

Delta Updates

Making Updates Leaner

OSS@Siemens | Zug | 2024

Felix Moessbauer & Jan Kiszka

About Us



Felix Moessbauer

<felix.moessbauer@siemens.com>

- Siemens Technology
- (In-house) Embedded Linux consultant & developer
- Contributes to major OSS projects for Siemens
- Tooling developer (static / dyn. analysis, build tools, ...)



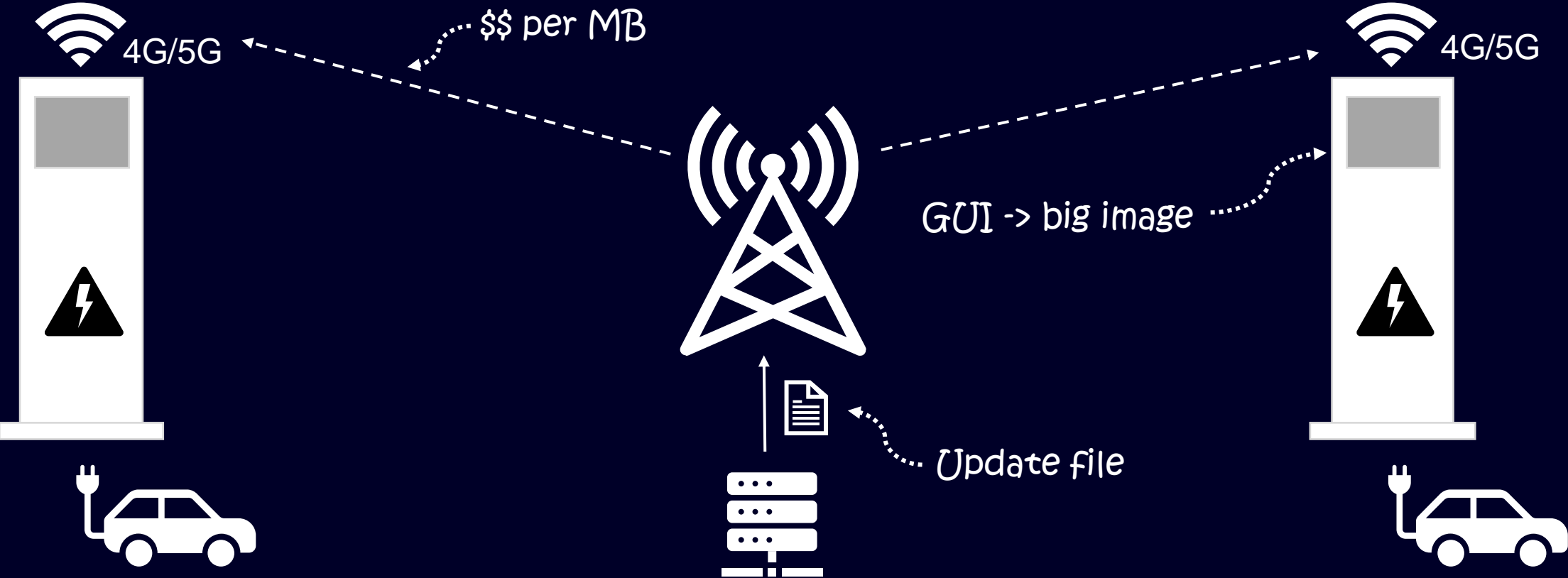
Jan Kiszka

<Jan.kiszka@siemens.com>

- Siemens Technology
- (In-house) Embedded Linux consultant & developer
- CIP kernel workgroup chair, isar-cip-core maintainer
- Maintainer and contributor to various OSS projects

Robust updating of industrial devices in the field

An example use-case



Civil Infrastructure Platform (CIP) Project

Provide an industrial-grade Linux

- Build on top of existing OSS projects
- Contribute to them to fill gaps
- Create new OSS projects where needed
- Provide integration patterns
- Enhance life-time of OSS where needed (LTS)
- Prepare security certifications

CIP-Core and Software Update Workgroups

- Implement pre-integration in isar-cip-core¹
 - Isar²: Debian image builder
 - Use Debian binaries where possible, build own ones where needed
- Define secure bootable & updateable reference images
- Concepts (not isar-cip-core recipes) can be transferred to Yocto/OE as well



CIVIL
INFRASTRUCTURE
PLATFORM

[1] <https://gitlab.com/cip-project/cip-core/isar-cip-core>, [2] <https://github.com/ilbers/isar>

Features and Concepts in CIP Software Update

State of the art

SWUpdate + wfx

- Software update for embedded system on device.
- Support Suricata mode and wfx (workflow executor) for remote update.



Reliable

- Dual-copy update pattern based on Round-Robin Handler.
- EFI Boot Guard as bootloader for robust boot path selection.
- Immutable rootfs + overlay for /etc



Secure

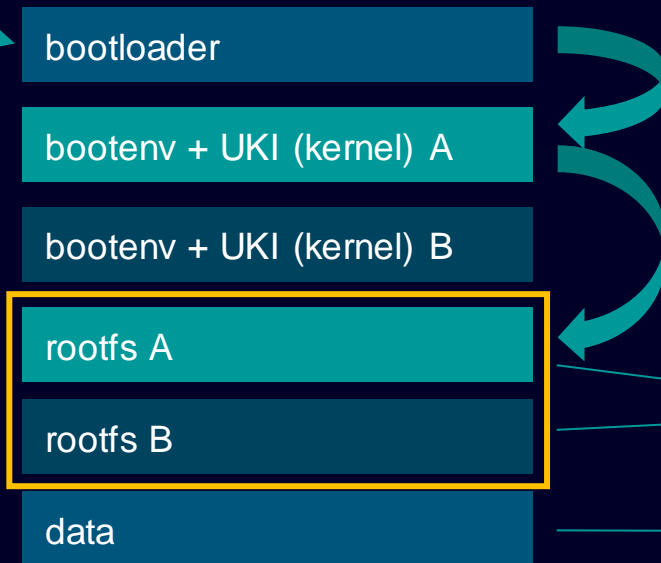
- Secure boot with EFI Boot Guard.
- SWUpdate image signing for image source verification.
- Disk encryption for data partitions (e.g. /var).
- Integrity verification of rootfs based on dm-verity.



CIP A/B rootfs Update Partition Layout



*Potentially big, needs
to be replaced on
every update*



Efibootguard: arms watchdog, reads bootenv, selects next target (UKI)

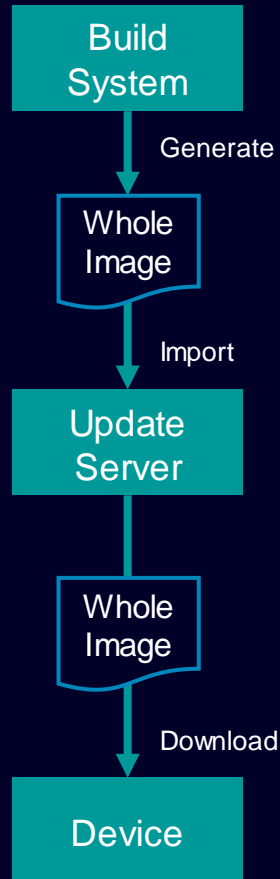
UKI = EFI-stub + Kernel + cmdline + Initrd
Potentially signed

Immutable rootfs (e.g. squashfs).
Can be integrity protected

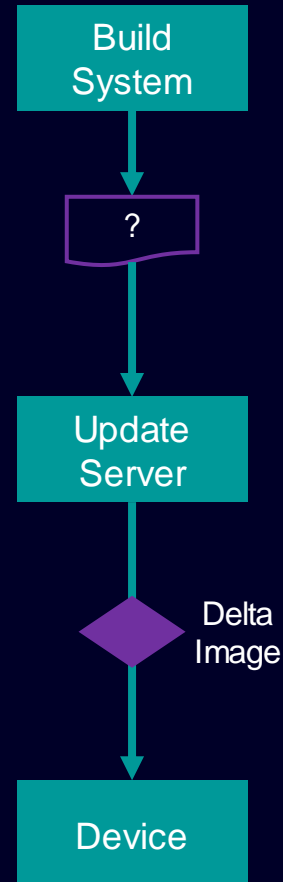
Persistent partition (not touched during update)

Delta Update and the Differences

Whole Image Update



Delta Update



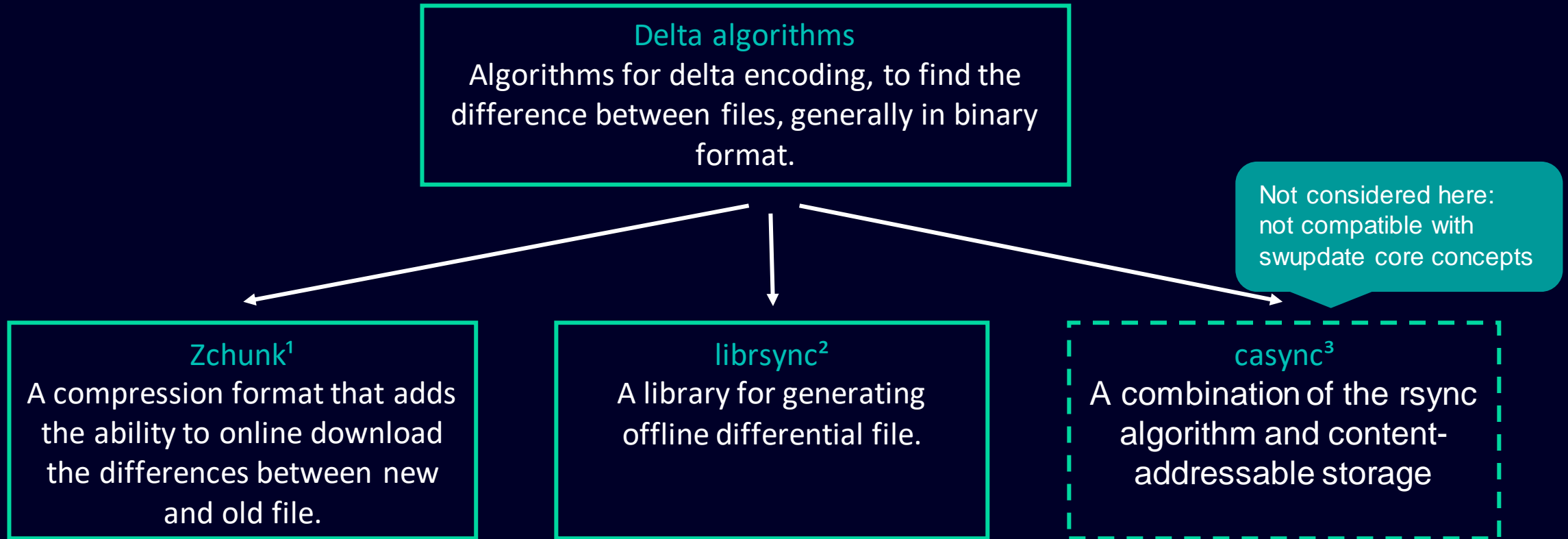
Delta update:

- Only changes of the image (delta image) need to be transferred.
- Resource efficient when the whole image is big, and changes are small.
- Necessary when bandwidth for update is limited.

Questions:

- Is it possible to support all the features for software update reliability and security on delta update pattern?
- What needs to be changed in the image building and update process?

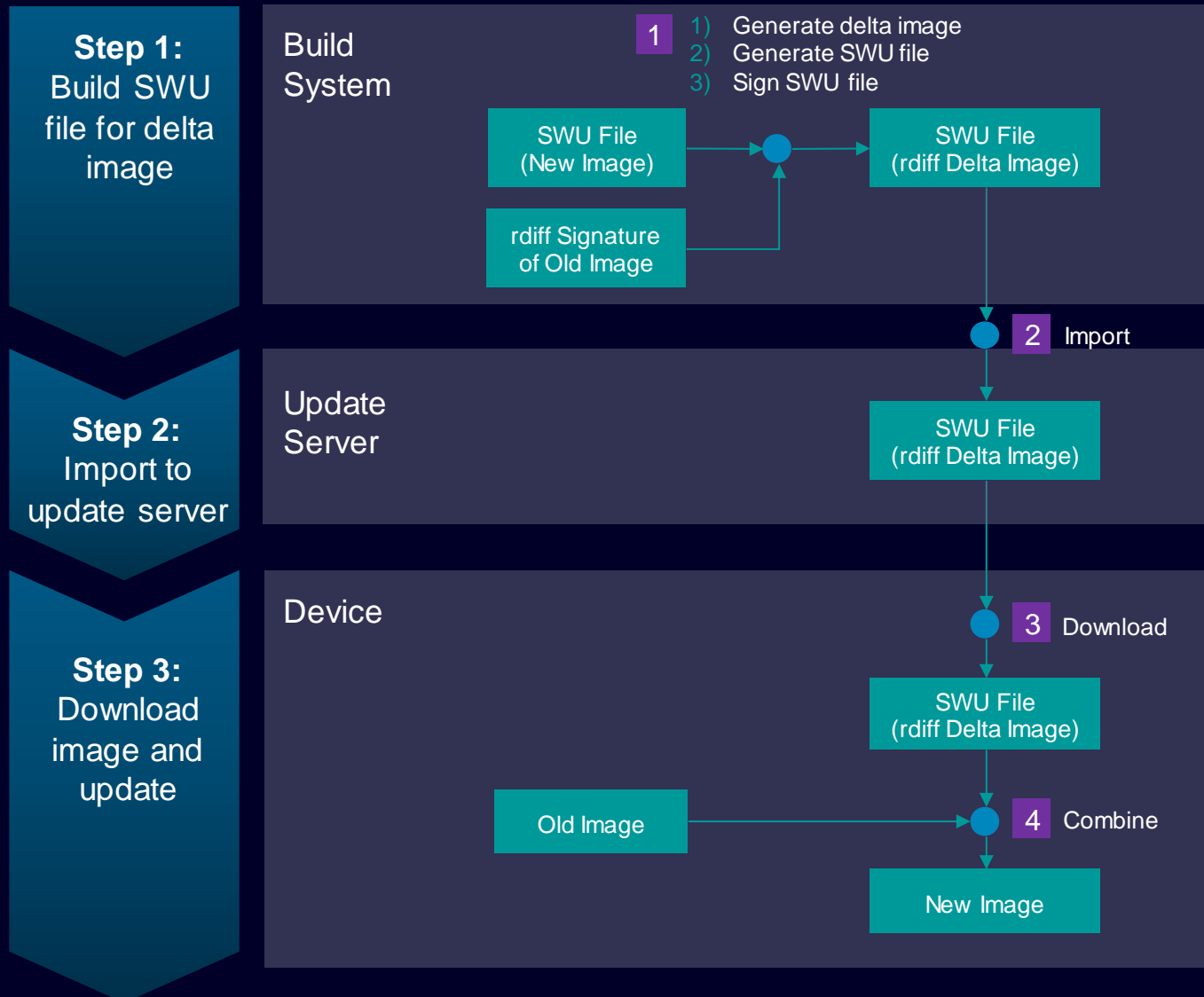
Delta Algorithms: librsync, Zchunk, casync



[1] <https://github.com/zchunk/zchunk>, [2] <https://github.com/librsync/librsync>, [3] <https://github.com/systemd/casync>
In depth comparison by Toshiba: https://docs.google.com/presentation/d/16iMgqzKczvWTufkWF_EwzaUMNhsQAmoi/

rdiff handler for SWUupdate

Delta Update based on librsync / rdiff – pre-generated



Description:

- Generate rdiff delta image in build system, from which the rdiff signature of old images should be accessible.
- Only support limited versions of delta update.

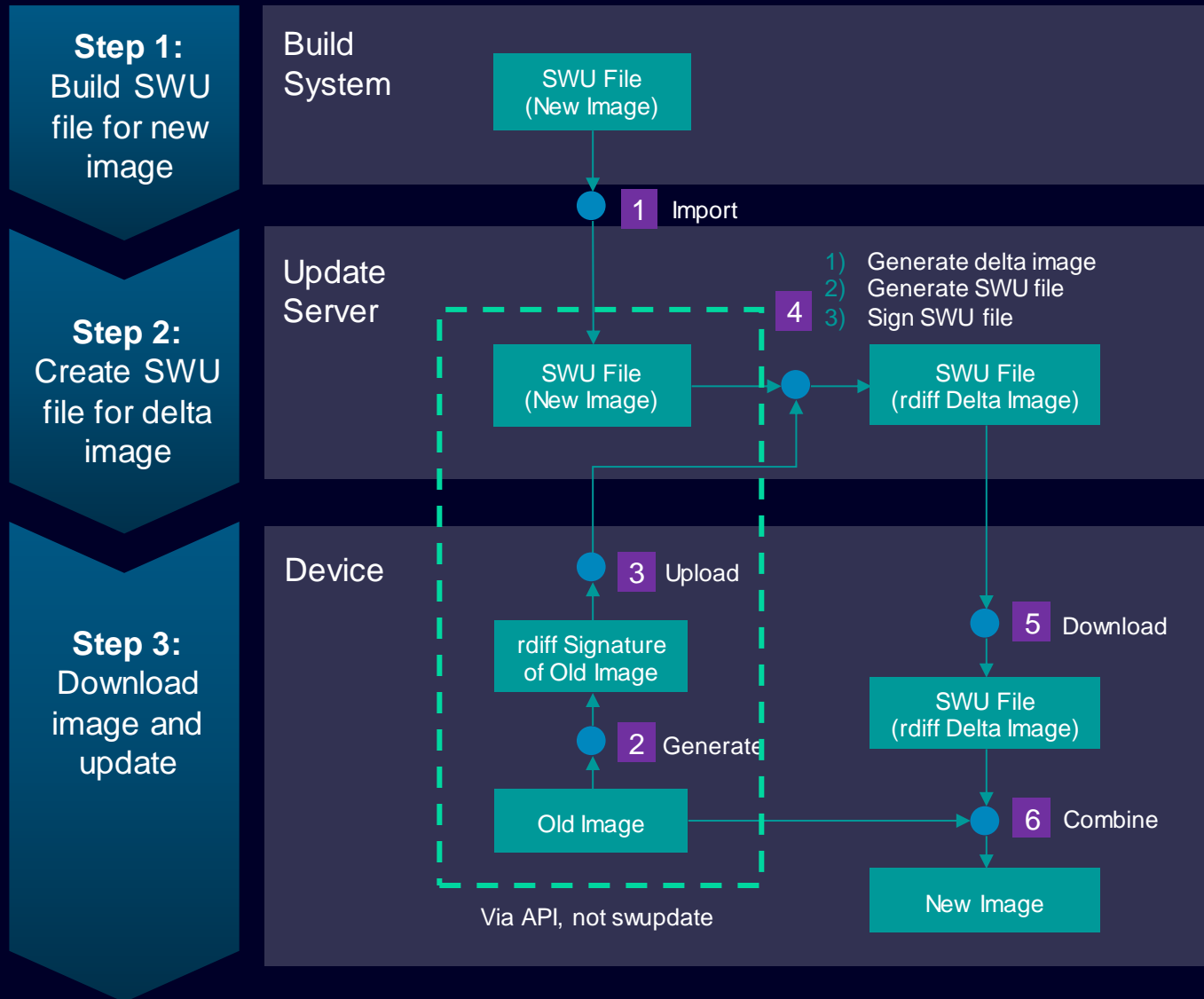
Pros:

- No change needed in update server
- SWU file signing is done in build system which avoids exposing key to update server
- SWU file download process already supported by SWUpdate.
- Offline update possible (e.g., via USB drive)

Cons:

- Only incremental updates are supported (e.g. 1.0 -> 1.1 -> 1.2, but not 1.0 -> 1.2)

Delta Update based on librsync / rdiff – on-demand generated



Description:

- Generate rdiff delta image in update server, based on rdiff signature of old image from device.

Pros:

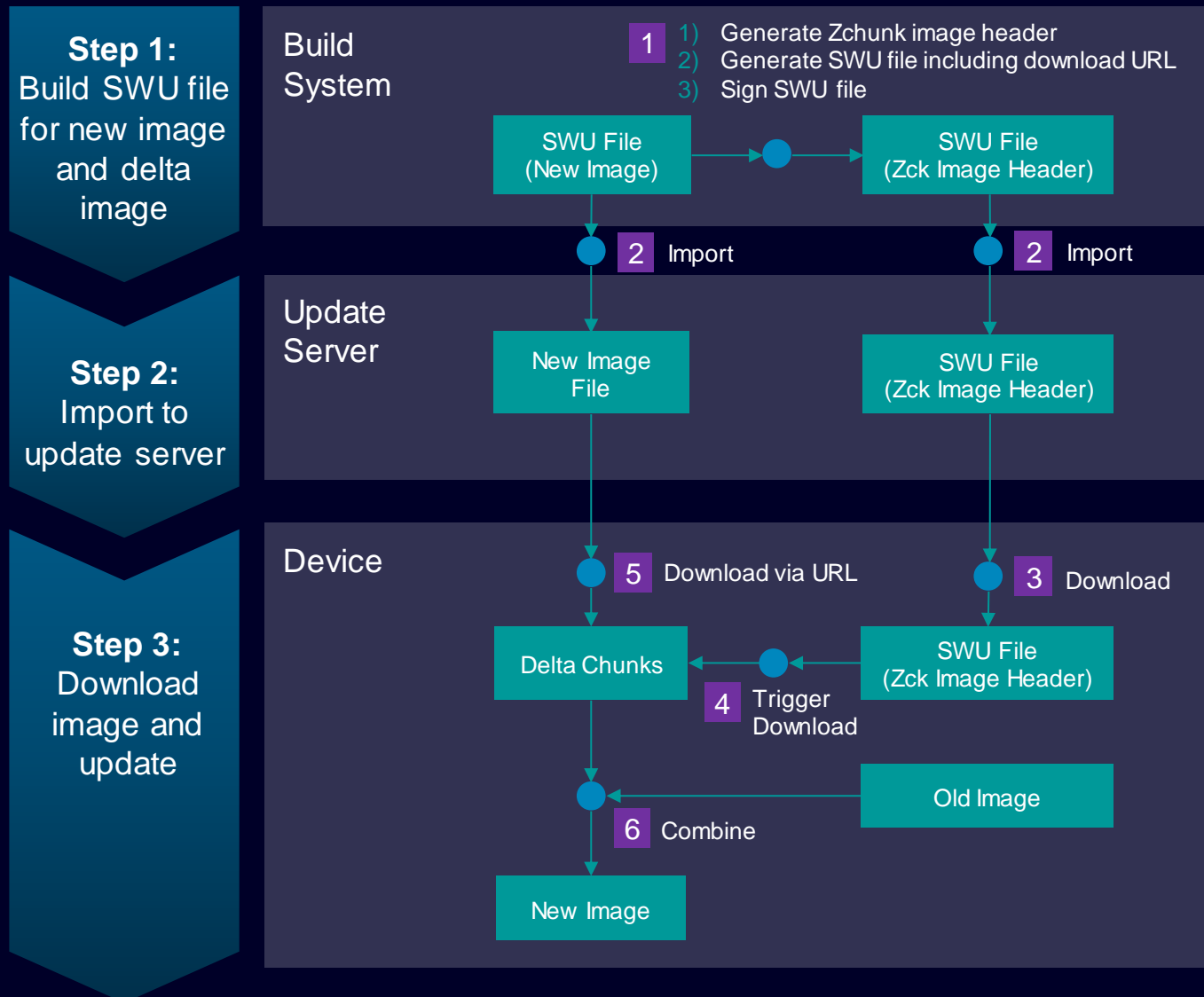
- Flexibly support different versions of images from devices.
- No change needed in build system.

Cons:

- Need to upload rdiff signature file from device, which is not part of original SWUpdate workflow.
- SWU file for delta image need to be signed in update server, which might cause security issue due to exposure of certificate/key for image signing.

Zchunk handler for SWUUpdate

Delta Update based on Zchunk



Preconditions:

- Generate Zchunk delta image in build system.
- Provide web service for downloading Zchunk delta chunks.

Pros:

- Flexibly support different versions of images from devices.
- SWU file signing can be in build system which can avoid security issue in update server.
- SWU file download process already supported by SWUpdate.

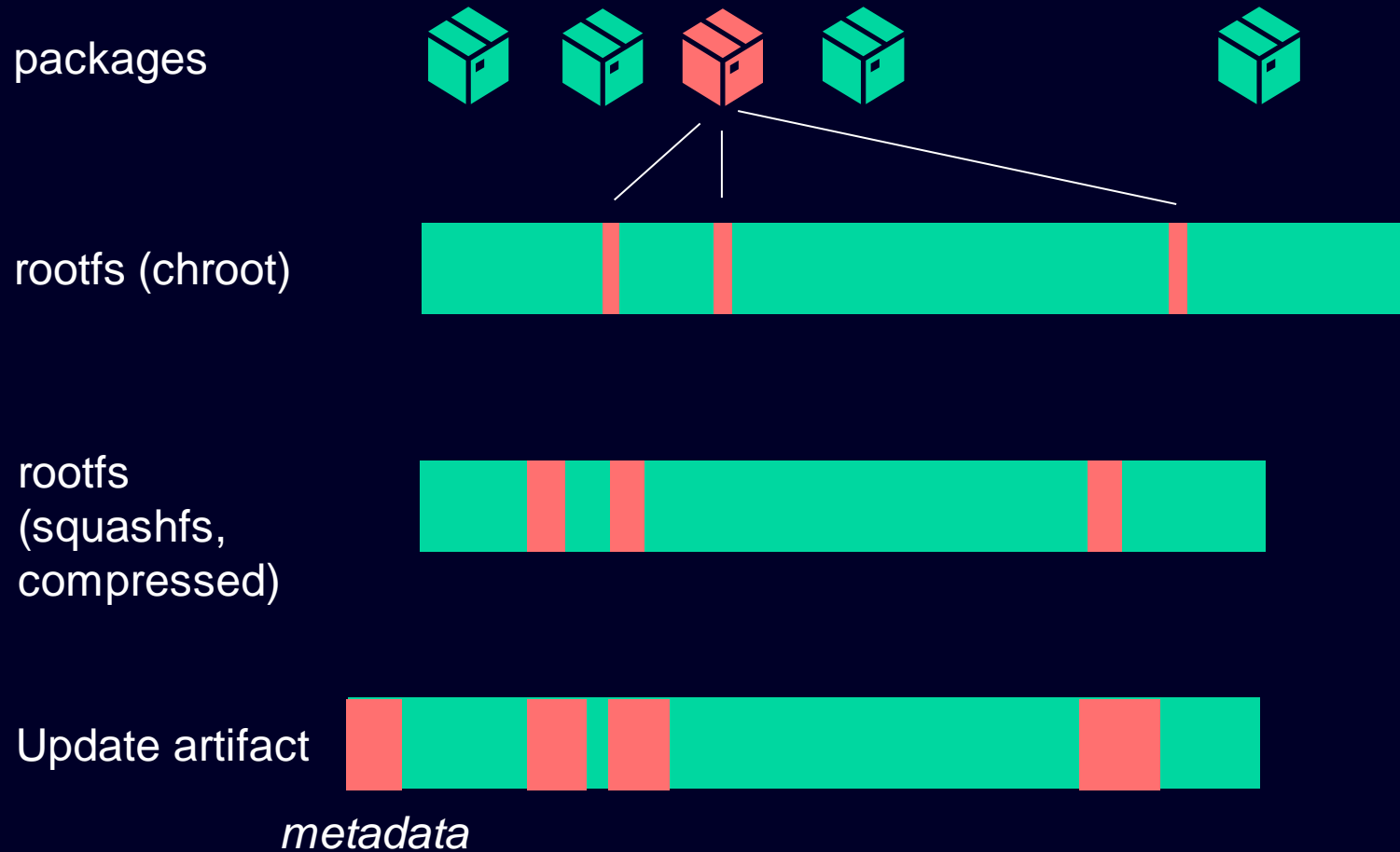
Cons:

- Need to import two image files and provide an additional web service for downloading delta chunks from update server, which is different from whole image update.
- Need to provide two update artifacts

— = *changed* — = *unchanged*

Optimizing the image building

Pre-conditions to make the update small



build reproducibly



Install reproducibly supported in isar-cip-core



compress continuously (use rsyncable algorithms)

Don't compress or use rsyncable algorithms. Use reasonable chunk size

Delta update in practice

Support in the SW Ecosystem?

isar-cip-core

- Required components ✓
- Generation of rdiff and zchunk artifacts ✓
- Released? "next" branch, v1.4-rc1



CIVIL
INFRASTRUCTURE
PLATFORM

SWUpdate + Handler

- SWUpdate ✓
- Suricata LUA Handler ✓
- Handler chaining for Round-Robin Handler with rdiff Handler & Delta Update ✓



Backends?

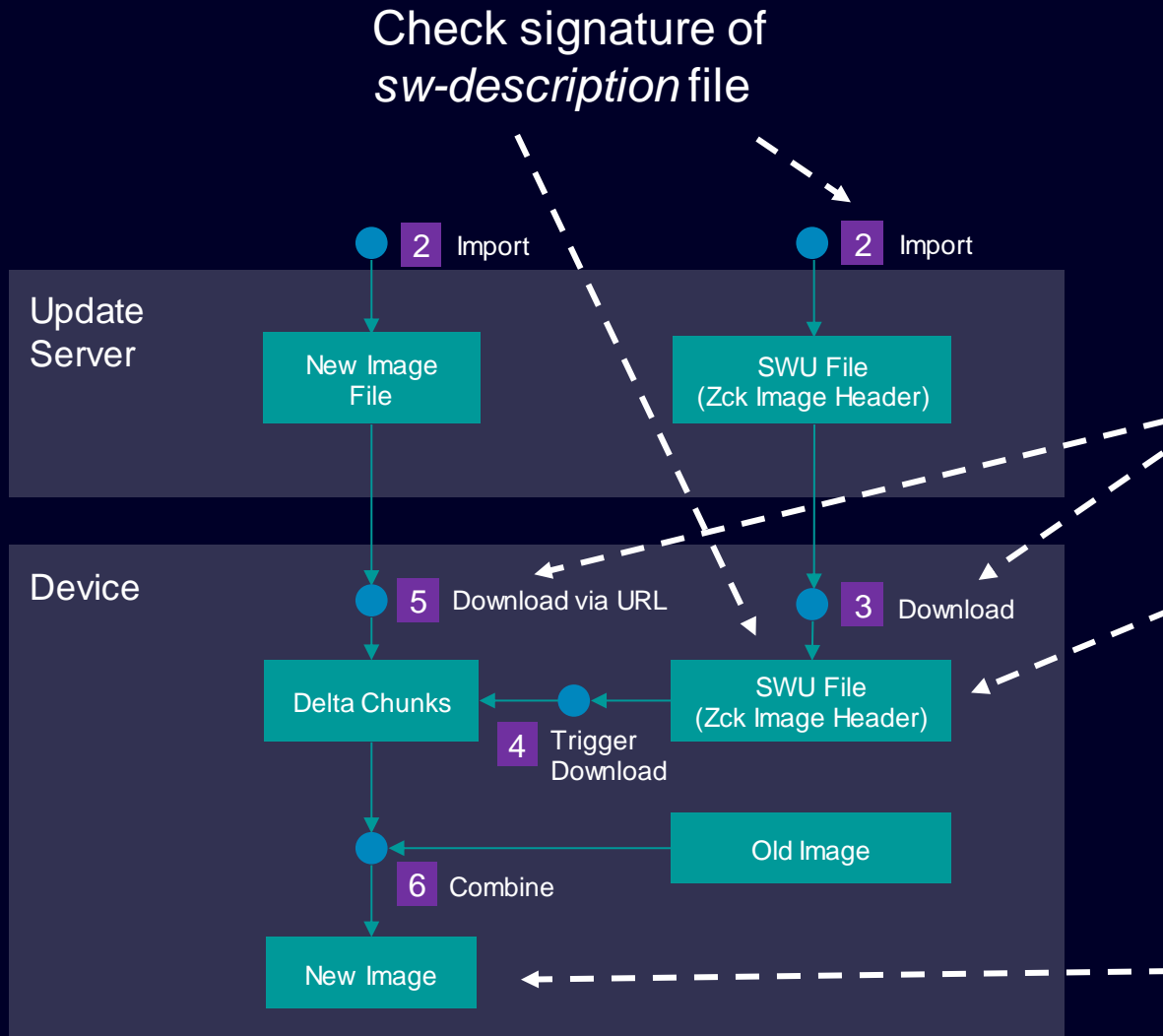
- rdiff: No backend change required ✓
- Zchunk:
 - Hawkbit (✗)
 - WFX ✓



Delta Updates & Security

FAQ: Securing the update path

- Step 1: Build the image
- Step 2: Import to update server
- Step 3: Download image and update



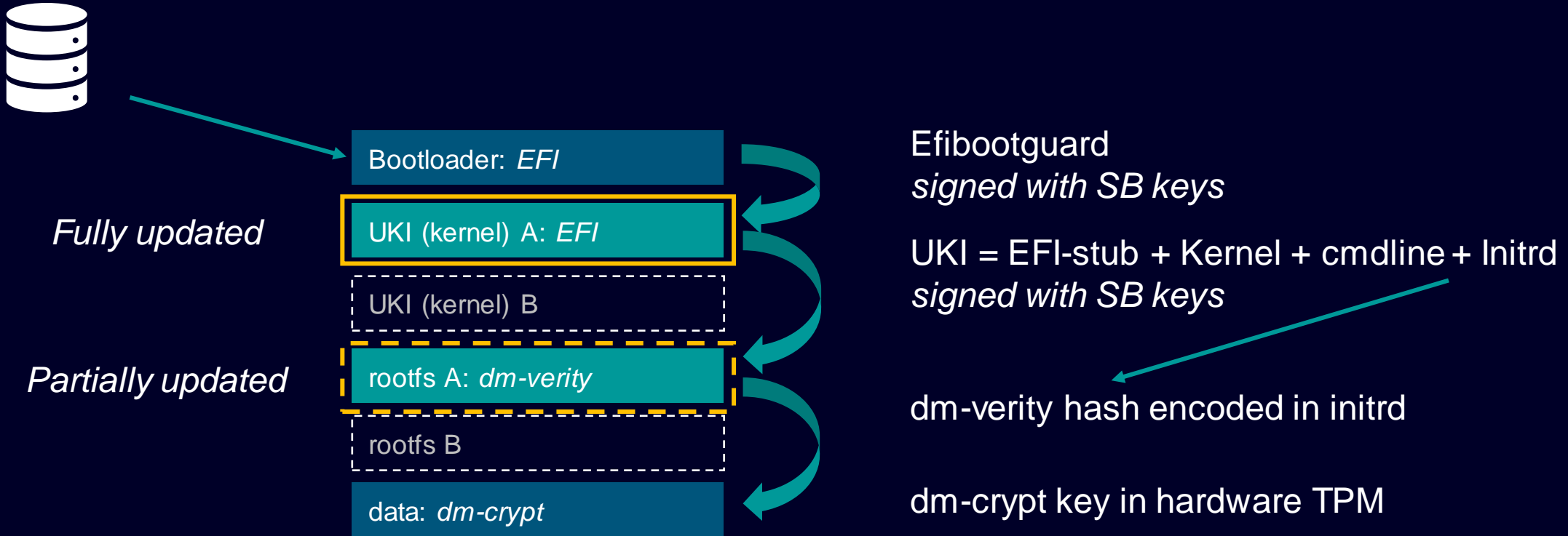
Downloads can be done via local reverse proxy

URL + zchunk header encoded in signed *sw-description*

dm-verity container checksums (see next slide)

FAQ: Delta Updates & Secure Boot

Why does it work?



Summary & Kudos

- **More resource efficient** when the whole image is big, and changes are small
- **Reliable:** Supports dual-copy update pattern
- **No impact on security:** secure boot, disk encryption, and dm-verity based integrity verification.
- **No free lunch:** Image generation and update workflows are different between rdiff and Zchunk, users need to decide which one to use according to requirements and use cases.

Siemens AG¹ / Siemens Ltd. China²

Felix Moessbauer¹

Jan Kiszka¹

Wang Qi²

Toshiba

Adithya Balakumar

Dinesh Kumar

Kazuhiro Hayashi



CIVIL
INFRASTRUCTURE
PLATFORM