

Software Update Solutions for the Yocto Project and OpenEmbedded

Leon Anavi

Konsulko Group

leon.anavi@konsulko.com

leon@anavi.org

Embedded Linux Conference Europe 2020

Konsulko
Group

- Services company specializing in Embedded Linux and Open Source Software
- Hardware/software build, design, development, and training services
- Based in San Jose, CA with an engineering presence worldwide
- <http://konsulko.com/>

Agenda

- The Yocto Project and OpenEmbedded
- Challenges for software updates of embedded Linux devices
- Overview of open source software update solutions
- Closer look at Mender.io, RAUC and libostree (OSTree)
- Conclusions
- Q&A

Embedded Linux Devices



Embedded Linux devices dominate various different industries. To save time and money, best practices are to create a custom distribution based on proven:

- Build system
- Software update mechanism

The Yocto Project

- Open source collaborative project of the Linux foundation for creating custom Linux-based systems for embedded devices using the OpenEmbedded Build System
- OpenEmbedded Build System includes BitBake and OpenEmbedded Core
- Poky is a reference distribution of the Yocto Project provided as metadata, without binary files, to bootstrap your own distribution for embedded devices
- Bi-annual release cycle
- Long term support (LTS) release covering two-year period

Yocto Project Releases



Codename	Version	Release Date	Support Level
Gatesgarth	3.2	Oct 2020	Dev
Dunfell	3.1	April 2020	Long Term Stable
Zeus	3.0	October 2019	Community
Warrior	2.7	April 2019	EOL
Thud	2.6	Nov 2018	EOL
Sumo	2.5	April 2018	EOL
Rocko	2.4	Oct 2017	EOL

- **Recipe:** The most common form of metadata. A recipe contains instructions as a list of settings and tasks for building packages that are then used to build the binary image. A recipe describes source code source, additional patches, dependencies for libraries or for other recipes as well as configuration and compilation options.
- **Layer:** A collection of related recipes and configurations. Layers also isolate information used when building for multiple architectures. Layers are hierarchical in their ability to override previous specifications.
- Documentation:
<https://www.yoctoproject.org/docs/latest/mega-manual/mega-manual.html>

Things to Consider for Software Updates (1/2)



- Are there any limitations of the disk space for the downloaded updates?
- Are there any limitations of the network bandwidth for the data transfer?
- Do you need a container-based solution?
- Do you need A/B or binary delta updates?
- How to upgrade: over the air, cable, USB stick, etc?
- Is the device mission critical?

Things to Consider for Software Updates (2/2)



- Is there Yocto/OpenEmbedded BSP for the hardware you use?
- Is software update technology compatible with the YP, OE and the BSP?
- Which Yocto Project released do you need for your product?
- How to update fleet of devices?

Popular open source solution for updates

- Mender
- RAUC
- SWUpdate
- Swupd
- UpdateHub
- Balena
- Snap
- OSTree
- Aktualizr
- Aktualizr-lite
- QtOTA
- Torizon
- Rpm-ostree (used in Project Atomic)

Common Embedded Linux Update Strategies



- A/B updates (dual redundant scheme)
- Delta updates
- Container-based updates
- Combined strategies

■ Combined Strategies

- Container technology has changed the way application developers interact with the cloud and some of the good practices are nowadays applied to the development workflow for embedded devices and IoT
- Containers make applications faster to deploy, easier to update and more secure through isolation
- Yocto/OE layer meta-virtualization provides support for building Xen, KVM, Libvirt, docker and associated packages necessary for constructing OE-based virtualized solutions
- There are use cases on powerful embedded devices where contains are combined with A/B updates of the base Linux distribution built with Yocto/OE

Mender

- Available as a free open source or paid commercial and enterprise plans
- A/B update scheme for open source and all plans as well as delta updates for professional and enterprise plans
- Back-end services (Hosted Mender)
- Written in Go, Python, JavaScript
- Yocto/OE integration through meta-mender and extra BSP layers:
<https://github.com/mendersoftware/meta-mender>
- Source code in GitHub under Apache 2.0



Mender Supported Devices

The following hardware platforms and development boards are supported:

- Raspberry Pi
- BeagleBone
- Intel x86-64
- Rockchip
- Allwinner
- NXP
- And more in: <https://github.com/mendersoftware/meta-mender-community>

meta-mender-community



🔍 dunfell 7 branches 0 tags

Go to file Add file Code

This branch is 30 commits ahead, 18 commits behind zeus. Pull request Compare

mirzak Merge pull request #175 from BoulderAI/dunfell+documentation-updates 3d2c631 11 days ago 341 commits

.github	Added a configuration file for stalebot	6 months ago
meta-mender-atmel	meta-mender-atmel: sama5d27_som1: rebase patch on zeus	2 months ago
meta-mender-beaglebone	switch to upstream zeus branch	3 months ago
meta-mender-clearfog	clearfog: fix missing \$ in local.append template	13 months ago
meta-mender-coral	coral: add missing CONFIG_SYS_REDUNDAND_ENVIRONMENT	3 months ago
meta-mender-intel	intel: update to zeus	7 months ago
meta-mender-nxp	meta-mender-nxp: imx7s-warp: rebase patch on zeus	2 months ago
meta-mender-odroid	odroid: update to zeus	7 months ago
meta-mender-qemu	[qemu] Update the Poky branch description in the README to dunfell	last month
meta-mender-raspberrypi	Bump meta-mender-raspberrypi to dunfell	21 days ago
meta-mender-rockchip	rockchip: update manifest file to point to thud branches	2 years ago
meta-mender-sunxi	meta-mender-sunxi: Update README	2 months ago
meta-mender-tegra	Update documentation and scripts for dunfell	12 days ago
meta-mender-up	add support for UP2 board	2 years ago
meta-mender-update-modules	Update srcrev for upstream fix; path to version-compare script... again	7 months ago
meta-mender-variscite	u-boot-variscite: Adjust patches to latest upstream version.	17 months ago

About

Community supported integration layers for Mender on various boards

Readme

Apache-2.0 License

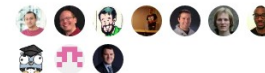
Releases

No releases published

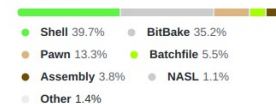
Packages

No packages published

Contributors 10



Languages



Mender A/B updates supports two client modes:

- Managed (default) - client running as a daemon polls the server for updates
- Standalone - updates are triggered locally which is suitable for physical media or any network update in pull mode

```
SYSTEMD_AUTO_ENABLE_pn-mender = "disable"
```

```
$ cd tmp/deploy/images/raspberrypi4
```

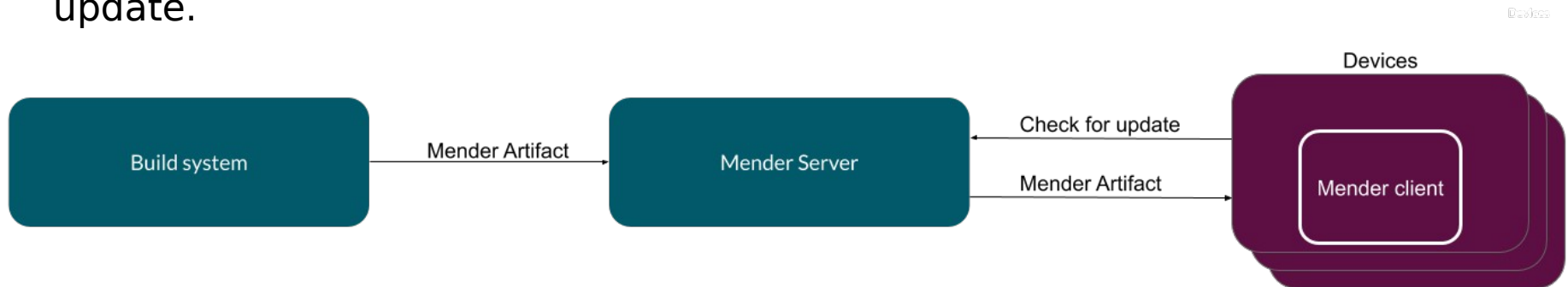
```
$ python3 -m http.server
```

```
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

```
$ mender -install http://example.com:8000/core-image-base-raspberrypi4.mender
```


Steps to install Mender A/B update on embedded Device:

- Apply update
- Reboot
- On the first boot after a successful update, the Mender client will commit the update.



- A lightweight update client that runs on an Embedded Linux device and reliably controls the procedure of updating the device with a new firmware revision
- Provides tool for the build system to create, inspect and modify update bundles
- Uses X.509 cryptography to sign update bundles
- Compatible with the Yocto Project, PTXdist and Buildroot



RAUC Licenses

- RAUC - GPLv2.1
<https://github.com/rauc/rauc>
- meta-rauc - MIT
<https://github.com/rauc/meta-rauc>
- rauc-hawkbite - GPLv2.1
<https://github.com/rauc/rauc-hawkbite>
- rauc-hawkbite-updater - GPLv2.1
<https://github.com/rauc/rauc-hawkbite-updater>

RAUC Integration Steps

- Select an appropriate bootloader
- Enable **SquashFS** in the Linux kernel configurations
- **ext4** root file system (RAUC does not have an ext2 / ext3 file type)
- Create specific partitions that matches the RAUC slots
- Configure Bootloader environment and create a script to switch RAUC slots
- Create a certificate and a keyring to RAUC's system.conf

RAUC Example with Raspberry Pi 4

- Integration layer:
<https://github.com/leon-anavi/meta-rauc-community/tree/master/meta-rauc-raspberrypi>
- Add layers to bblayers.conf and in local.conf:

```
MACHINE = "raspberrypi4"  
DISTRO_FEATURES_append = " systemd"  
VIRTUAL-RUNTIME_init_manager = "systemd"  
DISTRO_FEATURES_BACKFILL_CONSIDERED = "sysvinit"  
VIRTUAL-RUNTIME_initscripts = ""  
IMAGE_INSTALL_append = " rauc"  
IMAGE_FSTYPES="tar.bz2 ext4 wic.bz2 wic.bmap"  
SDIMG_ROOTFS_TYPE="ext4"  
ENABLE_UART = "1"  
RPI_USE_U_BOOT = "1"  
PREFERRED_PROVIDER_virtual/bootloader = "u-boot"  
WKS_FILE = "sdimage-dual-raspberrypi.wks"
```

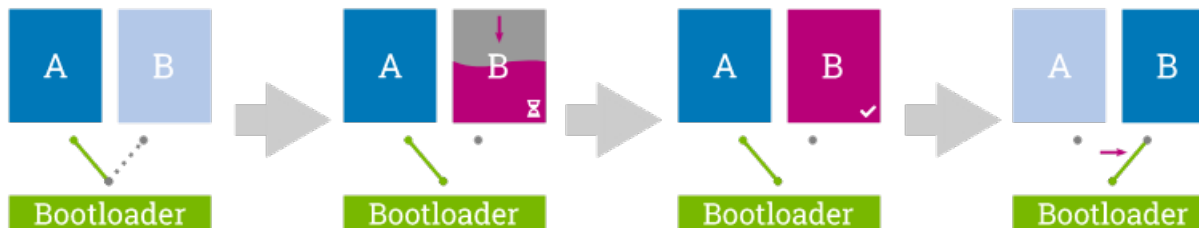
Manual RAUC Update of Raspberry Pi 4

- On the build system:

```
cd tmp/deploy/images/raspberrypi4/  
python3 -m http.server
```

- On the embedded device:

```
wget http://example.com:8000/update-bundle-raspberrypi4.raucb -P /tmp  
rauc install /tmp/update-bundle-raspberrypi4.raucb  
reboot
```



- A shared library and suite of command line tools for committing and downloading bootable filesystem trees
- Supports “git-like” model for incremental atomic upgrades of a filesystem using binary deltas.
- After an update a reboot is required
- Persistent data is kept in /var and /etc
- Previously was known as **OSTree**
- Exact steps for adapting an existing mainstream GNU/Linux distribution to libostree:
<https://ostreedev.github.io/ostree/adapting-existing/>

Libostree Source Code

- Written in C
- Language bindings available through GObject Introspection (GI)
- Compatible with multiple bootloader options: GRUB, U-Boot and initramfs
- Source code available at GitHub under GPLv2 license:
<https://github.com/ostreedev/ostree>
- More than 100 contributors
- Documentation:
<https://ostreedev.github.io/ostree/>

OSTree, Yocto and OpenEmbedded

gerrit.automotivelinux.org/gerrit/c/AGL/meta-agl/+/5813

Gerrit CHANGES DOCUMENTATION BROWSE

Merged as 8176e8b 5813 ostree: Add OSTree

Updated Jun 19, 2016

Owner Leon Anavi

Uploader Jan-Simon Moeller

Committer Jan-Simon Moeller

Assignee

Reviewers Jan-Simon Moeller
Walt Miner
Tadao Tanikawa

AND 6 MORE

CC

Repo | Branch AGL/meta-agl | master

Parent 13ea965

Topic No topic

Hashtags

Code-Review +2 Jan-Simon Moeller

Verified No votes.

Other labels

ci-image-build No votes.

ci-image-boot-test No votes.

ostree: Add OSTree

OSTree is a tool for managing bootable, immutable, versioned filesystem trees. It will be used to apply updates downloaded by RVI SOTA client.

Bug-AGL: SPEC-194

Change-Id: I286b9ce8631c6ef8632f134f89e85b462cb39fd3

Signed-off-by: Leon Anavi <leon.anavi@konsulko.com>



index : meta-openembedded

Collection of OpenEmbedded layers

about summary refs log tree **commit** diff stats

path: root/meta-oe/recipes-extended/ostree

author Alex Kiernan <alex.kiernan@gmail.com>

committer Khem Raj <raj.khem@gmail.com>

commit 59180f9f910cfd541d95f0619c597a870093dc1c (patch)

tree 0e3971fd7be7ed237ac0fb19421ff64bc946a84f /meta-oe/recipes-extended/ostree

parent 1849429a791457250236778793c36a12f0df3194 (diff)

download meta-openembedded-59180f9f910cfd541d95f0619c597a870093dc1c.tar.gz
meta-openembedded-59180f9f910cfd541d95f0619c597a870093dc1c.tar.bz2
meta-openembedded-59180f9f910cfd541d95f0619c597a870093dc1c.zip

2019-09-22 13:06:38 +0100
2019-09-22 08:13:17 -0700

ostree: Add recipe

Signed-off-by: Alex Kiernan <alex.kiernan@gmail.com>

Signed-off-by: Khem Raj <raj.khem@gmail.com>

Aktualizr and Aktualizr-lite

- Aktualizr is an open source client for embedded devices relying on OSTree to download and install updates. Developer by HERE (which acquired ATS Advanced Telematic Systems GmbH)
- Aktualizr is compatible with GENIVI SOTA and Uptane requirements
- Written in C++
- Source code available in GitHub under Mozilla Public License 2.0:
<https://github.com/advancedtelematic/aktualizr>
- Aktualizr-lite is a lightweight open source version developed by foundries.io which allows anonymous access and requires devices to be always up to date
<https://github.com/foundriesio/aktualizr-lite>

OSTree Based Solutions for Embedded Linux



- **HERE OTA Connect** with Aktualizr, meta-updater and appropriate BSP layers for Raspberry Pi, QEMU, Intel x86-64 (Minnowboard), RISC-V, TI and Renesas boards:
<https://docs.ota.here.com/getstarted/dev/index.html>
On 31 August 2020 HERE removed OTA Connect from their product portfolio
- **Automotive Grade Linux (AGL)** agl-sota feature based on meta-updater:
<https://wiki.automotivelinux.org/subsystem/agl-sota/ostree>
- **Foundries.io** with Aktualizr-lite, meta-updater and meta-Imp
<https://docs.foundries.io/latest/>
- **Torizon OTA** for Toradex Apalis, Colibri and Verdin i.MX devices with eMMC, using Aktualizr and layer meta-toradex-torizon
<https://labs.toradex.com/projects/torizon-over-the-air>

More OSTree Based Solutions

- QtOTA
<https://doc.qt.io/QtOTA/>
- Gnome Continuous
<https://wiki.gnome.org/Projects/GnomeContinuous>
- Project Atomic
<https://www.projectatomic.io/>
- Flatpak
<https://flatpak.org/>
- Pulp Platform
<https://pulpproject.org/>

Eclipse hawkBit

- Domain independent back-end framework for rolling out software updates to constrained edge devices as well as more powerful controllers and gateways connected to IP based networking infrastructure
- Written in Java
- Available in GitHub under EPL-1.0 License
- Compatible with **RAUC** and **SWUpdate**
- <https://www.eclipse.org/hawkbit/>



Eclipse hawkBit



The screenshot displays the hawkBit web interface, which is used for software provisioning. The interface is divided into several main sections:

- Deployment Management:** This section is the primary focus. It features a sidebar on the left with navigation options: Deployment, Rollout, Target Filters, Distributions, Upload, and System Config. The main area is titled "Deployment Management" and contains several panels:
 - Filters:** A "Simple Filter" section with options for "NO TAG", "Europe", and "Simulated".
 - Targets:** A table listing various simulated targets (e.g., dmfsimulated0, dmfsimulated1, etc.) with status indicators (green circles).
 - Distributions:** A table showing distribution sets like "Baseline" with versions 1 and 2.
 - Action History:** A table titled "Action History For dmfsimulated0" showing actions for "Baseline:2" and "Baseline:1" with columns for Active, Distributionset, Date and time, Status, Forced, and Actions.
 - Target Details:** A panel for "Target : dmfsimulated0" showing details like Controller id, Last poll, Address, and Security token.
 - Distribution set Details:** A panel for "Distribution set : Baseline:1" showing details like Type (OS only) and Required Migration Step (No).
- Upload Management:** A section on the right titled "Upload Management" with a "Software Module" table showing "A Firmware" with versions 1 and 2. Below this is an "Artifact Details of A Firmware:1" section with a table for file names, sizes, and last modified dates. A "Drop files to upload" area is visible at the bottom of this section.

Conclusions

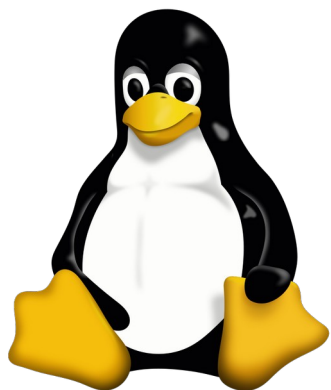
- Most open source solutions for software updates already have support for the Yocto Project and OpenEmbedded
- It is recommended to use actively maintained Yocto releases, for example a LTS
- Software updates depend on the bootloader, U-Boot is often preferred
- Mender is an excellent choice for A/B updates, alternatives are RAUC and SWUpdate
- Libostree is commonly used as a core technology in the various open source solutions for delta updates
- Combining A/B updates of the host OS with containers from meta-virtualization is nowadays also often used for embedded Linux devices

Recommended Related Talks



- Debian or Yocto Project?, Chris Simmonds, ELCE 2019
<https://elinux.org/images/3/39/Debian-or-yocto-csimmonds-elce-2019.pdf>
- A Comparison of Linux Software Update Technologies, Matt Porter, ELCE 2016
<https://www.konsulko.com/portfolio-item/comparison-of-linux-software-update-technologies/>
- How we added software updates to AGL, Phil Wise, ELC NA 2017
<http://events17.linuxfoundation.org/sites/events/files/slides/How%20we%20added%20software%20updates%20to%20AGL.pdf>
- Secure and Safe Updates for Your Embedded Device, Enrico Jörns, FOSDEM 2017
https://archive.fosdem.org/2017/schedule/event/secure_safe_embedded_updates/attachments/slides/1758/export/events/attachments/secure_safe_embedded_updates/slides/1758/Secure_and_Safe_Updates_For_Your_Embedded_Device.pdf
- Secure OTA Updates For Rich IoT Rity Platform Using Mender Update Modules, Bartosz Golaszewski, ELCE 2019, Lyon
<http://baylibre.com/pub/conferences/2019/ELC-E/ELCE-2019-secure-ota-updates-brgl.pdf>
- Evolution of (OTA) Update in the IoT World - Stefano Babic, ELC NA 2019
https://static.sched.com/hosted_files/ossna19/4f/Evolution_of_OTA_Update_in_the_IoT_world.pdf

Thank You!



Useful links:

- <https://mender.io/>
- <https://rauc.io/>
- <https://ostreedev.github.io/ostree/>
- <https://sbabic.github.io/swupdate/swupdate.html>
- <https://wiki.automotivelinux.org/agl-distro/libostree-demo>
- <https://docs.ota.here.com/ota-client/latest/index.html>
- <https://www.konsulko.com/getting-started-with-rauc-on-raspberry-pi-2/>

