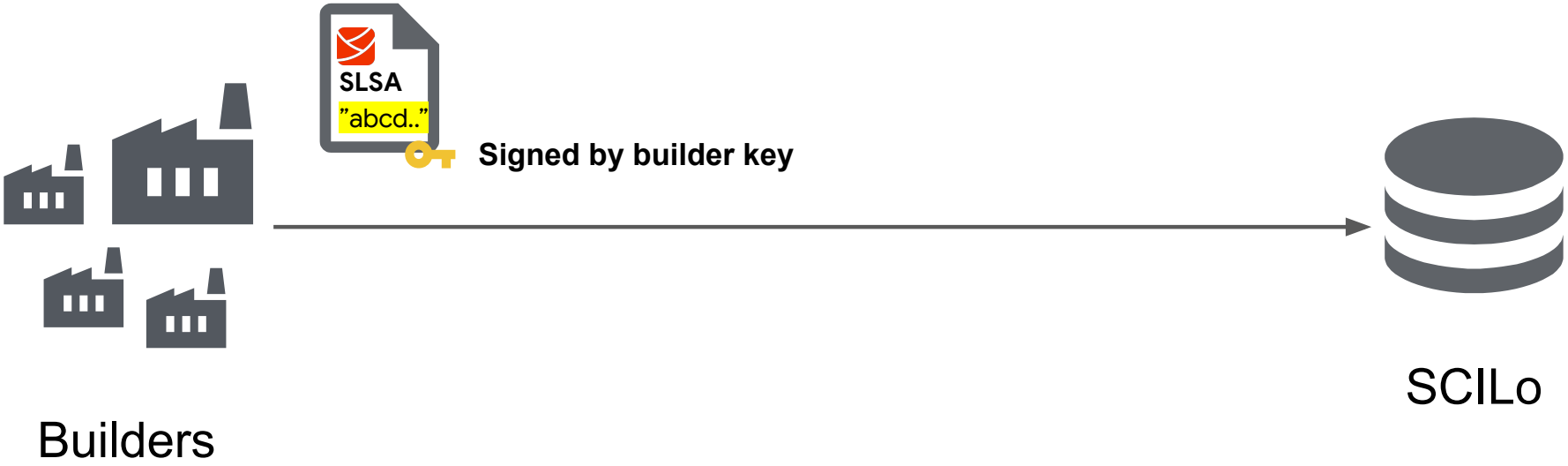
WAIT!! What graph?

BUILD! BUILD! BUILD!

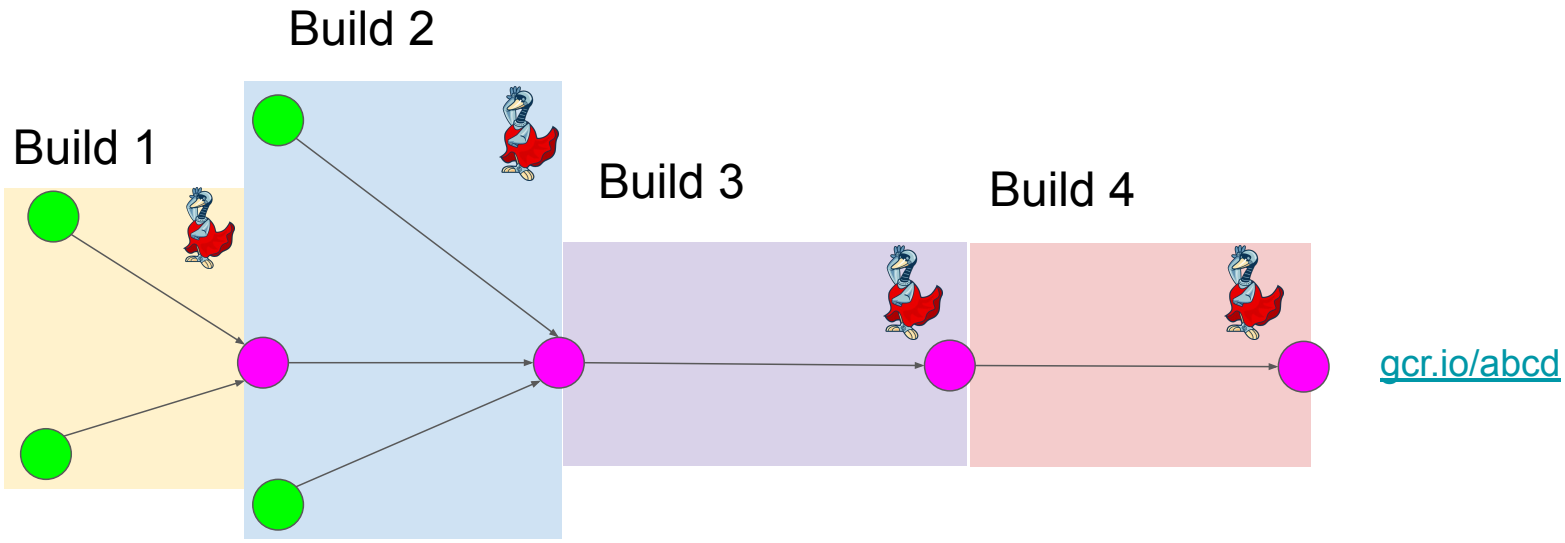
gcr.io/abcd

RECAP!

I built an artifact, here's the build provenance to show it was securely built **and what it was built from**



Using SLSA



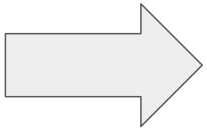
SLSA build metadata can be used to glue together lost pieces of SBOMs, creating more accurate SBOMs by composing them together!

https://slsa.dev/blog/2022/05/slsa-sbom

Retrieve Flow

Compliance Officer

Give me SBOMs for product **PixelOS**.



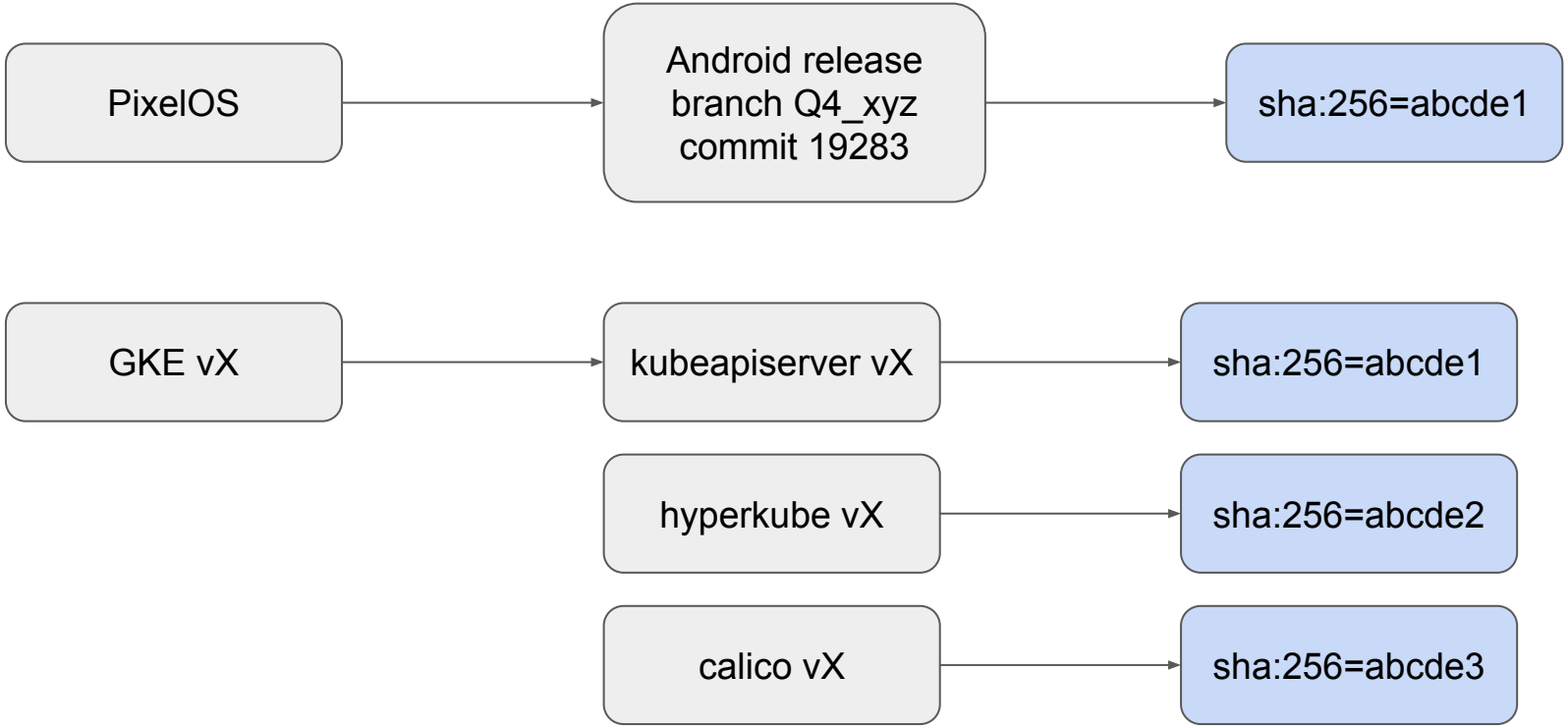
Product Owner

I am looking for ***URI: XYZ*** or *Hash:* ***sha256:abcd****…*

Translating request requirements is not easy

- Product mapping to software is HARD
- Effort needs to put into **maintaining software inventory**

Product mapping




Doing cool things with SBOMs
EO 14028 Compliance

04

Using SBOM

Using SBOMs

- Operationalizing an SBOM-based dependency inventory

SCILo

GUAC

+ Threat Intelligence
+ Organization Metadata

Incident Response (e.g. xz)

Quote from team: “We were able to figure out that we weren’t affected within 10 minutes”

Next, find where the package is used.

Purl

~ pkg:deb/debian/liblzma5@5.2.4-1

Dependent count

[redacted]

This search is limited to the latest builds of each ResourceUri.

ResourceUri

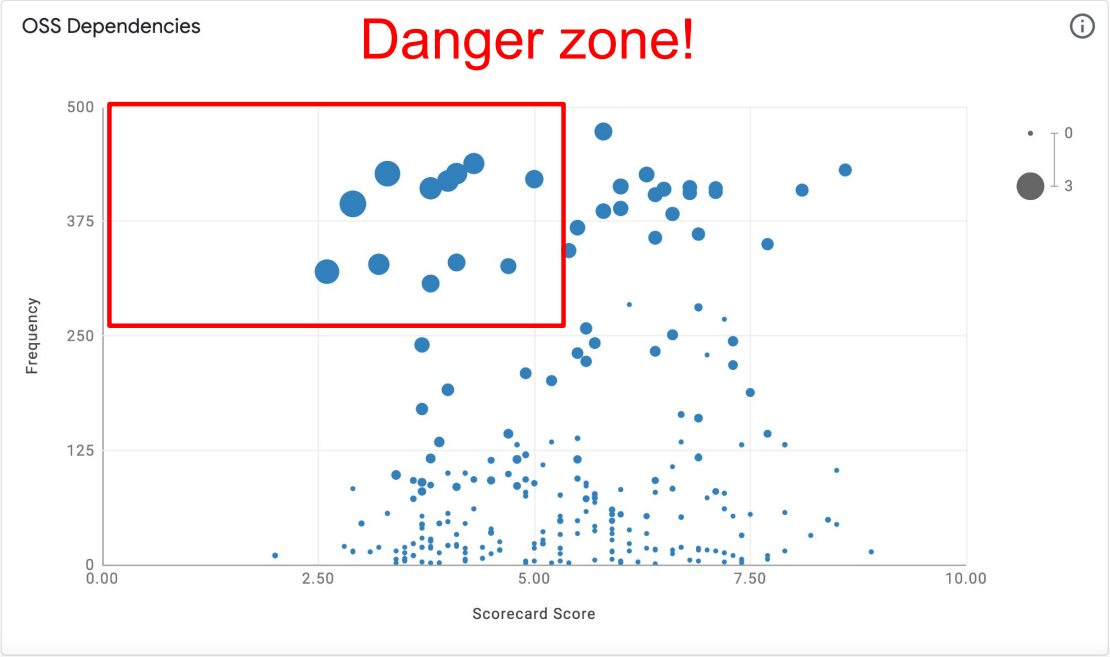
Point of contact not provided by SBOMs but VERY important!

ResourceUri	PointOfContact	
container_image://us-c[redacted]		
container_image://us-c[redacted]		{}
container_image://us-c[redacted]		{}
container_image://us-c[redacted]	{ "moma_team_id": [redacted] "buganizer_component": [redacted], [redacted] "team_email_addresses": ["overground-dev-null@google.com", "cloud-memcache-team@google.com"] }	{ "buganizer_component": "[redacted] }
container_image://us-c[redacted]		{}
container_image://us-c[redacted]		{}
container_image://us-c[redacted]		{}
container_image://us-c[redacted]		{}
container_image://gcr.[redacted]	{ "moma_team_id": [redacted] "buganizer_component": "[redacted] [redacted] "team_email_addresses": [[redacted]] }	{ "buganizer_component": [redacted]

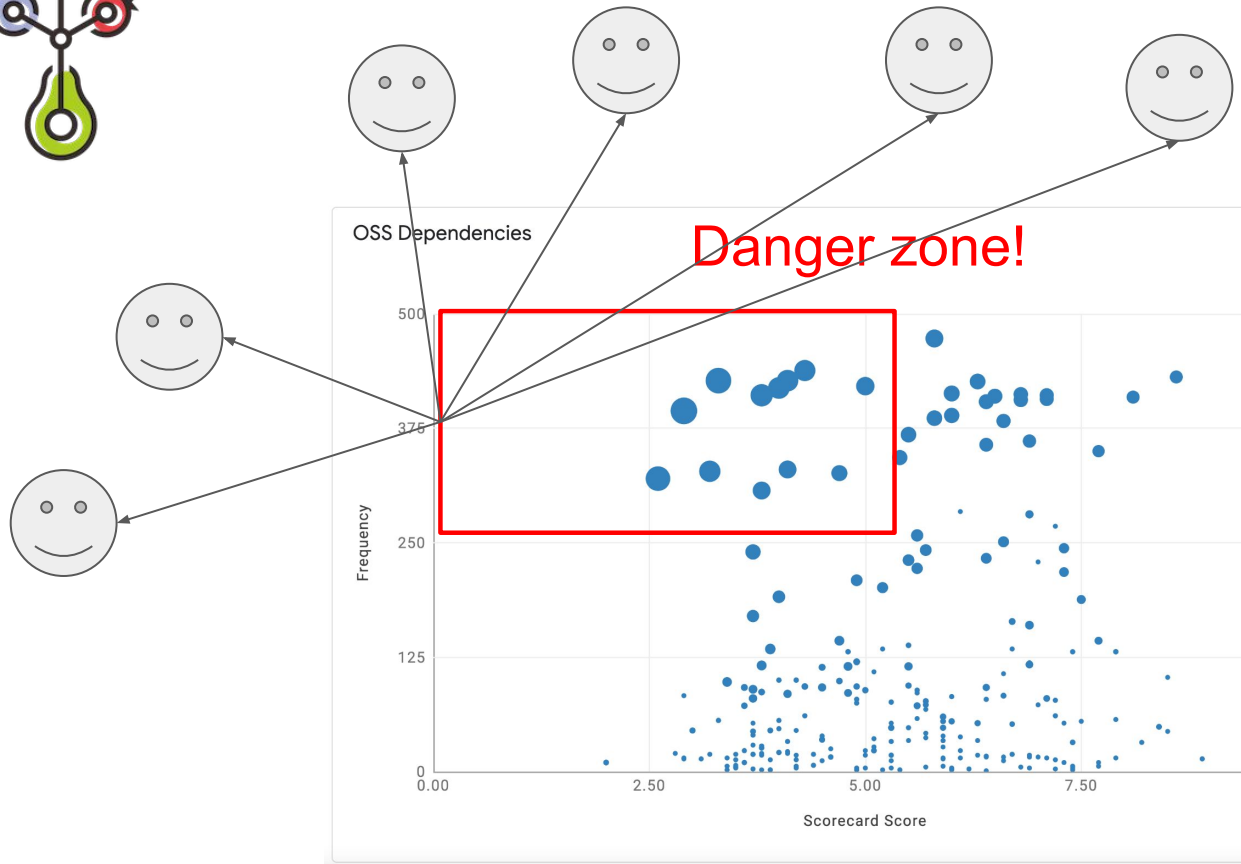
Because SBOMs are common formats across ecosystems, this queries across otherwise silo’ed systems

Fleet-wide Insights

Using GUAC + deps.dev, we mapped out fleet-wide dependency OpenSSF scorecard risks.



Fleet-wide Insights



Fleet-wide insights are only actionable if its scoped to someone that can take action

Here fleet = all container images across multiple orgs and ecosystems = little accountability

05

SBOMs Lessons Learned++

Using SBOMs: Lessons Learned

- Missing Software Identifiers

```
| | | "externalRefs": []
```

Using SBOMs: Lessons Learned

- Missing Software Identifiers
- SCA Shortcomings

```
1  {
2    "name": "extraneous-dev-dep",
3    "version": "0.0.0",
4    "dependencies": {
5      "d": "1.0.0"
6    }
7  }
```

pkg:npm/extraneous-dev-dep

Using SBOMs: Lessons Learned

- Missing Software Identifiers
- SCA Shortcomings

ljharb on Sep 19, 2023 Member ···

> Because it's a private package that just coincidentally has the same name as the malicious one, it is a false positive - so whatever tool is flagging this repo is broken, and you should strongly reconsider using a tool that is this naive about npm package names.

Duplicate of #303. Duplicate of #291. Duplicate of #288. Duplicate of #304. Duplicate of #305. Duplicate of #306. Duplicate of #309. Duplicate of #310. Duplicate of #311. Duplicate of #312. Duplicate of #314. Duplicate of #317. Duplicate of #318.

👎 9

Using SBOMs: Lessons Learned

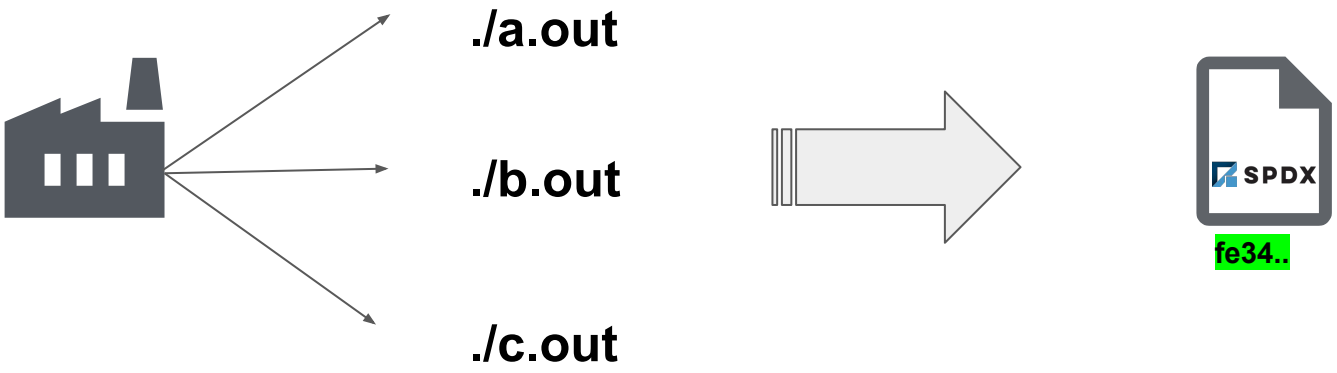
- Missing Software Identifiers
- SCA Shortcomings
- Identifier Shortcomings

Using SBOMs: Lessons Learned

- Missing Software Identifiers
- SCA Shortcomings
- Purl Shortcomings
- Focus on SBOM Quality

Using SBOMs: Lessons Learned

- Missing Software Identifiers
- SCA Shortcomings
- Purl Shortcomings
- Focus on SBOM Quality



What now?

2021 - 2024 SBOMs @ Google

From 0* to...

4M
SBOMs a week

200M+
SBOMs

- Security and compliance teams now using SBOMs to help triage security/compliance issues
- SBOMs being part of several organizations' governance posture
- 2 Build SBOM tools, 1 Analysis SBOM tool, and more coming!

*We had some SBOMs generated from compliance purposes

What’s next

- SBOM Quality Library?
- Collaborating with OSV-SCALIBR
- Guac Software Identifiers Project
- Better organization metadata (via attestations)