

Apertis

FOSS GNU/Linux OS for embedded

Denis Pynkin

Collabora

Aug 23, 2019





Presentation Outline

Apertis

Core components

- libostree

- Booting

- Update manager

Build infrastructure

- Debosc

- CI for OSTree

Distro features



- ▶ Aimed for automotive embedded
- ▶ Ubuntu Debian derivative*
- ▶ AppArmor LSM
- ▶ OSTree-based images
- ▶ Updates & rollbacks out-of-box*
- ▶ GPLv3-free core*
- ▶ Open & free CI
- ▶ SDK for apps developers

* Work in progress

Targets



Architecture:

- ▶ **x86_64**
Minnowboard, Qemu
- ▶ **arm64**
Renesas R-Car
- ▶ **armhf**
i.MX6 SabreLite

Targets



Output:

- ▶ Packages
- ▶ Rootfs
 - ▶ System
 - ▶ Development
- ▶ OSTree repo
- ▶ Images
 - ▶ minimal
 - ▶ target
 - ▶ SDK
- ▶ OSTree
- ▶ APT

Presentation Outline

Apertis

Core components

libostree

Booting

Update manager

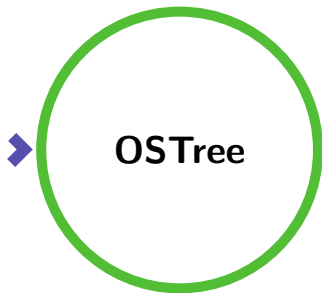
Build infrastructure

Debos

CI for OSTree

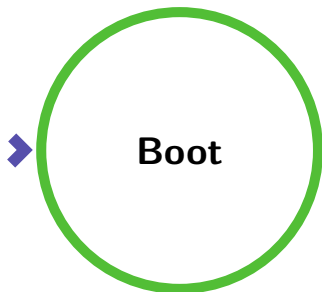
Open First

libostree



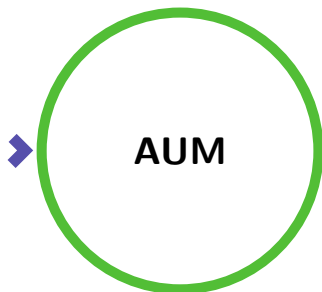
- ▶ Management library
- ▶ Numerous OSes and variants
- ▶ Atomic versions switch
- ▶ Stateless root filesystem
- ▶ Git-like system updates
- ▶ Multiple upgrade sources
- ▶ Reliable updates and rollbacks
- ▶ OTA & offline updates
- ▶ Disk efficient:
deduplication and hard links
- ▶ Operates at the filesystem layer

System boot



- ▶ Bootloaders (+rollback)
 - ▶ U-Boot
 - ▶ UEFI
- ▶ Linux kernel
- ▶ Initrd (+ostree)

Apertis update manager



- ▶ Update service
- ▶ CLI utility

Tasks:

- ▶ Offline updates
- ▶ OTA updates
- ▶ Validation of updates
- ▶ Rollback support
- ▶ Signature validation¹

¹ Work in progress

Presentation Outline

Apertis

Core components

libostree

Booting

Update manager

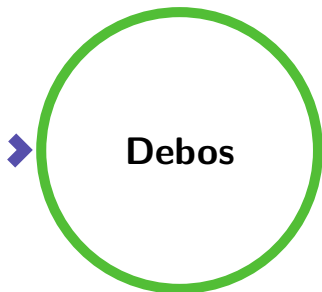
Build infrastructure

Debos

CI for OSTree

Open First

Image builder



- ▶ Easy to start
- ▶ Human-friendly 'yaml' recipe
- ▶ Simple actions
- ▶ Loosely coupled actions
- ▶ No 'root' privileges
- ▶ libostree integration
- ▶ External build environment (docker)

Actions

- ▶ debootstrap
- ▶ apt
- ▶ download
- ▶ overlay
- ▶ run
- ▶ raw
- ▶ pack
- ▶ unpack
- ▶ image-partition
- ▶ filesystem-deploy
- ▶ ostree-commit
- ▶ ostree-deploy

Recipe

YAML + go lang text template

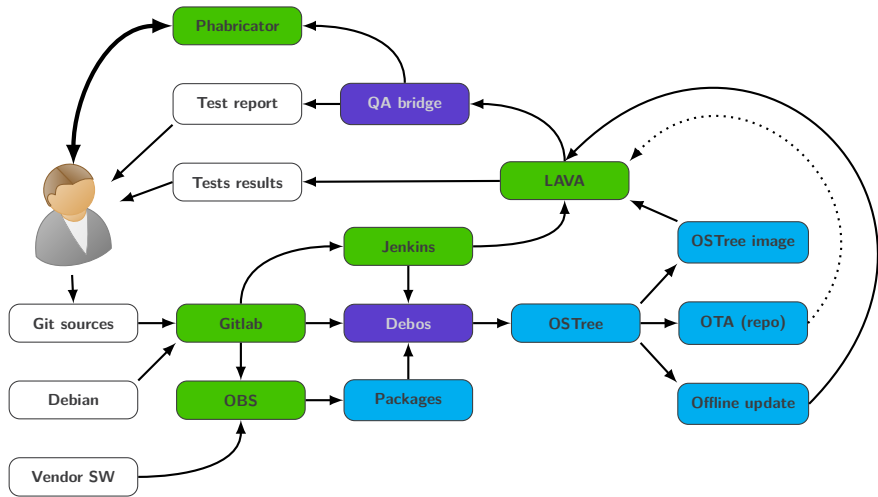
▶ Header

```
# Declare variable 'Var'  
{{ $Var := "Value" }}  
# Header  
architecture: arm64
```

▶ Actions list

```
actions:  
  - action: ActionName1  
    property1: true  
  - action: ActionName2  
    property2: {{$Var}}
```

CI for OSTree



Thanks!

Q & A



We are hiring! ;)

Appendix: Useful links



- ▶ Apertis:
<https://apertis.org>
- ▶ Design documentation:
<https://designs.apertis.org>
- ▶ Images recipes:
<https://gitlab.apertis.org/infrastructure/apertis-image-recipes>
- ▶ Images for download:
<https://images.apertis.org/>
- ▶ Debos:
<https://github.com/go-debos/debos>
- ▶ libostree:
<https://ostree.readthedocs.io>