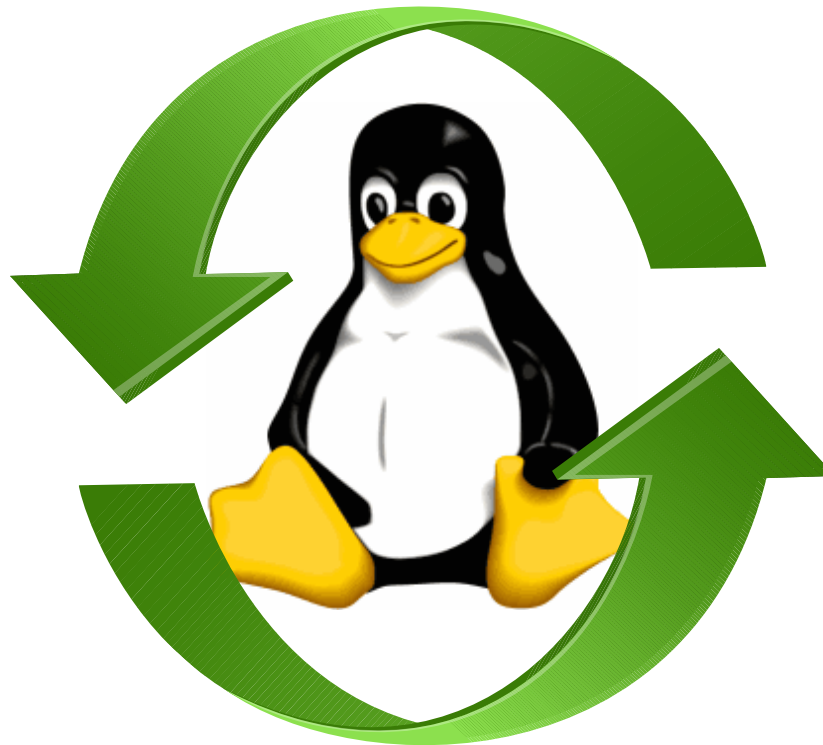


Secure and Safe Updates for Your Embedded Device

(And how to solve this with the RAUC update tool)



Enrico Jörns

FOSDEM 2017





Embedded, IoT, ... Updating?



Security fixes!!!



General bug fixes!!



New Features!



Structure

- Updating Embedded Systems
- Introducing RAUC
- DEMO
- Conclusion

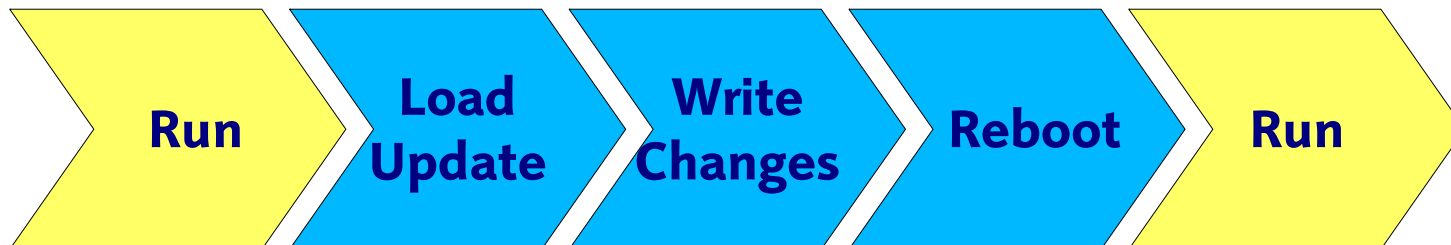
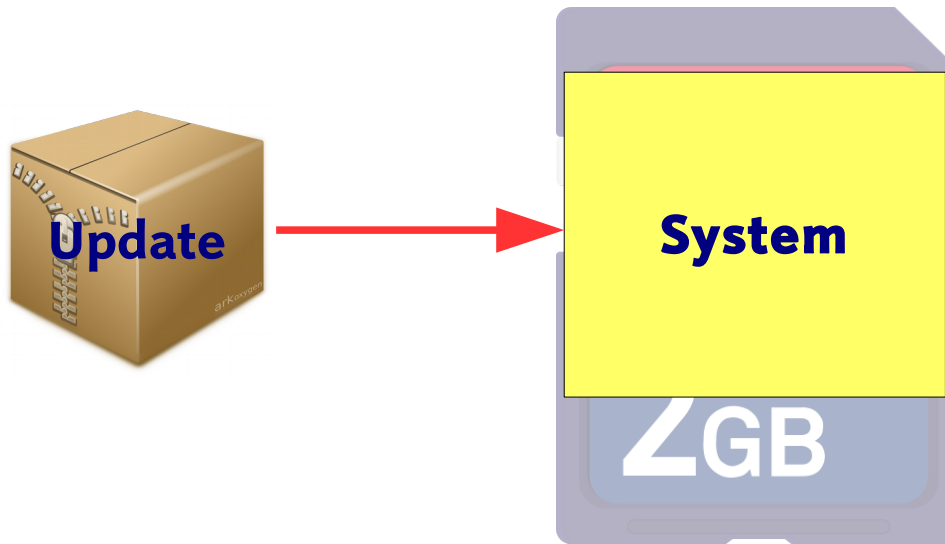


Updating – The simple View

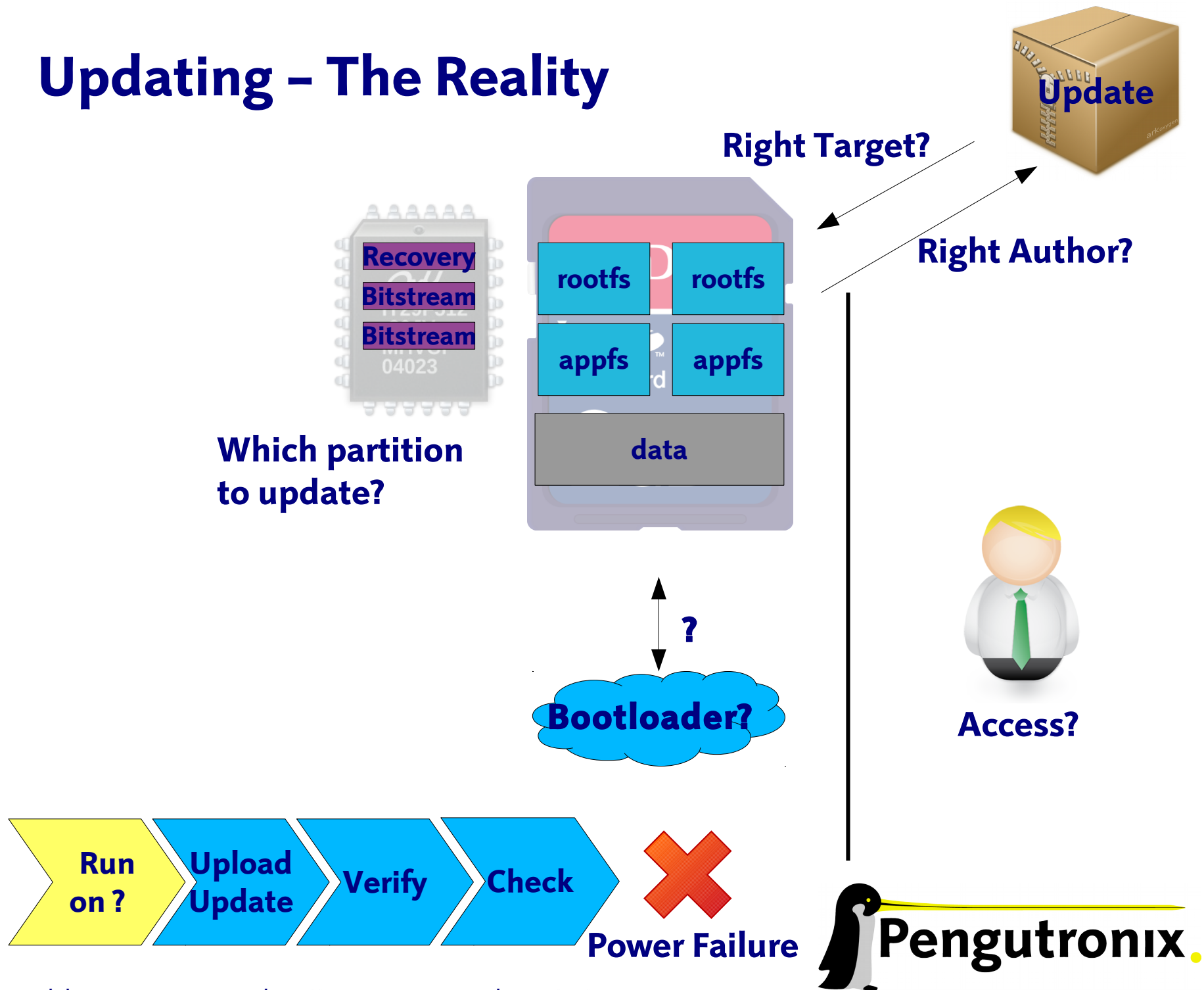


Updating...

Do not turn
off!

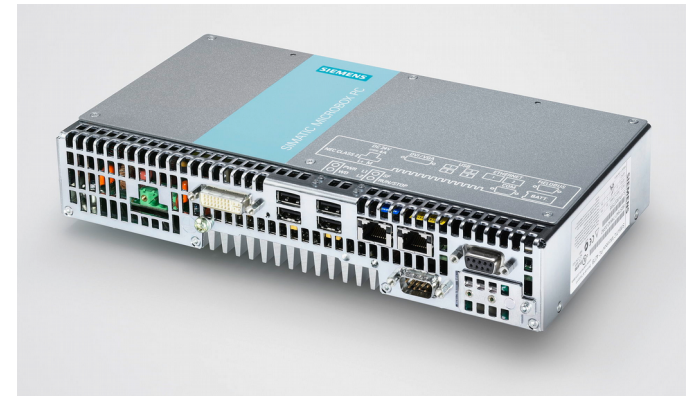
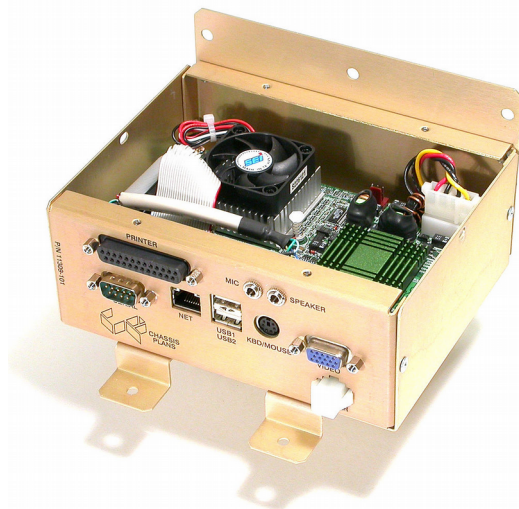


Updating – The Reality



What's so special on Embedded Updates?

And why not simply
write a bash script?..



Supervised Updates



Update?

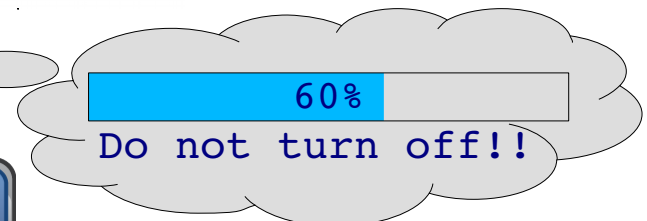
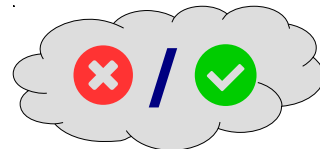
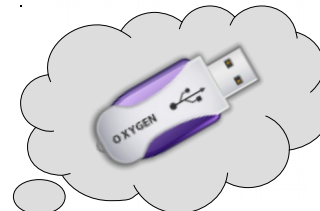
```
$ sudo apt-get upgrade
```



Unattended Updates



Update?



Updating Embedded Devices

Key Requirements:

- **Safety**
 - Update process should not fail!
 - Still updatable after failed update
 - Recover from runtime issues
- **Security**
 - Prevent unauthorized access

**! Install only
well tested
software state**

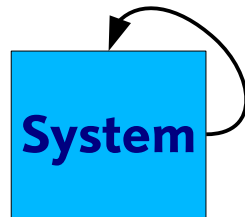
**! Do NOT brick
your device!**

Further Restrictions:

- Storage handling (NAND, NOR, eMMC, ...)
- Limited resources (storage, data rate, ...)

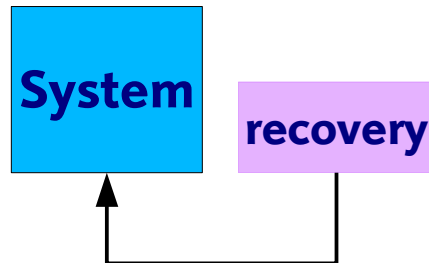


System Redundancy



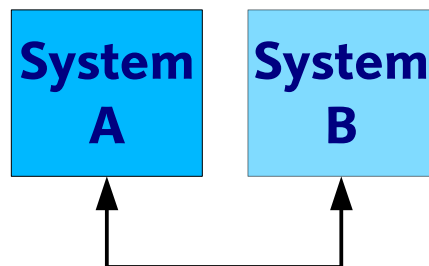
No redundancy

⚡ → Service technician



Asymmetric

⚡ → Updatable, Non-operating



Symmetric

⚡ → Updatable, Operating (old version)



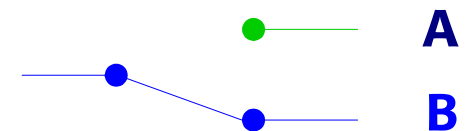
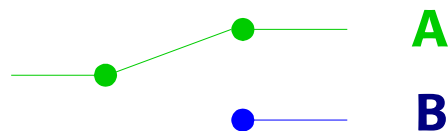
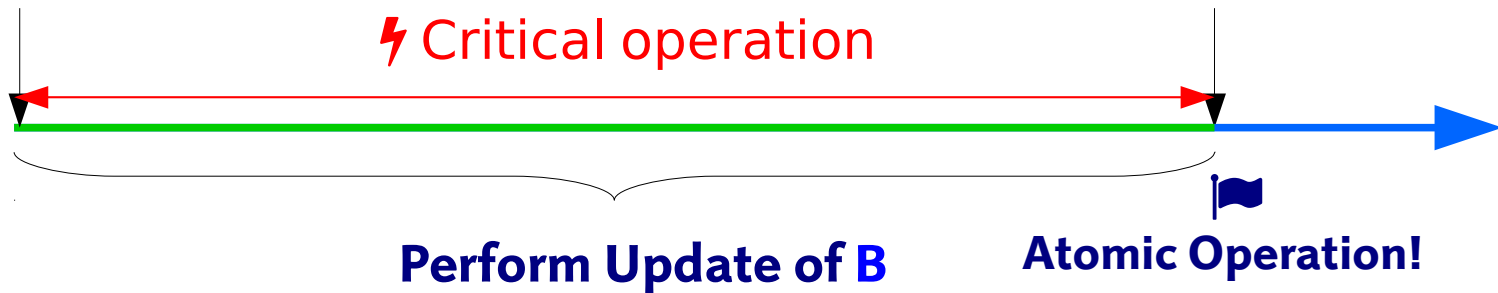
Update Atomicity



Start
Update
from **A**

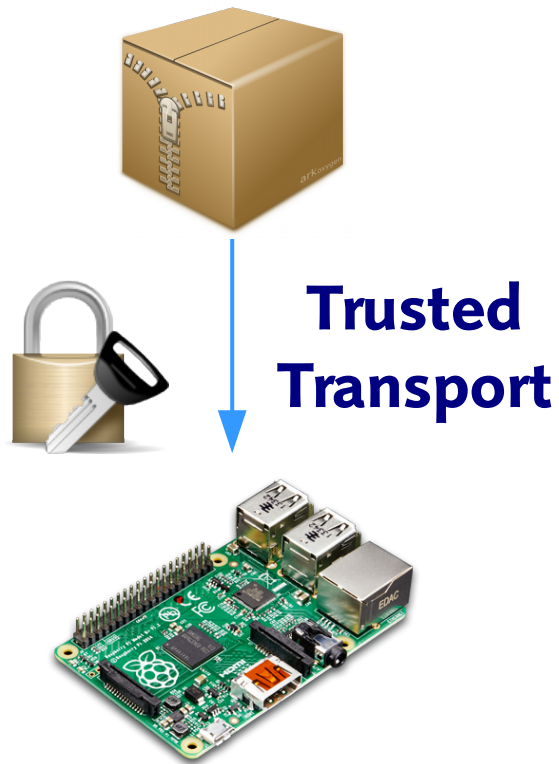
Change
Bootsource
to **B**

⚡ Critical operation

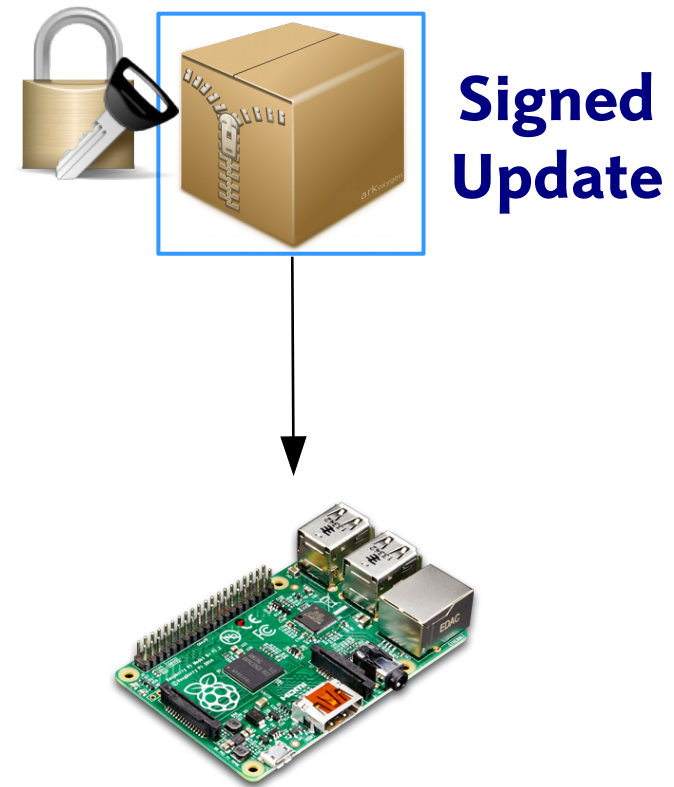


Secure Updates

→ Prevent unauthorized installation



✓ Download
✗ USB stick



✓ Download
✓ USB stick

How to not turn a Toaster into a Printer

Or: How to make sure you don't install the wrong image

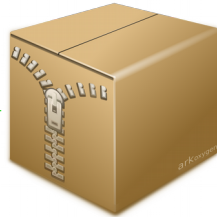
Check Hardware:



Same hardware might operate in different setups / devices

Explicit identification:

Update for
„PiToast 2000“



✓ accept

„PiToast 2000“

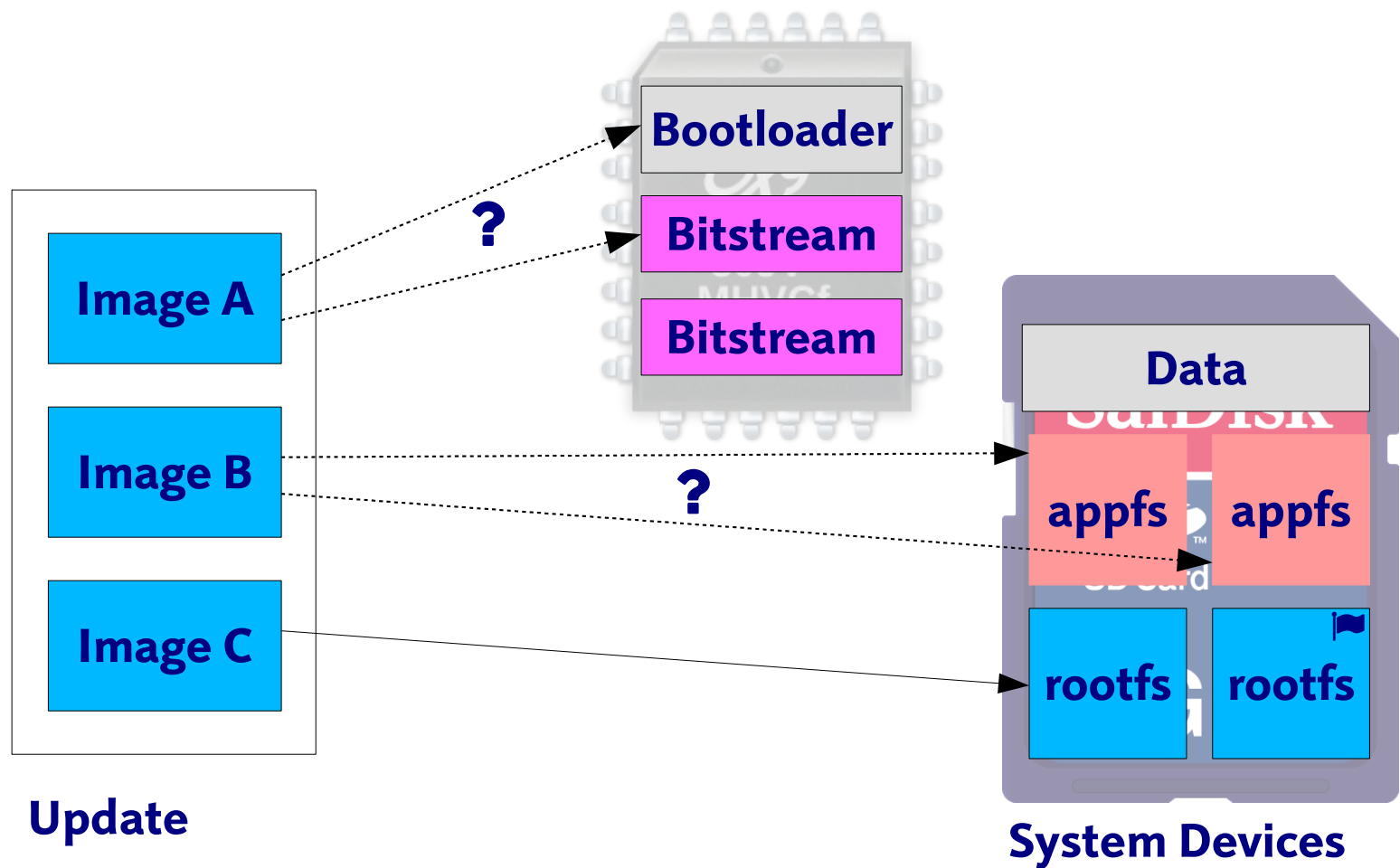


✗ reject

„PiPrint Plus“



Determine and handle Update Targets



→ Requires configuration, meta-information, custom handler



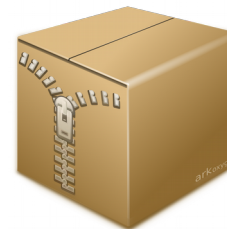
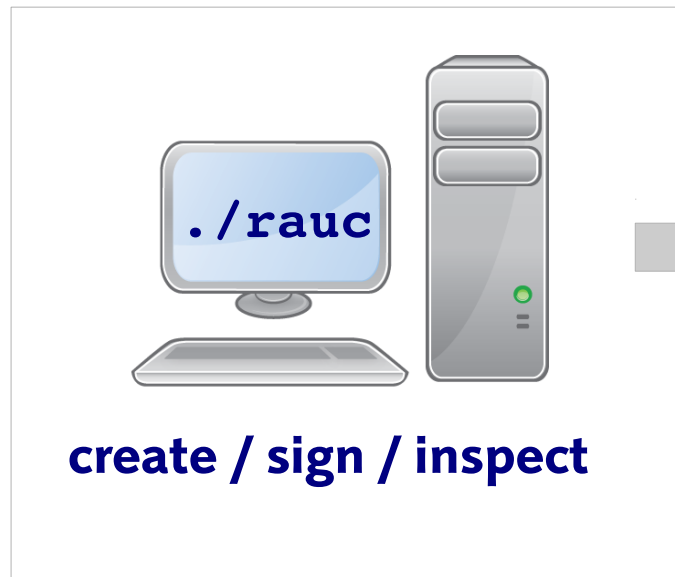
Other Things to take care of

- Error Handling
 - Check all function/subprocess return values!
 - Abort correctly
- Data Handling
 - Single storage location?
 - Redundant storage?
- Watchdog handling
- Update Format
 - Version handling?
- Extensibility
- ...

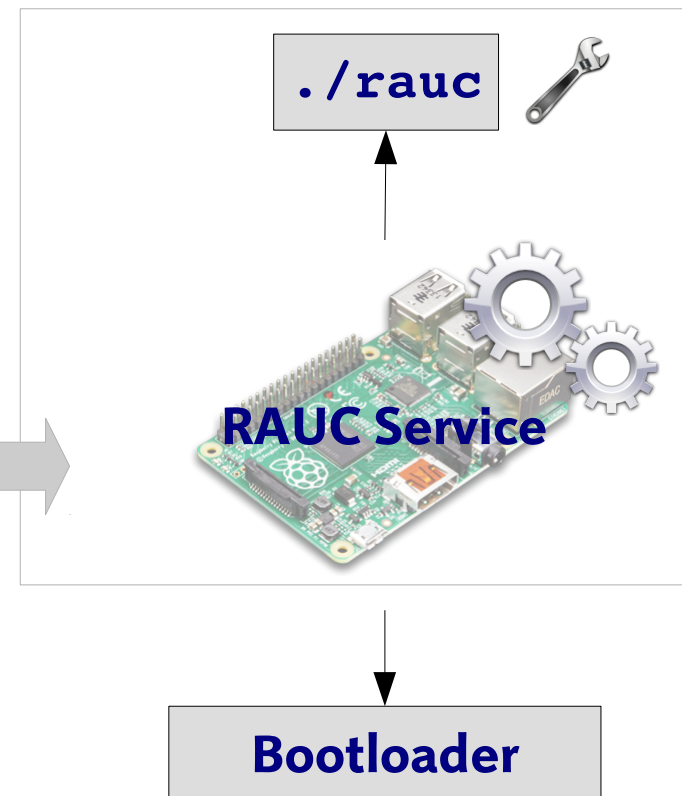


RAUC – Robust Auto-Update Controller

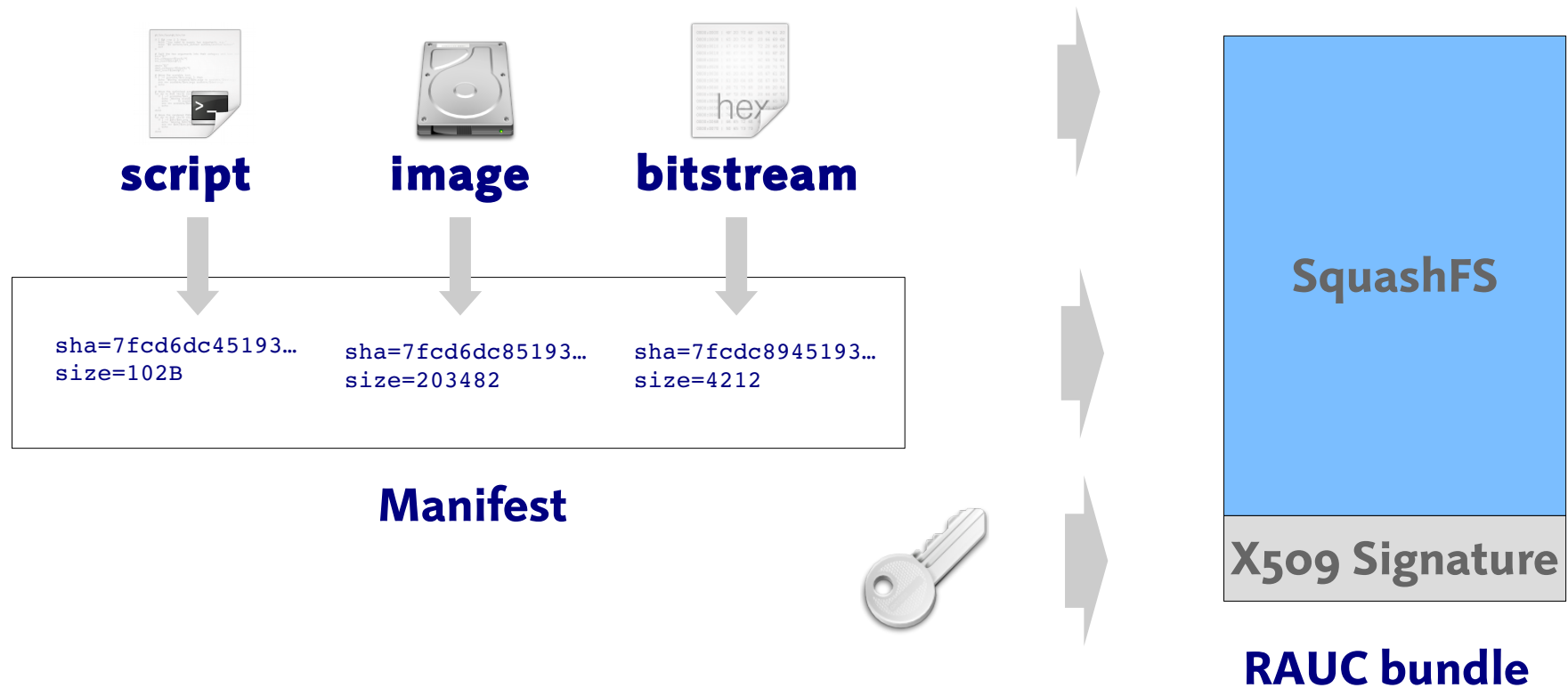
- LGPL-2.1
- C with GLib
- Autotools



Bundle

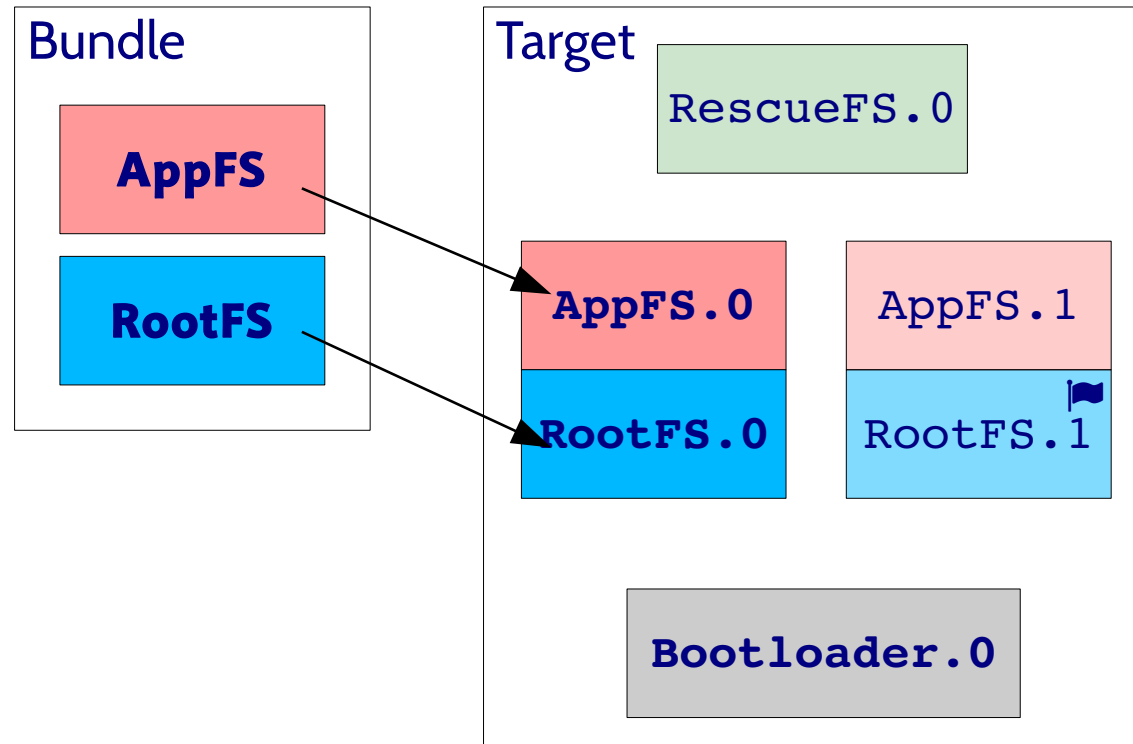


RAUC – Bundles



A Bundle describes the intended target state of the device

RAUC – Slot Handling



Slot: Anything we might want to update

RAUC – Configuration

```
[system]
compatible=MyProduct2000
bootloader=barebox

[keyring]
path=/etc/rauc/keyring.pem

[handlers]
post-install=/sbin/postinst.sh

[slot.rootfs.0]
device=/dev/sda0
type=ext4
bootname=system0

[slot.rootfs.1]
device=/dev/sda1
type=ext4
bootname=system1

[...]
```

System Configuration on Target

```
[update]
compatible=MyProduct2000
version=2017.02-4

[image.rootfs]
filename=rootfs.ext4
size=419430400
sha256=b14c1457dc1046...

[image.appfs]
filename=appfs.ext4
size=219430400
sha256=ecf4c031d01cb9...
hooks=post-install
```

Update Manifest



Using RAUC on your next Device

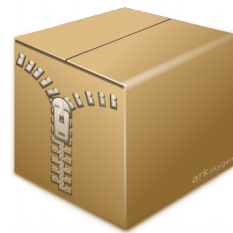
- **System Integration**
 - Yocto integration: meta-ptx layer (→ meta-rauc)
 - PTXdist integration: coming soon
 - Manual Integration (system.conf, keys, manifest)
- **Usage**
 - Create Bundle (Host)

```
rauc bundle --key=... content/ my-bundle.raucb
```
 - Install Bundle (Target)

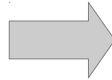
```
rauc install my-bundle.raucb
```
- Integrate into Application (using D-Bus)



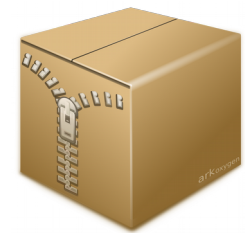
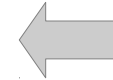
DEMO: Update Rollout with HawkBit



Yocto



hawkBit
Deployment Server



PTXdist

DDI API

Python Script

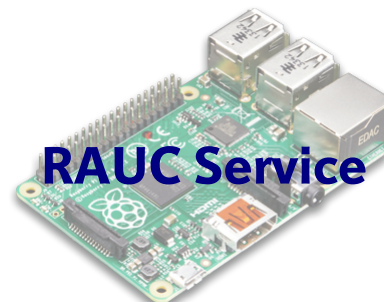
D-Bus

RAUC Service

Rootfs
A

Rootfs
B

Barebox
(Bootstate)



What we've learned...

- 1) Updates are mandatory
- 2) Updates are non-trivial
- 3) Only install well tested software states
- 4) Use (FOSS) update frameworks



Thanks!



RAUC on GitHub:

<https://github.com/jluebbe/rauc>



Documentation:

<http://rauc.readthedocs.io/>



meta-ptx Layer (will soon become meta-rauc):

<https://git.pengutronix.de/cgit/meta-ptx/>



Barebox Bootchooser:

<http://barebox.de/doc/latest/user/bootchooser.html>

