Memfault

Over-the-Air Updates for Embedded Linux Devices

Thomas Sarlandie

Thomas Sarlandie

Linux Tech Lead, Memfault

- Passion: building at the intersection of software and hardware
- Previously led software teams at Pebble and Fitbit
- 🦀🦀🦀 Rust-aficionado

pebble fitbit Memfault

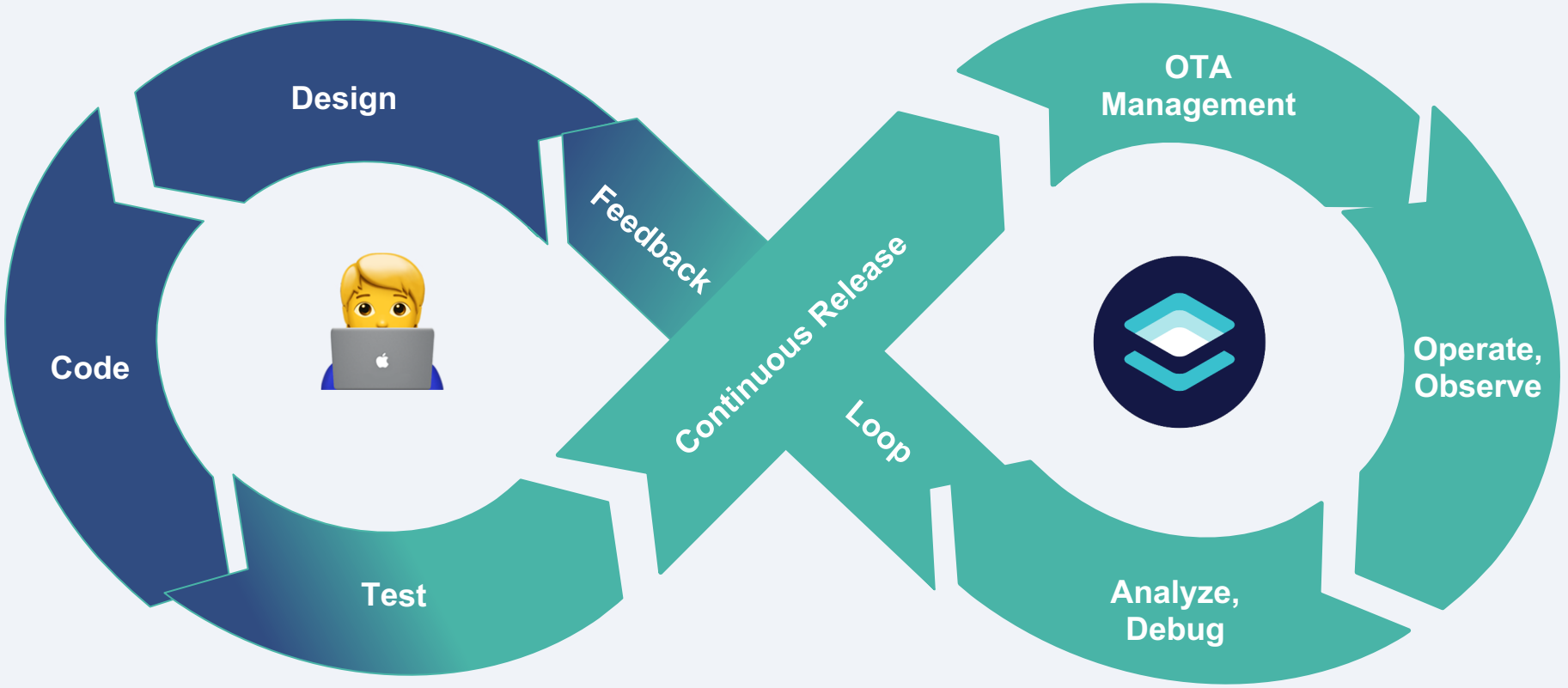
Agenda

- Requirements for a Robust Over-the-Air update system
- High-level Overview of the OTA Process
- Implementation with SWUpdate & U-Boot
- Q & A

Agenda

- **Requirements for a Robust Over-the-Air update system**
- High-level Overview of the OTA Process
- Implementation with SWUpdate & U-Boot
- Q & A

Over-the-Air updates



Requirements of your over-the-air update system

Predictable

Through new installs or multiple updates, devices always end-up in the exact same configuration.

Reliable

Will not leave the device in an “intermediate” state.

Will work even when other components are not working.

Secure

Will verify the origin of the update and maintain a secure-boot chain.

The case against package managers

Embedded Linux

Kernel

/etc
/usr
/bin

Full system updates

Desktop Linux

Kernel

Python

Browser

Package by package

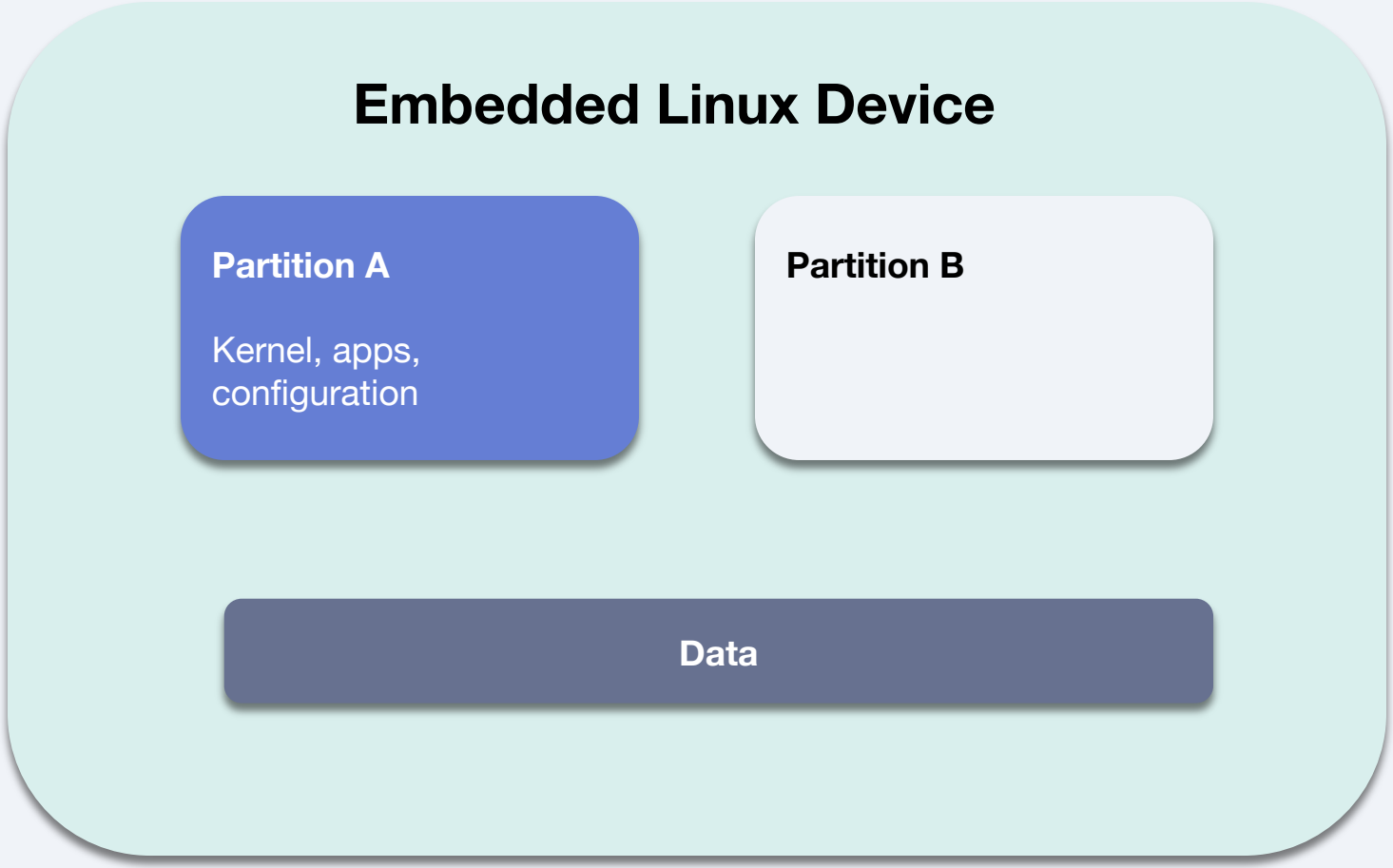
Reproducibility
Reliability

Flexibility

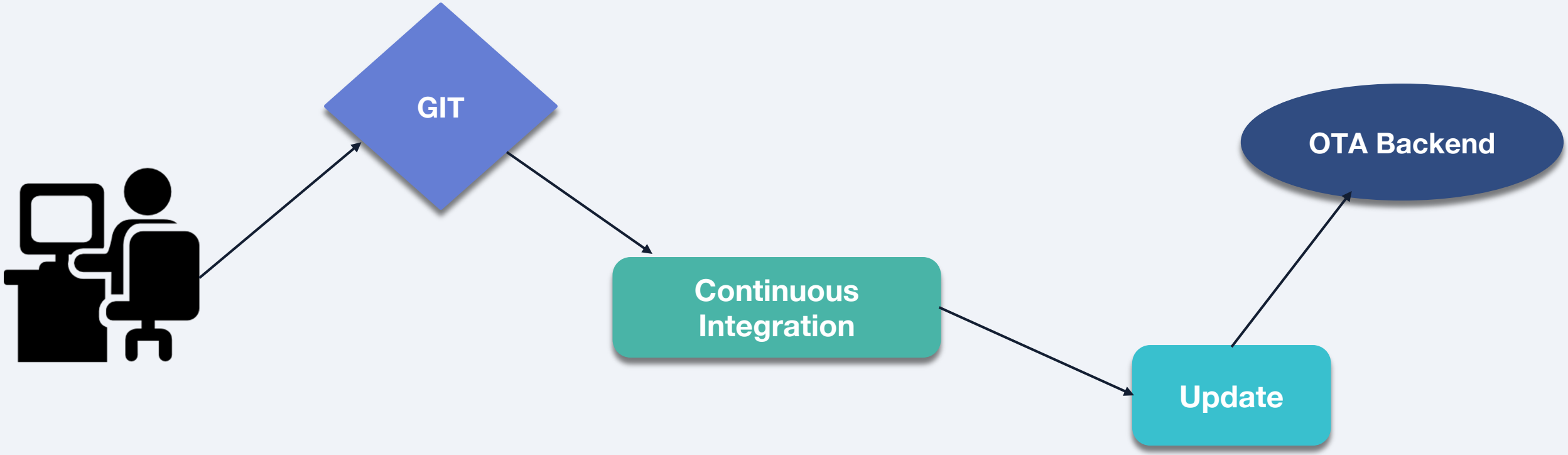
Agenda

- Requirements for a Robust Over-the-Air update system
- **High-level Overview of the OTA Process**
- Implementation with SWUpdate & U-Boot
- Q & A

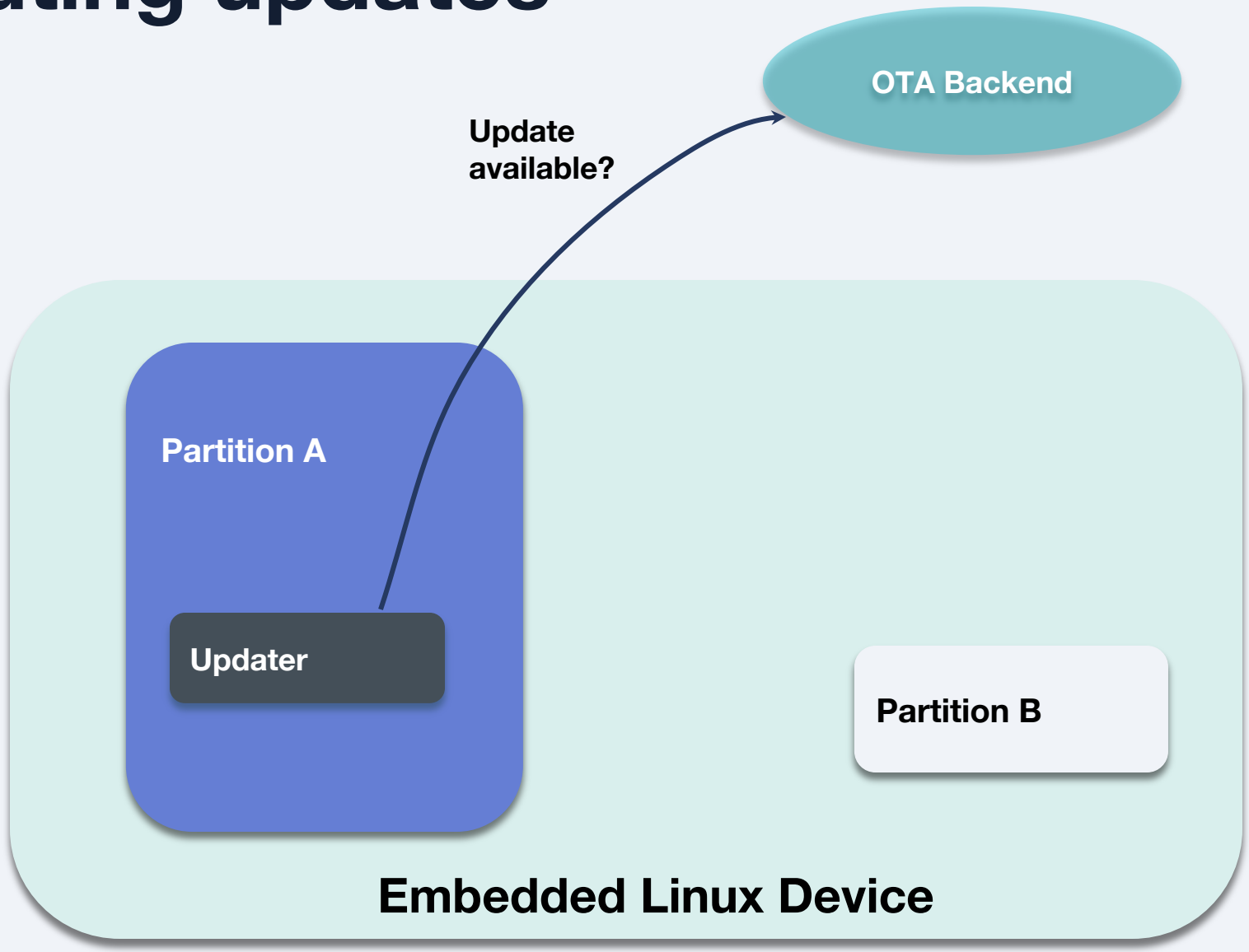
A/B Partition Scheme



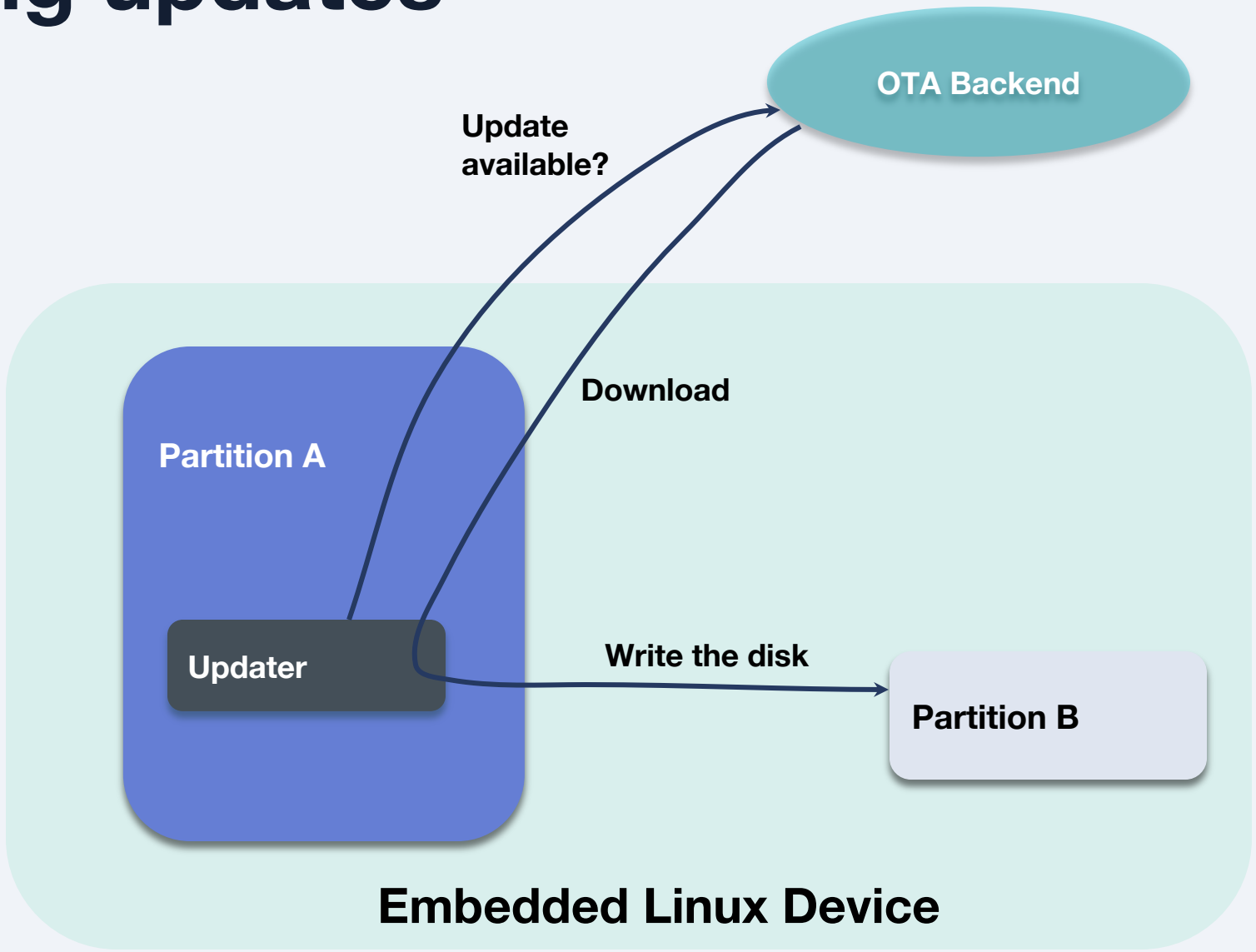
Preparing an update



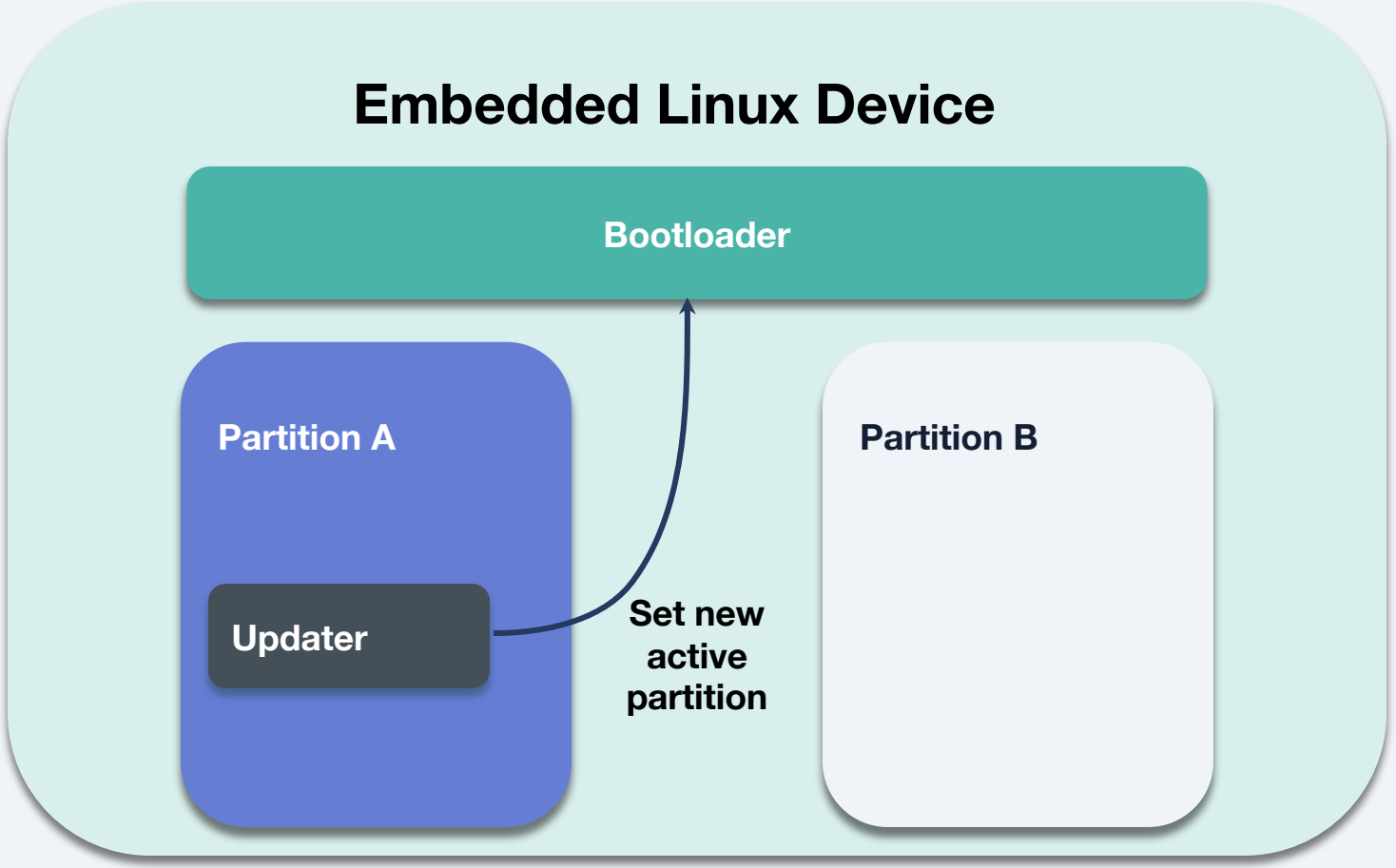
Distributing updates



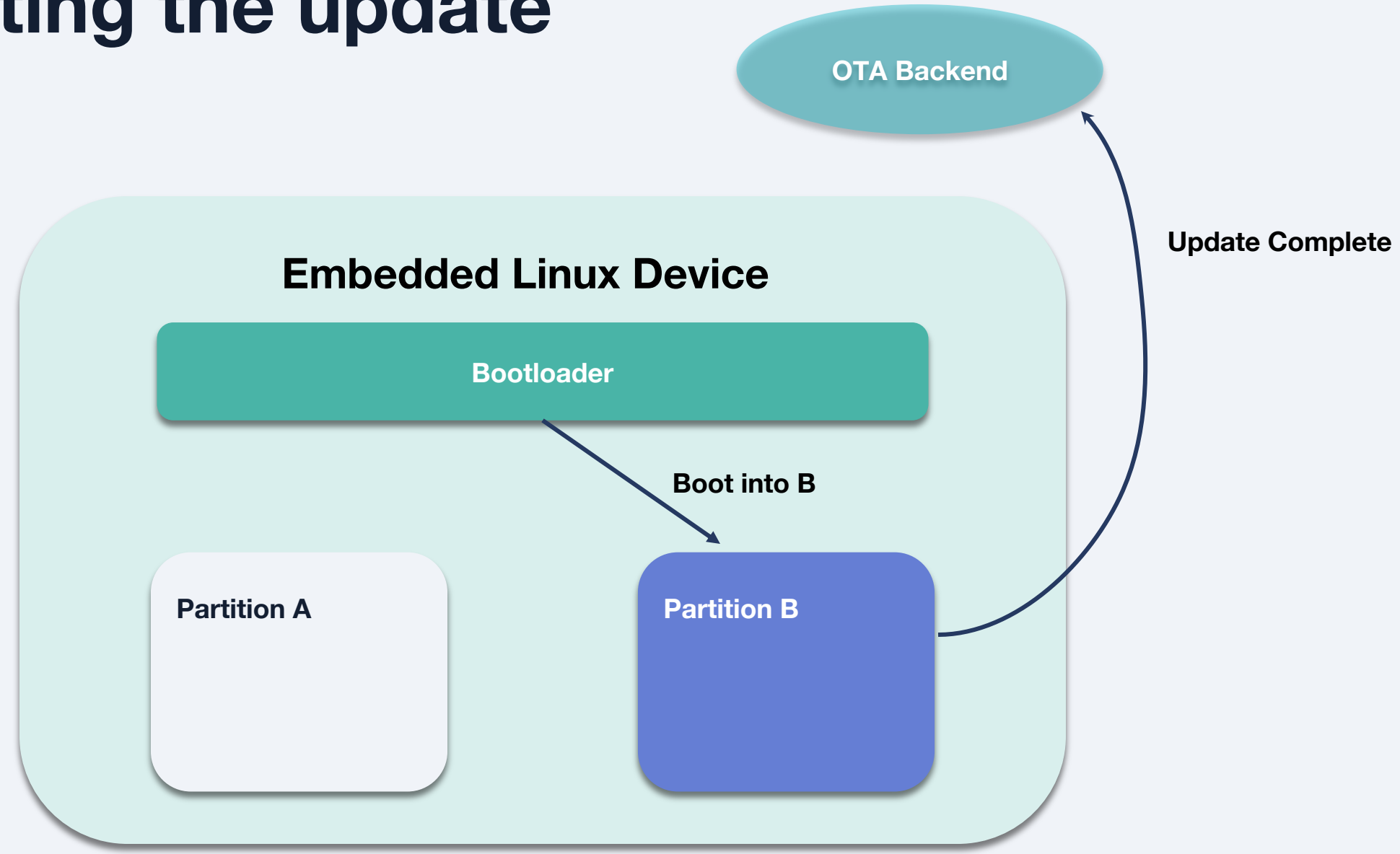
Installing updates



Rebooting into an update



Completing the update



Agenda

- Requirements for a Robust Over-the-Air update system
- High-level Overview of the OTA Process
- **Implementation with SWUpdate & U-Boot**
- Q & A

Our configuration today

Yocto

Linux distribution

Prepares the filesystem image that will be distributed in the update.

SWUpdate

Updater

Packages the update, downloads and installs it.

Memfault

OTA Backend

Distributes the update to eligible devices.

U-Boot

Bootloader

One of the most common bootloader today in the Linux embedded world.

Implementing A/B partitions with Yocto

Partitioning scheme

memfault-linux-sdk / meta-memfault-example / wic / **image-qemu.wks**

Memfault Inc Memfault Linux SDK 1.3.1-kirkstone (Build 10896) 470e5ae · 2 months ago History

Code Blame 12 lines (8 loc) · 457 Bytes Raw

```
part /boot --fstype=vfat --ondisk vda --offset 2048s --align 4 --fixed-size 63M --active --label boot

# rootfs slot A
part / --source rootfs --fstype=ext4 --ondisk vda --label root-a --align 4 --fixed-size 256M

# rootfs slot B
part --fstype=ext4 --ondisk vda --label root-b --align 4 --fixed-size 256M

# /media (auto-mounted via /etc/fstab)
part --fstype=ext4 --ondisk vda --label media --align 4 --fixed-size 448M

bootloader --ptable msdos --timeout 1
```

Use Open-Embedded Image Creator (wic)
and a kickstart file (.wks) to create a partitioned image.

```
root@qemuarm64:~# fdisk -l /dev/vda
Disk /dev/vda: 1024 MB, 1073741824 bytes, 2097152 sectors
16384 cylinders, 4 heads, 32 sectors/track
Units: sectors of 1 * 512 = 512 bytes

Device  Boot StartCHS    EndCHS        StartLBA     EndLBA    Sectors  Size Id Type
/dev/vda1 *  16,0,1      1023,3,32         2048     131071     129024 63.0M  c Win95 FAT32 (LBA)
/dev/vda2    1023,3,32   1023,3,32       131072     655359     524288  256M 83 Linux
/dev/vda3    1023,3,32   1023,3,32       655360    1179647     524288  256M 83 Linux
/dev/vda4    1023,3,32   1023,3,32      1179648    2097151     917504  448M 83 Linux

root@qemuarm64:~# mount |grep vda
/dev/vda2 on / type ext4 (rw,relatime)
/dev/vda1 on /boot type vfat (rw,relatime,errors=remount-ro)
/dev/vda4 on /media type ext4 (rw,relatime)
```

Only one of the two system partition is mounted.

Preparing the update package

```
$ bitbake core-image-minimal
...
$ bitbake swupdate-image

$ cpio -vt < tmp/deploy/images/qemuarm64/swupdate-image-qemuarm64.swu
-rw-r--r--   1 build    users        1111 Apr 25 00:53 sw-description
-rw-r--r--   1 build    users    48618608 Apr 25 00:53 base-image-qemuarm64.ext4.gz
```

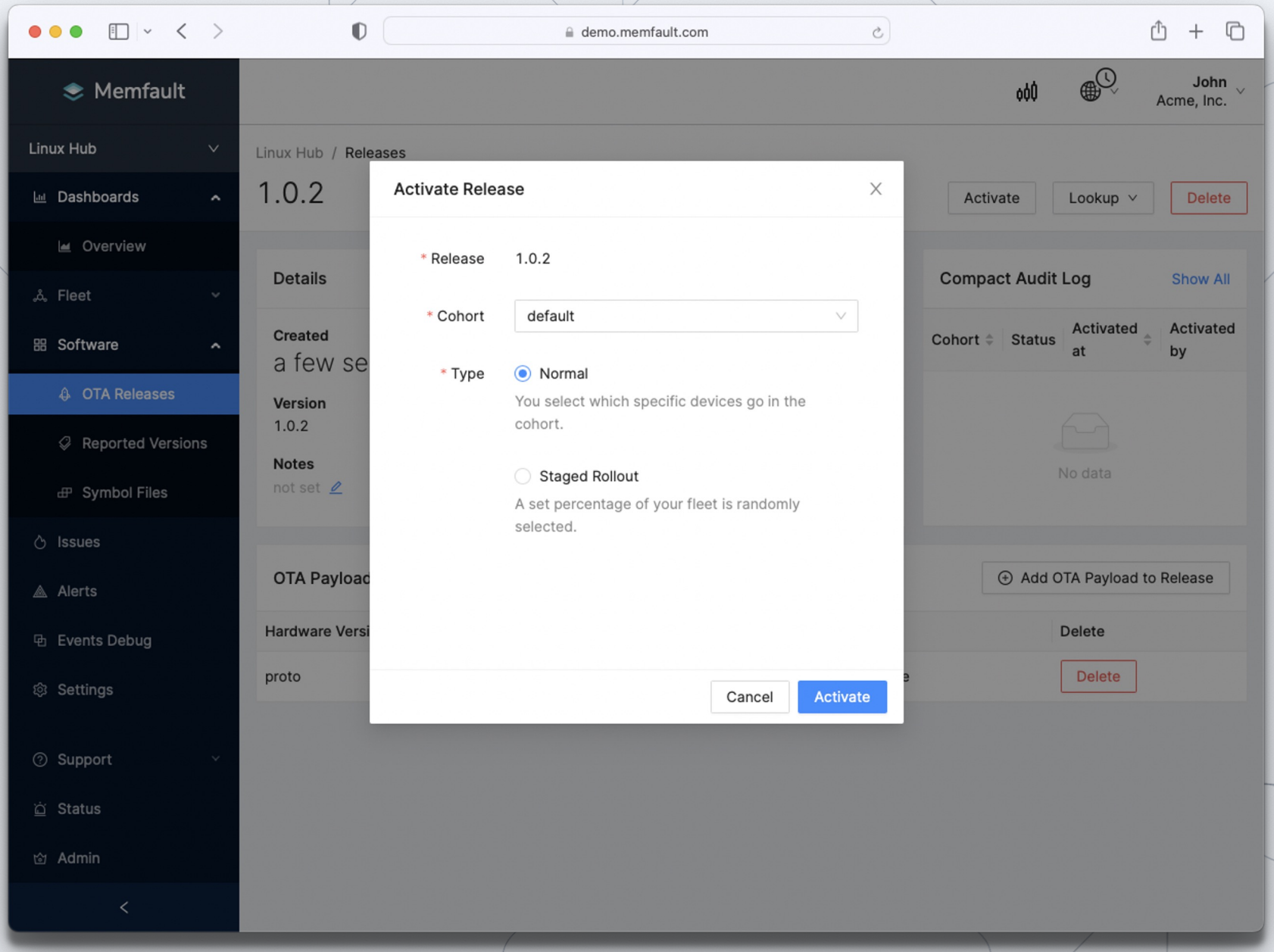
meta-swupdate includes a swupdate-image class

```
$ bitbake core-image-minimal
...
$ bitbake swupdate-image
```

```
$ cpio -vt < tmp/deploy/images/qemuarm64/swupdate-image-qemuarm64.swu
-rw-r--r--   1 build    users        1111 Apr 25 00:53 sw-description
-rw-r--r--   1 build    users    48618608 Apr 25 00:53 base-image-qemuarm64.ext4.gz
```

The update package contains a descriptor and a complete image of the system partition.

Distributing updates


demo.memfault.com
Memfault
John
Acme, Inc.
Linux Hub
Dashboards
Overview
Fleet
Software
OTA Releases
Reported Versions
Symbol Files
Issues
Alerts
Events Debug
Settings
Support
Status
Admin
Linux Hub / Releases
1.0.2
Activate
Lookup
Delete
Details
Created
a few se
Version
1.0.2
Notes
not set
Compact Audit Log
Show All
Cohort
Status
Activated at
Activated by
No data
OTA Payload
Add OTA Payload to Release
Hardware Versi
Delete
proto
Delete
Activate Release
* Release
1.0.2
* Cohort
default
* Type
Normal
You select which specific devices go in the cohort.
Staged Rollout
A set percentage of your fleet is randomly selected.
Cancel
Activate

Fetching updates

Running swupdate

```
root@qemuarm64:~# systemctl status swupdate
* swupdate.service - SWUpdate daemon
     Loaded: loaded (/lib/systemd/system/swupdate.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-05-10 00:21:02 UTC; 4min 41s ago
TriggeredBy: * swupdate.socket
       Docs: https://github.com/sbabic/swupdate
             https://sbabic.github.io/swupdate
   Main PID: 349 (swupdate)
      Tasks: 8 (limit: 424)
     Memory: 3.1M
     CGroup: /system.slice/swupdate.service
             |- 349 /usr/bin/swupdate -H qemuarm64 1.0 -e stable copy2 -f /tmp/swupdate.cfg -u " "
             `- 366 /usr/bin/swupdate -H qemuarm64 1.0 -e stable copy2 -f /tmp/swupdate.cfg -u " "

11:22:06 : SWUPDATE running :  [server_get_deployment_info] : No pending action on server.
11:23:06 : SWUPDATE running :  [start_suricatta] : Suricatta awakened.
11:23:06 : SWUPDATE running :  [channel_log_effective_url] : Channel s effective URL resolved to https:...qemu-tester
11:23:06 : SWUPDATE running :  [server_get_deployment_info] : No pending action on server.
```

Start SWUpdate Suricatta daemon.

Configuring SWUpdate-Suricatta

```
# cat /tmp/swupdate.cfg
...
suricatta :
{
  ...
  url = "https://device.memfault.com/api/v0/hawkbit";
  id = "sn-12345";
  gatewaytoken = "1gfiixxx";
};
identify = (
  {
    name = "memfault__current_version";
    value = "0.0.2";
  },
  {
    name = "memfault__hardware_version";
    value = "qemuarm64";
  },
  {
    name = "memfault__software_type";
    value = "main";
  } );
```

Generated configuration file contains details of the device.

Running swupdate

```
root@qemuarm64:~# systemctl status swupdate
* swupdate.service - SWUpdate daemon
     Loaded: loaded (/lib/systemd/system/swupdate.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-05-10 00:21:02 UTC; 4min 41s ago
TriggeredBy: * swupdate.socket
       Docs: https://github.com/sbabic/swupdate
             https://sbabic.github.io/swupdate
   Main PID: 349 (swupdate)
      Tasks: 8 (limit: 424)
     Memory: 3.1M
     CGroup: /system.slice/swupdate.service
             |- 349 /usr/bin/swupdate -H qemuarm64 1.0 -e stable copy2 -f /tmp/swupdate.cfg -u " "
             `- 366 /usr/bin/swupdate -H qemuarm64 1.0 -e stable copy2 -f /tmp/swupdate.cfg -u " "

11:22:06 : SWUPDATE running :  [server_get_deployment_info] : No pending action on server.
11:23:06 : SWUPDATE running :  [start_suricatta] : Suricatta awakened.
11:23:06 : SWUPDATE running :  [channel_log_effective_url] : Channel s effective URL resolved to https:...qemu-tester
11:23:06 : SWUPDATE running :  [server_get_deployment_info] : No pending action on server.
```

Run in mode “copy2”

Configure update mode

```
root@qemuarm64:~# cat /usr/lib/swupdate/conf.d/09-swupdate-args
rootfs=$(swupdate -g)

if [ "$rootfs" == '/dev/vda2' ]; then
    selection="-e stable,copy2"
else
    selection="-e stable,copy1"
fi
```

Running swupdate

```
root@qemuarm64:~# systemctl status swupdate
* swupdate.service - SWUpdate daemon
     Loaded: loaded (/lib/systemd/system/swupdate.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-05-10 00:21:02 UTC; 4min 41s ago
TriggeredBy: * swupdate.socket
       Docs: https://github.com/sbabic/swupdate
             https://sbabic.github.io/swupdate
   Main PID: 349 (swupdate)
      Tasks: 8 (limit: 424)
     Memory: 3.1M
     CGroup: /system.slice/swupdate.service
             |- 349 /usr/bin/swupdate -H qemuarm64 1.0 -e stable copy2 -f /tmp/swupdate.cfg -u " "
             `- 366 /usr/bin/swupdate -H qemuarm64 1.0 -e stable copy2 -f /tmp/swupdate.cfg -u " "

11:22:06 : SWUPDATE running :  [server_get_deployment_info] : No pending action on server.
11:23:06 : SWUPDATE running :  [start_suricatta] : Suricatta awakened.
11:23:06 : SWUPDATE running :  [channel_log_effective_url] : Channel s effective URL resolved to https:...qemu-tester
11:23:06 : SWUPDATE running :  [server_get_deployment_info] : No pending action on server.
```

Regularly poll server

Installing the update

```
# journalctl -u swupdate -f
00:32:16 SWUPDATE [channel_get_file] : Total download size is 47482 kB.
00:32:16 SWUPDATE [network_thread] : Incoming network request: processing...
00:32:16 SWUPDATE Software Update started !
00:32:16 SWUPDATE [network_initializer] : Software update started
00:32:17 SWUPDATE [extract_file_to_tmp] : Found file
00:32:17 SWUPDATE [extract_file_to_tmp] :         filename sw-description
00:32:17 SWUPDATE [extract_file_to_tmp] :         size 1111
00:32:17 SWUPDATE [get_common_fields] : Version 0.0.2
00:32:17 SWUPDATE [parse_hw_compatibility] : Accepted Hw Revision : 1.0
00:32:17 SWUPDATE [_parse_images] : Found compressed Image: base-image-qemuarm64.ext4.gz in device : /dev/vda3 for
handler raw
```

SWUpdate starts by downloading the descriptor

```
$ cat swupdate-description
software = {
    qemuarm64 = {
        stable: {
            copy1: {
                images: (
                    {
                        filename = "base-image-qemuarm64.ext4.gz";
                        type = "raw";
                        compressed = "zlib";
                        device = "/dev/vda2";
                    }
                );
                ...
            }
            copy2: {
                images: (
                    {
                        filename = "base-image-qemuarm64.ext4.gz";
                        type = "raw";
                        compressed = "zlib";
                        device = "/dev/vda3";
                    }
                );
                ...
            } } }
```

For each update mode, we specify how to write the update package

```
00:32:17 SWUPDATE [parse_hw_compatibility] : Accepted hw revision : 1.0
00:32:17 SWUPDATE [_parse_images] : Found compressed Image: base-image-qemuarm64.ext4.gz in device : /dev/vda3 for
handler raw
00:32:17 SWUPDATE [_parse_bootloader] : Bootloader var: rootpart = 3
00:32:17 SWUPDATE [check_hw_compatibility] : Hardware qemuarm64 Revision: 1.0
00:32:17 SWUPDATE [check_hw_compatibility] : Hardware compatibility verified
00:32:17 SWUPDATE [extract_files] : Found file
00:32:17 SWUPDATE [extract_files] :          filename sw-description.in
00:32:17 SWUPDATE [extract_files] :          size 1177 Not required: skipping
00:32:17 SWUPDATE [extract_files] : Found file
00:32:17 SWUPDATE [extract_files] :          filename base-image-qemuarm64.ext4.gz
00:32:17 SWUPDATE [extract_files] :          size 48618608 required
00:32:24 SWUPDATE [extract_padding] : Expecting 276 padding bytes at end-of-file
00:32:24 SWUPDATE [channel_log_effective_url] : Channel s effective URL resolved to https://ota-
cdn.memfault.com/3544/4307/6635564307?token=V1cfiiGBj1qdJrZSK0bYFDp7BwoVJ60plGCpQE3E6Ew&expires=1683702000
00:32:24 SWUPDATE [network_initializer] : Valid image found: copying to FLASH
00:32:24 SWUPDATE Installation in progress
00:32:24 SWUPDATE [install_single_image] : Found installer for stream base-image-qemuarm64.ext4.gz raw
00:32:30 SWUPDATE successful ! SWUPDATE successful !
...
[ 1768.757610] systemd-shutdown[1]: Syncing filesystems and block devices.
[ 1768.759655] systemd-shutdown[1]: Rebooting.
[ 1768.763348] reboot: Restarting system
```

The image is written directly to disk

Rebooting into the update

Rebooting into the update

```
# cat swupdate-description
...
  copy2: {
      ...
      images: ({ ... device = "/dev/vda3" ... })
      uboot: (
          {
              name = "rootpart";
              value = "3";
          }
      );
    }

# cat /tmp/swupdate.cfg
globals :
{
  postupdatecmd = "memfaultctl reboot --reason 3";
};
```

SWUpdate will also write a U-Boot environment variable.

Bootloader environment variables

memfault-linux-sdk / meta-memfault-example / recipes-bsp / u-boot / files / 0001-env-in-fat-defconfig-2022.01.patch

Memfault Inc Memfault Linux SDK 1.1.0-kirkstone (Build 4469)

Code Blame 52 lines (51 loc) · 1.47 KB

```
diff -Naur a/configs/qemu_arm64_defconfig b/configs/qemu_arm64_defconfig
--- a/configs/qemu_arm64_defconfig      2022-03-29 13:53:40.086225070 +0100
+++ b/configs/qemu_arm64_defconfig      2022-03-29 14:09:48.213001877 +0100
@@ -3,7 +3,7 @@
 CONFIG_ARCH_QEMU=y
 CONFIG_SYS_MALLOC_LEN=0x1000000
 CONFIG_NR_DRAM_BANKS=1
-CONFIG_ENV_SIZE=0x40000
+CONFIG_ENV_SIZE=0x4000
 CONFIG_ENV_SECT_SIZE=0x40000
 CONFIG_AHCI=y
 CONFIG_DISTRO_DEFAULTS=y
@@ -26,8 +26,10 @@
 CONFIG_CMD_TPM=y
 CONFIG_CMD_MTDPARTS=y
 CONFIG_OF_BOARD=y
-CONFIG_ENV_IS_IN_FLASH=y
-CONFIG_ENV_ADDR=0x4000000
+CONFIG_ENV_IS_IN_FAT=y
+CONFIG_ENV_FAT_INTERFACE="virtio"
+CONFIG_ENV_FAT_DEVICE_AND_PART="0:1"
+CONFIG_ENV_FAT_FILE="uboot.env"
 CONFIG_SCSI_AHCI=y
 CONFIG_AHCI_PCI=y
 CONFIG_DFU_TFTP=y
```

```
U-Boot 2022.01 (Jan 10 2022 - 18:46:34 +0000)

DRAM:  512 MiB
Flash: 64 MiB
In:    pl011@9000000
Out:   pl011@9000000
Err:   pl011@9000000
Net:   eth0: virtio-net#32
Loading Environment from FAT... OK
Hit any key to stop autoboot:  0

=> env print rootpart
rootpart=3
```

```
Poky (Yocto Project Reference Distro) 4.0.5 qemuarm64 ttyAMA0

qemuarm64 login: root

root@qemuarm64:~# fw_printenv rootpart
rootpart=3
```

Configured at compile time.
Available in the bootloader and at runtime.

Bootloader

Bootloader script

memfault-linux-sdk / meta-memfault-example / recipes-bsp / rpi-uboot-scr / files / **boot.cmd.in**

Memfault Inc Memfault Linux SDK 1.3.1-kirkstone (Build 10896) 470e5ae · 2 months ago History

Code Blame 6 lines (6 loc) · 386 Bytes Raw

```
1  saveenv
2  fdt addr ${fdt_addr} && fdt get value bootargs /chosen bootargs
3  if env exists rootpart;then echo Booting from mmcblk0p${rootpart};else setenv rootpart 2;echo rootpart not set, default to ${rootpart};fi
4  load mmc 0:${rootpart} ${kernel_addr_r} boot/@@KERNEL_IMAGETYPE@@
5  setenv bootargs "${bootargs} root=/dev/mmcblk0p${rootpart}"
6  @@KERNEL_BOOTCMD@@ ${kernel_addr_r} - ${fdt_addr}
```

The bootloader script defines the default boot partition, and implements the A/B switch.

Bootloader script

memfault-linux-sdk / meta-memfault-example / recipes-bsp / rpi-uboot-scr / files / **boot.cmd.in**

Memfault Inc Memfault Linux SDK 1.3.1-kirkstone (Build 10896) 470e5ae · 2 months ago History

Code Blame 6 lines (6 loc) · 386 Bytes Raw

```
1  saveenv
2  fdt addr ${fdt_addr} && fdt get value bootargs /chosen bootargs
3  if env exists rootpart;then echo Booting from mmcblk0p${rootpart};else setenv rootpart 2;echo rootpart not set, default to ${rootpart};fi
4  load mmc 0:${rootpart} ${kernel_addr_r} boot/@@KERNEL_IMAGETYPE@@
5  setenv bootargs "${bootargs} root=/dev/mmcblk0p${rootpart}"
6  @@KERNEL_BOOTCMD@@ ${kernel_addr_r} - ${fdt_addr}
```

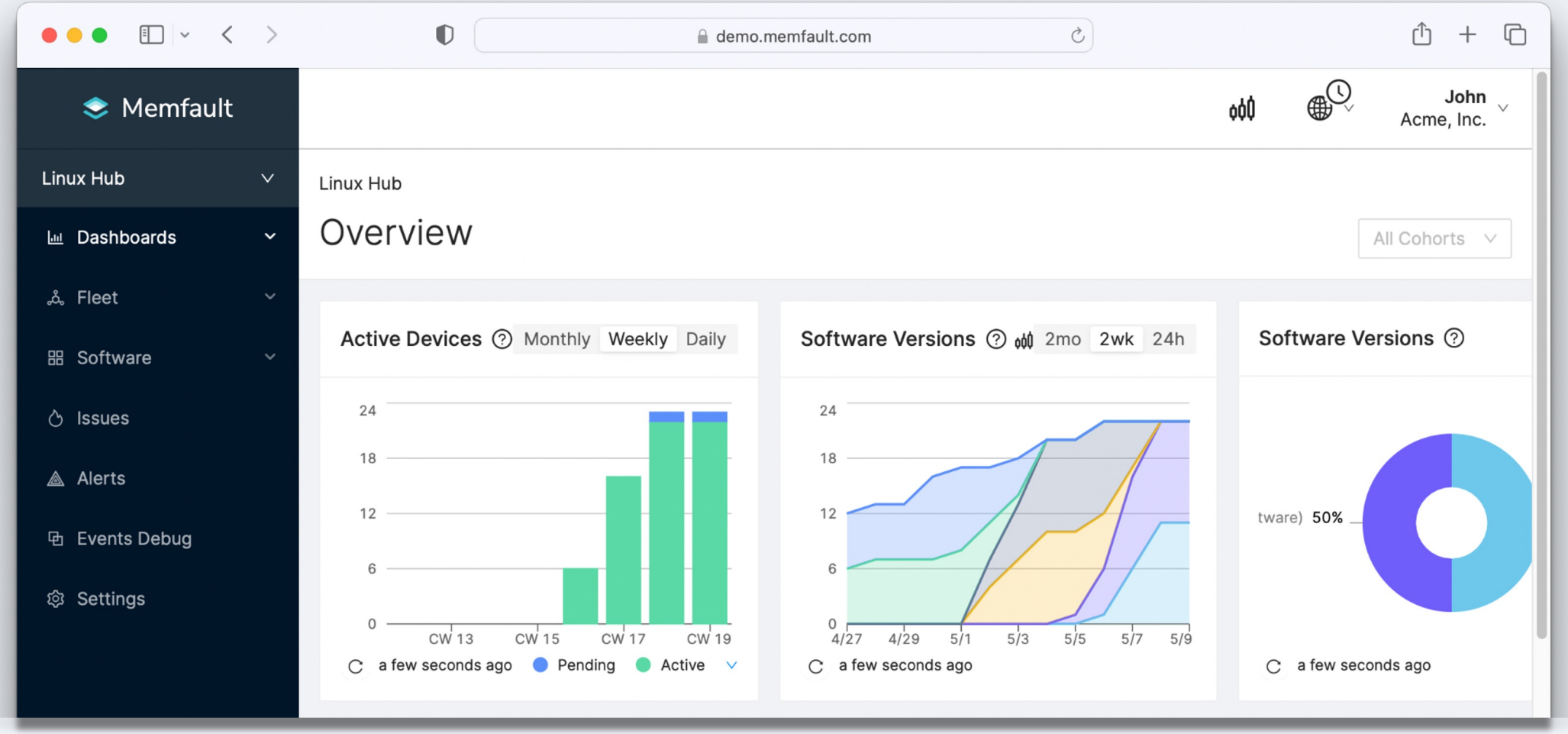
Kernel, Device-Tree and System partition are all loaded from the A/B partition. This enables us to update all of them with one package.

Notifying OTA Backend

```
root@qemuarm64:~# cat /usr/lib/swupdate/conf.d/09-swupdate-args
...
state=$(fw_printenv ustate | cut -f 2 -d'=')
if [ "$state" == 1 ]; then
 # Confirm update Success to server
    SWUPDATE_SURICATTA_ARGS="-c 2"
else
    SWUPDATE_SURICATTA_ARGS=" "
fi
```

```
# journalctl -u swupdate -f
00:32:52 SWUPDATE running :  [settings_into_dict] : Identify for configData: memfault__current_version --> 0.0.2
00:32:52 SWUPDATE running :  [settings_into_dict] : Identify for configData: memfault__hardware_version --> qemuarm64
00:32:52 SWUPDATE running :  [settings_into_dict] : Identify for configData: memfault__software_type --> main
00:32:52 SWUPDATE running :  [server_handle_initial_state] : Got state=2 from command line.
00:32:53 SWUPDATE running :  [channel_log_effective_url] : Channel s effective URL resolved to
https://device.memfault.com/api/v0/hawkbit/default/controller/v1/mf14
00:32:53 SWUPDATE running :  [server_set_config_data] : ConfigData:
https://device.memfault.com/api/v0/hawkbit/default/controller/v1/mf14/configData
00:32:53 SWUPDATE running :  [server_get_deployment_info] : No pending action on server.
00:32:53 SWUPDATE running :  [handle_feedback] : No active update available, nothing to report to server.
```

Conclusion


demo.memfault.com
Memfault
John
Acme, Inc.
Linux Hub
Dashboards
Fleet
Software
Issues
Alerts
Events Debug
Settings
Linux Hub
Overview
All Cohorts
Active Devices
Monthly
Weekly
Daily
24
18
12
6
0
CW 13
CW 15
CW 17
CW 19
a few seconds ago
Pending
Active
Software Versions
2mo
2wk
24h
4/27
4/29
5/1
5/3
5/5
5/7
5/9
a few seconds ago
Software Versions
tware) 50%
a few seconds ago

Memfault for Linux

OTA Backend

Compatible with SWUpdate, Mender and more.

Cohorts management

Progressive roll-out

Metrics

Collect and aggregate metrics from large fleet of devices

Correlate metrics to firmware version

Coredumps

Capture coredumps from crashes

Add symbol names, variables, threads, etc

Automatic de-duplication of crashes

and Reboot tracking, Log collection, Device attributes, powerful APIs.

Try this at home!

https://docs.memfault.com/docs/linux/quickstart

Memfault Linux SDK

- Docker container to easily build Yocto images
- Pre-configured for OTA with SWUpdate and U-Boot
- Runs inside QEMU or on RasperryPis

```
dev$ git clone git@github.com:memfault/memfault-linux-sdk.git
dev$ cd memfault-linux-sdk/docker
dev$ export MEMFAULT_PROJECT_KEY=abcdef
dev$ ./run.sh -b
docker$ bitbake memfault-image
...
docker$ bitbake swupdate-image
docker$ q


U-Boot 2022.01 (Jan 10 2022 - 18:46:34 +0000)

DRAM:  512 MiB
Flash: 64 MiB
In:    pl011@9000000
Out:   pl011@9000000
Err:   pl011@9000000
Net:   eth0: virtio-net#32
Loading Environment from FAT... OK
Hit any key to stop autoboot:  0
...
```

Thank You!

- Interrupt.com: OTA for Embedded Linux SWUpdate Manual
- memfault.com
- twitter.com/memfault
- interrupt-slack.herokuapp.com

- We’re hiring!

Memfault

Question: How to implement a boot counter?

We can implement a boot counter using the techniques presented earlier:

```
if env exists bootcount; then
  setexpr bootcount ${bootcount} + 1
  if test ${bootcount} -ge 3; then
    echo Bootcount limit reached - Rolling back...
    if test ${rootpart} -eq 2; then
      setenv rootpart 3
    else
      setenv rootpart 2
    fi
    saveenv
  else
    echo Bootcount is ${bootcount}
    saveenv
  fi
else
  echo Initializing bootcount at 1.
  setenv bootcount 1
  saveenv
fi
```

Question: Delta Updates

SWUpdate supports delta updates using ZChunk (FOSS)

- Update is again a full-filesystem image.
  - It needs to be converted to the zchunk format which will add a header with a checksum of each chunk in the update.

- SWUpdate will download only the header file and compare the checksum of each chunk in the currently active partition to the checksum in the new update
  - If the checksum has not changed, SWUpdate will use data from the active partition.
  - If the checksum has changed, SWUpdate will download only this chunk from the server.

- All download requests are executed using http range requests and grouped together to be efficient.

See https://sbabic.github.io/swupdate/delta-update.html and https://www.thegoodpenguin.co.uk/blog/delta-ota-update-with-swupdate/

Question: Verifying the source of the update

You can generate a public/private key to sign all updates and ask swupdate to verify the signature of the updates before installing them.

See https://sbabic.github.io/swupdate/signed_images.html