
COLLABORA
Apertis
Open Source Reference
Linux Distribution
Open First

COLLABORA

Héctor Orón Martínez

@zumbi

Debian Developer
Senior Software Engineer
hector.oron@collabora.com

Open First

COLLABORA

Open Source is ...

- Transparency.
- Reliability.
- Faster time to market.
- Better security.
- Cost effective.

COLLABORA

Open Source is ...

COLLABORA

Debian

- The oldest and largest OSS community
- >24 architectures
- Long Term Support (LTS) extends lifetime at least 5 years
- Solid stable baseline (high robustness)

Apertis

- Collabora supported
- Rebased on Ubuntu 16.04 LTS (“Xenial Xerus”)
- GPLv3 free runtime

COLLABORA

Apertis

Versatile **open source** infrastructure tailored to the automotive needs, fit for a wide variety of **electronic devices**.

- **Security** and **modularity** are two of its primary strengths.
- Provides a **feature-rich framework** for add-on software and **resilient upgrade** capabilities.
- Beyond an operating system, it offers new APIs, tools and cloud services.

COLLABORA

Apertis scope

- Free and open source middleware ...
  - ... for infotainment in vehicles (IVI)
  - ... for internet of things (IoT)
  - ... with long term security support (LTS)
  - ... over the air updates (OTA)

- Apertis could easily be adapted ...
  - ... for cloud services (Cloud)
  - ... for virtual reality (VR, AR, xR)
  - ... for machine learning, artificial intelligence (ML, AI)
  - ... big data analysis (big data)

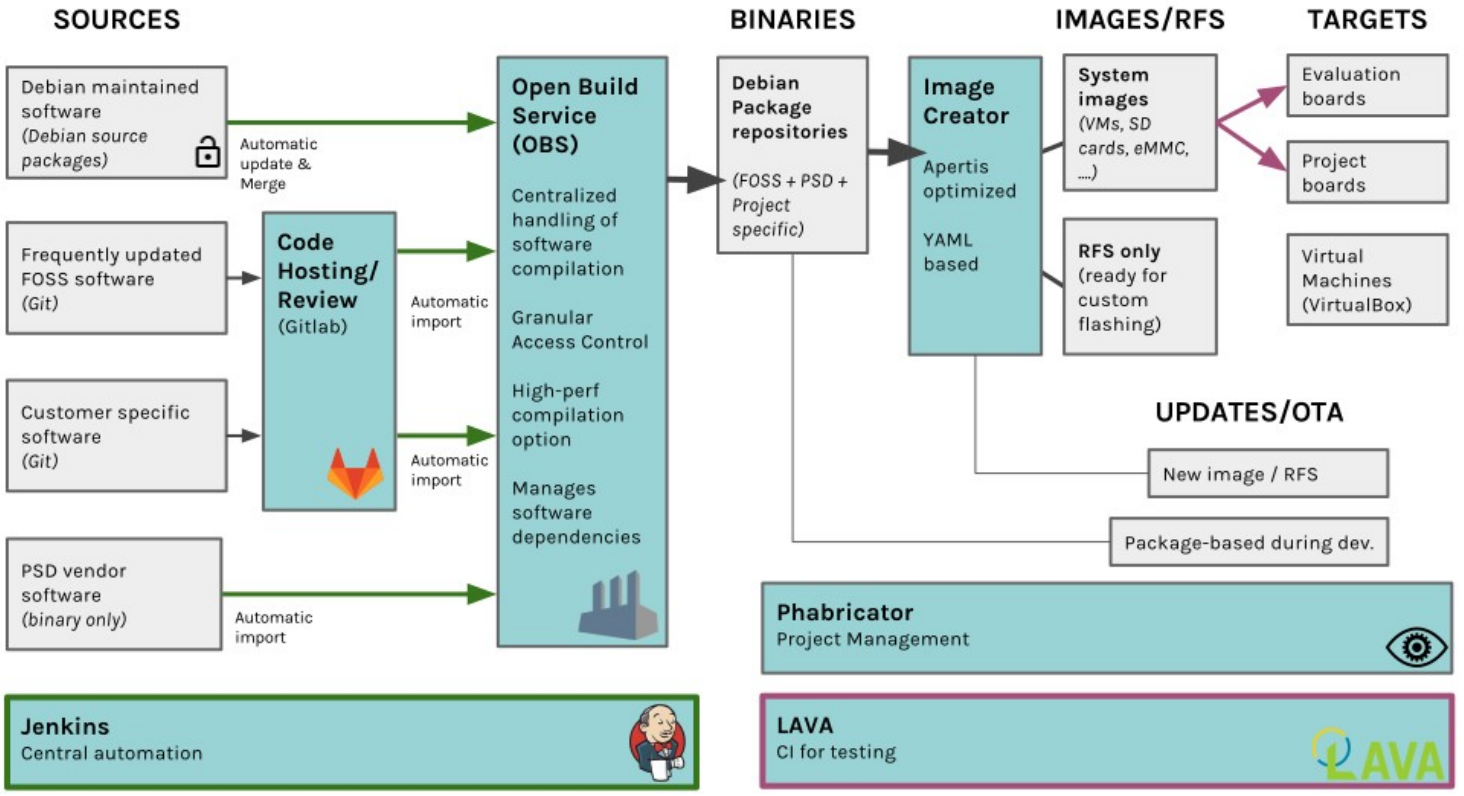
COLLABORA

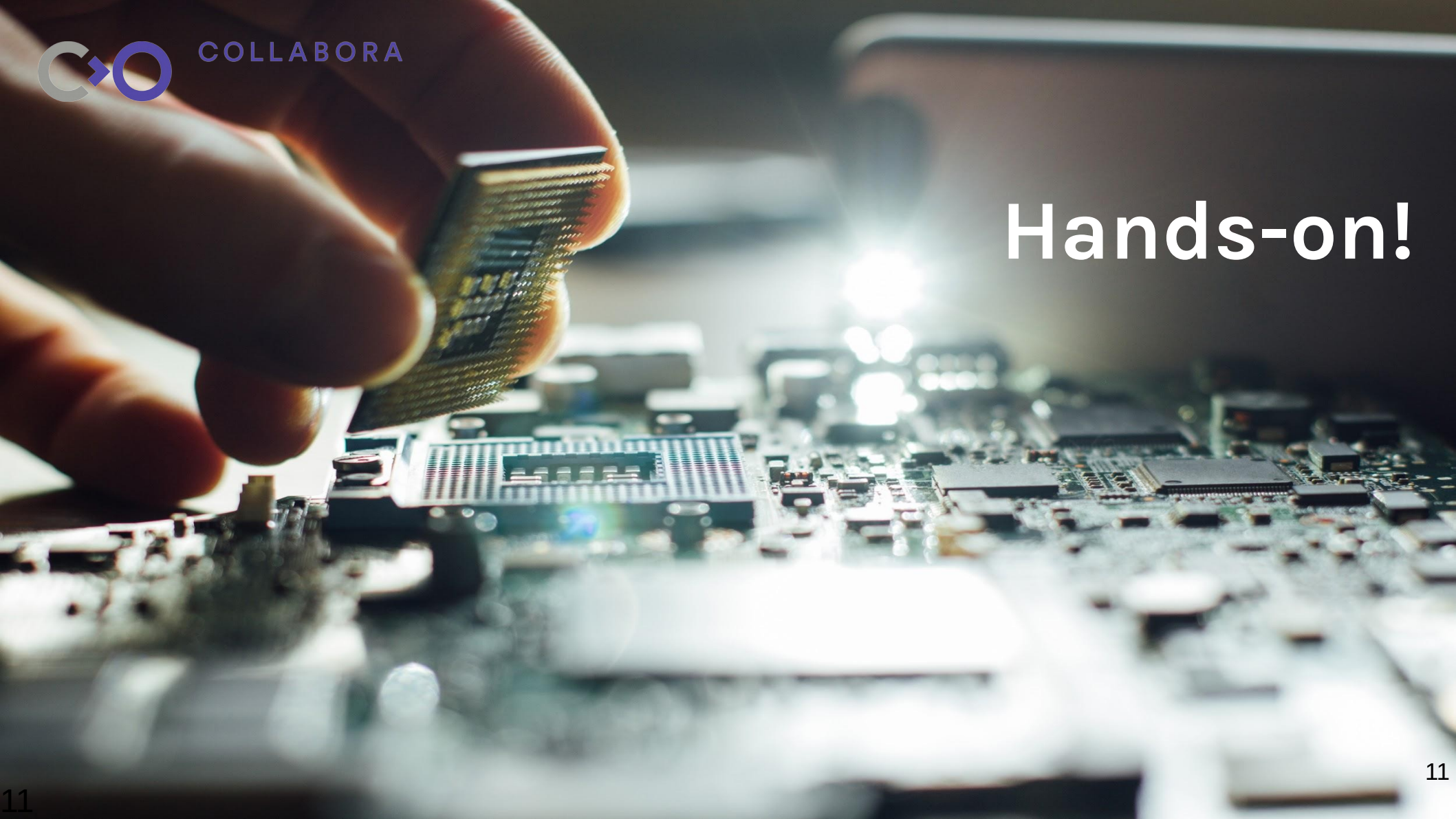
Apertis infrastructure

- Code review system (**Phabricator**, GitLab)
- Application build system (**Jenkins**)
- Package build system – eases integration in OS (**OBS**)
- Image build system (Jenkins, **debos** in **Docker** containers)
- Test system
  - Unit tests (at application build time)
  - System tests (**LAVA**)
- License check system (FOSSology, BlackDuck)

COLLABORA

Apertis infrastructure





11

COLLABORA

Apertis SDK

- Download SDK from Apertis images site:
  - https://images.apertis.org/
  - Look for amd64 > SDK > VDI
- Uncompress the downloaded VM image
- Workshop will be based on the following tutorial:
  - https://people.collabora.com/~zumbi/bosch-linux-days/
  - https://gitlab.apertis.org/zumbi/apertis-bld-workshop

COLLABORA

Apertis workshop materials

- Workshop will be based on the following tutorial:
  - https://people.collabora.com/~zumbi/bosch-linux-days/
- Workshop source materials:
  - https://gitlab.apertis.org/zumbi/apertis-bld-workshop
  - https://gitlab.apertis.org/zumbi/apertis-bld-image-recipes

Apertis SDK

- Add training repository with extra packages (overlay repository)

```
$ echo "deb https://repositories.apertis.org/apertis 18.09 demo" \
  | sudo tee -a /etc/apt/sources.list.d/demo.list
```

Apertis SDK

- Install and configure software

```
$ sudo apt install qtcreator weston qt5-default nano geany
```

Apertis SDK

- SDK image can be easily customized and tuned for your needs following standard procedures to create Apertis images
  - Modifying image recipes

COLLABORA

Apertis SDK

- SDK can be used for:
  - Platform application development
  - Development toolkit (toolchain, debuggers, emulators)
    - GCC, GDB, autotools, cmake, weston, etc.
  - Package creation and upload to main archive
  - Generating images
  - Interfacing to the Apertis infrastructure

COLLABORA

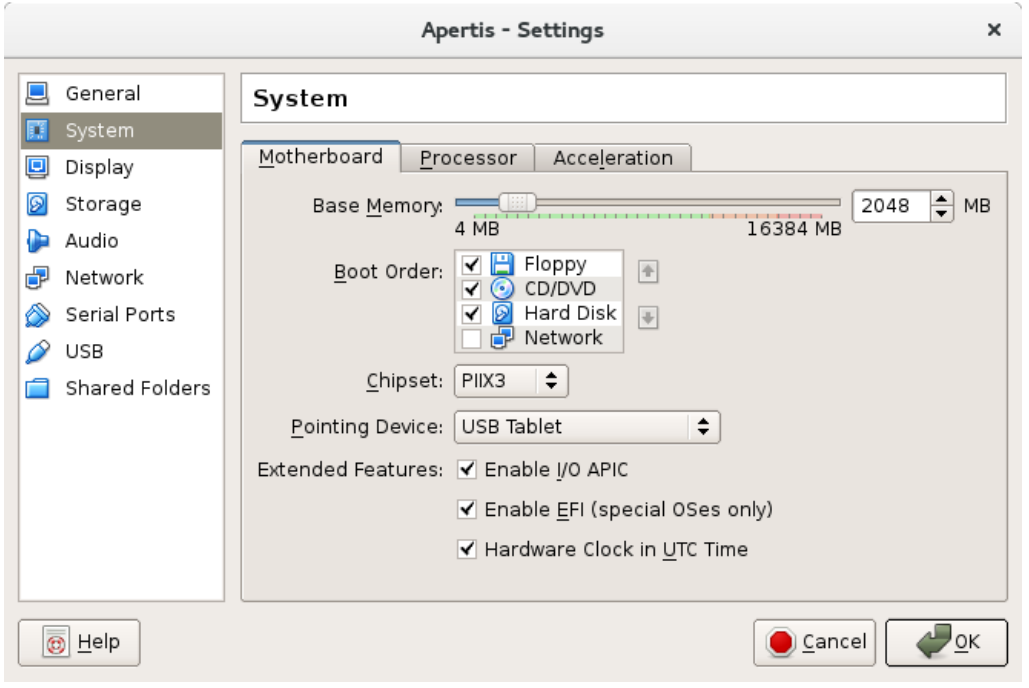
Apertis SDK

- Download and install VirtualBox
- Download online
- Add image to VirtualBox
  - Type: Linux
  - Version: Ubuntu (64-bit)

COLLABORA

Apertis SDK

- Update your VM settings
  - More RAM
  - Enable I/O APIC
  - Enable EFI



Apertis SDK

- Install Guest Additions for the virtual machine
- Devices > Insert Guest Additions CD image…
- In the SDK, open desktop shortcut (VBOXADDITIONS…)
- Right click in file browser and open terminal
- In the terminal, run

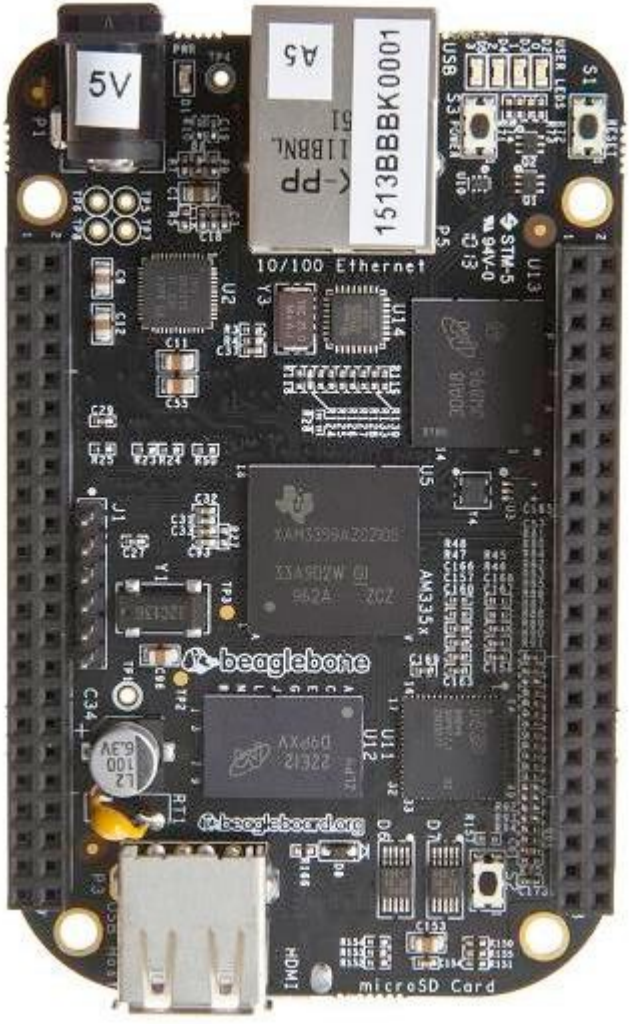
```
$ sudo ./VboxLinuxAdditions.run
```

- Reboot the SDK

COLLABORA

Reference hardware

BeagleBone Black (BBB) - Rev C



COLLABORA

Agenda

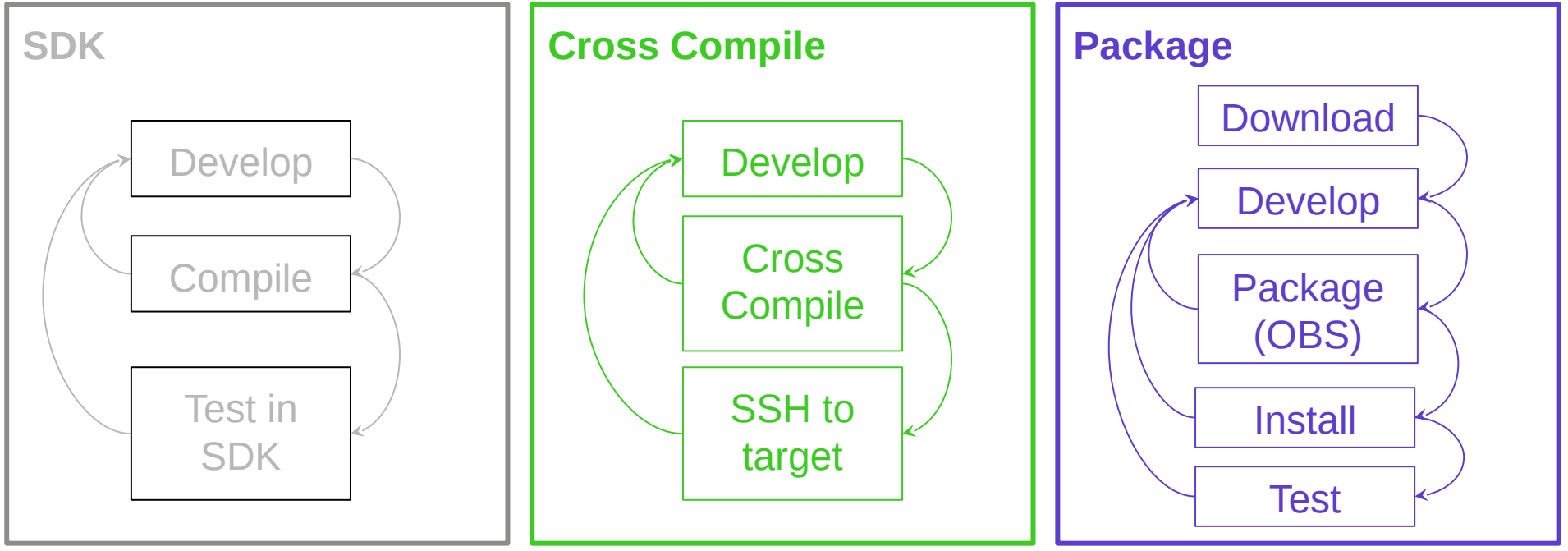
- Apertis application development

Apertis application development

- Application development can be done from SDK
- SDK is built using the same libraries and applications as the target image.
  - Additional packages provided to aid development.
- A number of editors and IDEs available:
  - Eclipse
  - QT Creator
  - Vim
  - Emacs...

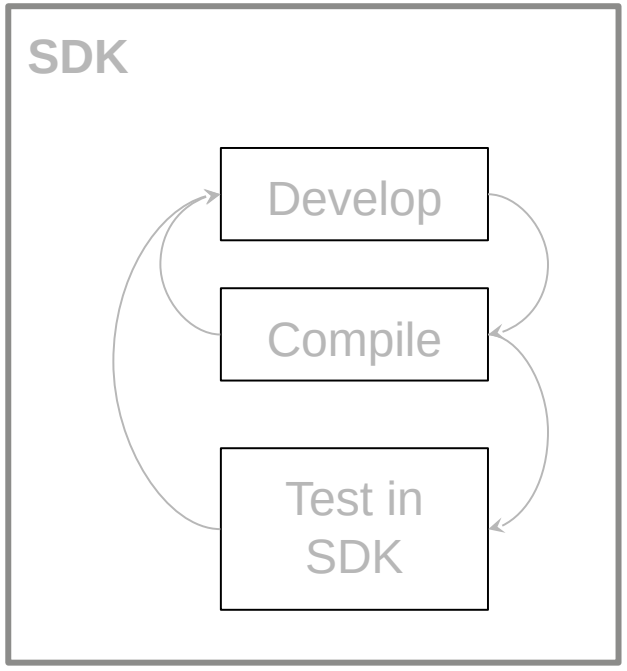
COLLABORA

Apertis application development



COLLABORA

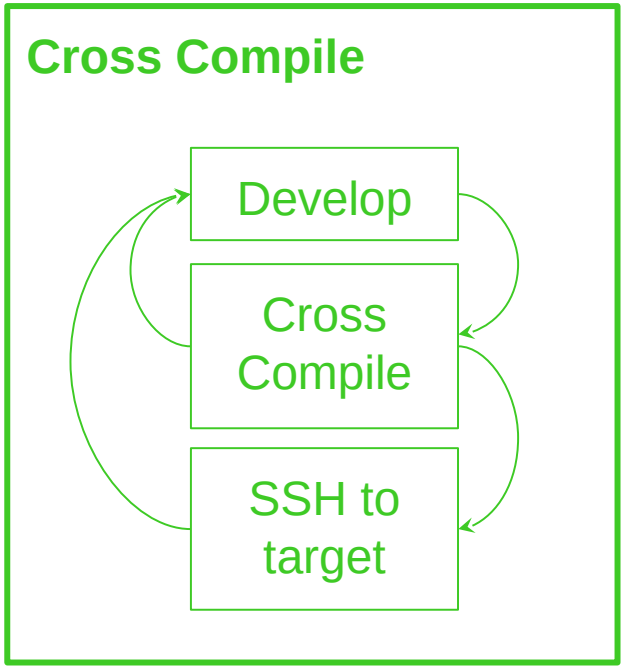
Apertis application development



- Get application source for *debconsolepkg*
- Build application using own tooling
  - CMake
- Debug application
  - Using GDB

COLLABORA

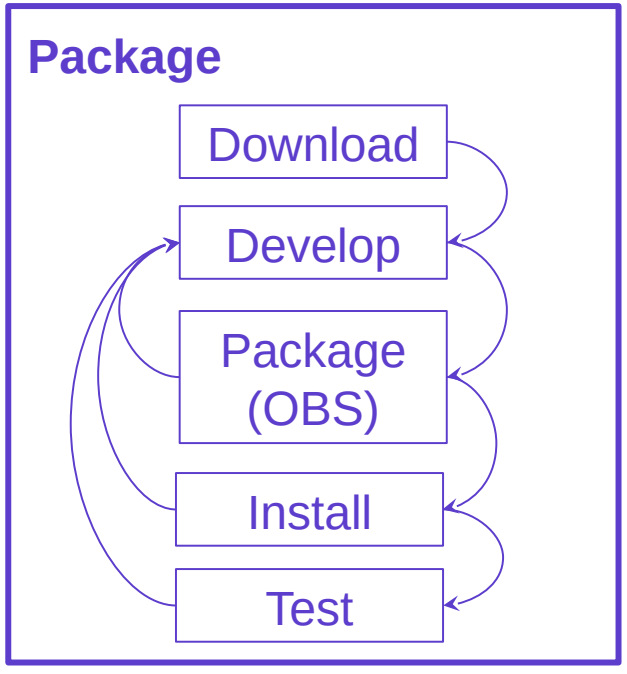
Apertis application development



- Get application source for *debconsolepkg*
- Generate cross building configuration
- Build application using own tooling
  - CMake
- Deploy application to target
  - Using secure copy (scp) command

COLLABORA

Apertis application development

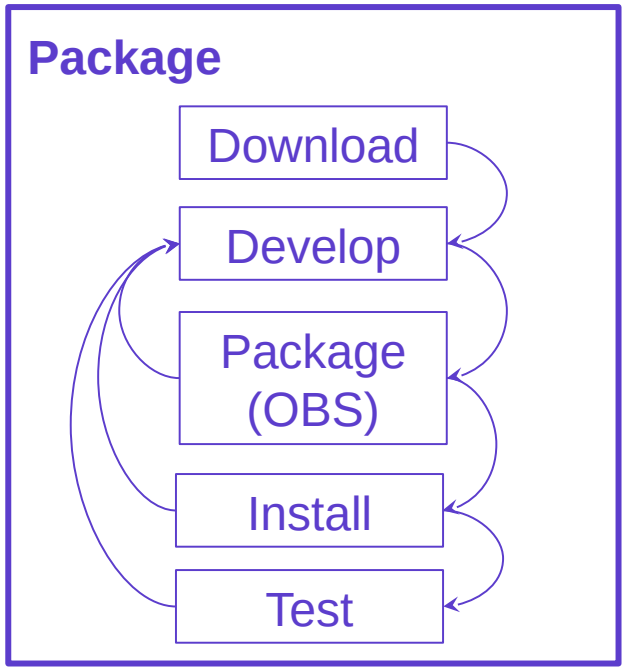


- Build application the Debian way

```
$ apt source debconsolepkg
$ cd debconsole-1.0
$ dpkg-buildpackage -us -uc
```

COLLABORA

Apertis application development



- Upload to OBS using osc
- Copy orig, debian meta and dsc file in application directory

COLLABORA

Agenda

- Introduction to Debian packaging
- Apertis documentation resources

Introduction to Debian packaging

- Packaging
  - structure
  - versioning
  - workflow
- Building packages
- Installation and testing

Introduction to Debian packaging

The three central concepts are:

- upstream tarball
- source package
- binary package

Introduction to Debian packaging

- **upstream tarball**
  - Usually, people package software that someone else, called the **upstream developer**, has written
  - The upstream developers will make a release of their software in the "**upstream source** archive file" or tarball
  - A **tarball** is the *.tar.gz* or *.tgz* file upstream makes (*tarball* is a bit of computer jargon), and it can also be compressed to *.tar.bz2*, *.tb2* or *.tar.xz*. The tarball is exactly what Debian takes and builds a package out

COLLABORA

Introduction to Debian packaging

- **source package:**
  - Add **debian directory** to the unpacked compressed tarball
  - **Essential** to have:
    - *debian/control* - package relationships with other packages, list of binary packages, software description, etc.
    - *debian/rules* - set of rules to configure, build and install software
    - *debian/copyright* - list of licenses used (see DEP5 recommendation)
    - *debian/changelog* - list of changes, it is usually good to explain why changes are being made
  - Review **Debian Policy** for further details

Introduction to Debian packaging

- **binary package**:
  - From source package build the Debian **binary package** which is what actually gets installed:

    ```
    $ dpkg-buildpackage -us -uc -rfakeroot
    ```

  - DEB_BUILD_OPTIONS
    - environment variables can be set to allow different build behaviours, such avoid running tests, setting parallel build jobs, etc.

COLLABORA

Introduction to Debian packaging

- debian/control

```
Source: hello
Section: devel
Priority: optional
Maintainer: Santiago Vila <sanvila@debian.org>
Standards-Version: 3.9.6
Build-Depends: debhelper (>= 9.20120311)
Homepage: http://www.gnu.org/software/hello/

Package: hello
Architecture: any
Depends: ${shlibs:Depends}, ${misc:Depends}
Conflicts: hello-traditional
Replaces: hello-traditional, hello-debhelper (<< 2.9)
Breaks: hello-debhelper (<< 2.9)
Description: example package based on GNU hello
 The GNU hello program produces a familiar, friendly greeting.  It
 allows non-programmers to use a classic computer science tool which
 would otherwise be unavailable to them.
 .
 Seriously, though: this is an example of how to do a Debian package.
 It is the Debian version of the GNU Project's `hello world' program
 (which is itself an example for the GNU Project).
```

COLLABORA

Introduction to Debian packaging

- debian/rules

```
#!/usr/bin/make -f
%:
	dh $@

override_dh_auto_clean:
	[ ! -f Makefile ] || $(MAKE) distclean

override_dh_installdocs:
	dh_installdocs NEWS
```

COLLABORA

Introduction to Debian packaging

- debian/copyright
  - DEP5

```
Upstream-Name: wayland
Upstream-Contact: Kristian Høgsberg <krh@bitplanet.net>
Source: https://wayland.freedesktop.org/releases/

Files: debian/*
Copyright: © 2011 Cyril Brulebois <kibi@debian.org>
License: X11

Files: *
Copyright: © 2008-2012 Kristian Høgsberg
           © 2010-2012 Intel Corporation
           © 2011 Benjamin Franzke
           © 2012 Collabora, Ltd.
           © 2012 Jonas Ådahl
           © 2002 Keith Packard
           © 1999 SuSE, Inc.
License: X11

License: X11
 Permission to use, copy, modify, distribute, and sell this software and its
 documentation for any purpose is hereby granted without fee, provided that
 the above copyright notice appear in all copies and that both that copyright
 notice and this permission notice appear in supporting documentation, and
 that the name of the copyright holders not be used in advertising or
 publicity pertaining to distribution of the software without specific,
 written prior permission.  The copyright holders make no representations
 about the suitability of this software for any purpose.  It is provided "as
 is" without express or implied warranty.
 .
 THE COPYRIGHT HOLDERS DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
 INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO
 EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY SPECIAL, INDIRECT OR
 CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
 DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
 TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE
 OF THIS SOFTWARE.
```

COLLABORA

Introduction to Debian packaging

- debian/changelog

```
hello (2.10-1) unstable; urgency=low

  * New upstream release.
  * debian/patches: Drop 01-fix-i18n-of-default-message, no longer needed.
  * debian/patches: Drop 99-config-guess-config-sub, no longer needed.
  * debian/rules: Drop override_dh_auto_build hack, no longer needed.
  * Standards-Version: 3.9.6 (no changes for this).

 -- Santiago Vila <sanvila@debian.org>  Sun, 22 Mar 2015 11:56:00 +0100

hello (2.9-2) unstable; urgency=low

  * Apply patch from Reuben Thomas to fix i18n of default message.
    This is upstream commit c4aed00. Closes: #767172.
  * The previous change in src/hello.c trigger a rebuild of man/hello.1
    that we don't need. Add a "touch man/hello.1" to avoid it.
  * Use Breaks: hello-debhelper (<< 2.9), not Conflicts,
    as hello-debhelper is deprecated.
  * Restore simple watch file from old hello package that was lost
    when the packages were renamed.
  * Update 99-config-guess-config-sub patch.

 -- Santiago Vila <sanvila@debian.org>  Thu, 06 Nov 2014 12:03:40 +0100

hello (2.9-1) unstable; urgency=low

  * New upstream release. Closes: #744195.
  * Source now contains README-dev. Closes: #621716.
  * Reworded short description.
  * Renamed source and binary from "hello-debhelper" to "hello".
```

COLLABORA

Introduction to Debian packaging

- **Package installation**

```
$ sudo dpkg -i ../foobar_1.0-1_amd64.deb
$ sudo apt install ../foobar_1.0-1_amd64.deb
```

- **Package testing**
  - Run the application
  - List package contents

```
$ dpkg -L foobar
```

COLLABORA

Continuous Integration

Increasing productivity and quality control

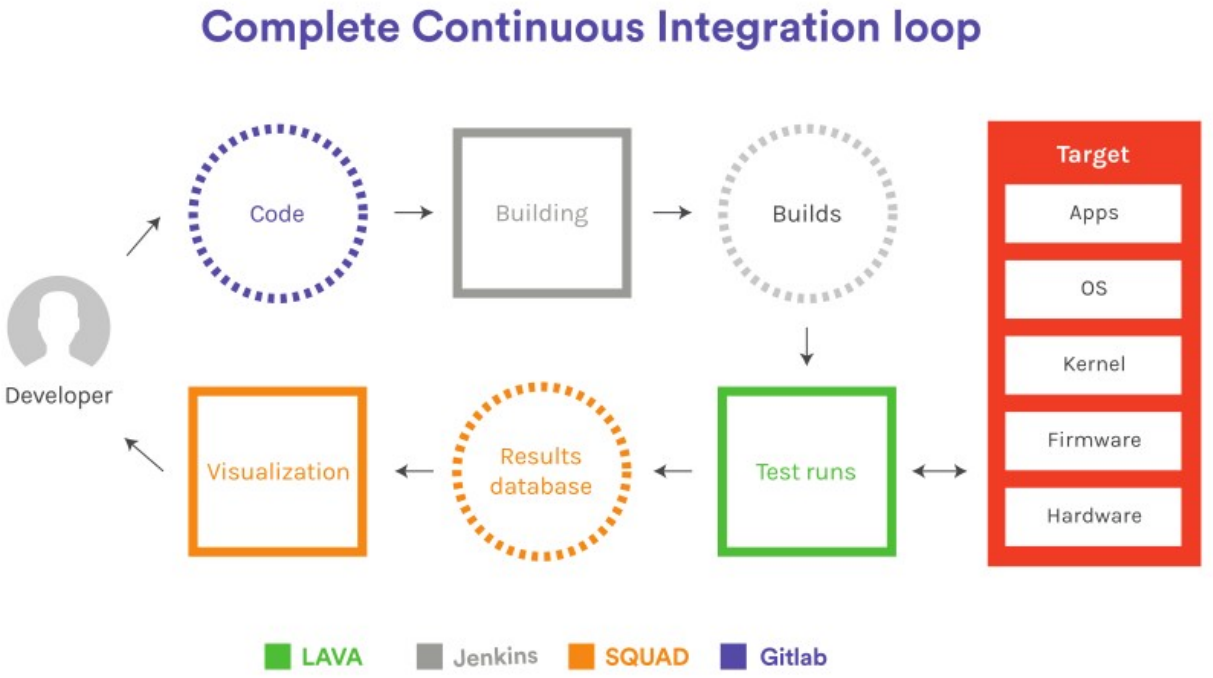
Continuous integration (CI)

is the practice of **merging**

all developer working copies

to a **shared mainline**

several times a day



Continuous Integration – best practices

- **Maintain** a code repository & **automate** the build
- Make the build **self-testing** & keep it **fast**
- Everyone **commits** to the baseline **every day** & every commit should be **built**
- **Test** in a clone of the **production environment**
- Make it easy to get the latest **deliverables**
- **Results** of the latest build visible to everyone
- Automate **deployment**

COLLABORA

Apertis documentation resources

- **Apertis** provides documentation in:
  - devhelp: offline documentation in SDK
  - Apertis documentation portal - designs.apertis.org
    - Apertis work-in-progress and obsolete documentation portal
  - Apertis wiki - wiki.apertis.org

COLLABORA

Debian documentation resources

- **Debian** documentation and useful sites
  - https://www.debian.org/doc/devel-manuals
  - http://snapshot.debian.org/
  - https://tracker.debian.org/
  - https://sources.debian.org/

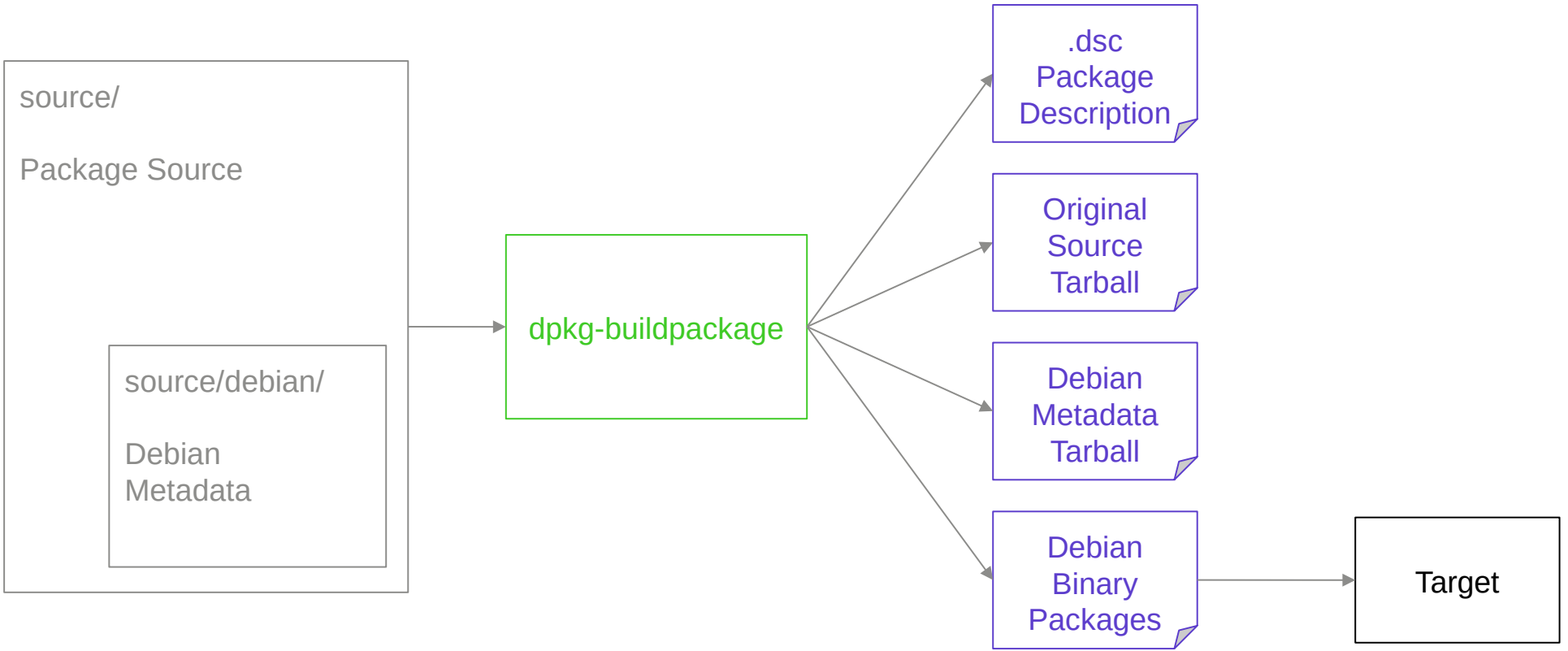
COLLABORA

Agenda

- Apertis application integration
- Modifying existing packages

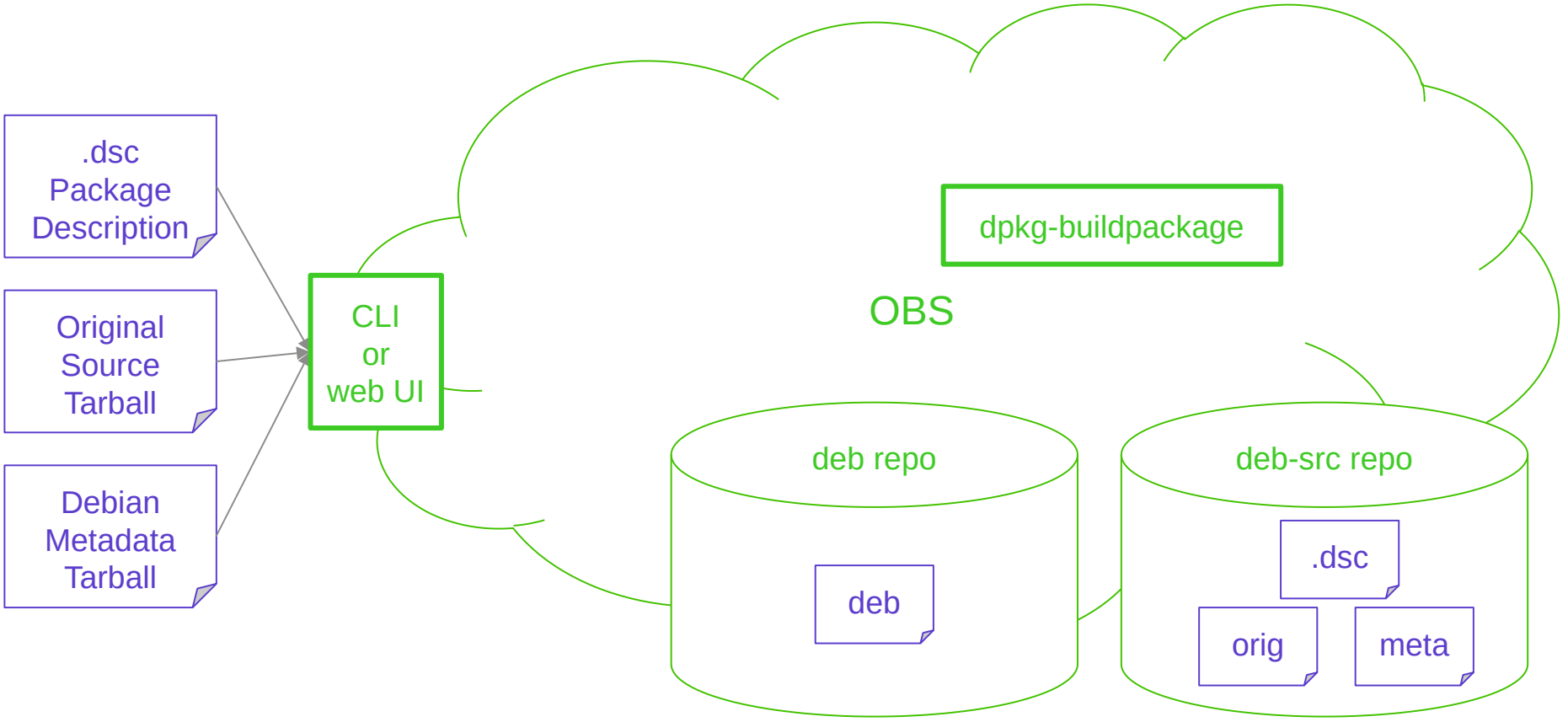
COLLABORA

Apertis application integration



COLLABORA

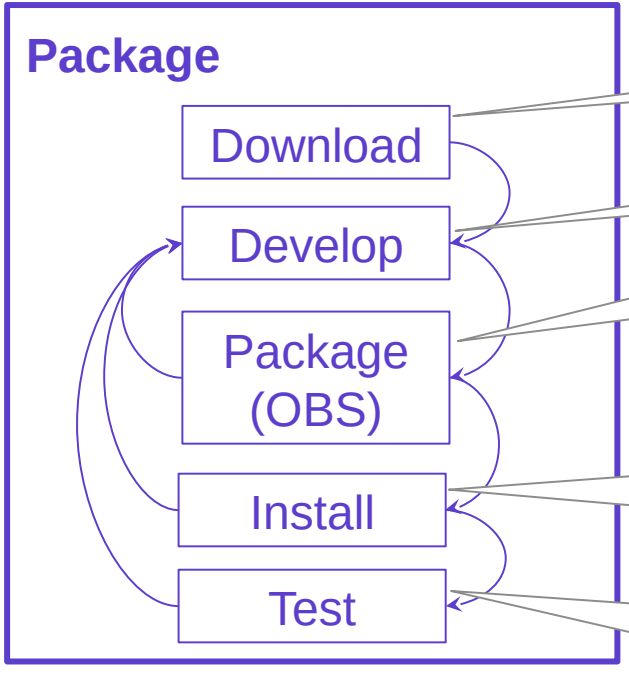
Apertis application integration



COLLABORA

Modifying existing packages

Cover:



- Existing source packaging can be pulled from OBS
- In house projects and frequently updated packages will be in git

- Use SDK

- Upload new source packaging to OBS
- OBS will build the official binary package.

- “deb” package pulled from repo and installed
- apt update / apt install from target
- Baked into new version of the image

- Automated testing via LAVA
- Manual testing on target

COLLABORA

Agenda

- Apertis images
  - Apertis reference images
  - Apertis adding features (ospack)
  - Apertis cross build support (sysroot)
  - Apertis hardware enablement (hwpack)

Apertis reference images

- Images are built from:
  - **ospack:** hardware, but not architecture, independent
  - **hwpack:** hardware dependent software
- Images are built using debos and the process is controlled via a debos yaml recipe

Adding features (ospack)

- ospack is effectively a root filesystem that contains unpacked packages
- ospacks are generated using debos
  - Debos controlled via yaml recipes
  - Modifications like adding new repositories, running certain commands or installing extra packages can be done describing in the yaml file
- Automating ospack builds can be done using Jenkins service

COLLABORA

Cross build support (sysroot)

- For cross building purposes, a directory containing libraries and headers for target system must be populated
- A pre-built sysroot is provided, but is likely to need to be extended as development progresses to include extra headers
- Extended sysroots can be assembled in the SDK using debos
- Extensions should be added to jenkins so that the sysroots can be automatically rebuilt as components are updated

Hardware enablement (hwpack)

- A hwpack provides hardware-specific packages and settings that can be combined with ospacks
  - Multiple ospacks can be combined with an hwpack to implement different feature sets on the same hardware
  - An ospack can be combined with multiple hwpacks to implement the same feature set on different hardware
- Typically such components are:
  - Bootloader
  - Linux kernel
  - Graphic stack

COLLABORA

Agenda

- Apertis images
  - Apertis image creation
  - Customizing images during development

COLLABORA

Apertis image creation

- Create partitions and filesystems
- Deploy ospack on the image
- Combine the deployed ospack with the hwpack packages
- Prepare the bootloader
- List empty blocks for more efficient flash writes
- Compress the image

COLLABORA

Partitions and filesystems

```
- action: image-partition
  imagename: {{ $image }}
  imagesize: 3G
  partitiontype: gpt
  mountpoints:
    - mountpoint: /
      partition: root
      flags: [ boot ]
  partitions:
    - name: root
     fs: ext4
     start: 8MB
     end: 100%
```

Deploying the ospack on the image

```
- action: unpack
  compression: gz
  file: {{ $ospack }}

[...]

- action: filesystem-deploy
  description: Deploying filesystem onto image
```

Combining the hwpack

```
- action: apt
  description: Install hardware support packages
  packages:
  - linux-image-armmp
  - u-boot-omap
```

Preparing the bootloader

```
- action: raw
  description: Flash MLO
  offset: {{ sector 256 }}
  origin: filesystem
  source: /usr/lib/u-boot/am335x_boneblack/MLO
- action: raw
  description: Flash u-boot
  origin: filesystem
  offset: {{ sector 768 }}
  source: /usr/lib/u-boot/am335x_boneblack/u-boot.img
```

COLLABORA

Listing empty blocks

```
- action: run
  description: Create bmap file
  postprocess: true
  command: bmaptool create {{ $image }} > {{ $image }}.bmap
```

Compressing the image

```
- action: run
  description: Compress image
  postprocess: true
  command: gzip -f {{ $image }}
```

COLLABORA

Customizing images during development

- Create new overlay for packages:

  ```
  $ mkdir -p overlay/packages
  ```

- Copy linux and u-boot binary packages into overlay

  ```
  $ cp ~/linux-image-*_armhf.deb \ ~/u-boot-omap_*_armhf.deb
  overlay/packages
  ```

- Copy overlay into build environment

  ```
  - action: overlay
    source: overlays/packages
  ```

COLLABORA

Customizing images during development

- Install packages into image

```
- action: run
  chroot: true
  description: Install hardware support packages
  command: apt -y install ./*deb
```

- Remove packages from image, after installed

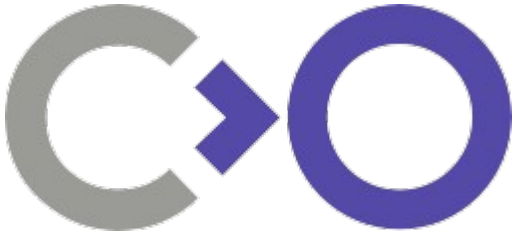
```
- action: run
  chroot: true
  description: Remove hardware support packages
  command: rm -f ./*deb
```

COLLABORA

Apertis image creation

- Create partitions and filesystems
- Deploy ospack on the image
- Combine the deployed ospack with the hwpack packages
- Prepare the bootloader
- List empty blocks for more efficient flash writes
- Compress the image

```
$ sudo debos apertis-image.yaml
```



Vielen Dank!

(Thank you!)

Héctor Orón Martínez <hector.oron@collabora.com>

www.collabora.com