Generating SBOMs with isar

Felix Mößbauer, Christoph Steiger
Siemens AG

SIEMENS

About us

Felix Moessbauer

- Senior key expert @Siemens FT
- Embedded Linux consultant & developer

@fmoessbauer

mastodon.social / @fmoessbauer

felix.moessbauer@siemens.com

Christoph Steiger

- Software Engineer @Siemens FT

@Urist-McGit

christoph.steiger@siemens.com

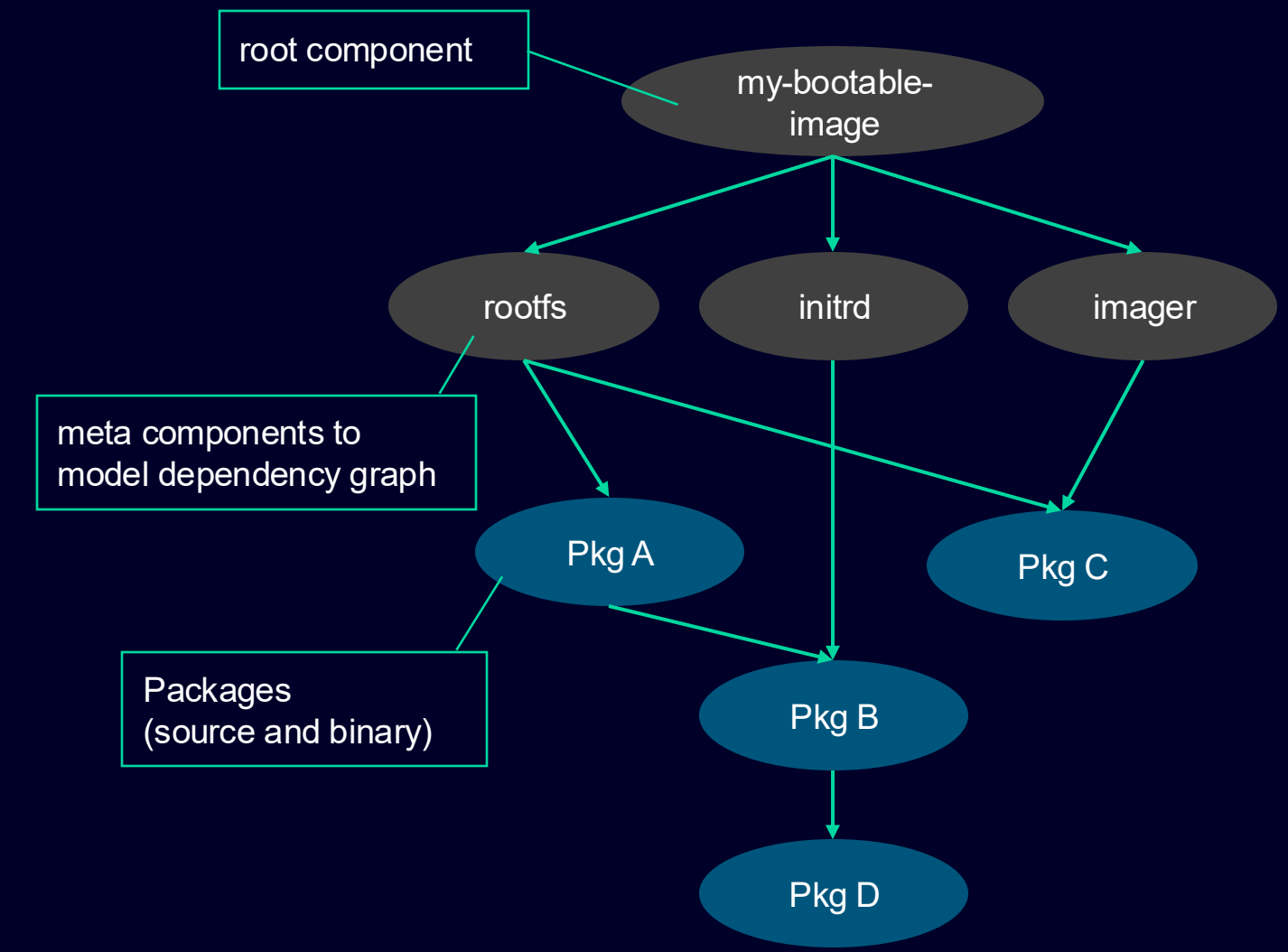
SIEMENS

The Basics – What is an SBOM?

SBOMs (Software Bill of Material) is a **standardized file format** to exchange information about the components used in a software or system.

This information is needed for regulatory reasons, license clearing, and security (e.g., CVE tracking).

Commonly used formats are **SPDX** and **CycloneDX** (CDX)



SIEMENS

Why yet another SBOM generator?

Requirements

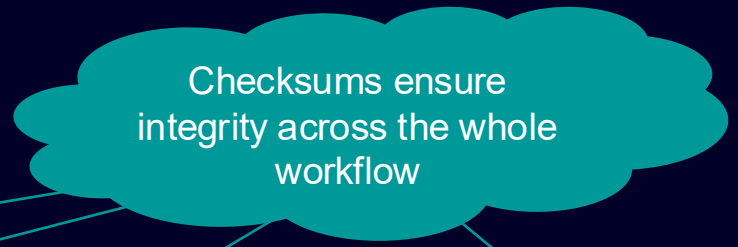
- Needs to be OSS
- Easily buildable with Debian toolchain (and Debian dependencies)
- Can be integrated into ISAR
- Precise mapping of Debian components (including dependency graph, build-using, …)

Alternatives considered

- Internal tooling (script collection)
  - No dependency graph
  - No precise mapping
- Syft / Grype
  - Written in go -> huge dependency graph -> not packageable
  - No dependency mapping
  - No precise mapping

Using debsbom as input to license clearing
A common workflow

Recap "License clearing": the process of cataloging every component and dependency in the project, identifying each component's license, and ensuring all license terms are satisfied before shipping.



generate

`debsbom` runs as part of the isar build (offline) and generates a CycloneDX or SPDX SBOM

download

Reads an SBOM and downloads the referenced source and binary packages. Everything is secured by checksums

source-merge

Debian source packages consist of multiple files and contain patches to upstream components (e.g., to backport security fixes). The `source-merge` merges all related artifacts into a single archive and optionally applies patches

repack

For license clearing and archiving, the SBOM is converted into a Siemens **standard-bom** along with a source-archive that contains a copy of all referenced packages

Examples: https://siemens.github.io/debsbom/examples.html

SIEMENS

Securing the Supply Chain

Package Metadata:

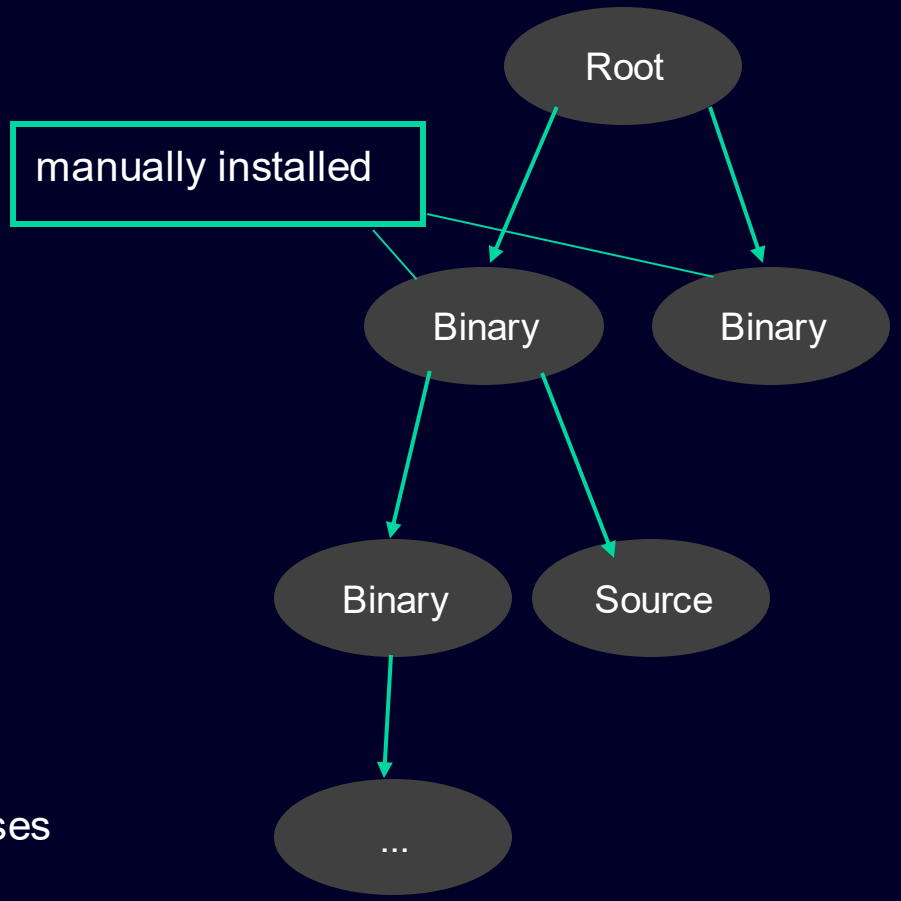
- Required are name, version and architecture
- Maintainer information is important, but unreliably maintained by Debian
- Mapping of Debian fields to SBOM fields is arbitrary to some extent
- debsbom provides no license information

Package Relationships:

- Three relevant relationship types:
  - Binary dependency
  - Source dependency
  - Built-Using dependency
- Full dependency graph with manually installed packages as roots

Package Identification:

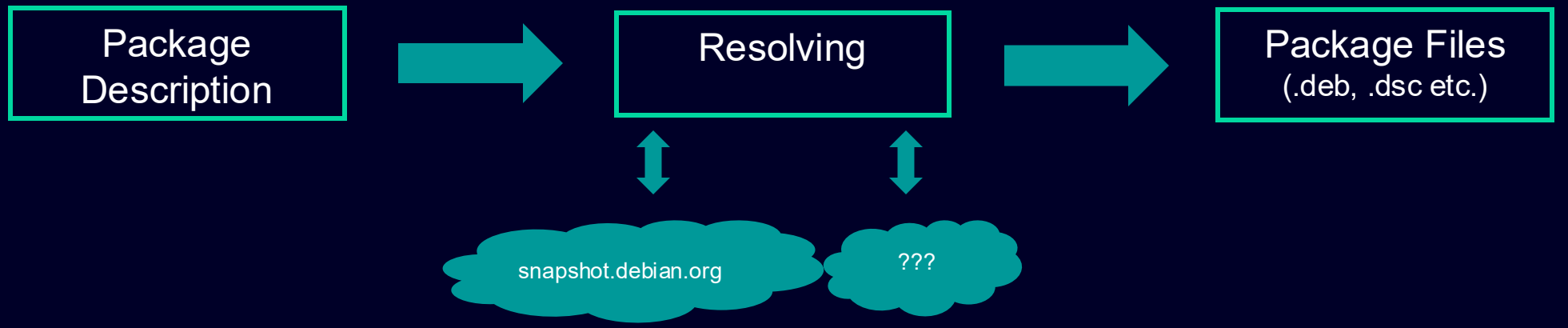
- Surprisingly hard, but (name, version, architecture) tuple enough in most cases
- PURLs encode this tuple: `pkg:deb/debian/htop@3.4.1-5?arch=amd64`
- Checksums are required to handle all edge cases



SIEMENS

Pluggable Resolvers

… or where to find Debian Packages



- debsbom supports resolving with the Debian snapshot mirror out of the box
- Resolving is use-case specific: custom package repositories, Artifactory etc.
- debsbom provides plugin infrastructure to implement custom resolvers
- For isar: local apt repository and Debian snapshot mirror for everything else

SIEMENS

Examples
Recap: License Clearing

Generate SBOM:

```
debsbom generate -r /path/to/the/rootfs -t cdx -o sbom.cdx.json
```

Download all source packages:

```
debsbom download --sources sbom.cdx.json
```

Merge all files of a source package into a single tarball:

```
debsbom source-merge --apply-patches sbom.cdx.json
```

Repack it into the Siemens Standard BOM format:

```
debsbom repack --format standard-bom sbom.cdx.json sbom.packed.cdx.json
```

Summary

- **No guessing:** debsbom precisely describes packages and metadata, integrity protected using checksums
- **Dependency graph**: debsbom generates a precise dependency graph, useful for later analysis
- **end-2-end**: SBOM generation, package downloading and repacking
- **Integrated into isar:** SBOMs are automatically generated and describe whole .wic image

Siemens AG

Felix Moessbauer
felix.moessbauer@siemens.com

Christoph Steiger
christoph.Steiger@siemens.com

siemens / debsbom

Urist-McGit / debsbom-plugin-examples

SIEMENS