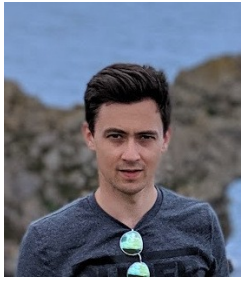
Delta-like Streaming of (encrypted) OTA Updates for RAUC

FOSDEM 2023

Enrico Jörns – e.joerns@pengutronix.de

About Me & Pengutronix



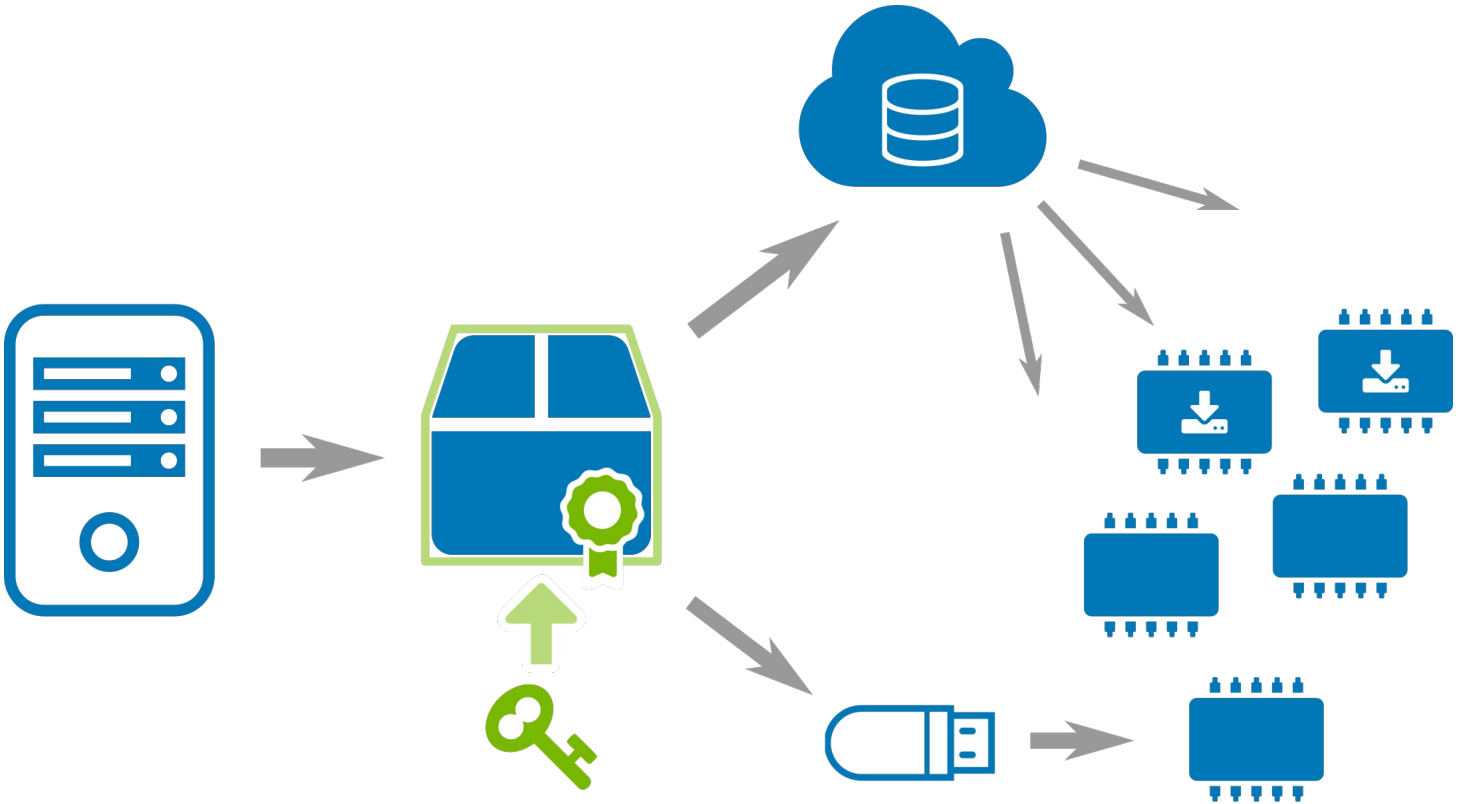
- Embedded software developer
- RAUC co-maintainer
- Integration Team Lead at Pengutronix

  - Embedded Linux consulting & support since 2001
  - 7000+ patches in Linux kernel

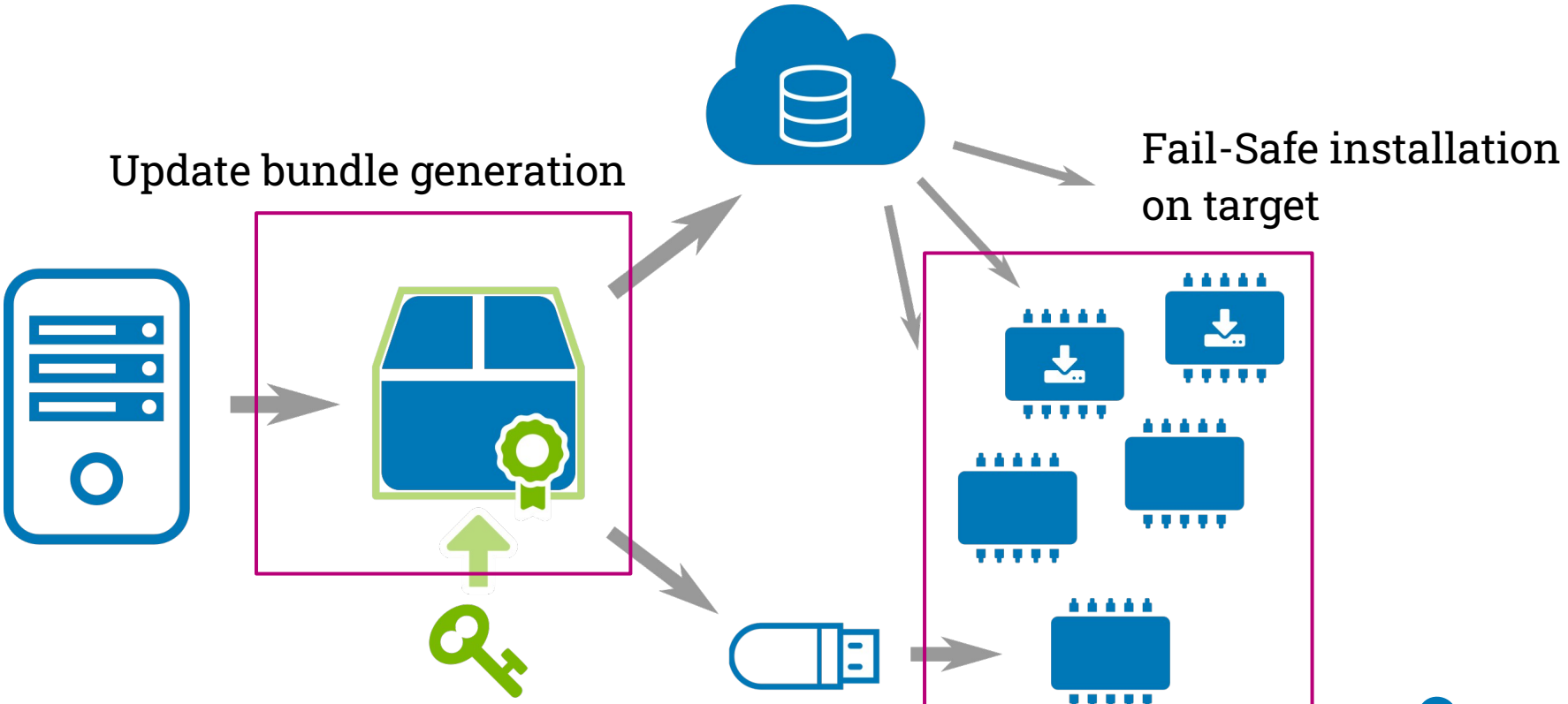
Structure

- Introduction + RAUC overview
- RAUC bundle format
- HTTP(S) bundle streaming
- Adaptive updates
- Encrypted bundles
- App updates
- Outlook & ecosystem

(OTA) Field Updates



RAUC – Scope


Update bundle generation
Fail-Safe installation
on target

RAUC

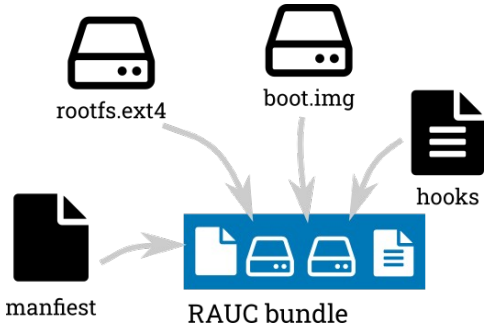
- Embedded Linux update *framework*
- Fail-safe, authenticated, image-based (A/B) installation

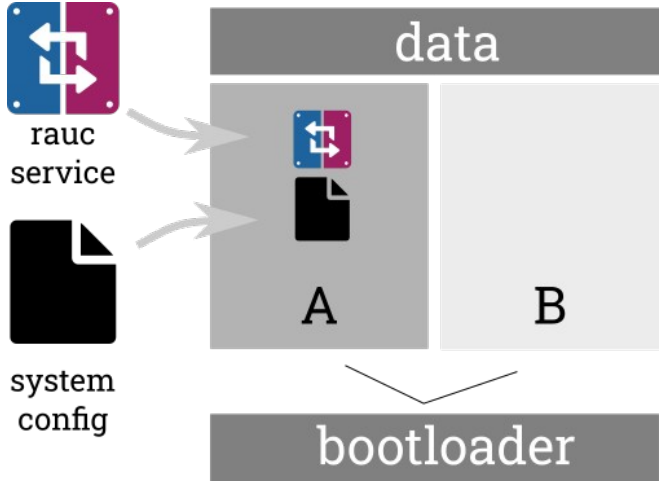
</> Written in C

(with glib, OpenSSL, curl, ...)

LGPL-2.1 License

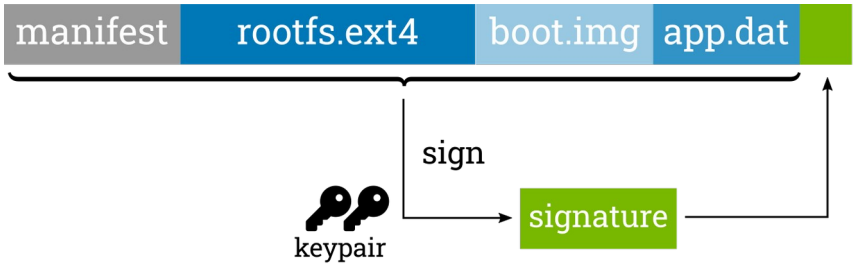
https://github.com/rauc/rauc





Bundle Format

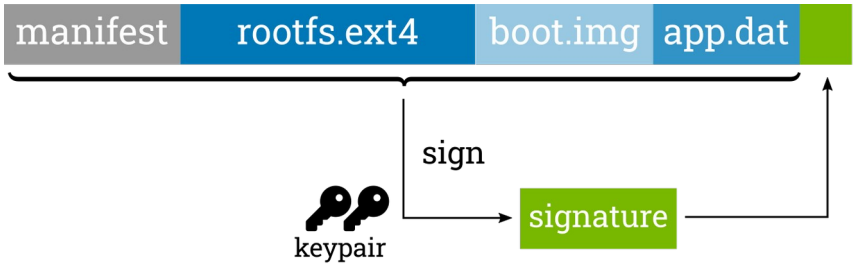
Limitations of Initial Bundle Format



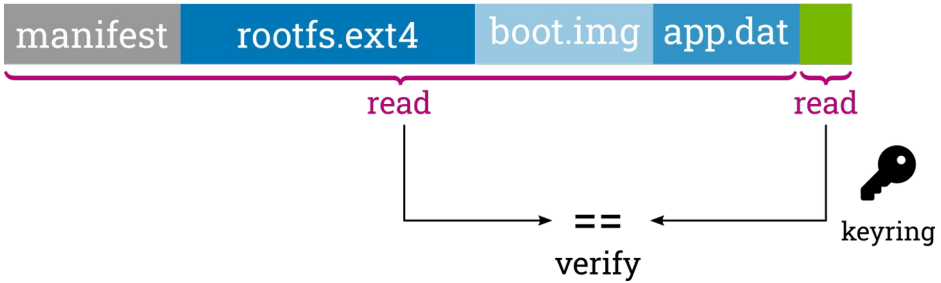
Bundle Generation

- SquashFS containing
  - Manifest, images, hooks, …
- Attached signature

Limitations of Initial Bundle Format



Bundle Generation



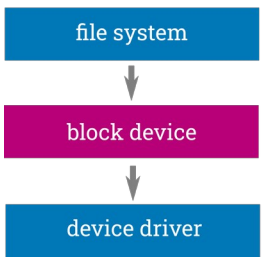
Bundle Authentication

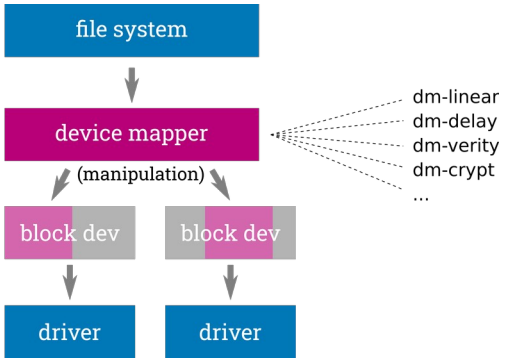
- SquashFS containing
  - Manifest, images, hooks, …
- Attached signature

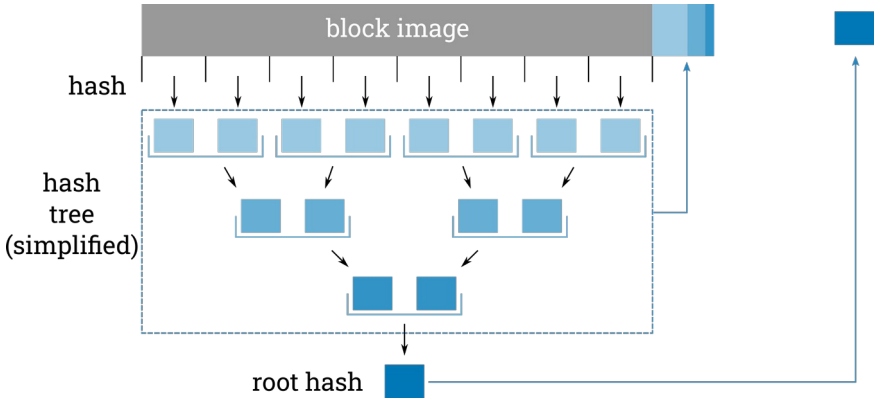
✖ Read entire bundle for verification!

  - Slow, not streamable!

Background – dm-verity

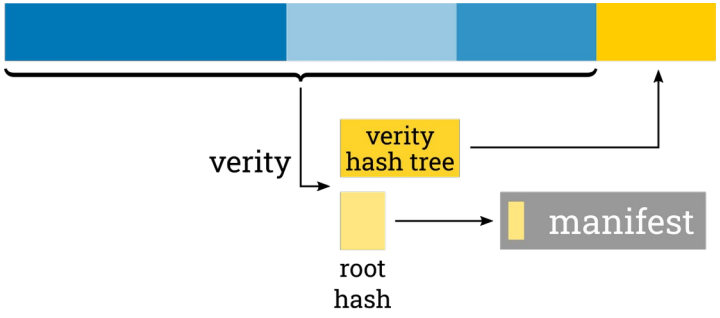






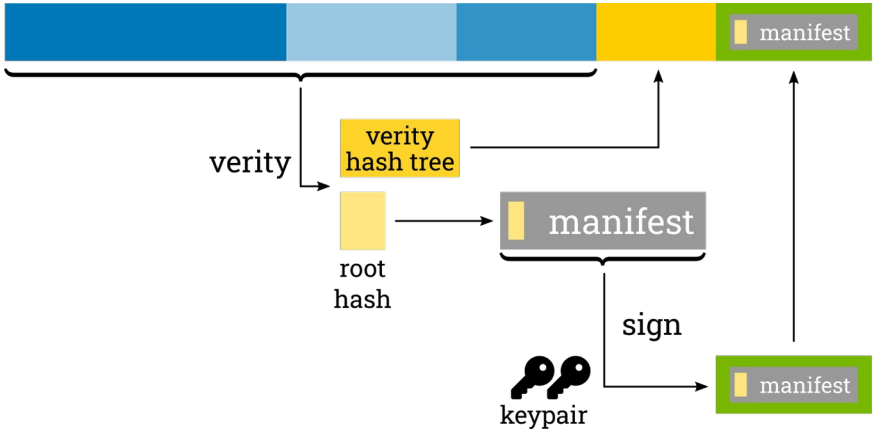
- Linux kernel device mapper
  - Generic block device manipulation abstraction
- Hash-table to authenticate read-only block devices

Verity Bundle Format – Creation



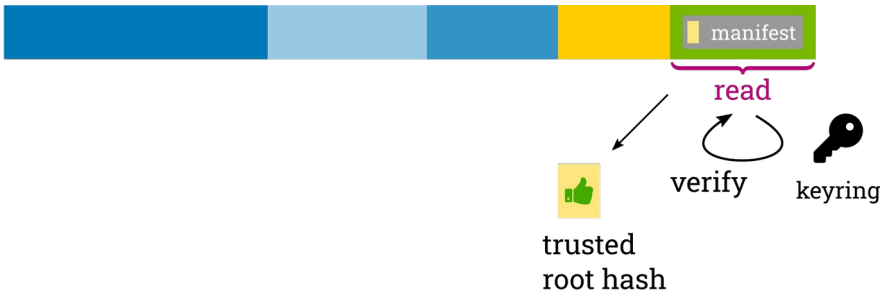
- dm-verity (Merkle) hash tree + root hash generated
- Root hash → trust anchor
  - Stored in manifest

Verity Bundle Format – Creation



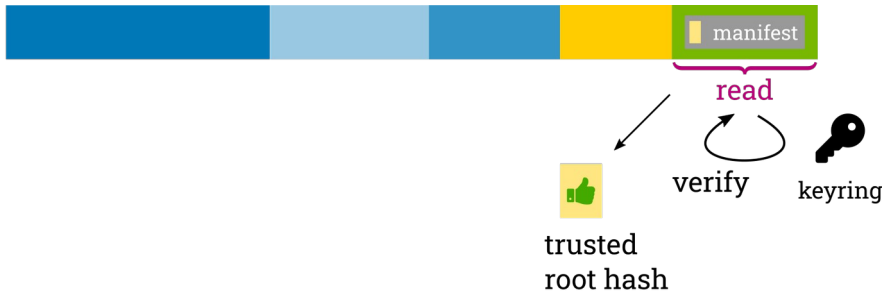
- dm-verity (Merkle) hash tree + root hash generated
- Root hash → trust anchor
  - Stored in manifest
- Enveloping signature
  - Contains manifest

Verity Bundle Format – Authentication

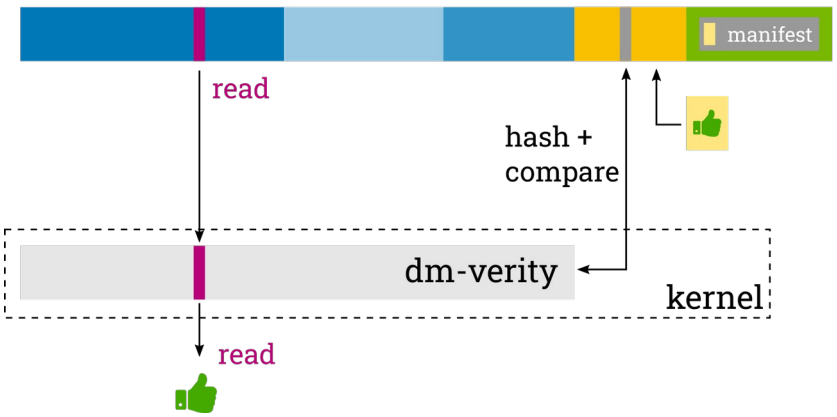


- ✔ Fast Manifest authentication
  - Authenticates verity root hash!

Verity Bundle Format – Authentication



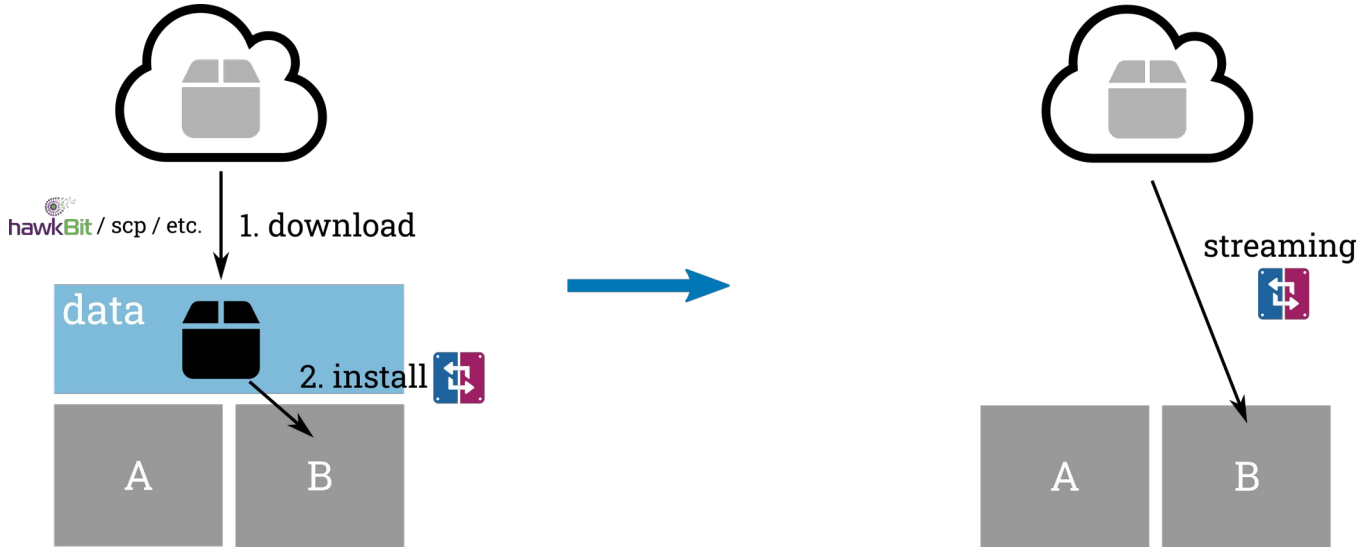
- ✔ Fast Manifest authentication
  - Authenticates verity root hash!



- Per-chunk verification
- ✔ Authenticated random access!

Bundle Streaming

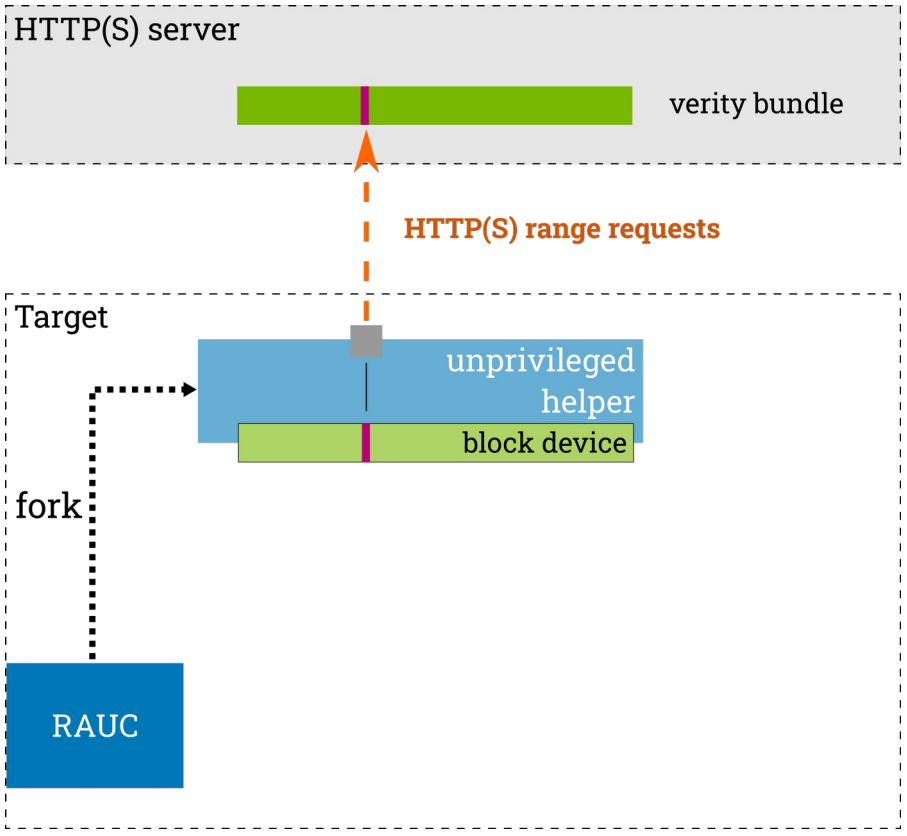
RAUC Streaming Support



- ✖ Full bundle download (by external application)
- ✖ Intermediate storage required

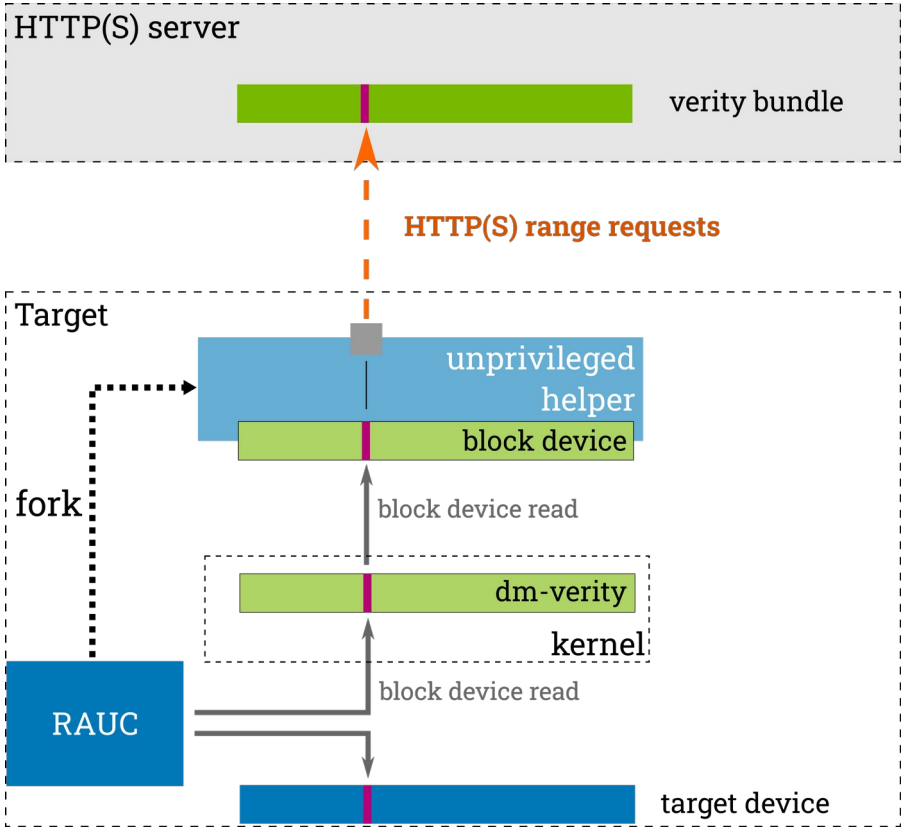
- ▪ Built-in on-demand download of required bundle data
- ✔ No intermediate storage

HTTP(S) Bundle Streaming



- Unprivileged helper process (using libcurl)
- HTTP(S) range requests
  - Supported by all common webservers and many CDNs

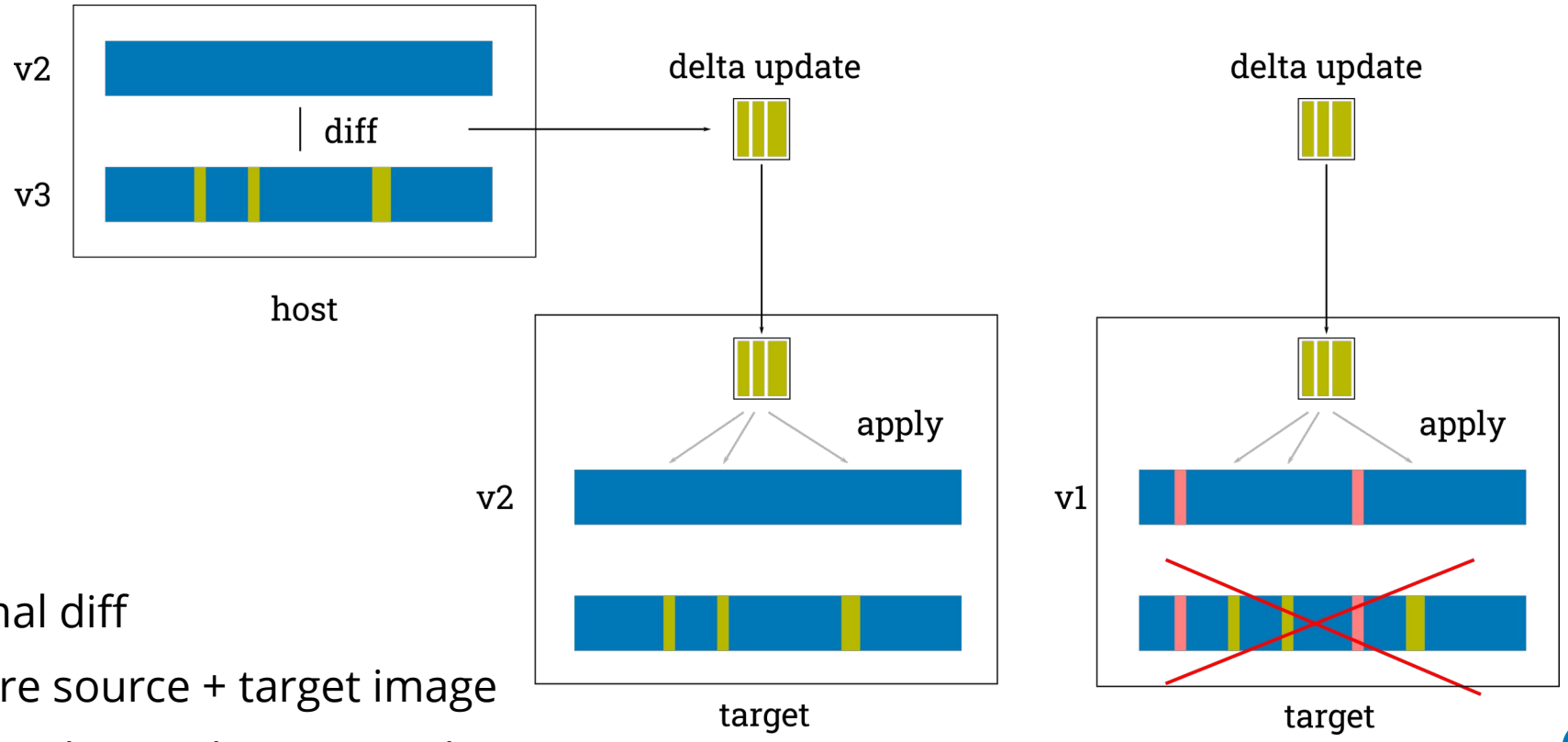
HTTP(S) Bundle Streaming



- Unprivileged helper process (using libcurl)
- HTTP(S) range requests
  - Supported by all common webservers and many CDNs
- ✔ Authenticated random access to remote bundle
- ✔ No intermediate storage

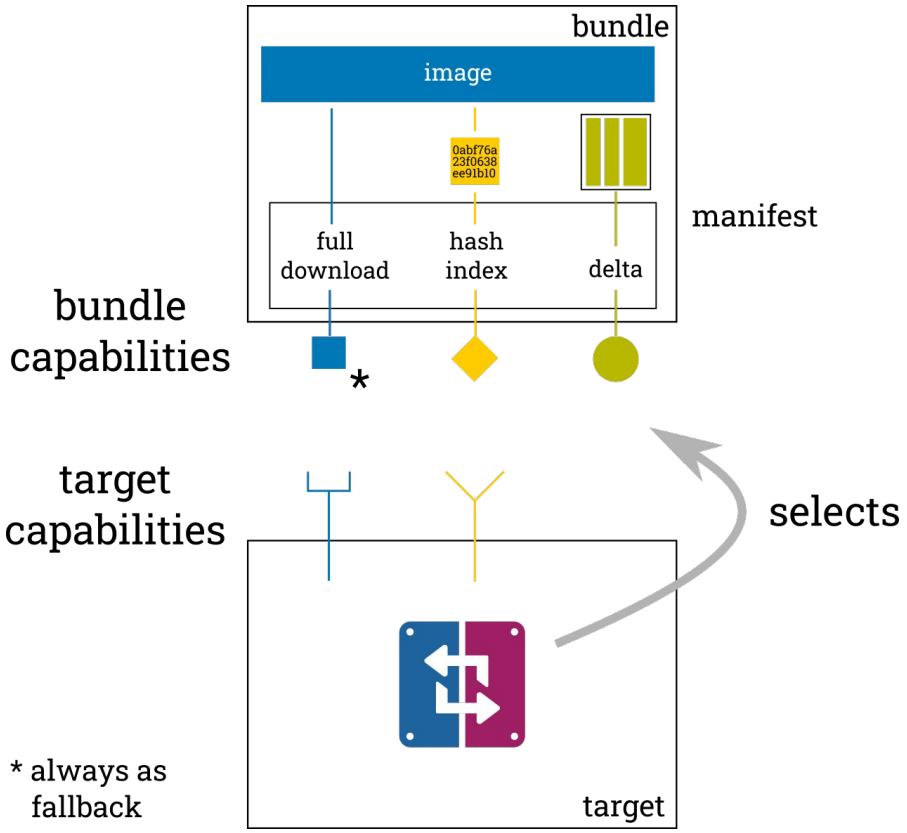
Saving Download Bandwidth

Conventional Delta-Updates



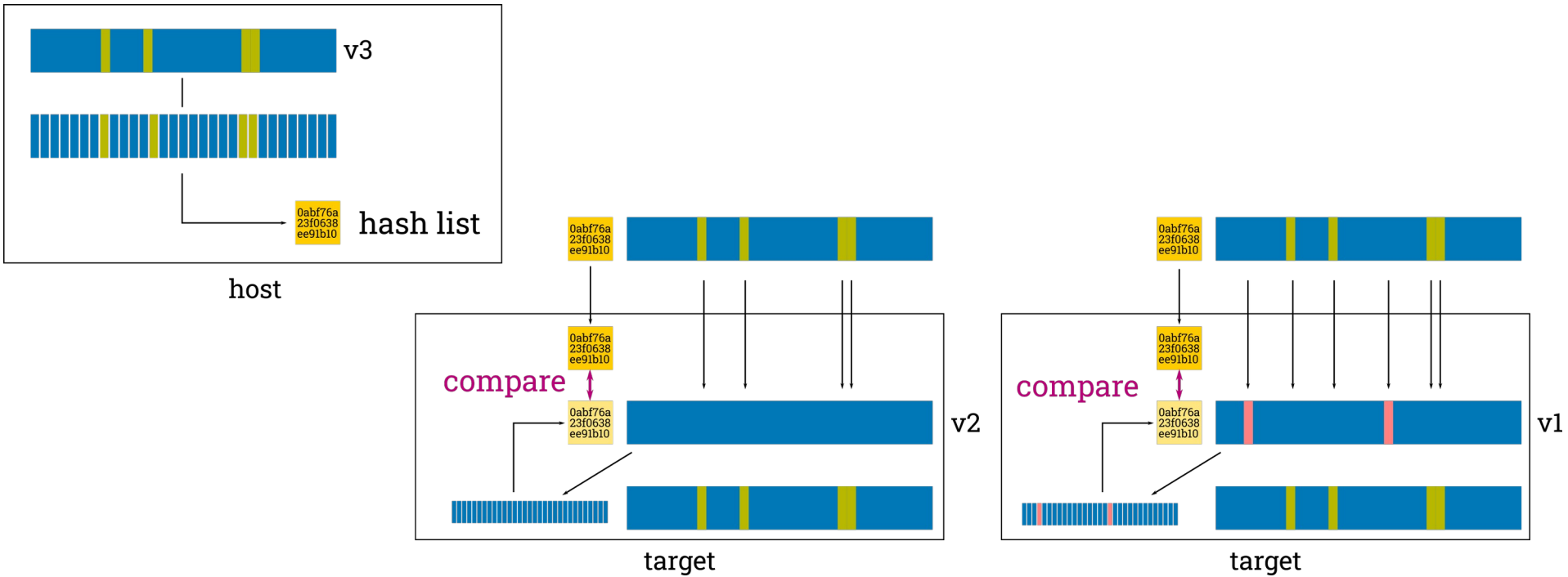
- ✔ optimal diff
- ✖ require source + target image
- ✖ allow only step-by-step updates!

RAUC Adaptive Updates – Concept



- Bundle provides optional download optimization
  - Additional data stored in bundle
- RAUC selects appropriate option
  - Downloads only the required data (streaming!)

Adaptive Update Method – hash-index

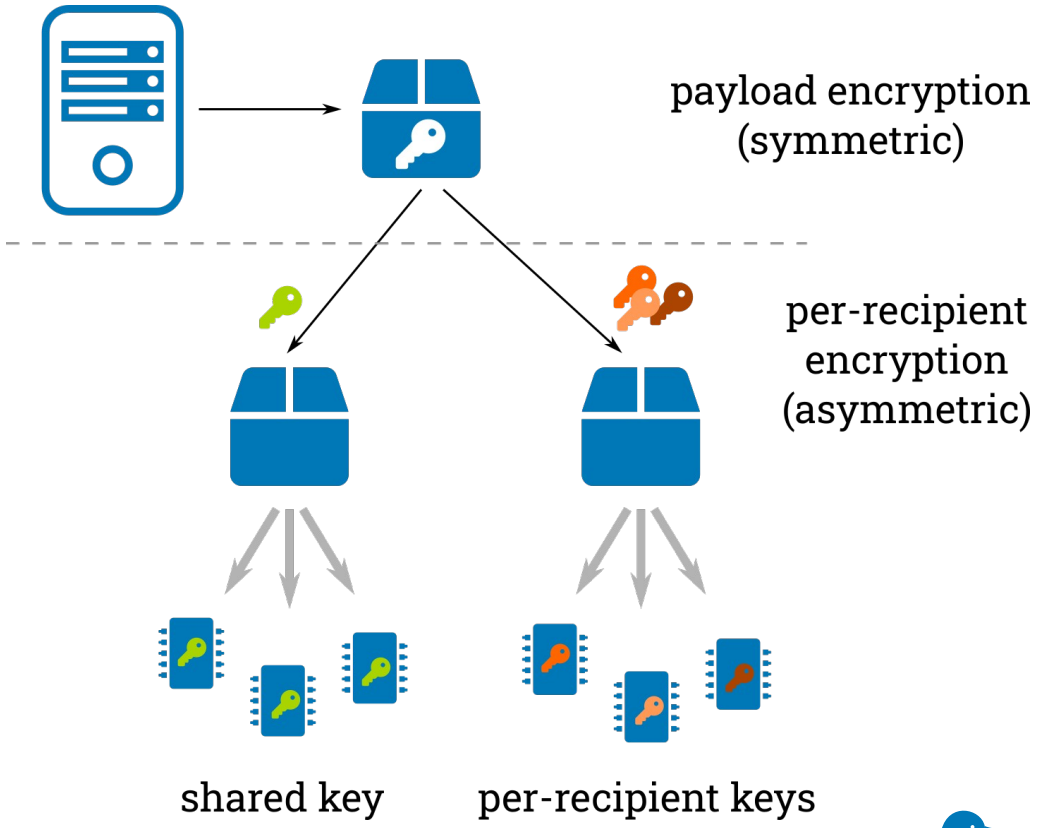


- ✔ `block-hash-index` implemented
- ✔ file-based variant via rsync + checksums planned

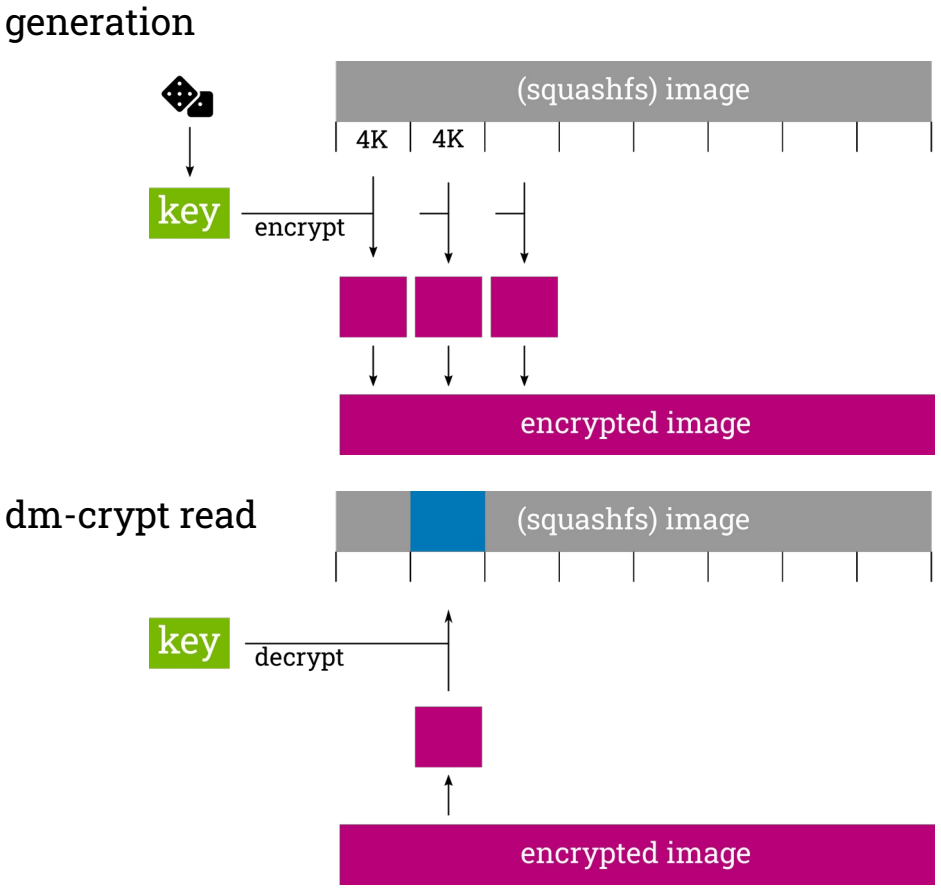
Bundle Encryption

Bundle Encryption

- Hide sensitive data or application IP from third-party
- Two-Stage approach in RAUC

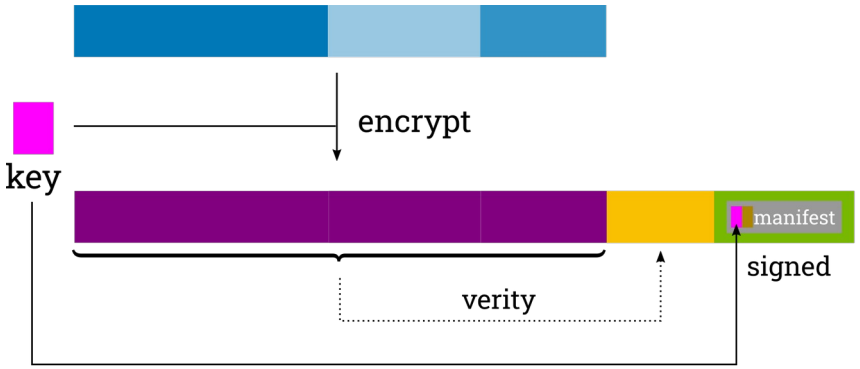


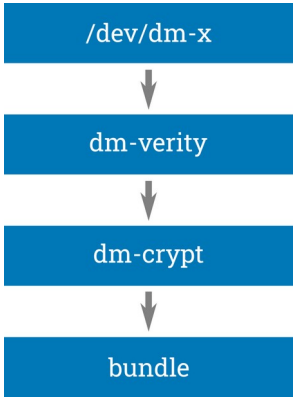
Bundle Payload Encryption – dm-crypt



- We just add another layer…
- Device mapper: dm-crypt (Symmetric with AES-256)
- Transparent decryption during installation

Bundle Payload Encryption



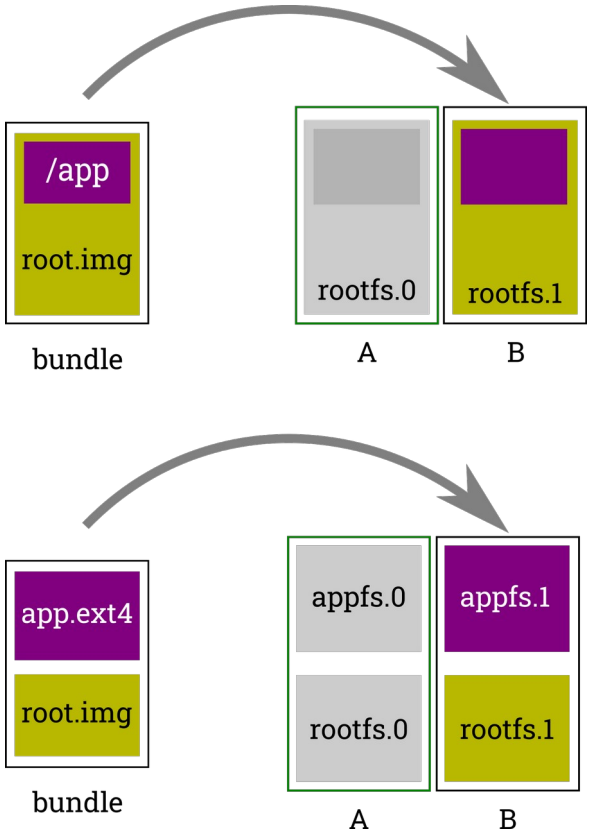


- Block-wise authenticated decryption
- Random access
- ✔ Streamable!

„App”-Updates

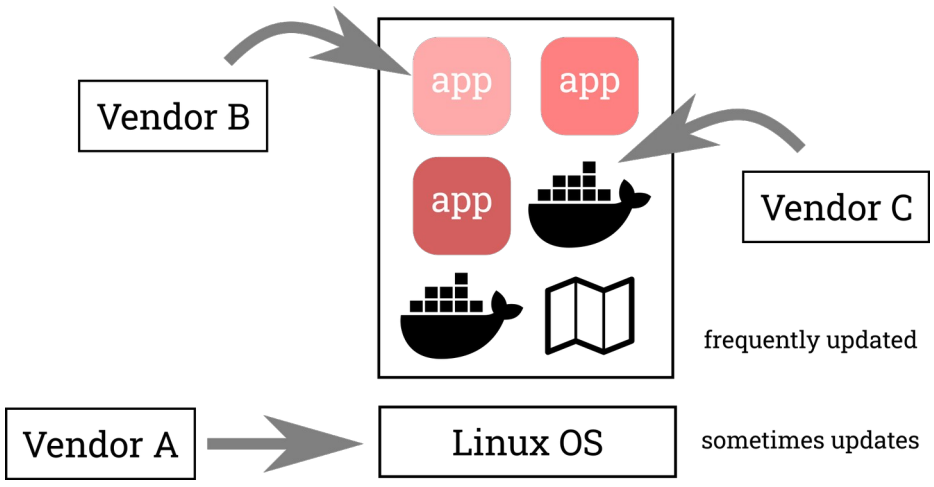
(In development)

Application Updates – Situation So Far



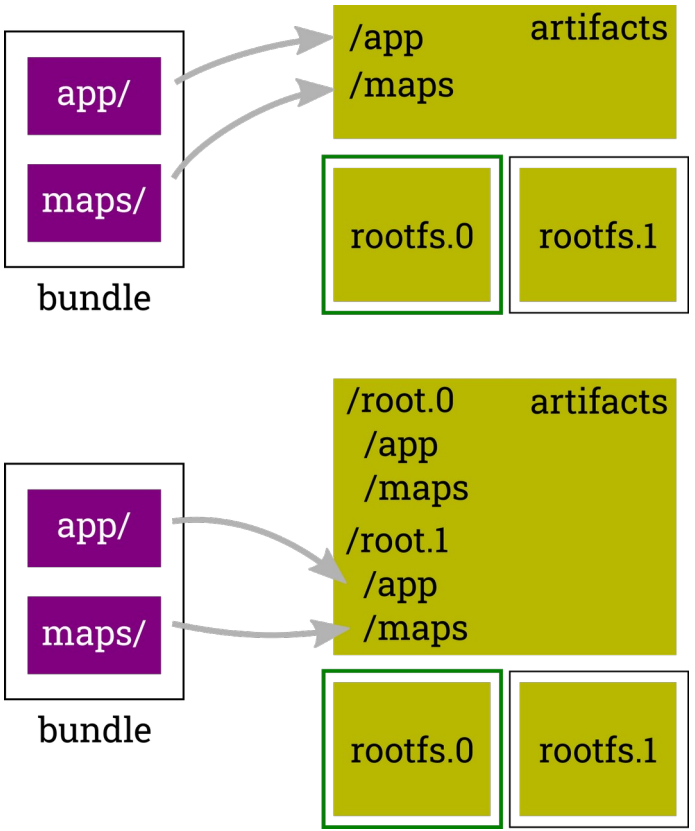
- Assumes monolithic system
  - Single (main) application
  - Linked against rootfs libs
- Application: part of rootfs or on separate slot
  - Always update both to have consistent state

‘App’ Updates



- Capability of updating
  - Apps, Containers, Maps, ...
  - Independent of rootfs (w/o reboot)
  - More frequent than OS
  - By different vendors
- Was beyond the scope of RAUC so far..

RAUC Artifact Updates



- New ‘artifacts’ slots
  - File- / Directory-based
  - Can hold multiple artifacts
- Atomic updates (symlinks)
- Redundant for dependency on rootfs
- Streaming support through rsync checksum files

https://github.com/rauc/rauc/issues/969

Outlook & Community

Current & Future Developments

- Switch to Meson (already merged)
- Custom meta-data in manifest
  - Exposed in ‘rauc info’ + D-Bus API
- Fine-grained installation progress
  - Tar extract progress

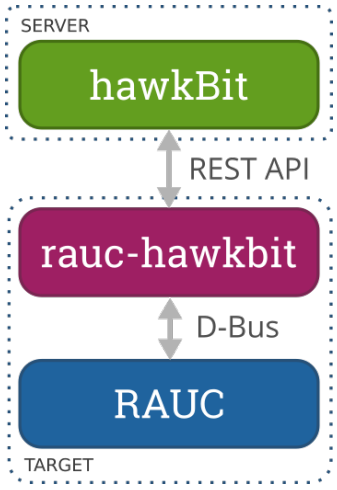
```
[update]
compatible=My Product Name
description=Verbose Text
…

[meta.pengutronix]
location=Brussels
release-notes=https://..
release-channel=beta

[meta.vendor]
key=value
```

Ecosystem: rauc-hawkbit-updater

- Eclipse hawkBit:
  Open Source back-end framework for software rollouts
- RAUC adapter in C started by Prevas (2018)
- Moved to RAUC GitHub org (2020)



✔ Compatible with streaming

https://github.com/rauc/rauc-hawkbit-updater

Community: meta-rauc-community

- Bitbake layer collections for example integrations
- Maintained by Leon Anavi
- Supported boards:
  - qemux86-64
  - raspberrypi
  - Sunxi
  - Tegra

https://github.com/rauc/meta-rauc-community

Community: RAUC-related Projects / Products

- Valve Steam Deck
  - RAUC + desync (casync variant in Go)
  - Patches mainlined by Collabora
- Home Assistant Operating System
  - Buildroot updated with RAUC
- Oniro
  - Eclipse project for distributed systems
- DB Linux4ICEs
  - Linux distro for ICE train info panels

https://www.heise.de/select/ct/2022/27/2223808321839973008

Home Assistant

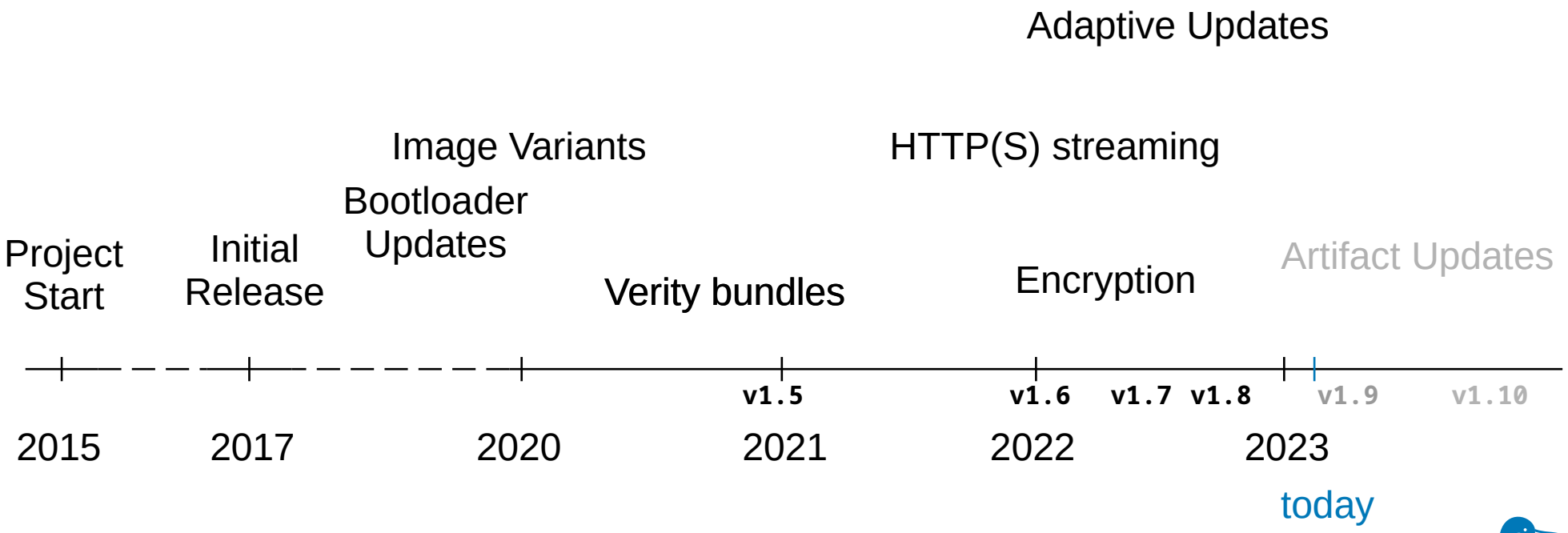
oniro

Thank You!

Questions?

Join the discussion and get help on: #rauc IRC/Matrix channel

Timeline


Adaptive Updates
Image Variants
HTTP(S) streaming
Bootloader Updates
Project Start
Initial Release
Verity bundles
Encryption
Artifact Updates
v1.5
v1.6
v1.7
v1.8
v1.9
v1.10
2015
2017
2020
2021
2022
2023
today

RAUC – Configuration Basics

```
[update]
compatible=Test System
version=2023.02

[bundle]
format=verity

[image.rootfs]
image=rootfs.ext4
```

Update manifest
→ in bundle

```
[system]
compatible=Test System
bootloader=u-boot

[slot.rootfs.0]
device=/dev/mmcblk0p1
...

[slot.rootfs.1]
device=/dev/mmcblk0p2
...
```

System configuration
→ on target

```
root@qemu-test:# rauc status
=== System Info ===
Compatible:  Test Config
Variant:
Booted from: rootfs.0 (A)

=== Bootloader ===
Activated: rootfs.0 (A)

=== Slot States ===
o [rootfs.1] (/tmp/rootdev, raw, inactive)
        bootname: B
        boot status: good
    [appfs.1] (/tmp/appdev, raw, inactive)

x [rootfs.0] (/dev/root, raw, booted)
        bootname: A
        mounted: /
        boot status: good
    [appfs.0] (/dev/null, raw, active)
```

→ Introspectable!