

There's a new update! But there are a few unsaved projects and you are gone for, like, five minutes now. May I update and restart anyway?



I'll take that as a yes.



LOLNEIN.com

This may also take a couple hours.



LOLNEIN.com

IoT Geräte updaten – jetzt aber mal richtig

Konzepte, Möglichkeiten und Praxiserfahrungen zum Thema Software Updates am Beispiel eines embedded Linux Systems



IoT Geräte updaten – jetzt aber mal richtig

Konzepte, Möglichkeiten und Praxiserfahrungen zum Thema Software Updates am Beispiel eines embedded Linux Systems



Florian Fischer

- Master in Space Science and Technology
- 1,5 Jahre Applikationsingenieur bei National Instruments
- Seit 2 Jahren Entwicklungsingenieur bei Helbling
- Meine Passion gilt SmartHome und IoT



Agenda

1. Firmware Update
2. Update Konzepte
3. Open Source Lösungen
4. Einsatz von RAUC in der Praxis
5. Fazit
6. Fragen

1. FIRMWARE UPDATE

1.1 Gründe für Updates

- Neue Funktionen
- Bugfixes
- Beheben von Sicherheitslücken



Linux : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : [CVE Number Descending](#) [CVE Number Ascending](#) [CVSS Score Descending](#) [Number Of Exploits Descending](#)

[Copy Results](#) [Download Results](#)

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2019-17351	400		DoS	2019-10-07	2019-10-11	4.9	None	Local	Low	Not required	None	None	Complete
An issue was discovered in drivers/xen/balloon.c in the Linux kernel before 5.2.3, as used in Xen through 4.12.x, allowing guest OS users to cause a denial of service because of unrestricted resource consumption during the mapping of guest memory, aka CID-6ef36ab967c7.														
2	CVE-2019-17133	120		Overflow	2019-10-04	2019-10-10	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
In the Linux kernel through 5.3.2, cfg80211_mgd_wext_giwessid in net/wireless/wext-sme.c does not reject a long SSID IE, leading to a Buffer Overflow.														
3	CVE-2019-17075	119		DoS Overflow	2019-10-01	2019-10-08	7.1	None	Remote	Medium	Not required	None	None	Complete
An issue was discovered in write_tpt_entry in drivers/infiniband/hw/cxgb4/mem.c in the Linux kernel through 5.3.2. The cxgb4 driver is directly calling dma_map_single (a DMA function) from a stack variable. This could allow an attacker to trigger a Denial of Service, exploitable if this driver is used on an architecture for which this stack/DMA interaction has security relevance.														

1.2 Updates sind doch ein alter Hut.....



...Uconnect systems may experience a reboot every 45-60 seconds. Our Engineering teams are investigating the cause and working towards a resolution....

Dear Lockstate Customer,
.... Your lock is among a small subset of locks that had a fatal error rendering it inoperable. After a software update was sent to your lock, it failed to reconnect to our web service making a remote fix impossible.



oder vielleicht doch nicht?

1.3 Anforderungen an Updates

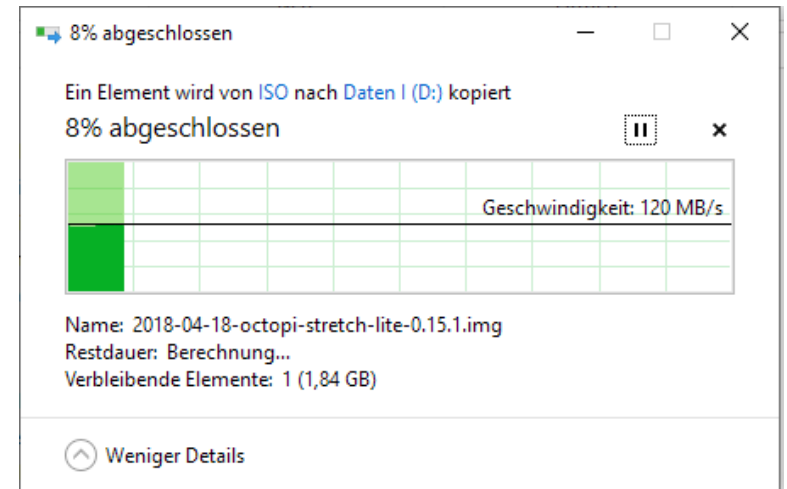
- Robust
- Zuverlässig
- Ressourcensparend
- Schnell
- Sicher

2. UPDATE KONZEPTE

2.1 Datei-basiert

Dateien werden direkt auf das Gerät kopiert

- Einfacher Ansatz
- Nicht atomar
- Behandelt keine Abhängigkeiten
- Meist Eigenlösung



2.2 Paket-basiert

Updates werden durch Paketmanager in Form von Paketen installiert

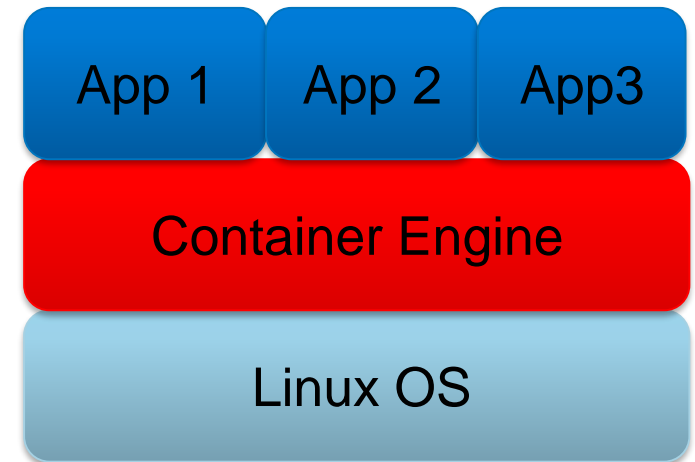
- Im Desktopbereich Standard
- Nicht atomar
- Abhängigkeiten können nicht immer aufgelöst werden
- Schwer zu testen



2.3 Container-basiert

Anwendungen laufen in Containern, welche komplett ausgetauscht werden können

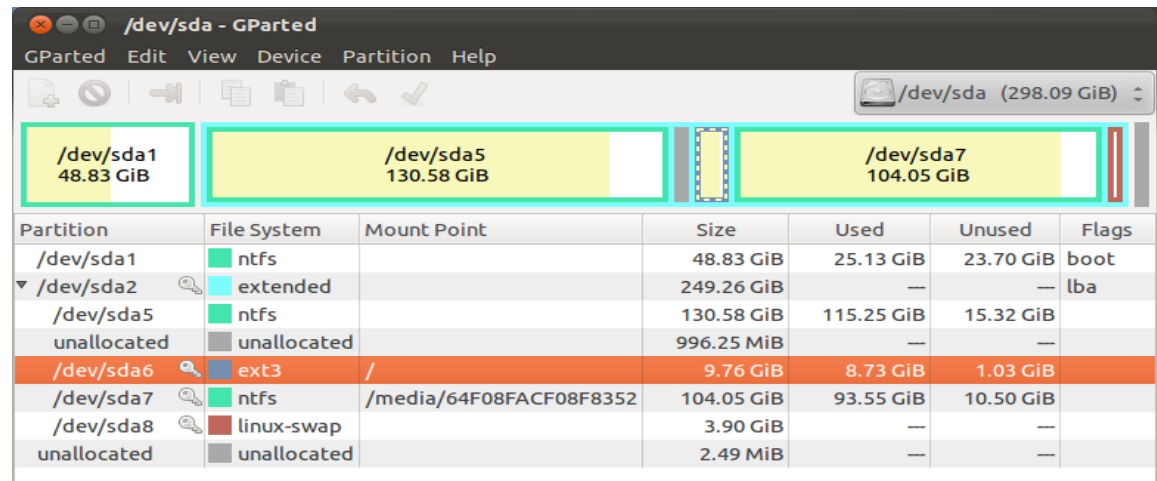
- Benötigt spezielle Container Engine
- Vereinfacht Updates
- Container enthält alle Abhängigkeiten
- Einfaches Testen
- OS Update auf anderem Weg



2.4 Image-basiert

Auf Partitionsebene wird das Image ausgetauscht

- Benötigt mindestens 2 Partitionen
- Garantiert gleichen Softwarestand
- Reboot nötig
- Atomar
- Rollback möglich



Partition	File System	Mount Point	Size	Used	Unused	Flags
/dev/sda1	ntfs		48.83 GiB	25.13 GiB	23.70 GiB	boot
▼ /dev/sda2	extended		249.26 GiB	—	—	lba
/dev/sda5	ntfs		130.58 GiB	115.25 GiB	15.32 GiB	
unallocated	unallocated		996.25 MiB	—	—	
/dev/sda6	ext3	/	9.76 GiB	8.73 GiB	1.03 GiB	
/dev/sda7	ntfs	/media/64F08FACF08F8352	104.05 GiB	93.55 GiB	10.50 GiB	
/dev/sda8	linux-swap		3.90 GiB	—	—	
unallocated	unallocated		2.49 MiB	—	—	

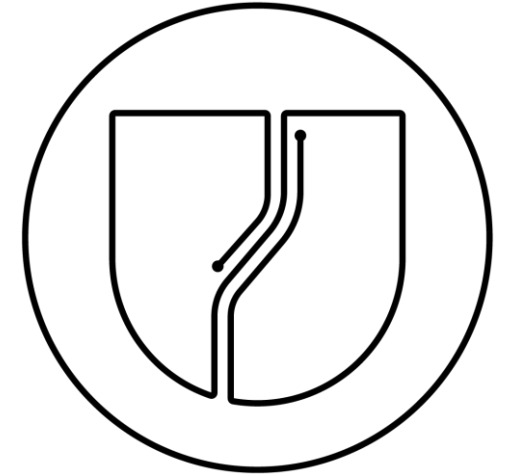
2.5 Vergleich von Update Konzepten

	Update Größe	Speicher- bedarf	Atomar	Abhängig- keiten	Tests	Simpel	Power fail save
Datei	klein	klein	nein	nein	schlecht	ja	nein
Paket	klein	mittel	nein	ja	sehr aufwändig	nein	nein
Container	mittel	mittel	ja	ja	gut	ja	ja
Image	groß	groß	ja	ja	gut	nein	ja

3. OPEN SOURCE LÖSUNGEN

3.1 SWUpdate:

- Partition und Datei basierte Updates
- Flexibles Partitionsschema
- Lokale und Remote Updates
- Yocto Integration
- Keine integrierter Rollback Mechanismus
- Signierte und verschlüsselte Updates



<https://github.com/sbabic/swupdate>

3.2 Mender:

- Partition-basierte Updates
- Benötigt mindestens 4 Partitionen
- Read-only Rootfs
- Lokale und Remote Updates
- Eigener Backend Server verfügbar
- Yocto Integration
- Signierte Updates
- Integrierter Rollback Mechanismus



<https://mender.io/>

3.3 RAUC:

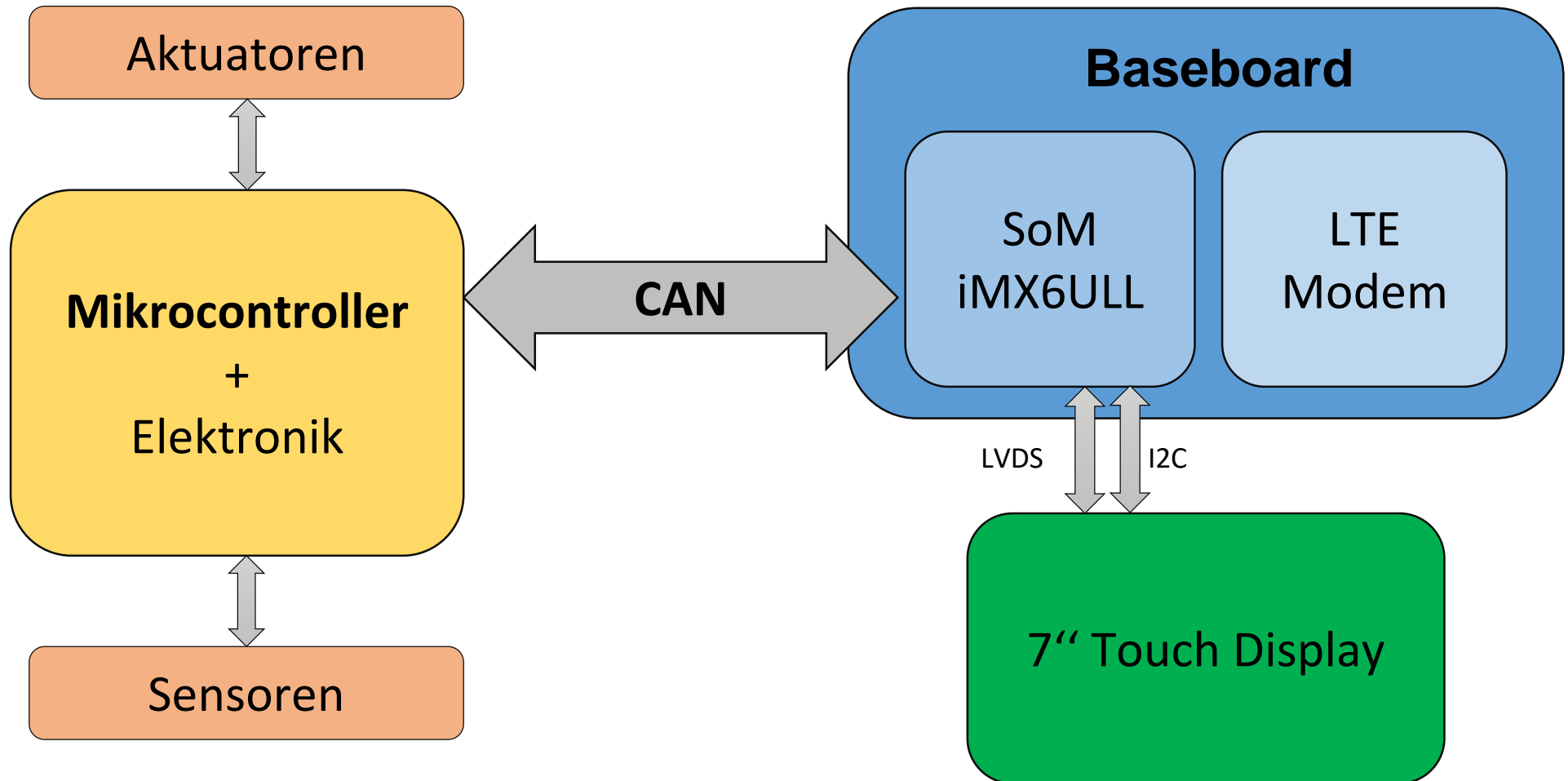
- Partition und Datei basierte Updates
- Flexibles Partitionsschema
- Read-only Rootfs
- Lokale und Remote Updates
- D-Bus Interface
- Yocto Integration
- Integrierter Rollback Mechanismus
- Signierte Updates



<https://www.pengutronix.de/de/software/rauc.html>

4. EINSATZ VON RAUC IN DER PRAXIS

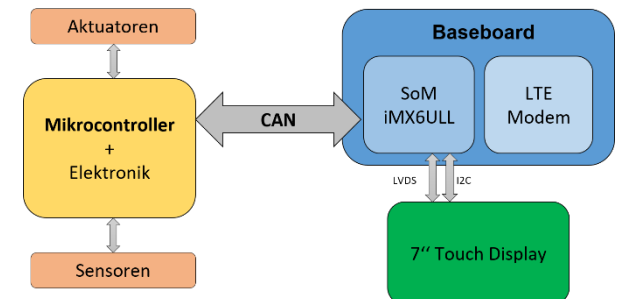
4.1 Aufbau des Systems



4.2 Anforderungen an das System

- Yocto als Buildsystem
- Update des Betriebssystems und der Anwendungen
- Bereitstellen der Update Pakete über AWS
- Update Frequenz halbjährlich bis jährlich
- Update der Mikrocontroller Firmware
- D-Bus Anbindung

yocto
PROJECT



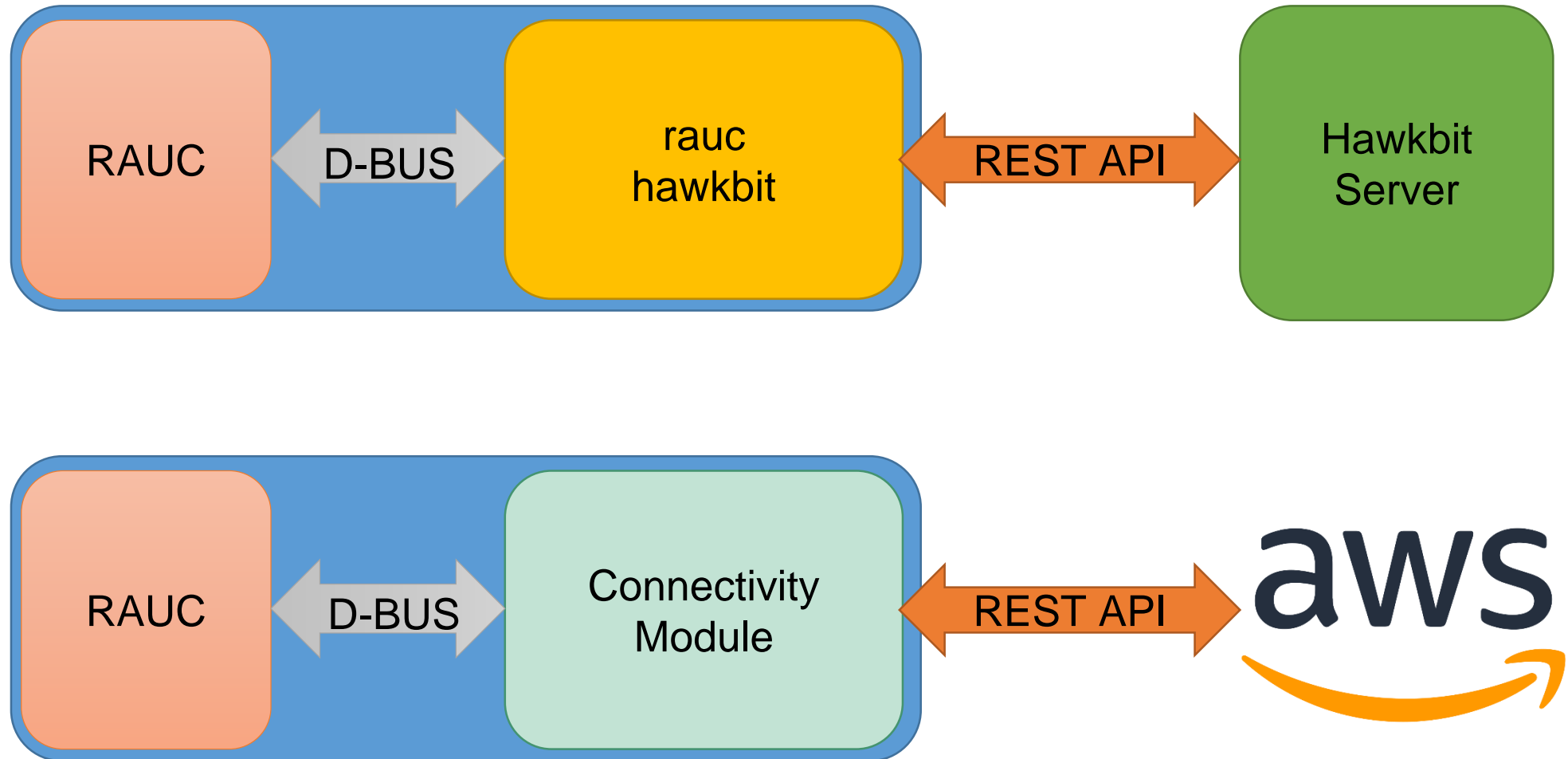
4.3 Integration des Updateprozesses

- Bereitstellung des Update Paketes
- Einspielen des Update Paketes
- Betrieb
- Update des Mikrocontrollers

4.4 Bereitstellung des Update Paketes

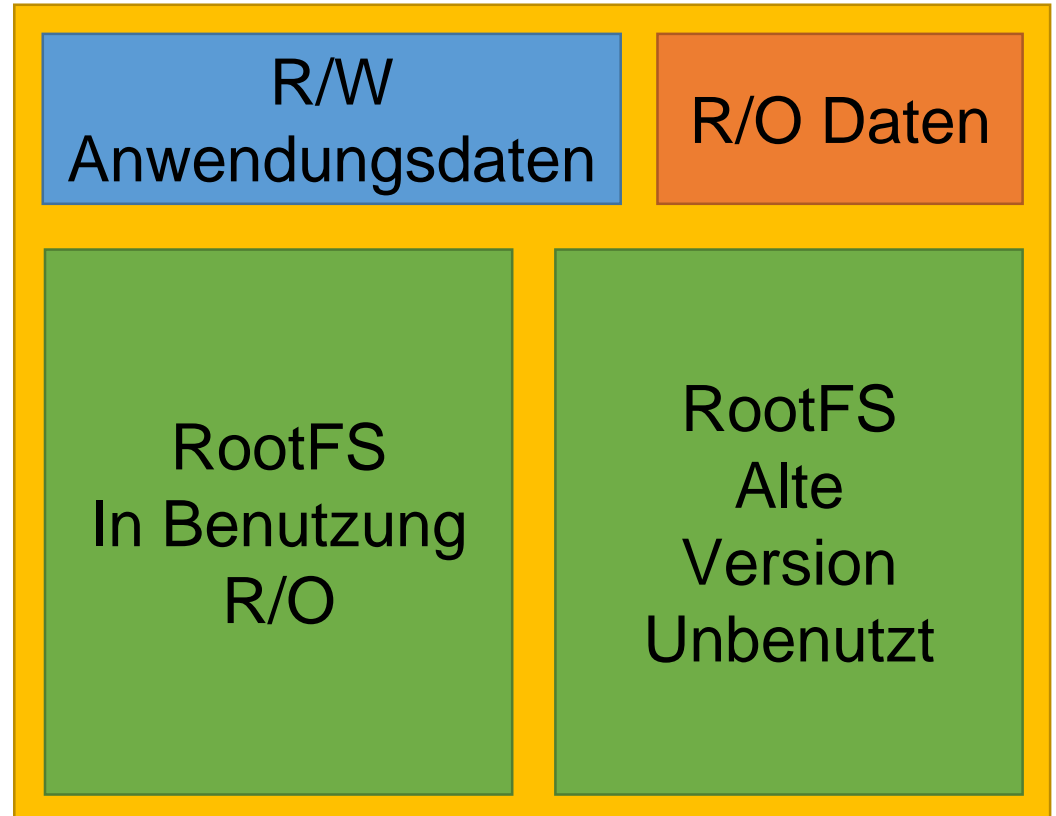
- Buildprozess mit Yocto
- *meta-rauc* Layer
 - Erstellen des Update Paketes
 - Signieren des Paketes
 - Installiert RAUC Client im Zielsystem

4.6 Einspielen des Update Paketes



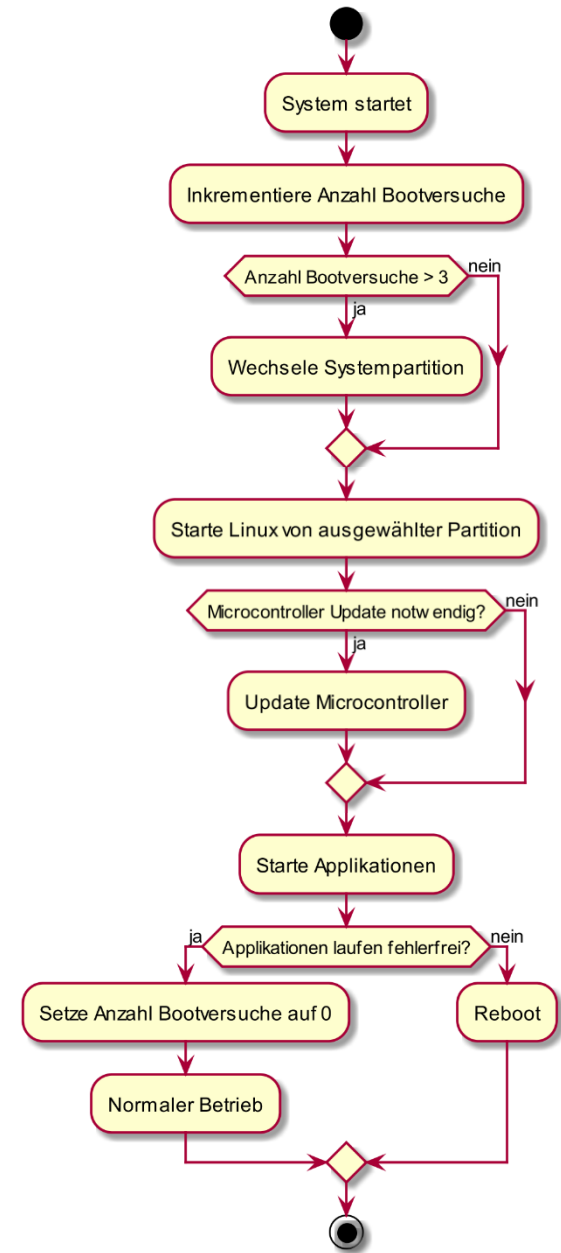
4.5 Einspielen des Update Paketes

- Zwei Systempartitionen
 - Aktuelles System
 - Vorheriges System
- Beschreibbare Partition für Anwendungsdaten
- Schreibgeschützte Partition für systemspezifische Daten



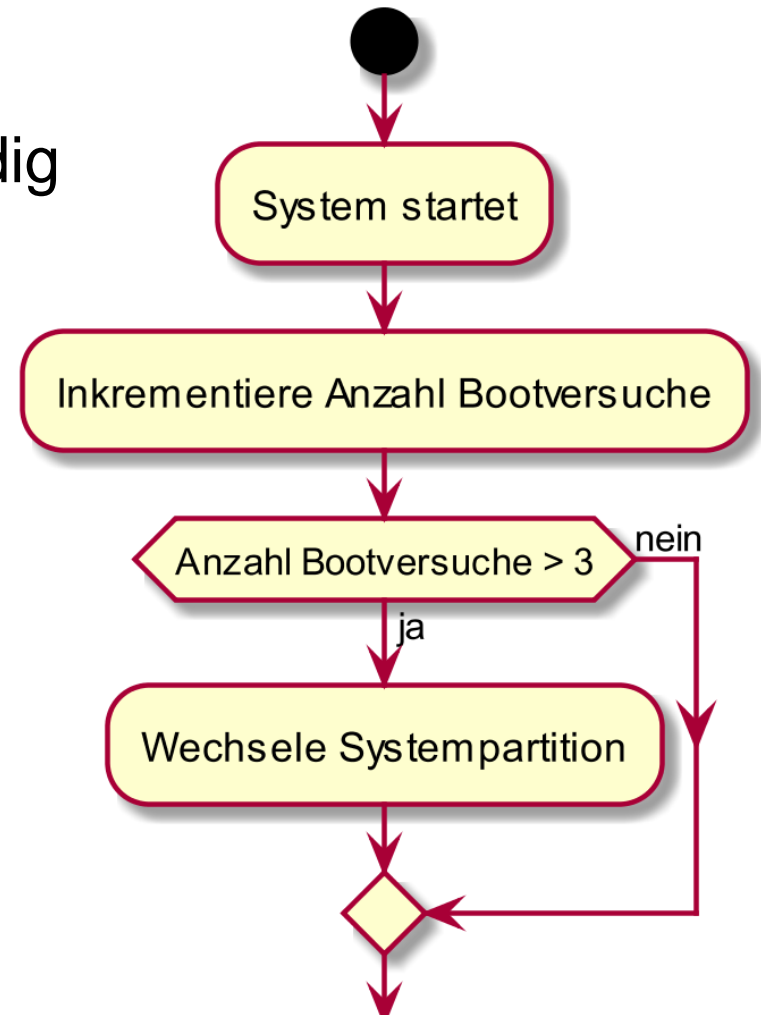
4.7 Ablauf

- Bootloader / Laden des Betriebssystems
- Mikrocontroller Update
- Starten der Anwendungen



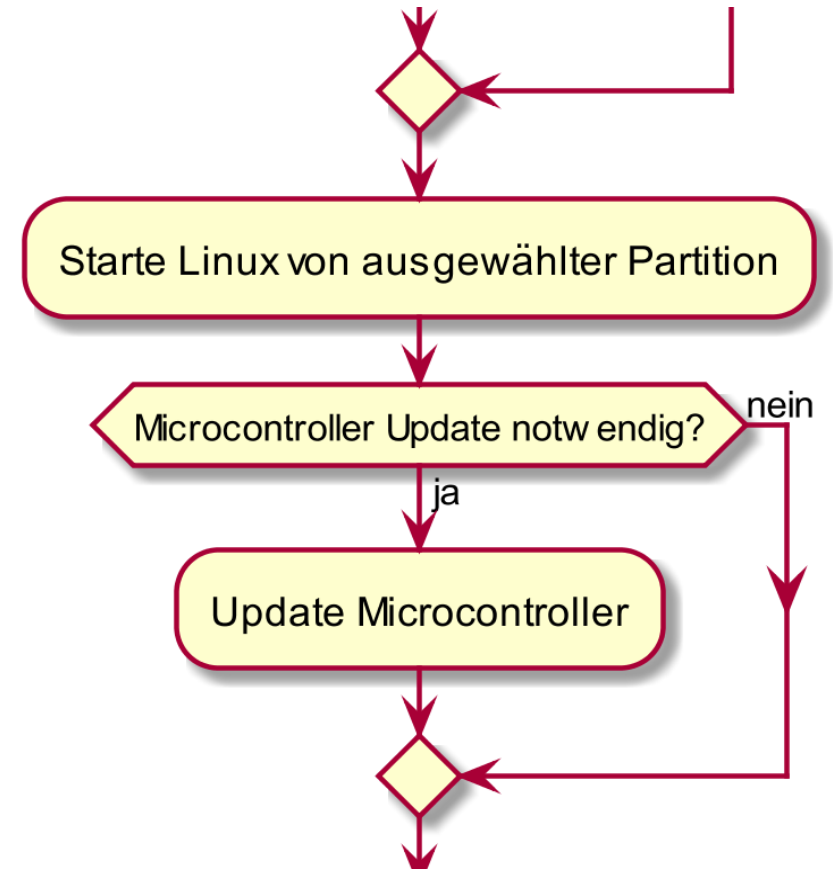
4.7.1 Starten des Systems

- Modifikationen am Bootloader notwendig
- Zusätzliche Bootvariable
- Logik zum Wechsel der Partition
- Skript in Uboot



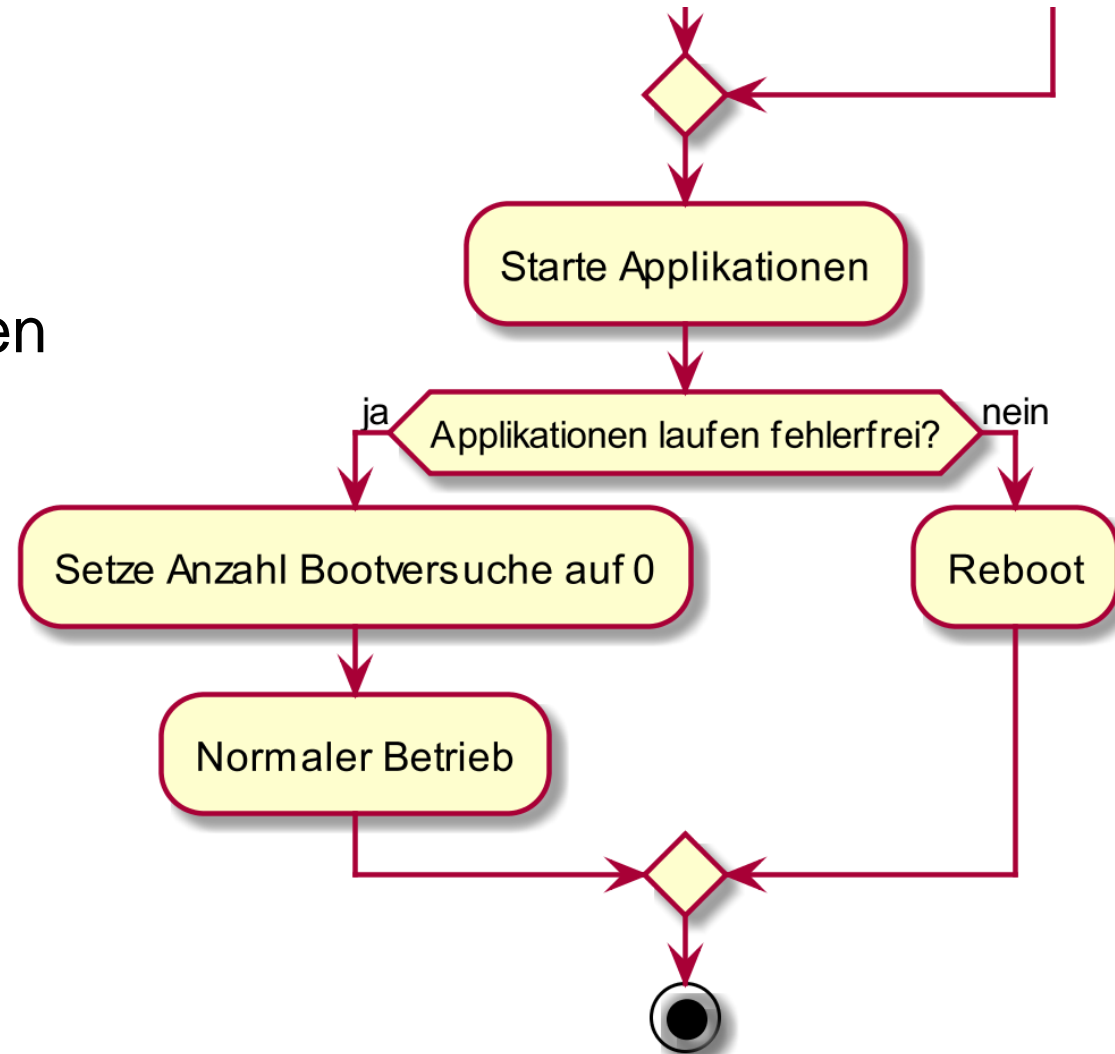
4.7.2 Mikrocontroller Check

- Abfrage Version FW Mikrocontroller
- Mikrocontroller Firmware ist Teil des Images
- Keine Inkompatibilität möglich



4.7.3 Applikationsstart

- Software Watchdog
- Systemd Startabhängigkeiten
- Setzen der Bootvariable



5. FAZIT

Fazit

- Updates sind absolut notwendig
- Updates sind nicht trivial
- Open Source Lösungen beachten
- RAUC ist eine flexible und alltagstaugliche Lösung

6. FRAGEN?

Vielen Dank für Ihre Aufmerksamkeit!



Ihr Ansprechpartner

Fischer Florian

M.Sc.

Entwicklungsingenieur

Helbling Technik GmbH

Leonrodstraße 52

D-80636 München

Telefon +49 (0) 89 45 929 250

florian.fischer@helbling.de

www.helbling.de

Helbling Technik

Aarau • Bern • Wil SG • Zürich • München • Boston • Shanghai