



———— CIVIL ————
INFRASTRUCTURE
———— PLATFORM ————

Implementing UEFI-based Secure Boot + OTA Update for Embedded ARM Devices

Jan Kiszka and Dr. Christian Storm, Siemens AG
Embedded Linux Conference Europe, September 14th 2022

About Us ...



Jan Kiszka



[<Jan.Kiszka@siemens.com>](mailto: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

Christian Storm



[<Christian.Storm@siemens.com>](mailto:Christian.Storm@siemens.com)

- Siemens Technology
- (In-house) Embedded Linux consultant & developer
- Member of CIP Work Group Software Update
- Contributor to OSS projects

... That Topic, Again !? ;)



THE **LINUX** FOUNDATION
PROJECTS

**CIVIL
INFRASTRUCTURE
PLATFORM**

Secure Boot and Over-the-Air Updates – That's simple, no?

Jan Kiszka, Siemens AG
Embedded Linux Conference North America 2020, June 30th 2020

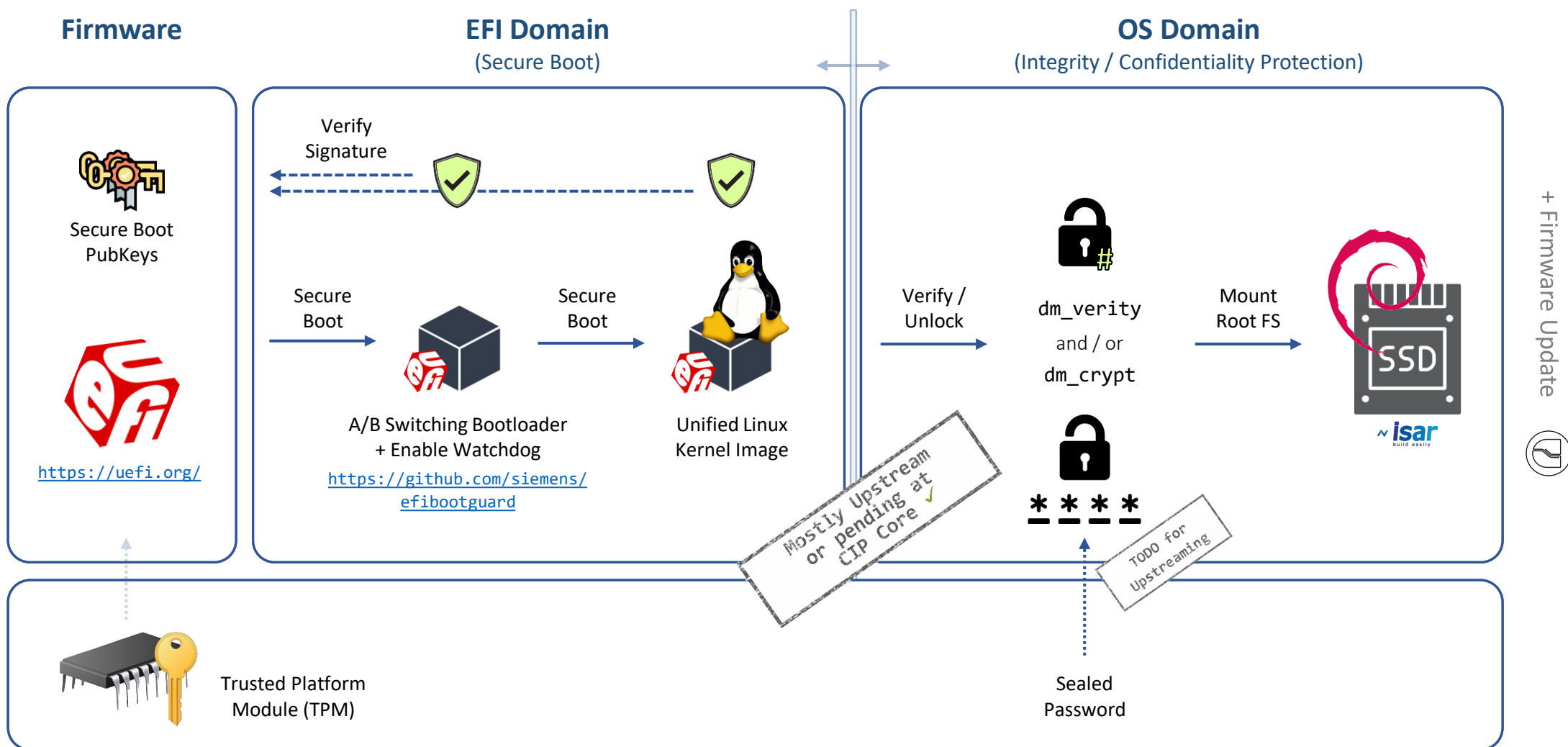
[https://events.linuxfoundation.org/archive/2020/
embedded-linux-conference-north-america/](https://events.linuxfoundation.org/archive/2020/embedded-linux-conference-north-america/)

From ROM Firmware to Over-the-Air Updates ...

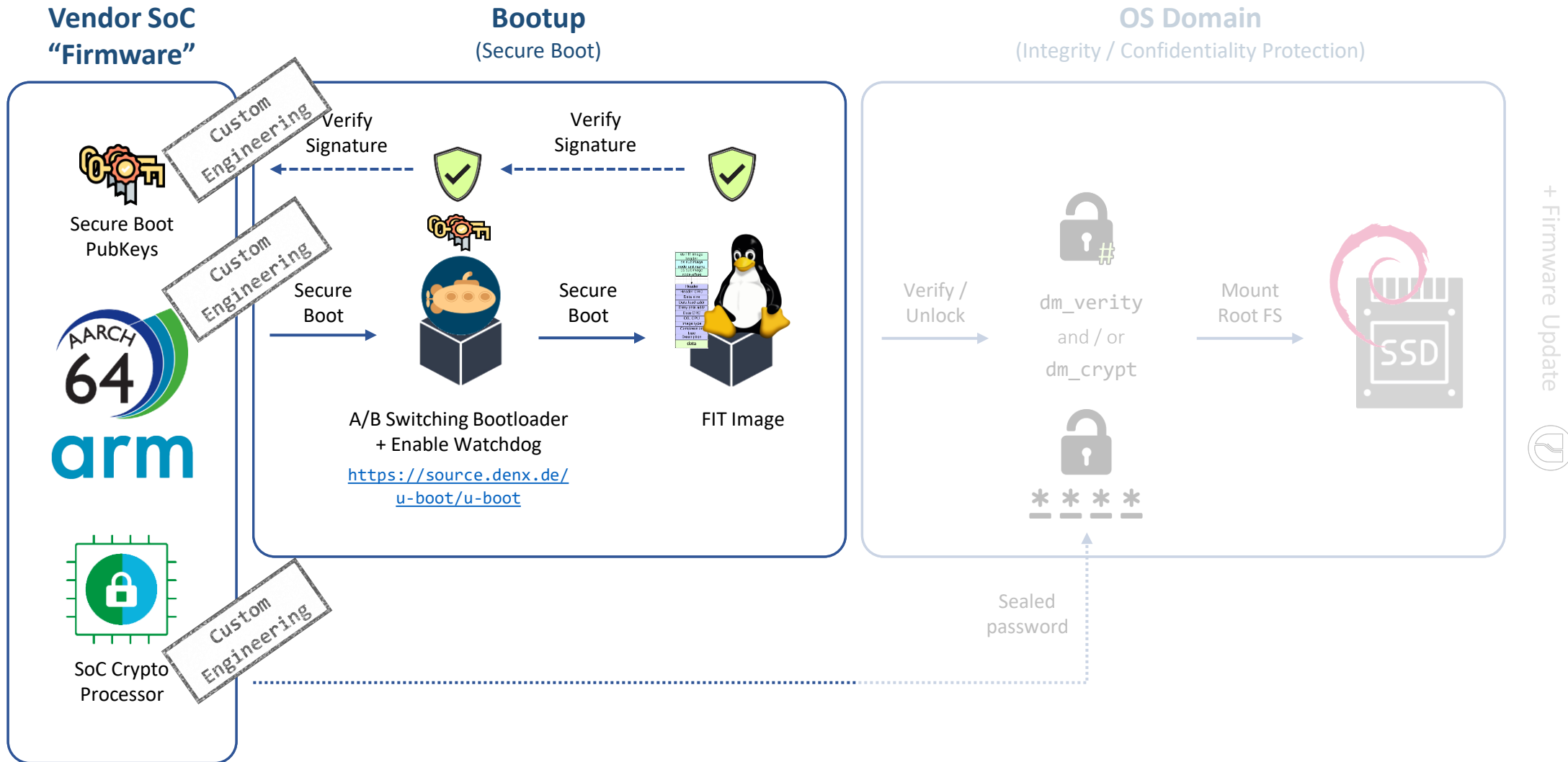


- ... we've come a long way:
 - Connectivity is standard
 - Security is standard
 - Security updates are inevitable (mandated by IEC 62443 e.g.)
 - Unattended updates are required
 - Robust updates (atomic, roll-back capable)
- Having integrated + CIP Core-upstreamed it for x86/UEFI, it's time to bring ARM{,64} on par ...

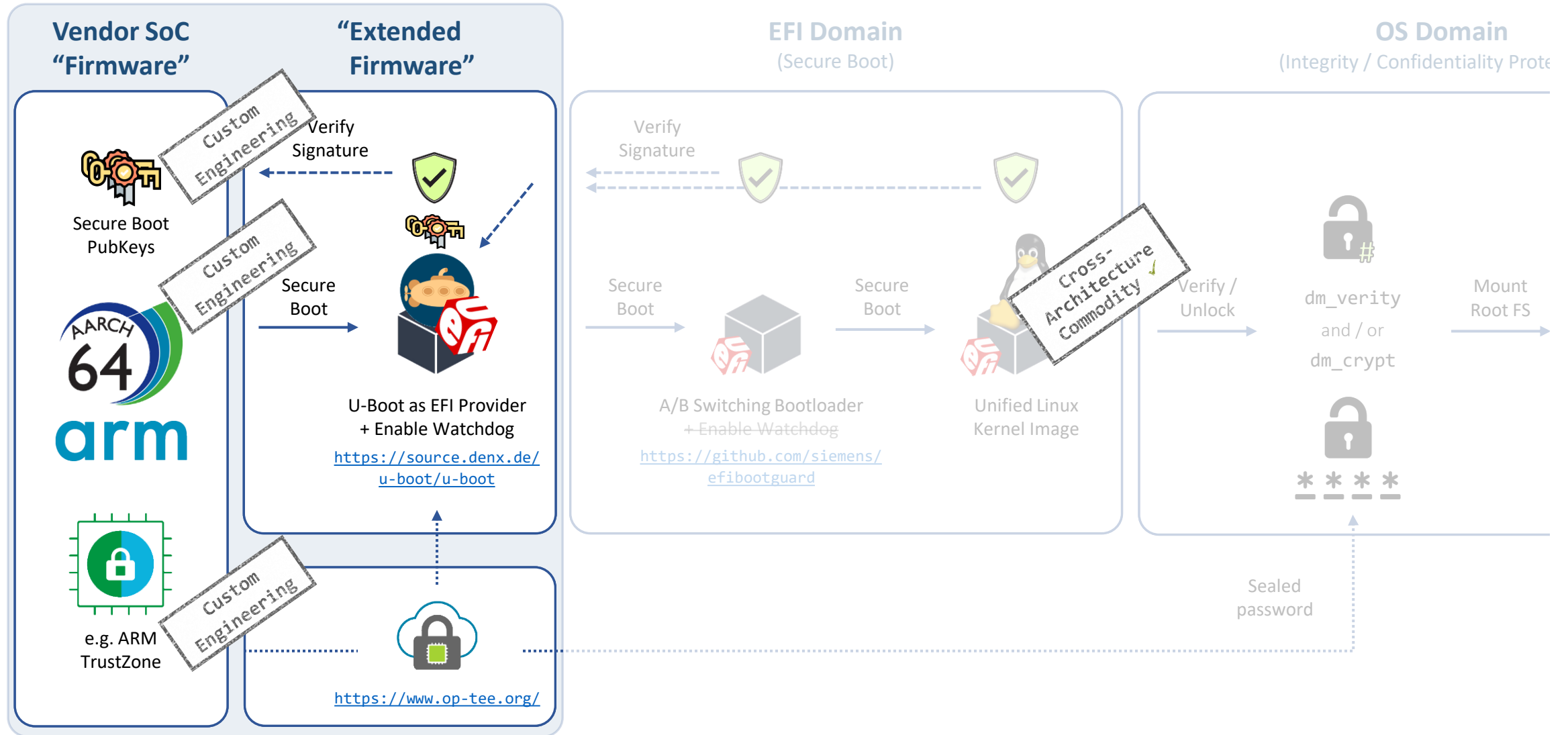
Current State on x86/UEFI



State on ARM64: Vendor / SoC Custom Engineering



... So Reduce Specifics & Increase Cross-Arch Commodity



UEFI Providers for Non-x86



- **EDK II**

- UEFI reference and common base on x86
- ARM and ARM64 support
- Not commonly supported by SoC vendors, limited driver support



- **U-Boot**

- De-facto firmware standard, vendor-backed, plenty of drivers
- Fairly advanced UEFI support, but not yet broadly in use

Enabling U-Boot with UEFI



- CONFIG_EFI=y often default
- Secure boot: use compiled-in keys (static chain) or efivars in secure storage [1]
- Don't forget hardening
 - Lock console (e.g. via CONFIG_BOOTDELAY=-2)
 - Limit to UEFI boot (e.g. via CONFIG_BOOTCOMMAND)
 - Turn off unused features, specifically filesystems
 - Make U-Boot env read-only or limit writing to selected, uncritical variables
 - See isar-cip-core [2] and meta-iot2050 [3]
- Sign/lock firmware artifacts via vendor-specific hardware mechanisms
- Start hardware watchdog (to be taken over by Linux)
- Test, don't trust! [4]



CC BY-SA 4.0, Heinrich Schuchardt

- 1) <https://u-boot.readthedocs.io/en/latest/develop/uefi/uefi.html#configuring-uefi-secure-boot>
- 2) <https://gitlab.com/cip-project/cip-core/isar-cip-core/-/blob/master/recipes-bsp/u-boot>
- 3) <https://github.com/siemens/meta-iot2050/tree/master/recipes-bsp/u-boot>
- 4) <https://source.denx.de/u-boot/u-boot/-/commit/634f6b2fb1056021fba603ccb7488d1864787576>

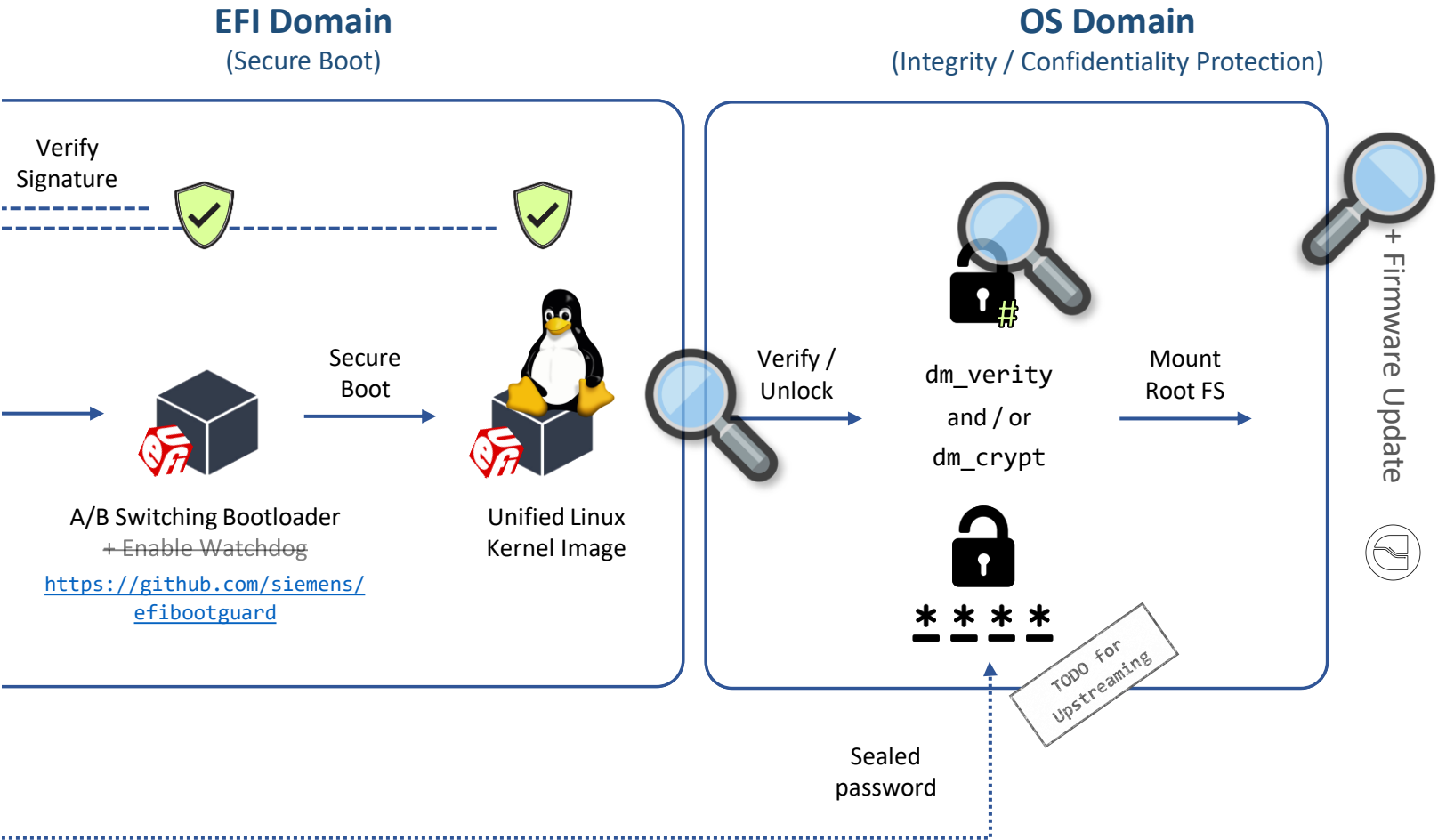
EFI Boot Guard, Unified Kernel Images



- EFI Boot Guard on ARM64 and ARM ✓
- Unified kernel image via systemd
 - Commonly shipped binutils do not work yet (objcopy for PE...)
 - Can bundle one (1) device tree
 - Some scenarios require one image for multiple boards (=DTBs)
- EFI Boot Guard as unified kernel image generator
 - Python-based image generation
 - Choose compatible DTB from multiple options
- Both unified kernel stubs need EFI_DT_FIXUP_PROTOCOL
 - U-Boot only, still a to-do for EDK II
 - EBBR standardization pending (see <https://github.com/ARM-software/ebbr/issues/68>)

<https://github.com/siemens/efibootguard>

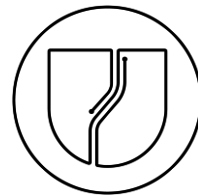
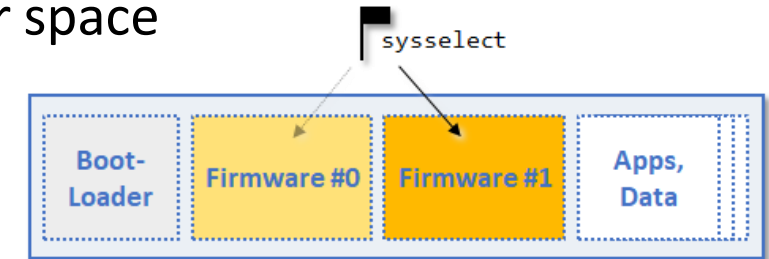
Cross-Arch Commodity: OS Domain + Firmware Update



- How to integrate Over-the-Air Firmware Update?
- How does a Unified Kernel Image find its Root Filesystem?
- How to realize Root Filesystem Integrity Protection?

Firmware + Software Update

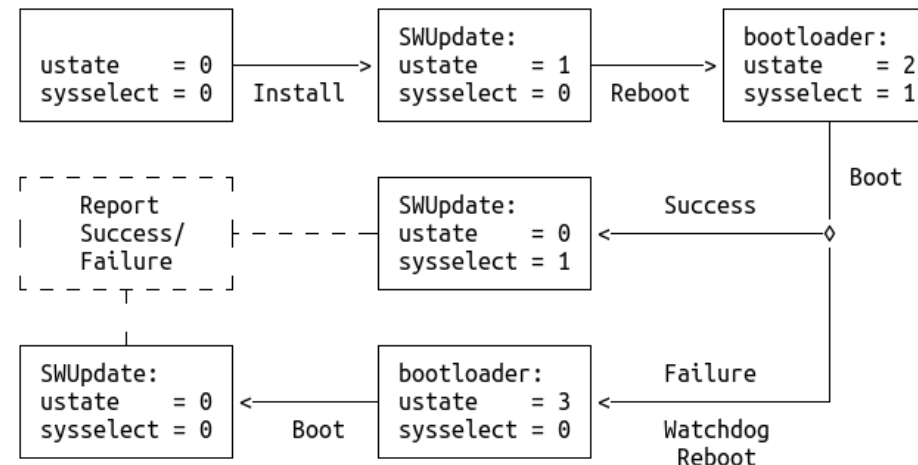
- Updateability – The one probably most important modern Product feature
- Commonly, an A/B scheme is used, favoring time/availability over space
- A robust solution deeply integrates with the Bootloader (arming the hardware watchdog + A/B switching, rollback)
- Pre-integrated in CIP Core
- Not only device “firmware” is updated via this mechanism ...



SWUpdate

A versatile, flexible, and extensible OSS on-device agent framework for Software Update on Unix Embedded Systems.

<https://swupdate.org/>



Matching an (Updated) Kernel to its Root Filesystem



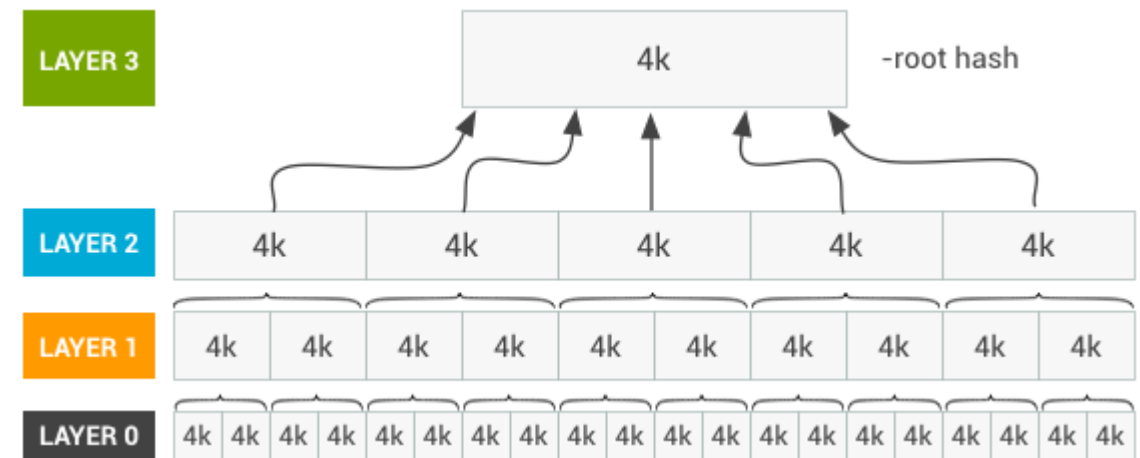
- A Kernel plus its Root Filesystem is one (update) package!
- There's a kernel option: `root=/dev/partition` (or `PARTUUID` or ...)
- But it needs to be run-time dynamic A/B agnostic ...
... and cannot use untrusted sensitive data (Secure Boot)
- ✗ Filesystem UUID – depends on Filesystem support, not always available
- ✗ PARTUUID – Partition table write, also not always available
- ✓ Add custom `IMAGE_UUID=<UUID>` in Root Filesystem's `/etc/os-release`,
iterate over filesystems & match it in `initramfs` with encoded `<UUID>`
– which is integrity-protected by Secure Boot + signed Unified Kernel Image



Root Filesystem Integrity Protection



- Immutable read-only Filesystem (squashfs) + Authenticity enforcement
- Device-Mapper (dm): Virtual layering of block devices
- dm-verity: Integrity checking of block devices using Merkle Tree
 - Each node is the hash of its children, except leaves = actual data
 - Hashes “trickle up” to the root hash: changed data \Rightarrow changed root hash
- Matching Kernel \leftrightarrow Root Filesystem with a twist: $\langle \text{UUID} \rangle = \text{root hash}$



<https://source.android.com/docs/security/verifiedboot/dm-verity>

Upstream First! at CIP



- There are many building blocks to align proper
- Upstream integration is codified “big picture” shared knowledge
- Touches many CIP projects as a natural fit
- Testing and Reusability: Upstream First! at CIP

1	2	3	4	5	6	
SLTS kernel	Real-time	Testing	CIP Core	Security WG	Software update WG	
✓	✓	✓	✓	✓	✓	Industrial grade
✓		✓	✓		✓	Sustainability
✓		✓	✓	✓	✓	Security

CIP Projects / Workgroups (WG) and their scopes

Reference Integration and Usage



- isar-cip-core

- Pre-integration of presented concepts
- Demo on QEMU (x86/ARM64/ARM) and BeagleBone Black



<https://gitlab.com/cip-project/cip-core/isar-cip-core>

- meta-iot2050

- Example Debian image for SIMATIC IOT2050
- Contains device-specific delta to isar-cip-core
- Eat your own dogfood ;)

<https://github.com/siemens/meta-iot2050>



Summary



- Secure Boot + robust software updates = No rocket science!
... but too much for a casual Friday to not ruin your weekend(s)
- Most puzzle pieces are available and Open Source Software
... alignment, configuration, and vertical integration is the merit
- CIP strives to provide reusable building blocks
 - Blueprints / Pre-integrations
 - Testing and long-term maintenance
- Join us at cip-dev@lists.cip-project.org
- Let's make it (even more) upstream commodity!

Questions !?

