The Cost of Security: Measuring and Reducing Boot-Time Impact

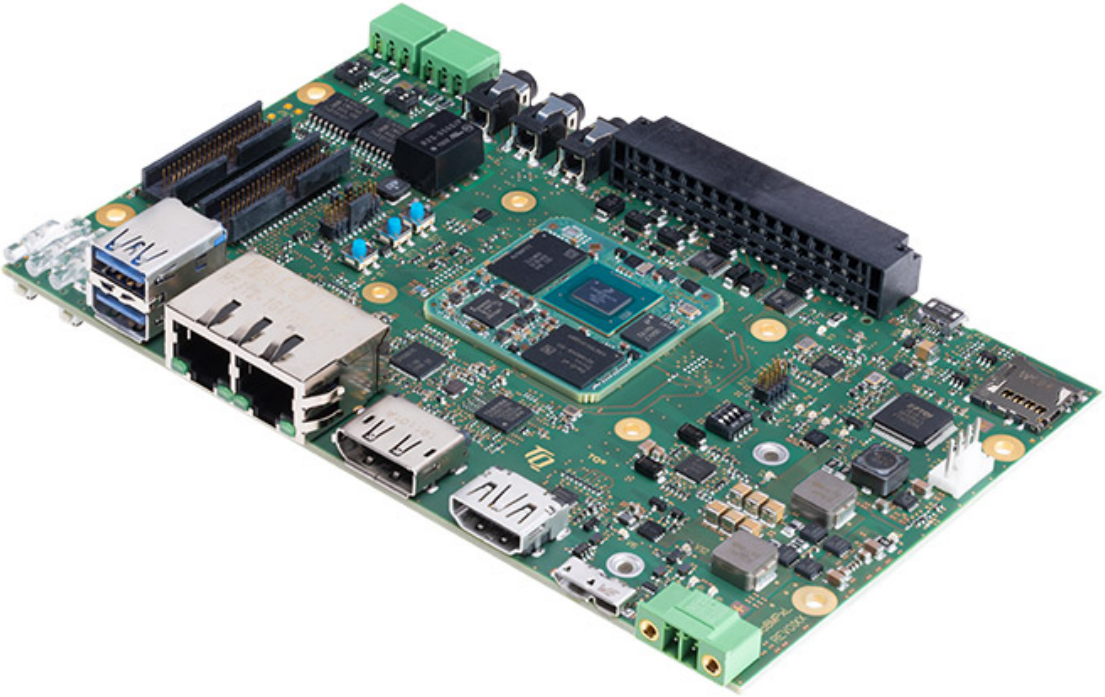
Michael Olbrich – m.olbrich@pengutronix.de

About Me

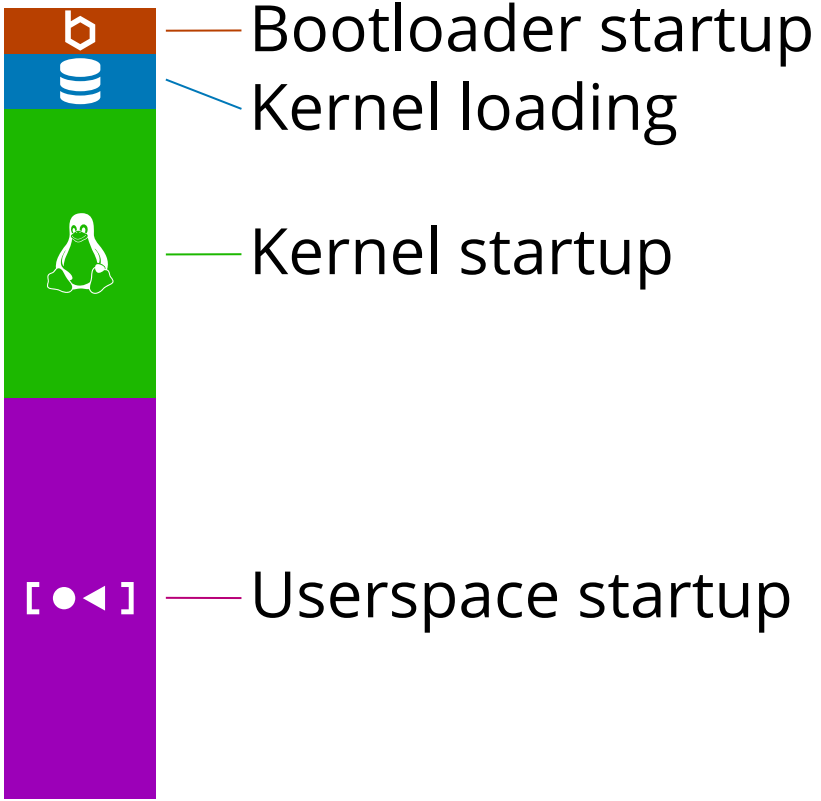
- Michael Olbrich
- Embedded Linux developer
- @ Pengutronix since 2006
- Graphics team lead

Test Hardware

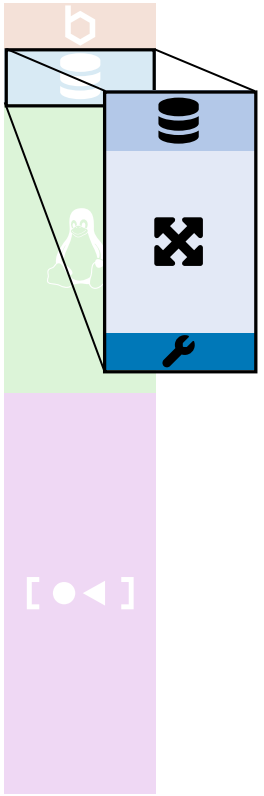
- TQ MBa8MPxL
- i.MX 8M Plus
  - 4x Cortex A53 @ 1.8GHz
  - 2 GiB LPDDR4 RAM
- eMMC up to HS400



Boot Stages


Bootloader startup
Kernel loading
Kernel startup
Userspace startup

Boot Stages



Read kernel from eMMC (10.5 MB)

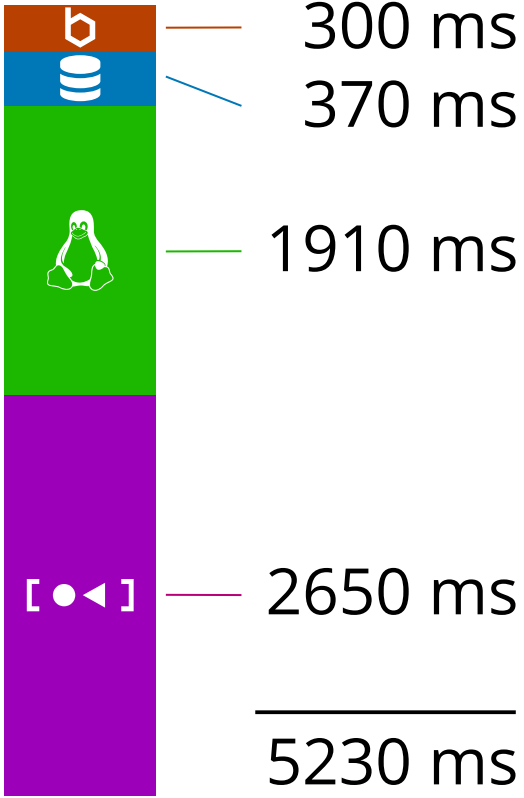
Extract lzo compressed kernel

Setup kernel for booting

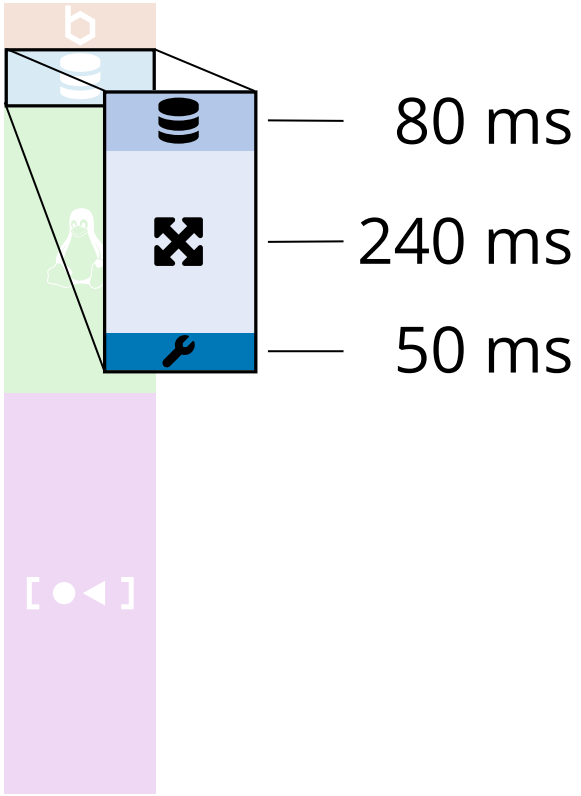
How to Measure Boot Time

- Timestamps from serial console output
- Kernel Timestamps (clock monotonic)
- Offset between:
  - Bootloader starts the kernel
  - Timestamp zero of clock monotonic
  - Here: 220 ms

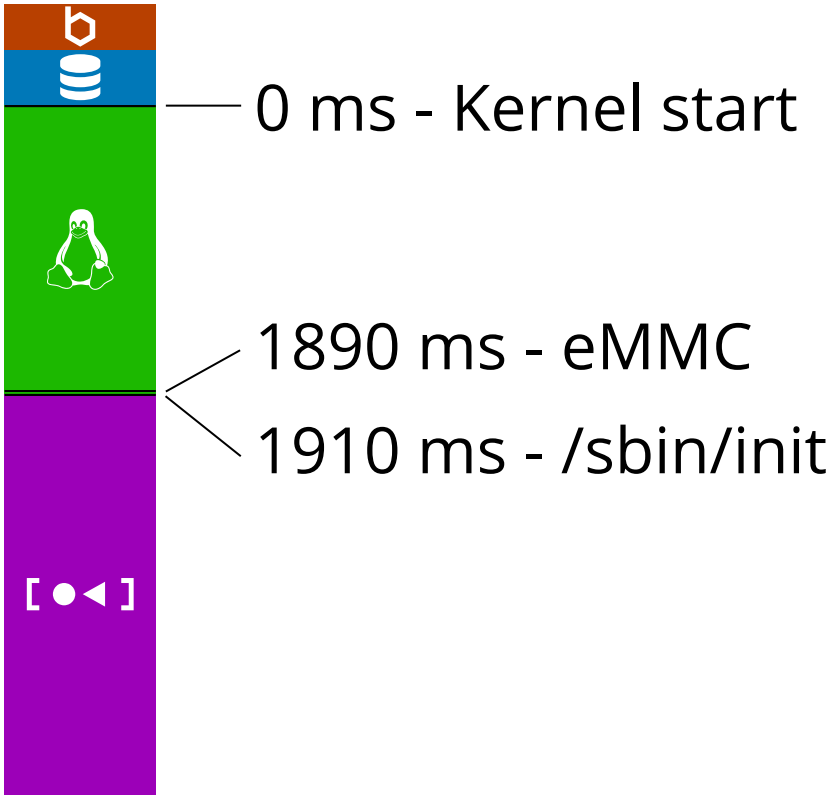
Reference Measurement



Reference Measurement



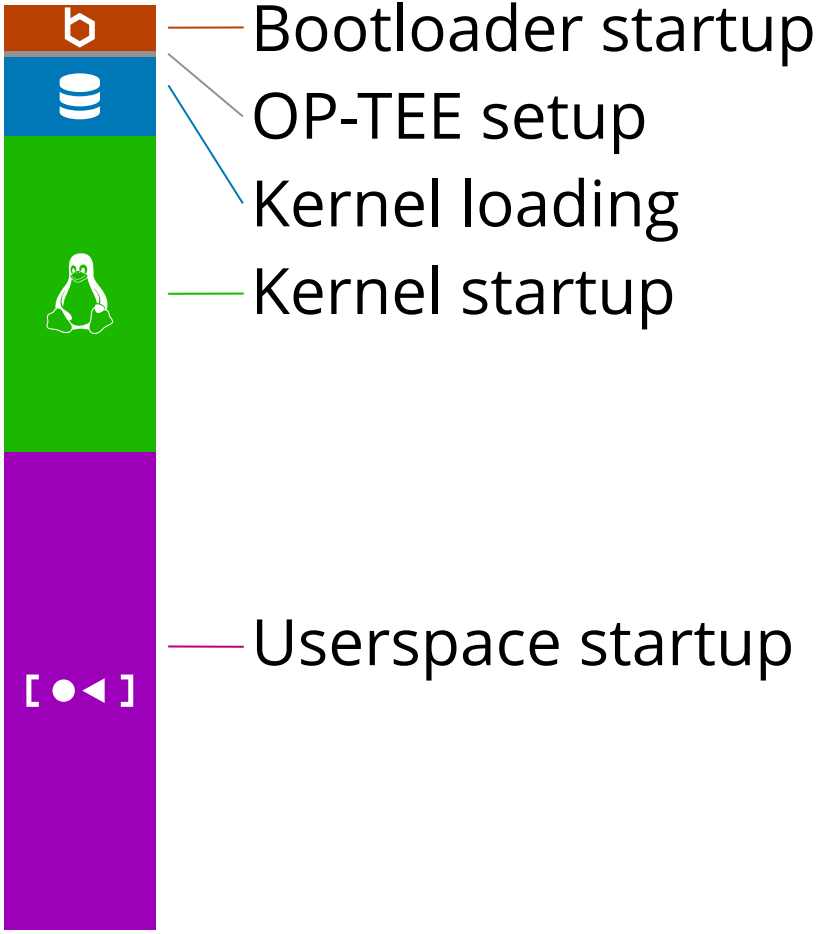
Reference Measurement



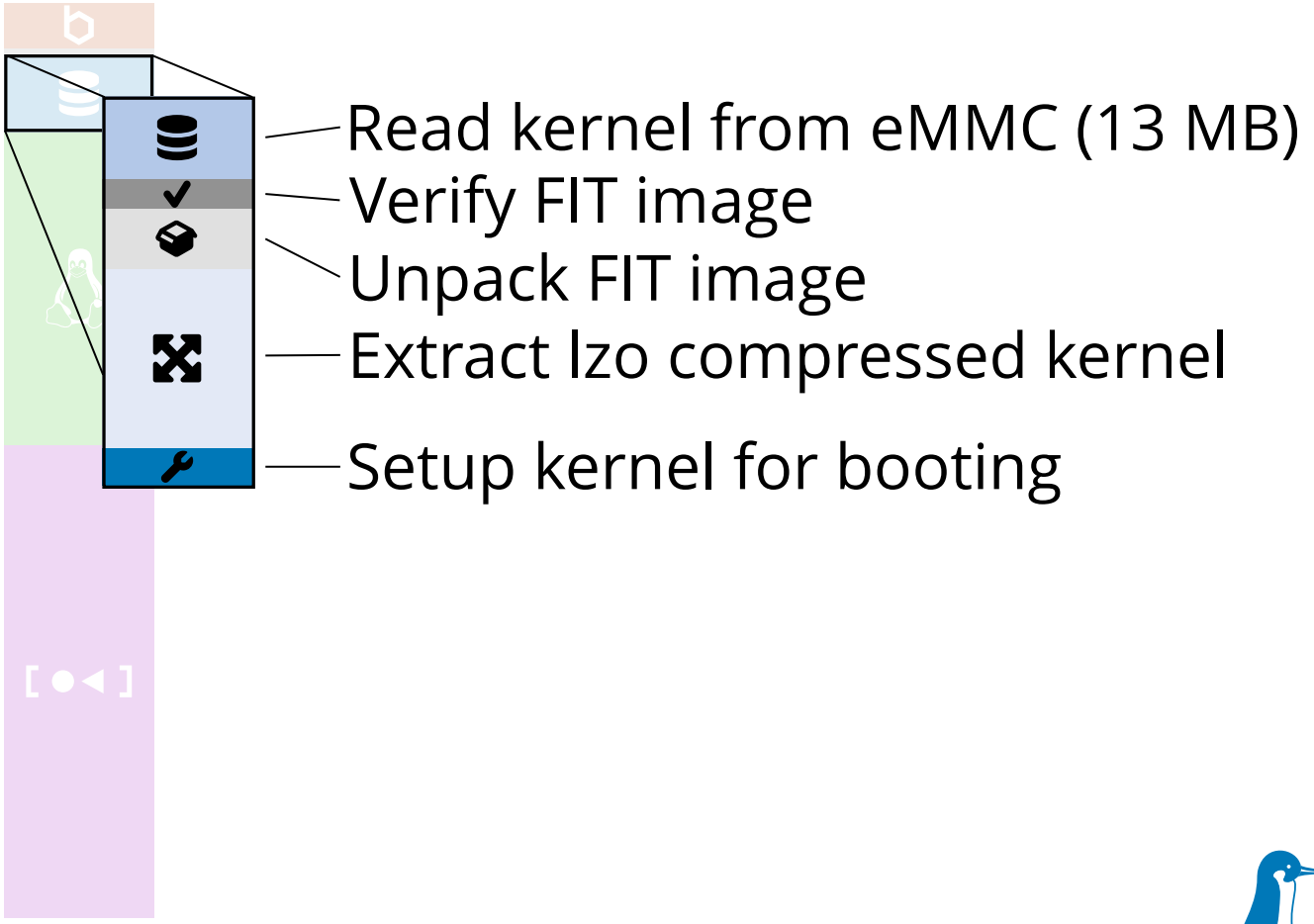
Extra Steps with Secure Boot

- OP-TEE must be loaded and initialized
- The kernel must be verified
  - Implemented with a signed FIT image
- The root filesystem must be verified
  - dm-verity is used for the rootfs
  - dm-verity is configured in an initramfs
  - Initramfs is created using the Yocto initramfs-framework

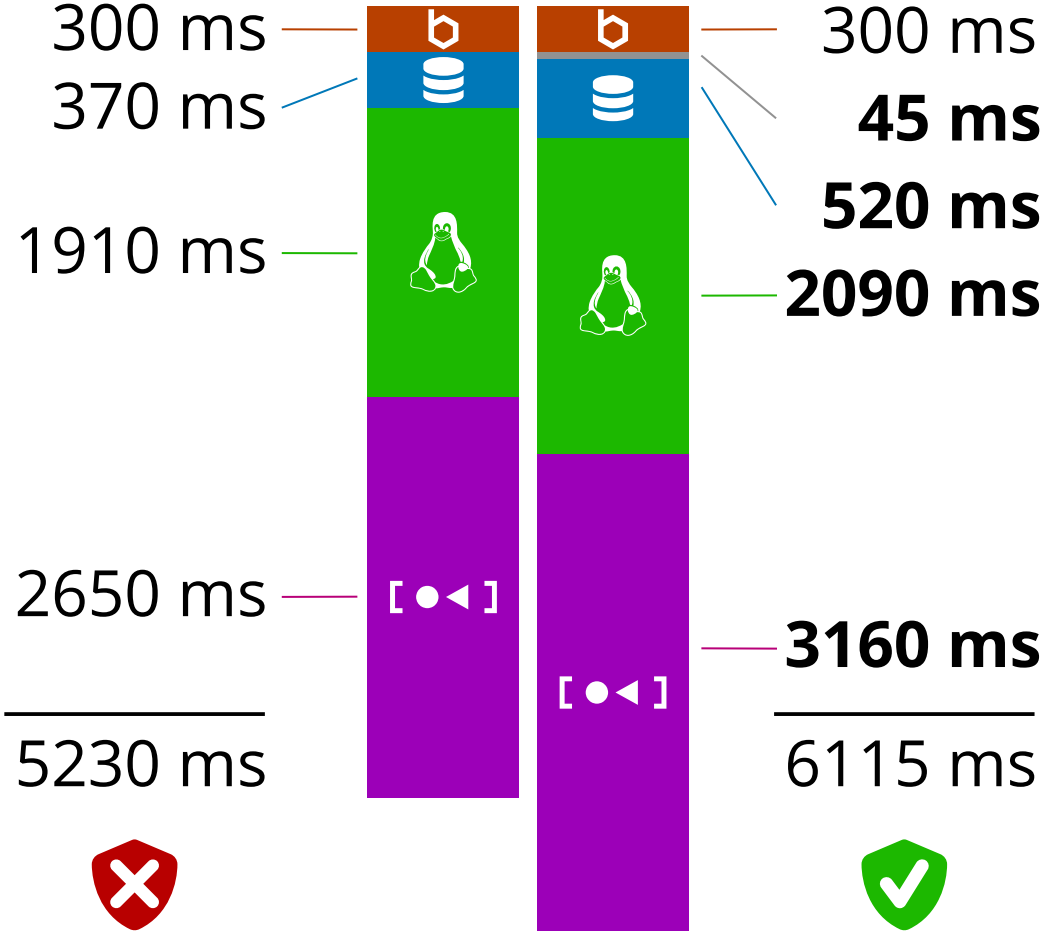
Boot Stages with Secure Boot



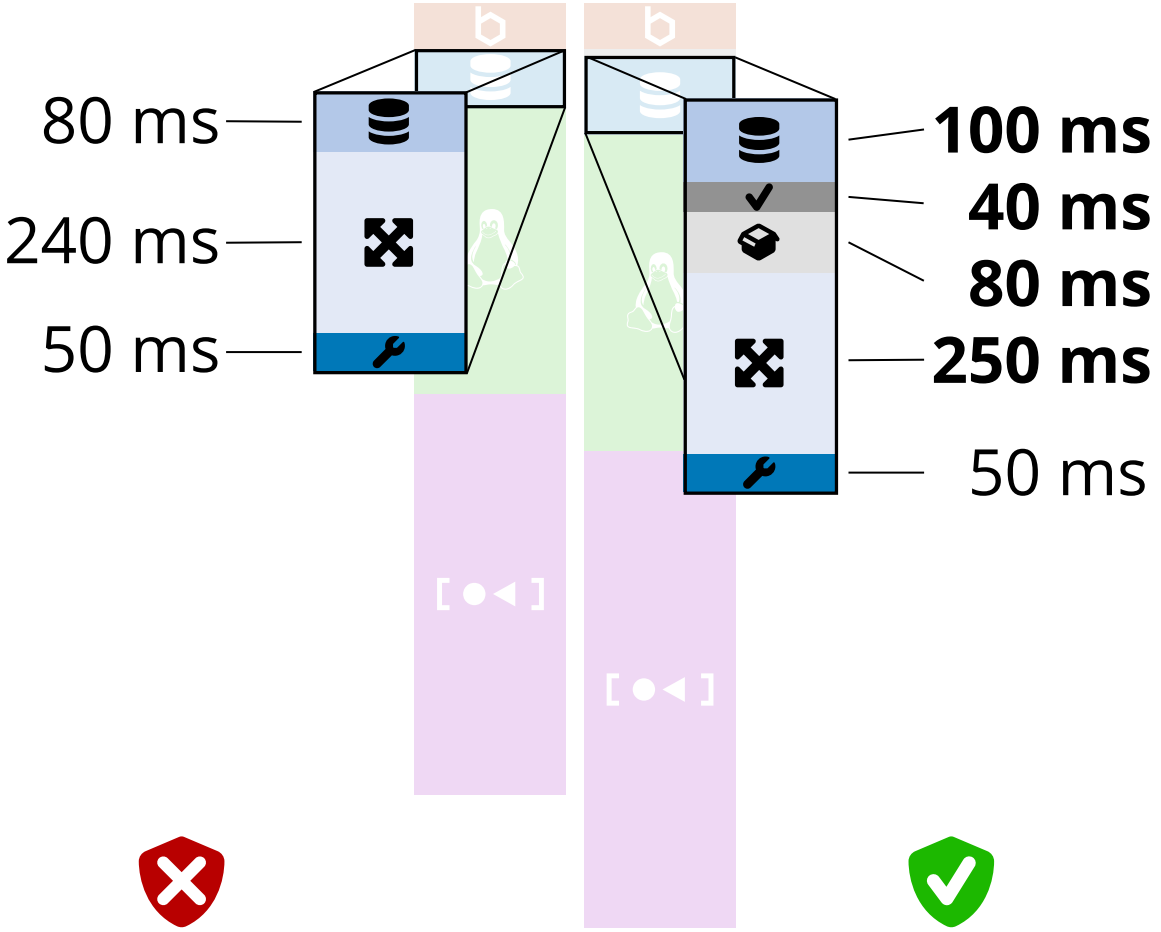
Boot Stages with Secure Boot



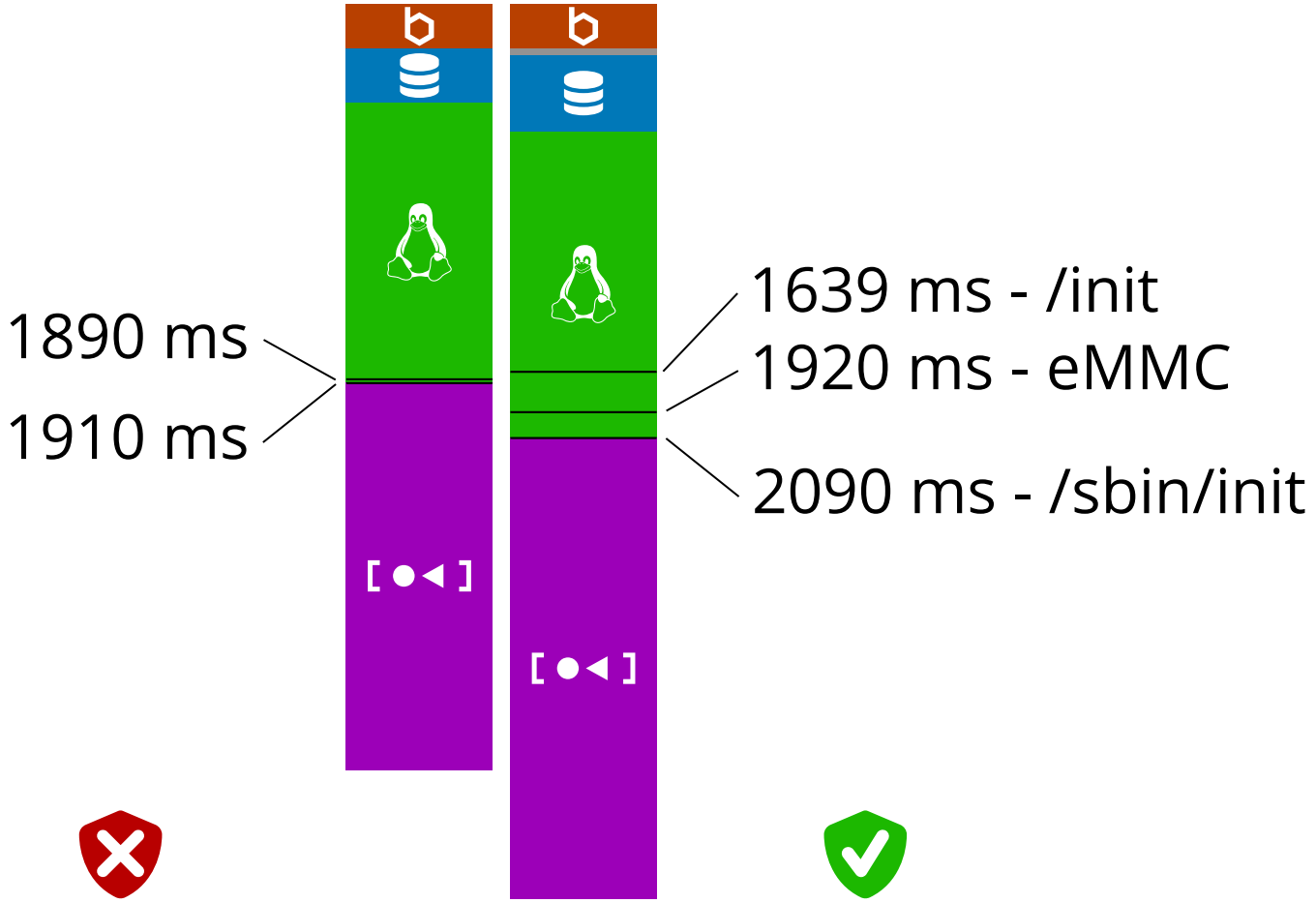
Secure Boot Measurement



Secure Boot Measurement



Secure Boot Measurement

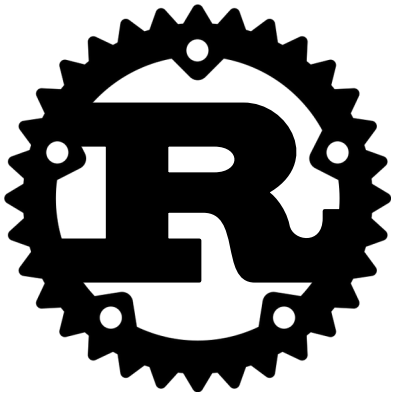


Optimization – Initramfs (old)

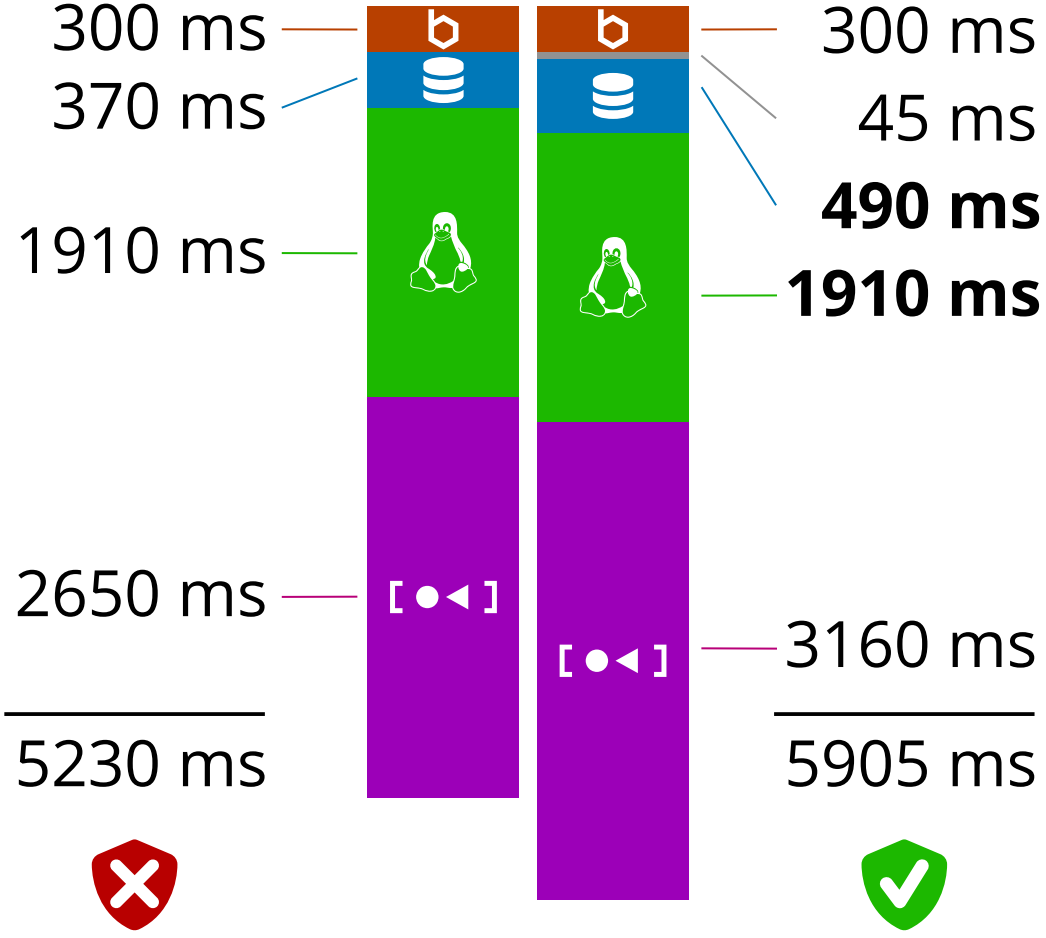
- Built with Yocto initramfs-framework
- 2.5 MB zstd compressed
- 7 MB uncompressed
- Busybox + dmsetup + required libraries
- Shell scripts

Optimization – Initramfs (new)

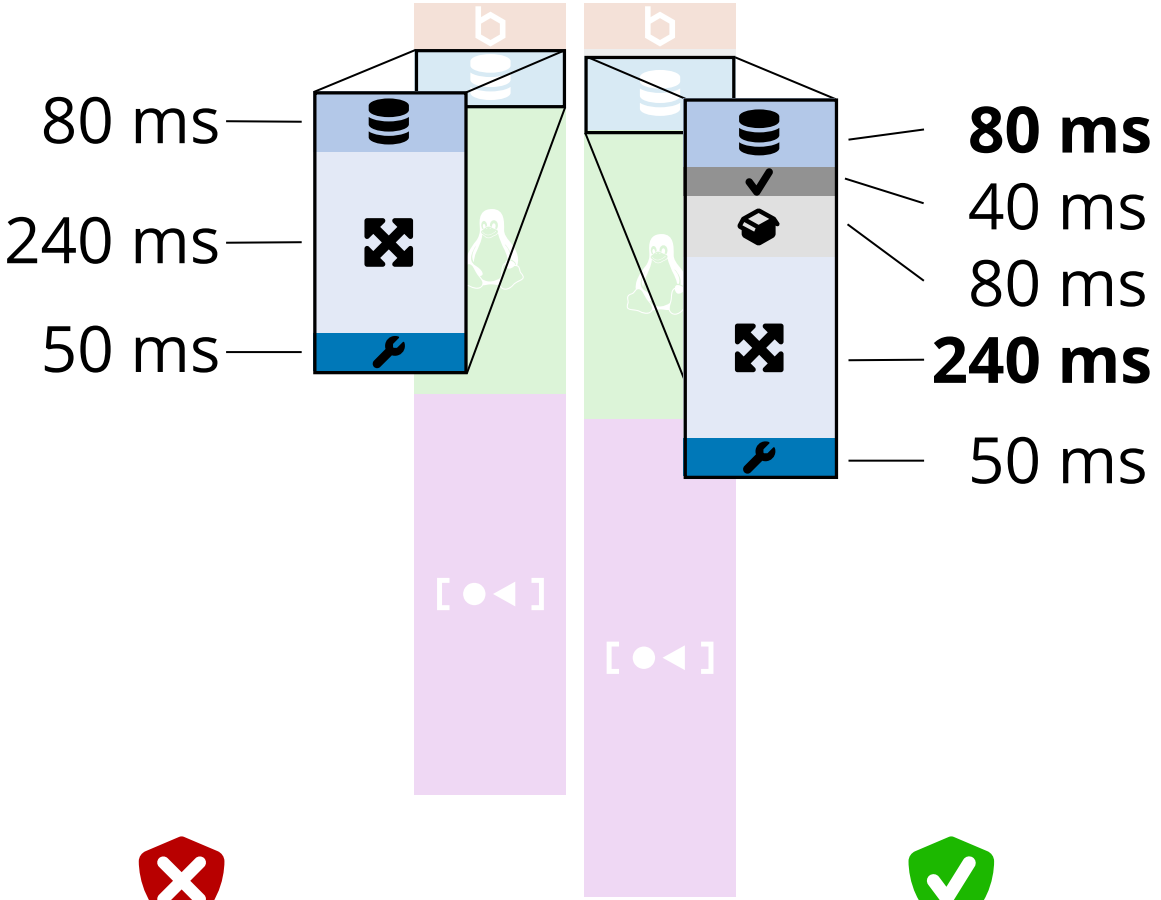
- Single binary initramfs
- 57 kB zstd compressed
- 130 kB uncompressed
- Written in Rust
- https://github.com/pengutronix/rsinit



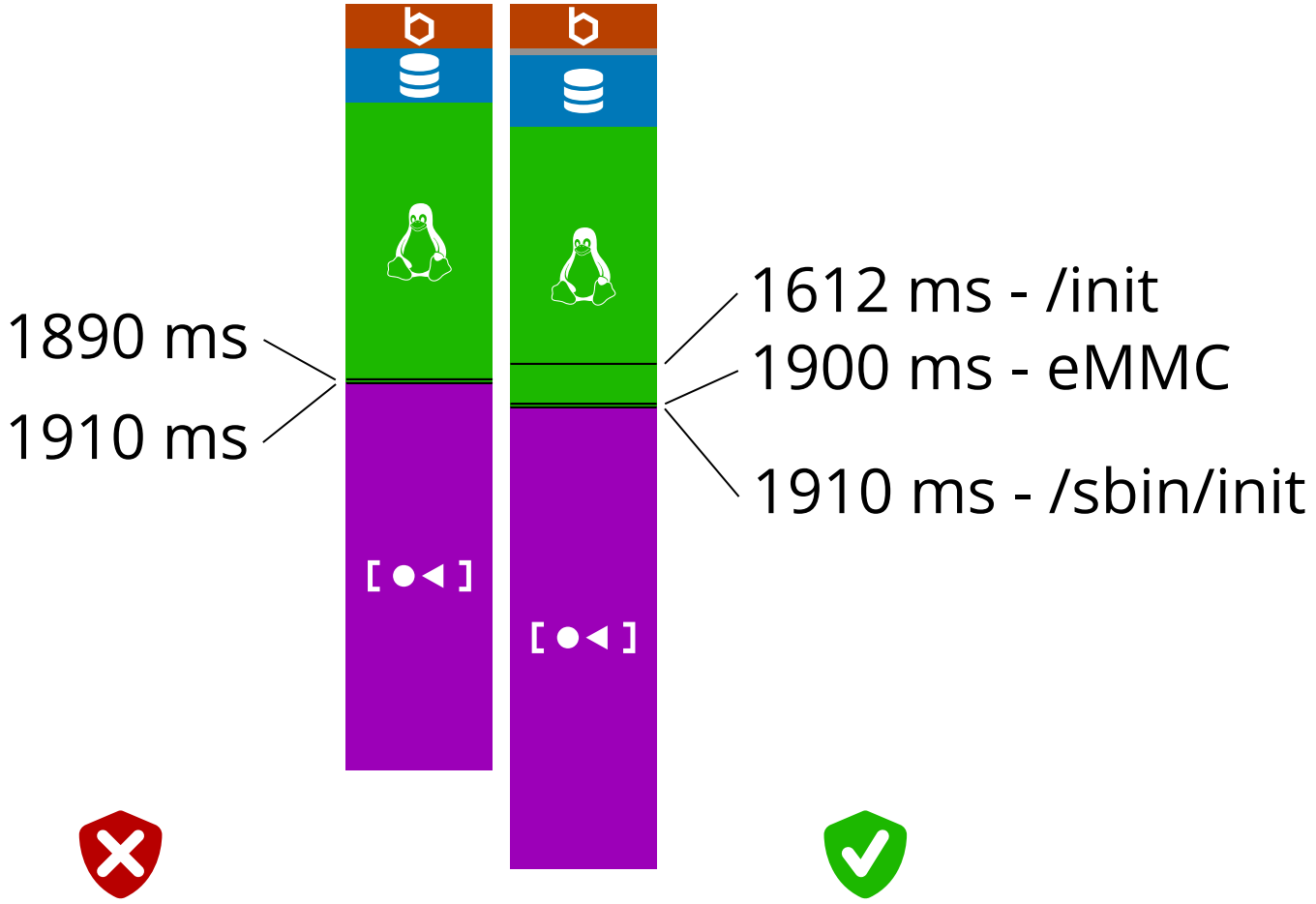
Secure Boot Measurement



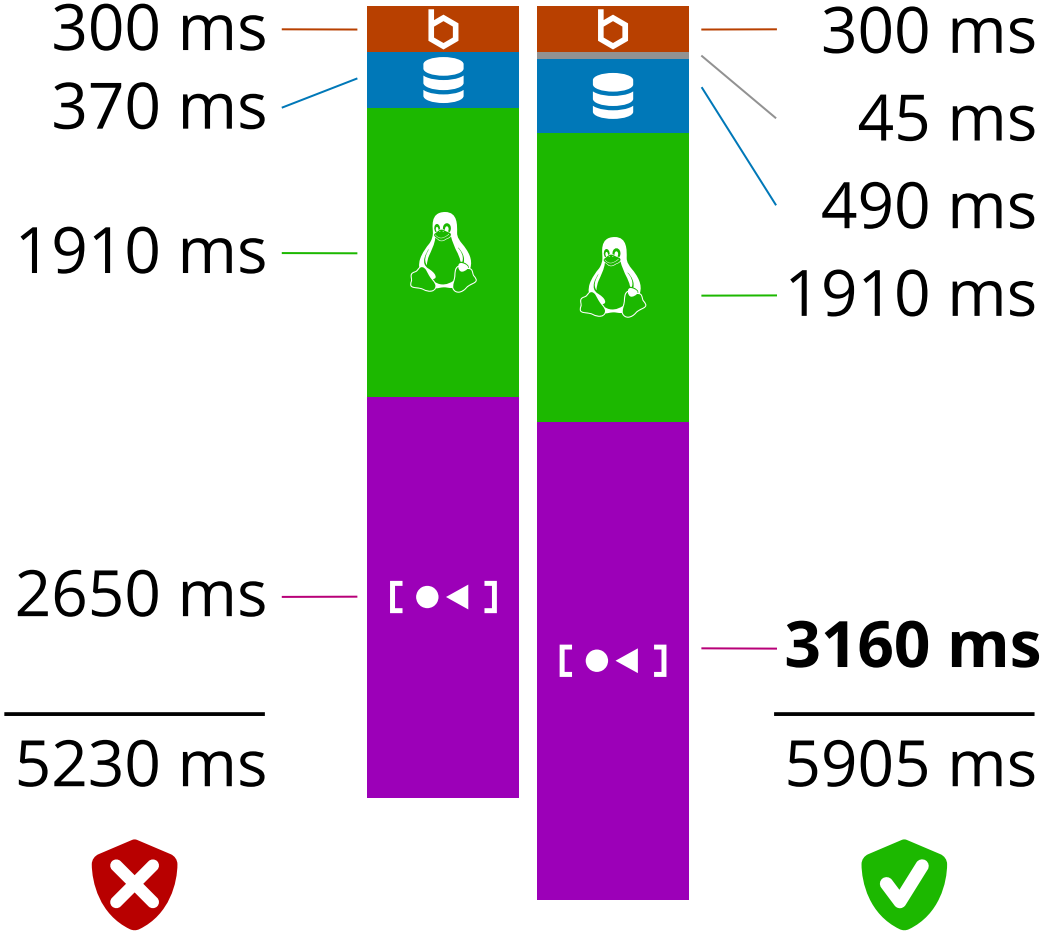
Secure Boot Measurement



Secure Boot Measurement



Secure Boot Measurement



dm-verity Overhead

From the Linux kernel log:

```
[...]
device-mapper: verity: sha256 using ahash "sha256-caam"
[...]
```

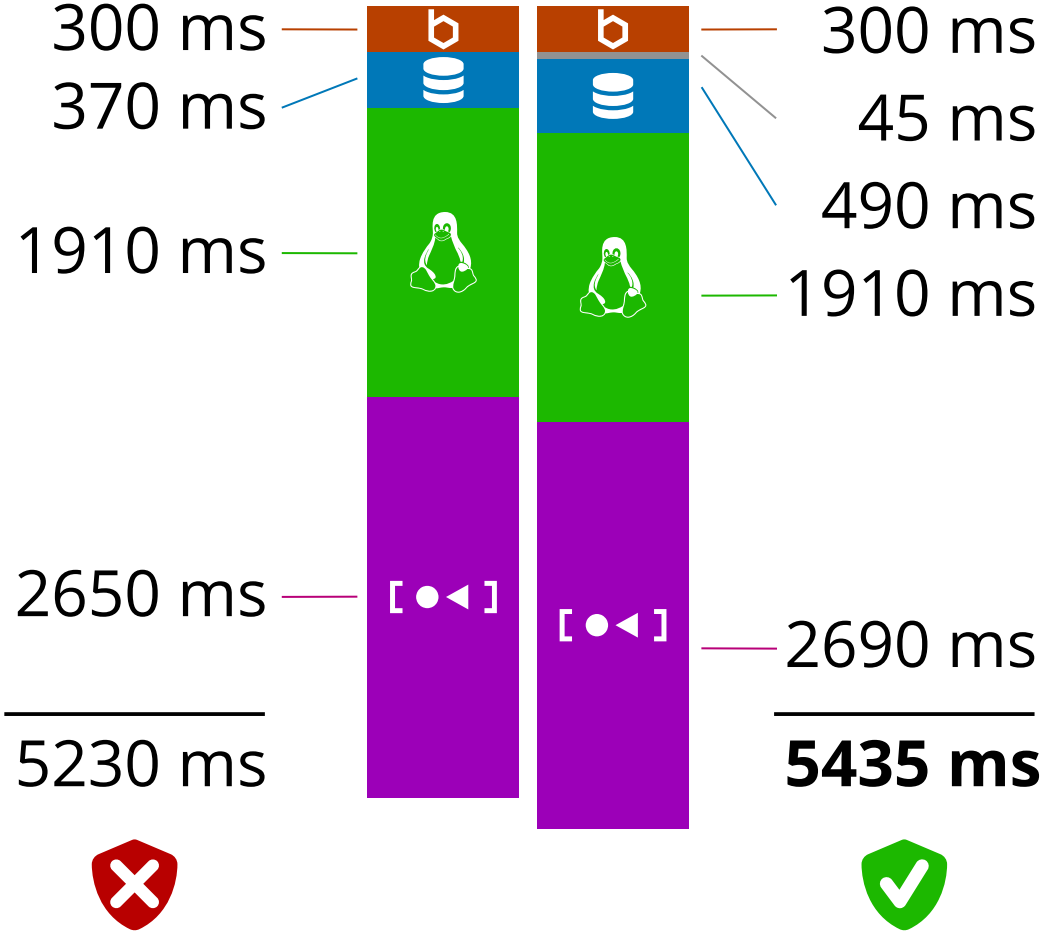
sha256 Implementations

- sha256-generic
  - Generic C implementation
- sha256-caam
  - CAAM – Cryptographic Acceleration and Assurance Module
- sha256-ce
  - ARMv8 Cryptographic Extensions

dm-verity Overhead – Userspace Boottime

- Without dm-verity: 2650 ms
- sha256-caam: 3160 ms (+510 ms)
- sha256-generic: 2920 ms (+270 ms)
- sha256-ce: 2690 ms (+40 ms)

Secure Boot Measurement



Conclusions

- Some extra overhead, but not too bad
- A small initramfs is important
- The correct hash implementation makes a big difference

What's next?

- Measure the impact of the bootloader verification
- Optimize OP-TEE setup?
- Optimize FIT image loading?
- Proper Yocto integration for rsinit

Thanks!

Questions?