Main steps to build an embedded Linux system

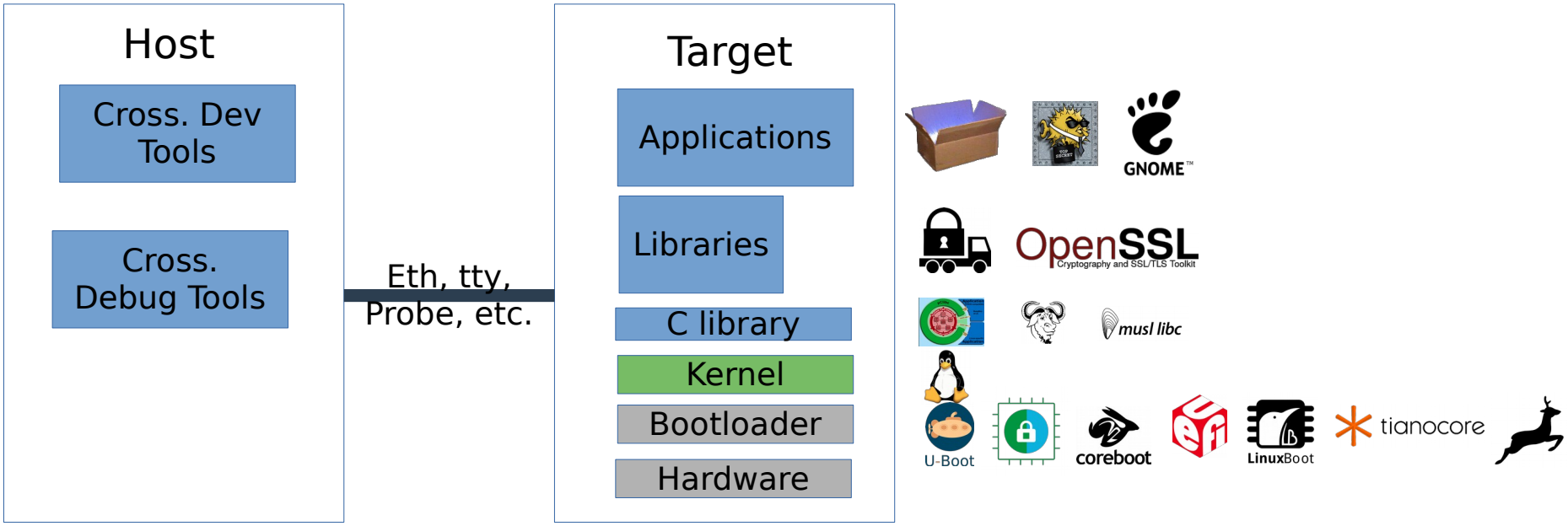
Capitole du libre 2019

thomas.perrot@tupi.fr

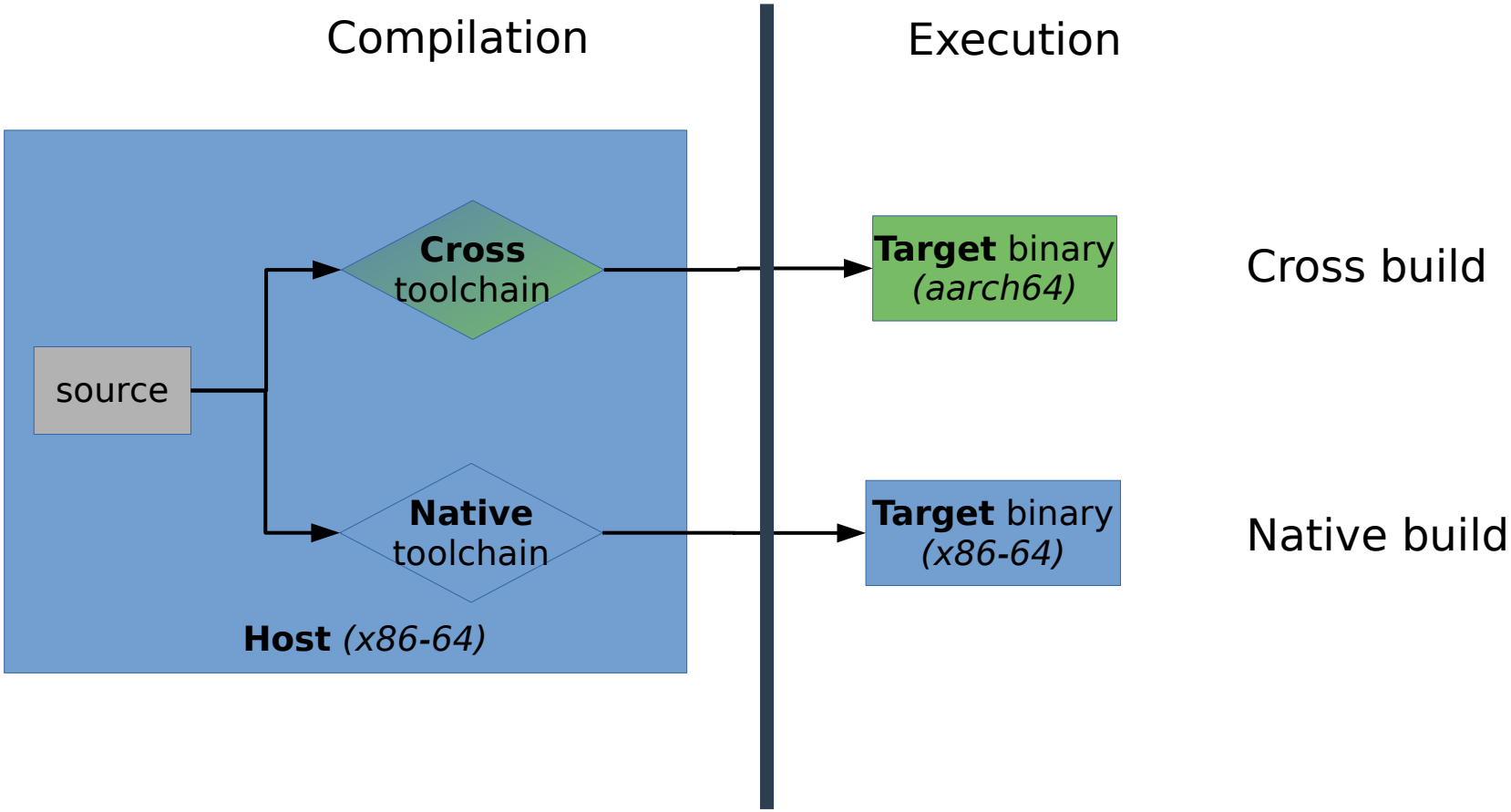
Why do that ?

- Functional needs
- Target limitation
- The challenge ;-)
- Etc.

Generic Linux system architecture



The cross-compilation



Main steps

1) Cross-compilation toolchain
2) Bootloader
3) Linux kernel
4) The root filesystem
5) The bootable image

Cross-compilation toolchain

- From scratch: very painful
- Pre-built toolchain
  - Host
  - Linaro
  - Etc.
- Meta-toolchain
  - Crosstool-NG
  - Yocto
  - Etc.

sudo dnf install glibc-arm-linux-gnu-devel ...

Linaro

gcc-linaro-7.2.1-2017.11-x86_64_aarch64-linux-gnu.tar.xz
gcc-linaro-7.2.1-2017.11-x86_64_aarch64-linux-gnu.tar.xz.asc

https://downloads.yoctoproject.org/releases/yocto

yocto PROJECT

poky-glibc-x86_64-core-image-minimal-cortexa8hf..>	09-Oct-2019 23:52	2146
poky-glibc-x86_64-core-image-minimal-cortexa8hf..>	09-Oct-2019 23:52	203
poky-glibc-x86_64-core-image-minimal-cortexa8hf..>	09-Oct-2019 23:52	391493863

bitbake meta-toolchain

crosstool-NG ct-ng build-all

Cross-compilation toolchain

- **Components**
  - Binutils
  - C libraries
  - C compiler
  - Debugger
  - Kernel headers
- **Flags / Options**
  - ABI
  - Offload
  - Optimization

Bootloader

- **Boot sequence**
  - ROM code:
    - Loads from SRAM
    - load the first stage from a storage or a device
  - First stage:
    - Small
    - Loads from SRAM
    - Init DDR and required devices (AP, PM…)
    - Load the second stage
  - Second stage:
    - Loads from RAM
    - Init some devices (watchdog, ethernet, usb, eMMC…)
    - Load the Linux kernel
  - **Binaries format**
  - **Initiatives to converge**

Bootloader

- **Vendor BSPs**
  - ARM Trusted Firmware
  - Barebox
  - Coreboot
  - Linuxboot
  - Grub
  - Tianocore
  - U-boot
  - Etc…
- **Customization**

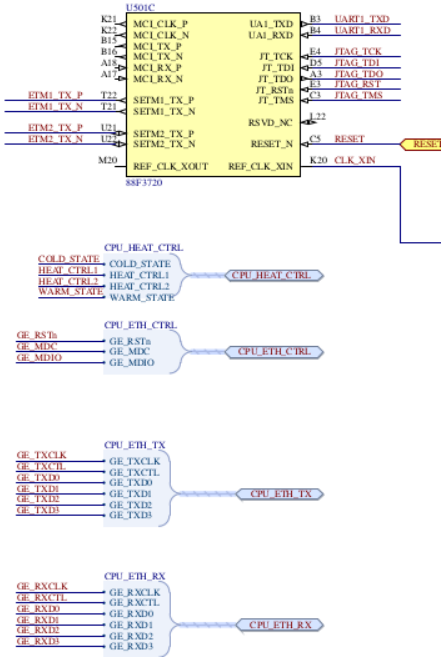
Linux Kernel

- **Vendor BSP**
- **Customization**
  - Defconfig
  - Device tree
  - Cmdline

console=ttyS0,115200 root=/dev/mapper/root ro

```
gpio1: gpio1 {
        gpio-controller;
        #gpio-cells = <2>;
};
[...]

data-gpios = <&gpio1 12 0>,
             <&gpio1 13 0>,
             <&gpio1 14 0>,
             <&gpio1 15 0>;
```

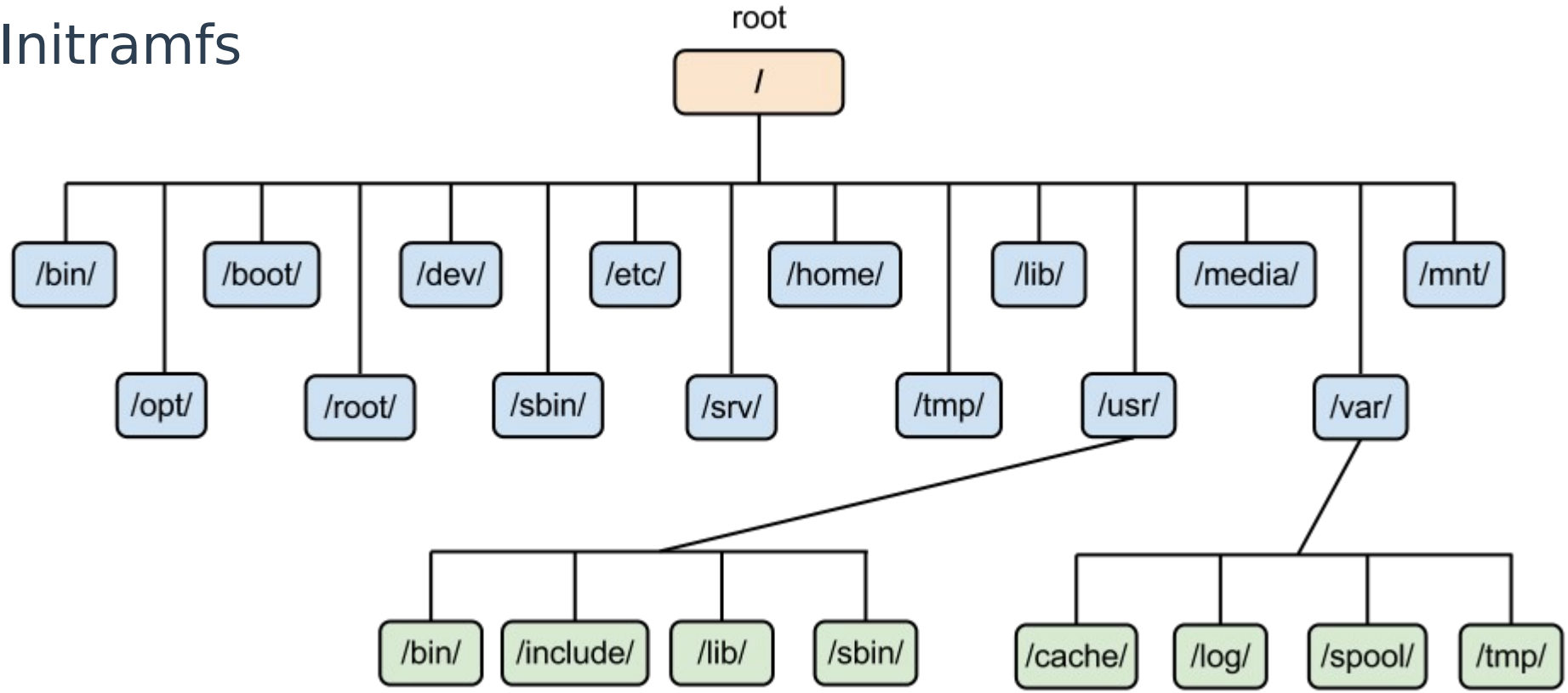
```
CONFIG_NUMA_BALANCING=y
CONFIG_MEMCG=y
CONFIG_MEMCG_SWAP=y
CONFIG_BLK_CGROUP=y
CONFIG_CGROUP_PIDS=y
CONFIG_CGROUP_HUGETLB=y
CONFIG_CPUSETS=y
CONFIG_CGROUP_DEVICE=y
CONFIG_CGROUP_CPUACCT=y
CONFIG_CGROUP_PERF=y
CONFIG_USER_NS=y
CONFIG_SCHED_AUTOGROUP=y
CONFIG_BLK_DEV_INITRD=y
CONFIG_KALLSYMS_ALL=y
# CONFIG_COMPAT_BRK is not set
CONFIG_PROFILING=y
CONFIG_ARCH_AGILEX=y
```




```
pwm: pwm {
        #pwm-cells = <2>;
};

[...]

bl: backlight {
        pwms = <&pwm 0 5000000>;
        pwm-names = "backlight";
};
```

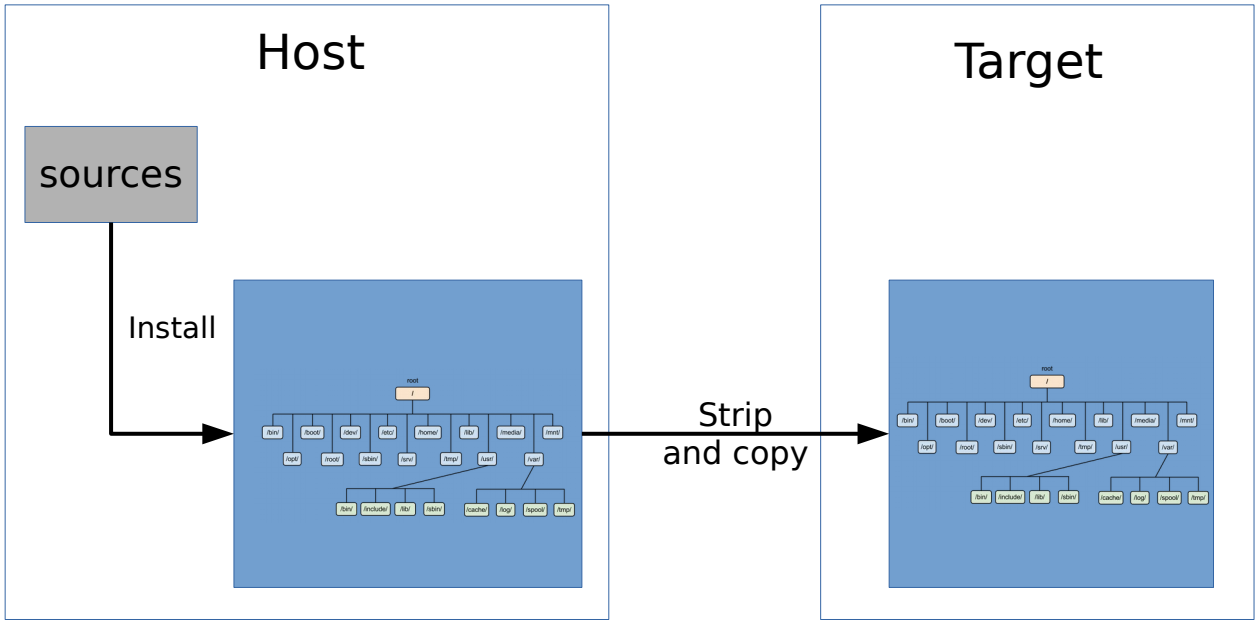
The rootfs filesystem

- Organize data
- Local or remote
- Initramfs

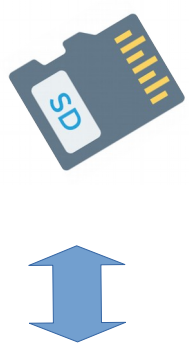


The rootfs filesystem

- For each components
  - Downloading
  - Configuring
  - Patching
  - Compiling
  - Installing
  - Stripping
- In the right order
- With the right rights
- Boring :-(



The bootable image

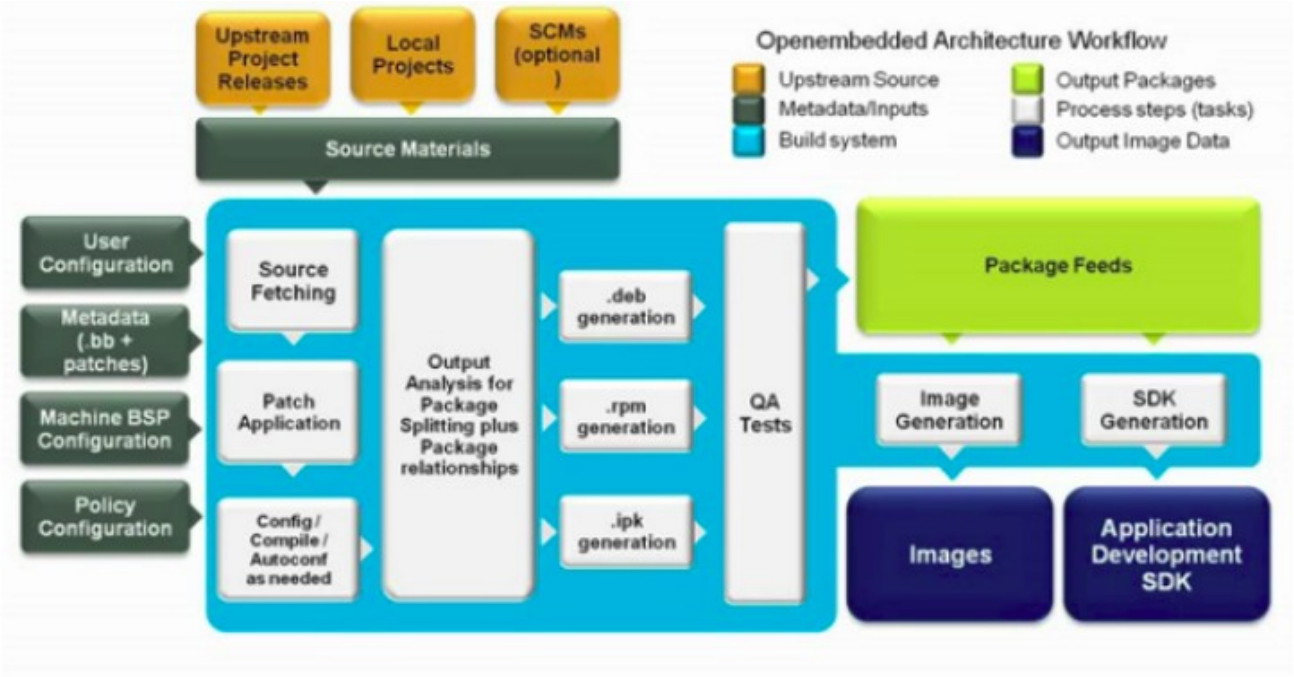


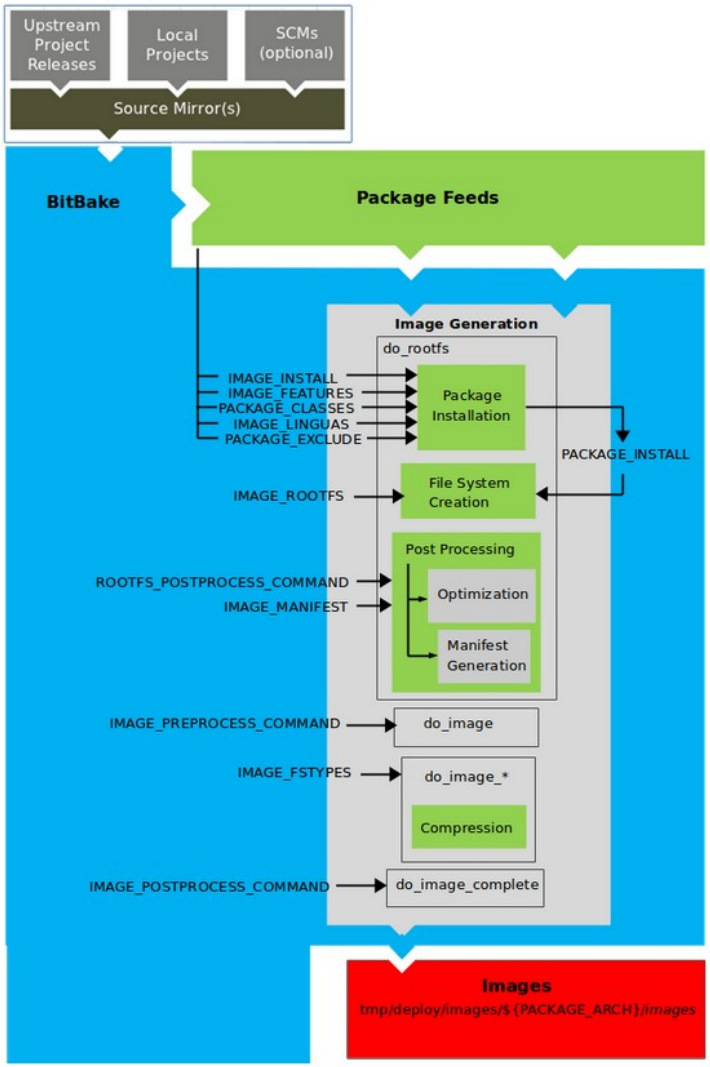

```
# The disk layout used is:
#  - --------- --------------
# | | u-boot  |    rootfs    |
#  - --------- --------------
# ^ ^         ^              ^
# | |         |              |
# 0 1kiB    4MiB + rootfs + IMAGE_EXTRA_SPACE (default 10MiB)
#
part u-boot --source rawcopy --sourceparams="file=u-boot.imx" --ondisk mmcblk --no-table --align 1
part / --source rootfs --ondisk mmcblk --fstype=ext4 --label root --align 4096

bootloader --ptable msdos
```

System building

- Yocto & OpenEmbedded
- Buildroot 
- Ptxdist
- Etc.

Yocto / OE


Upstream Project Releases
Local Projects
SCMs (optional)
Source Materials
Openembedded Architecture Workflow
Upstream Source
Metadata/Inputs
Build system
Output Packages
Process steps (tasks)
Output Image Data
User Configuration
Metadata (.bb + patches)
Machine BSP Configuration
Policy Configuration
Source Fetching
Patch Application
Config / Compile / Autoconf as needed
Output Analysis for Package Splitting plus Package relationships
.deb generation
.rpm generation
.ipk generation
QA Tests
Package Feeds
Image Generation
SDK Generation
Images
Application Development SDK


Upstream Project Releases
Local Projects
SCMs (optional)
Source Mirror(s)
BitBake
Package Feeds
Image Generation
do_rootfs
IMAGE_INSTALL
IMAGE_FEATURES
PACKAGE_CLASSES
IMAGE_LINGUAS
PACKAGE_EXCLUDE
Package Installation
PACKAGE_INSTALL
IMAGE_ROOTFS
File System Creation
Post Processing
ROOTFS_POSTPROCESS_COMMAND
IMAGE_MANIFEST
Optimization
Manifest Generation
IMAGE_PREPROCESS_COMMAND
do_image
IMAGE_FSTYPES
do_image_*
Compression
IMAGE_POSTPROCESS_COMMAND
do_image_complete
Images
tmp/deploy/images/${PACKAGE_ARCH}/images

Any questions ?