



ubiquitous crunch

Yocto Project getting started

SUMMER SCHOOL

Outline

What is Yocto Project

Key concepts

Layers and BSP

Recipes

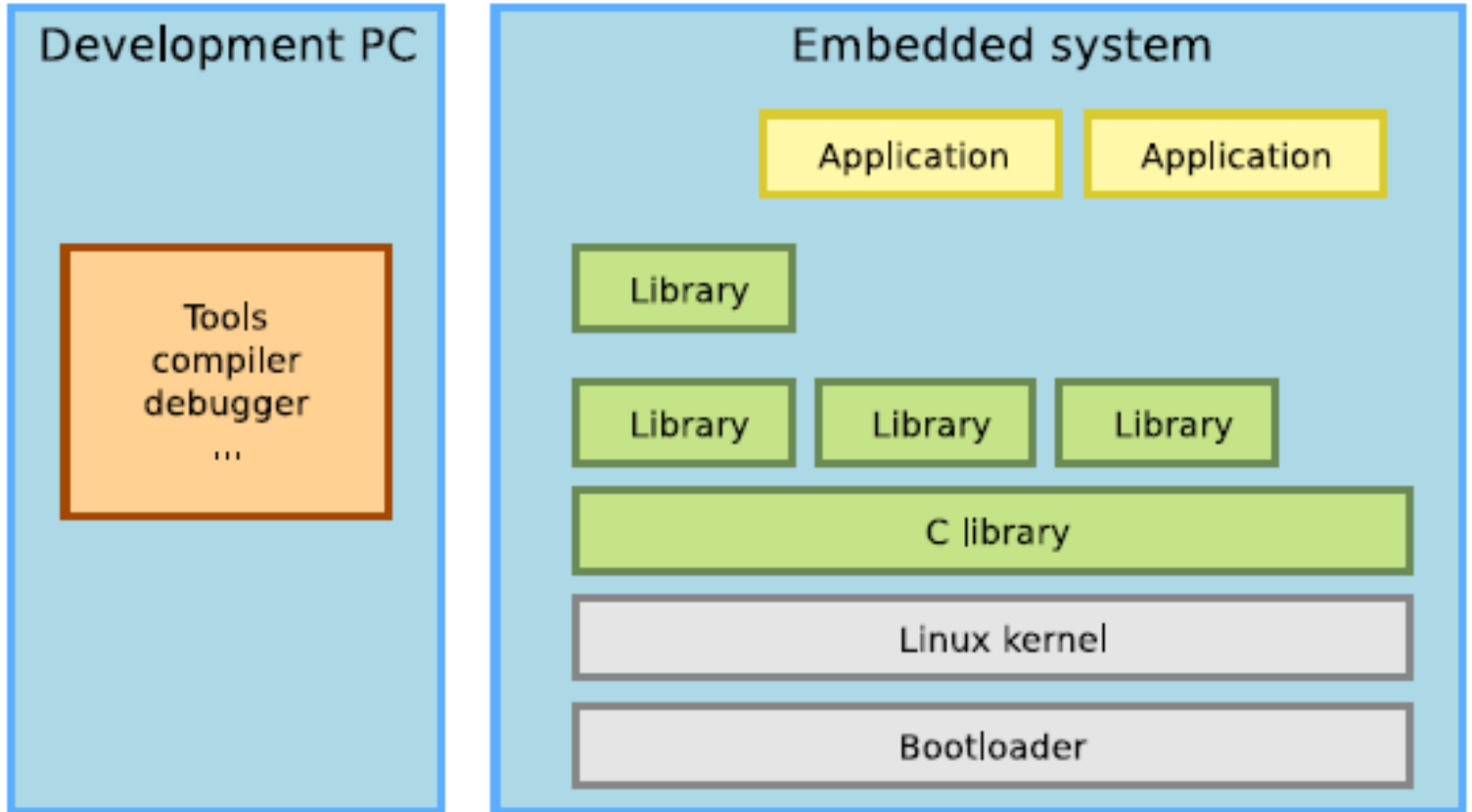
Application development

Hob GUI

Devshell

Questions

Linux Architecture



General philosophy

Support for the major architectures

- Add support for a few machines besides qemu ones

Only provides core recipes, uses layers to get support for more packages and machines

Custom modification stay in separate layer

Build system as flexible as possible

Open community governed by the Yocto Project Advisory Board

Yocto Project output

Builds distribution, the main output being a package feed

- The package management system is optional on the target
- It is possible to install and update only part of the system

Generates not only a root filesystem image but a full OS distribution. Supports many more filesystems extensions

With the help of images classes or wic it can create disk images

Able to generate and SDK for further application development and other purposes

What more to know about this?

Support for a variety of architectures: ARM, PPC, MIPS and x86 for both 32 & 64 bits ones

Huge open source community and really powerful players behind it

Support for binary packages: deb/rpm/ipk

Releases done at every 6 months in April and October/November

The starting place for most BSP (board support package) providers

Resources

Yocto Project tries to establish itself as a standard for the Linux community and the industry in the same time:

Mailing lists:

- openembedded-core@lists.openembedded.org
- poky@yoctoproject.org
- yocto@yoctoproject.org

Source code repositories:

- <http://git.yoctoproject.org>
- <http://git.openembedded.net>

Documentation:

- <http://www.yoctoproject.org/documentation>

Bug reports and feature requests:

- <http://bugzilla.yoctoproject.org>

Yocto Project Administration



WIND RIVER



ENEAA software

JUNIPER NETWORKS

Mentor Graphics

montavista



openembedded



Yocto Project presentation @ OSCON 2012
© 2012 The Linux Foundation. All rights reserved.
All logos are trademarks of their respective owners.

Yocto Project support

Silicon Vendors



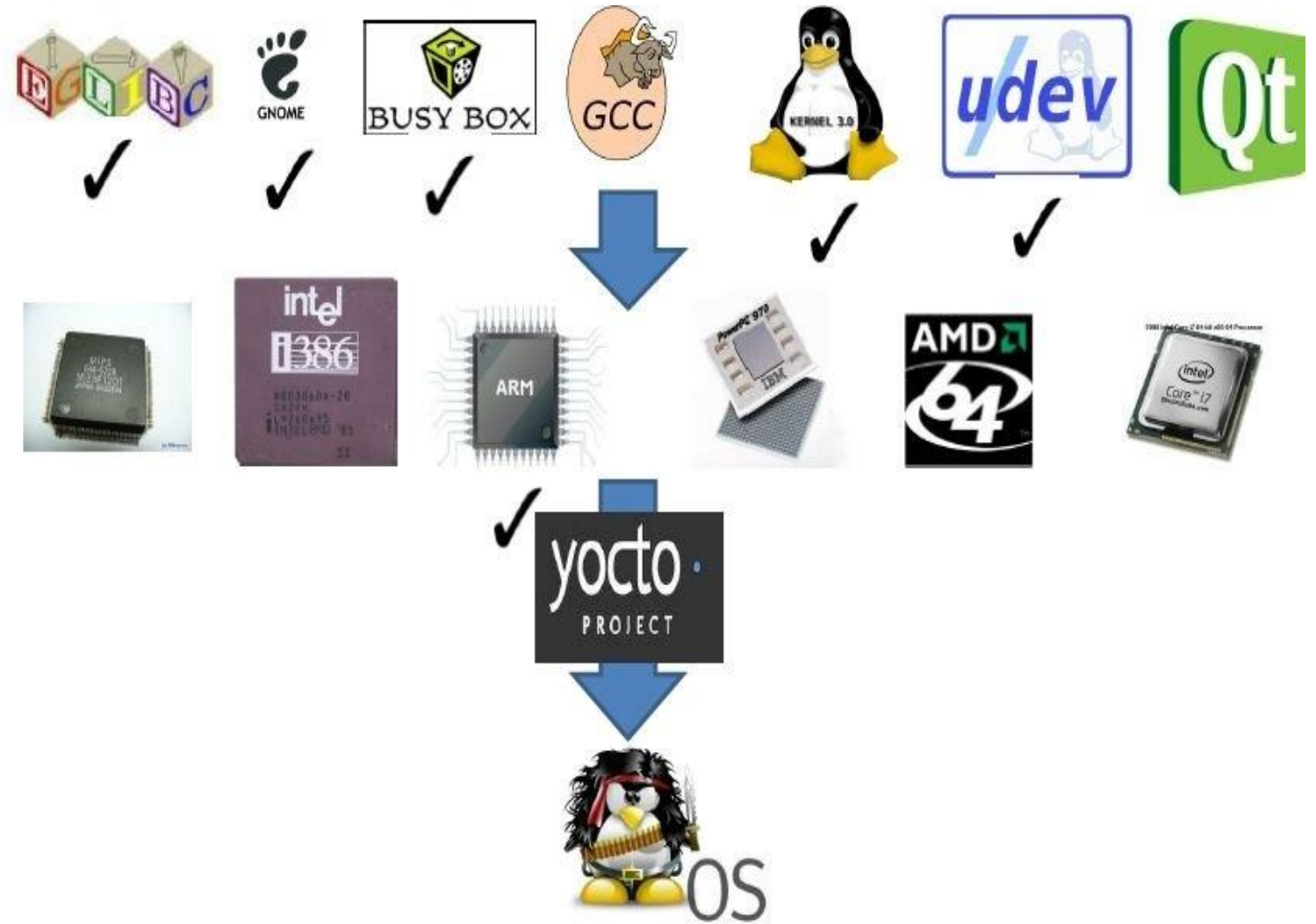
OSVs



Embedded Tools, Consulting Services, Users...



Yocto Project development



Keywords

Hob

Classes

Build directory

Configuration file

Image

Append files

Cross development toolchain

Upstream

Metadata

Poky

Bitbake

Tasks

Layer

Recipe

Oe-core

Package

Source directory

Poky

The reference distribution for the Yocto Project

Includes the Bitbake build system and OpenEmbedded core metadata

Used to bootstrap your own custom Linux distribution

After downloading a number of recipes and configuration files are visible for Bitbake to process

Similar to OpenEmbedded & Buildroot

Recipes of interest

u-boot – recipes-bsp: instructions necessary to build the u-boot package

Linux kernel – recipes-kernel: instructions necessary to build the Linux kernel package

gcc, binutils, glibc – recipes-devtools, recipes-core: instructions necessary to build the toolchain specific packages

core-image-minimal: reference build image

Available configuration

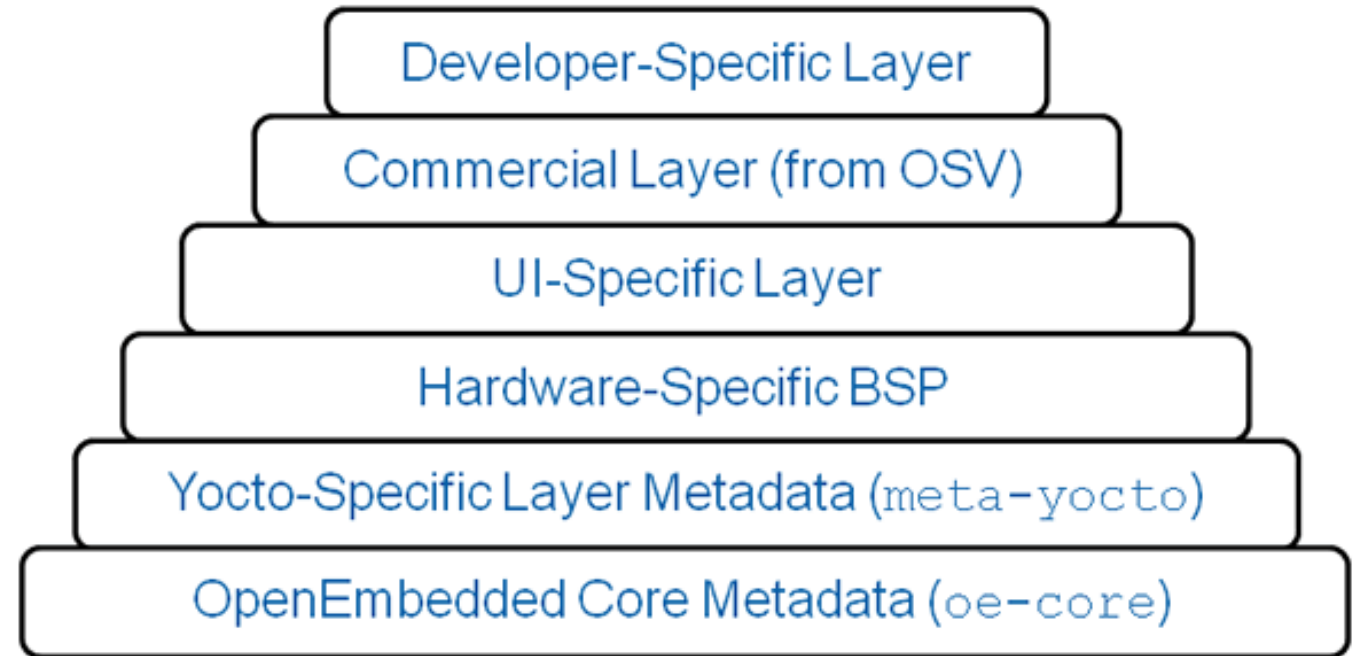
Machine: target specific configuration

Distro: distribution specific configuration

Layer.conf: layer specific configuration

Local.conf: build directory specific configuration

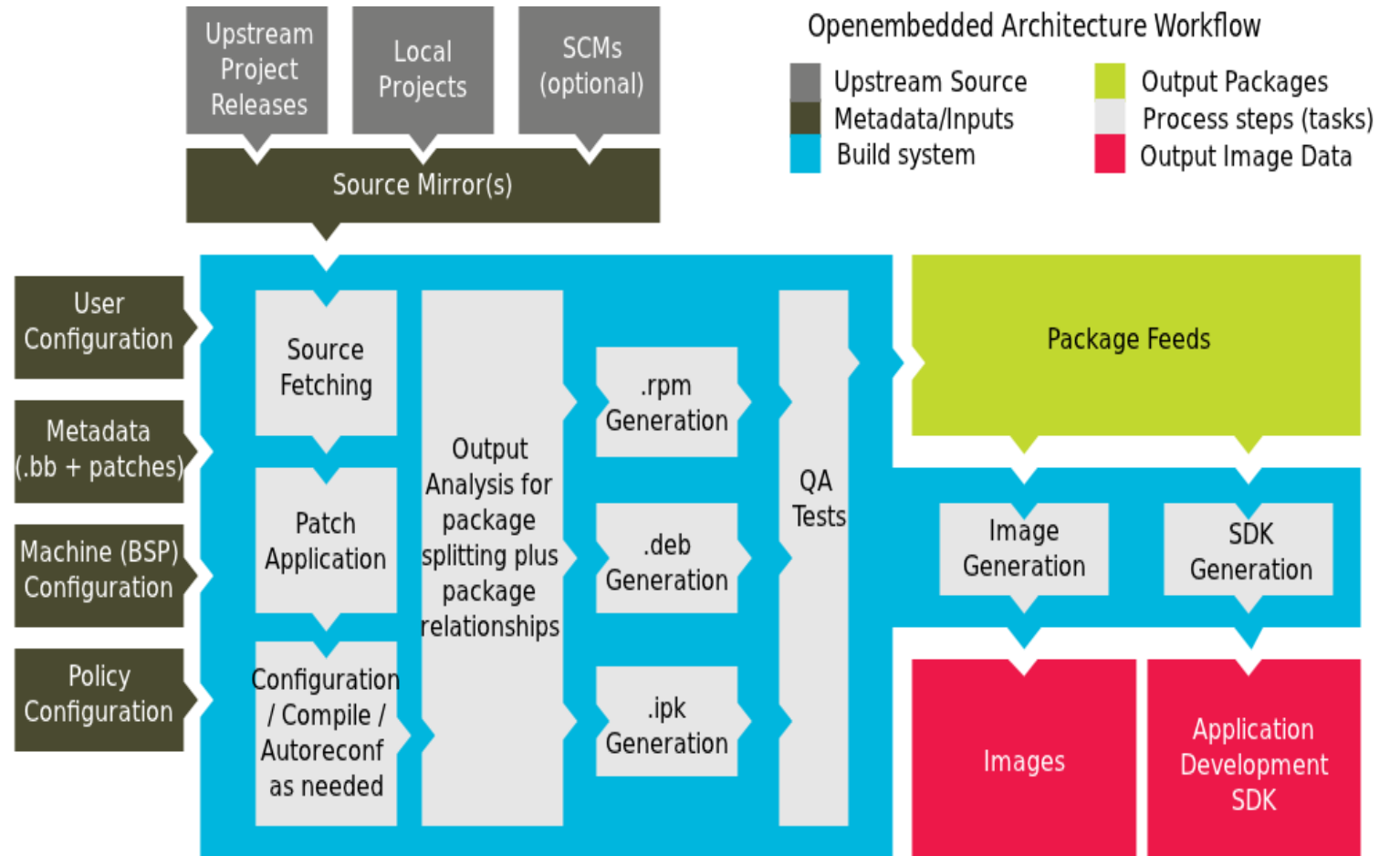
Layers architecture



Poky tasks execution



Development details



Development models

1. System development workflow
2. Application development workflow
3. Modifying temporary source code
4. OS Image development using Hob or Toaster
5. Using a development shell

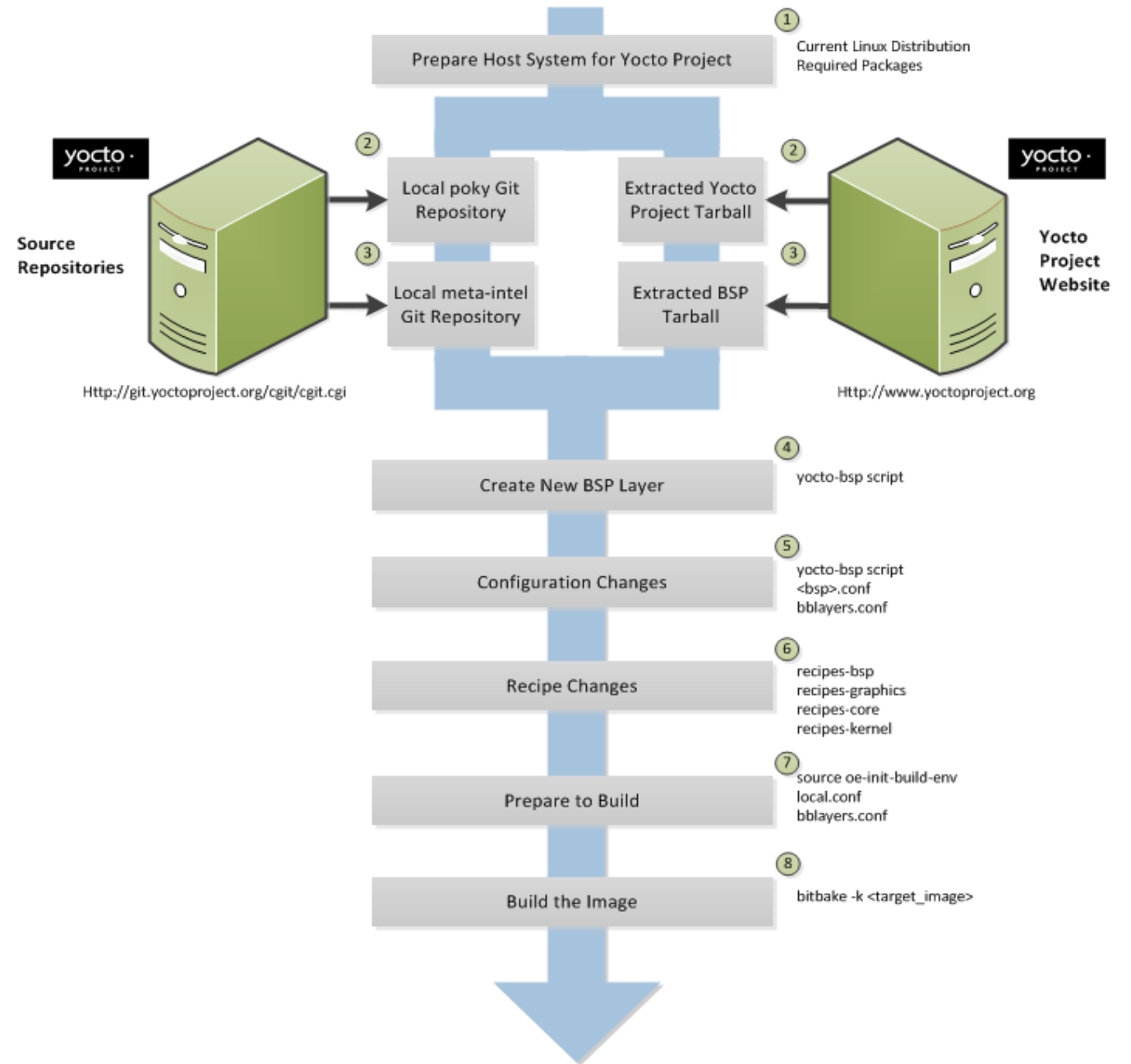
System development

System development involves modification/creation of an image for a specific hardware target.

A. Developing a Board Support Package (BSP)

B. Modifying the Kernel

Developing a BSP



BSP layer

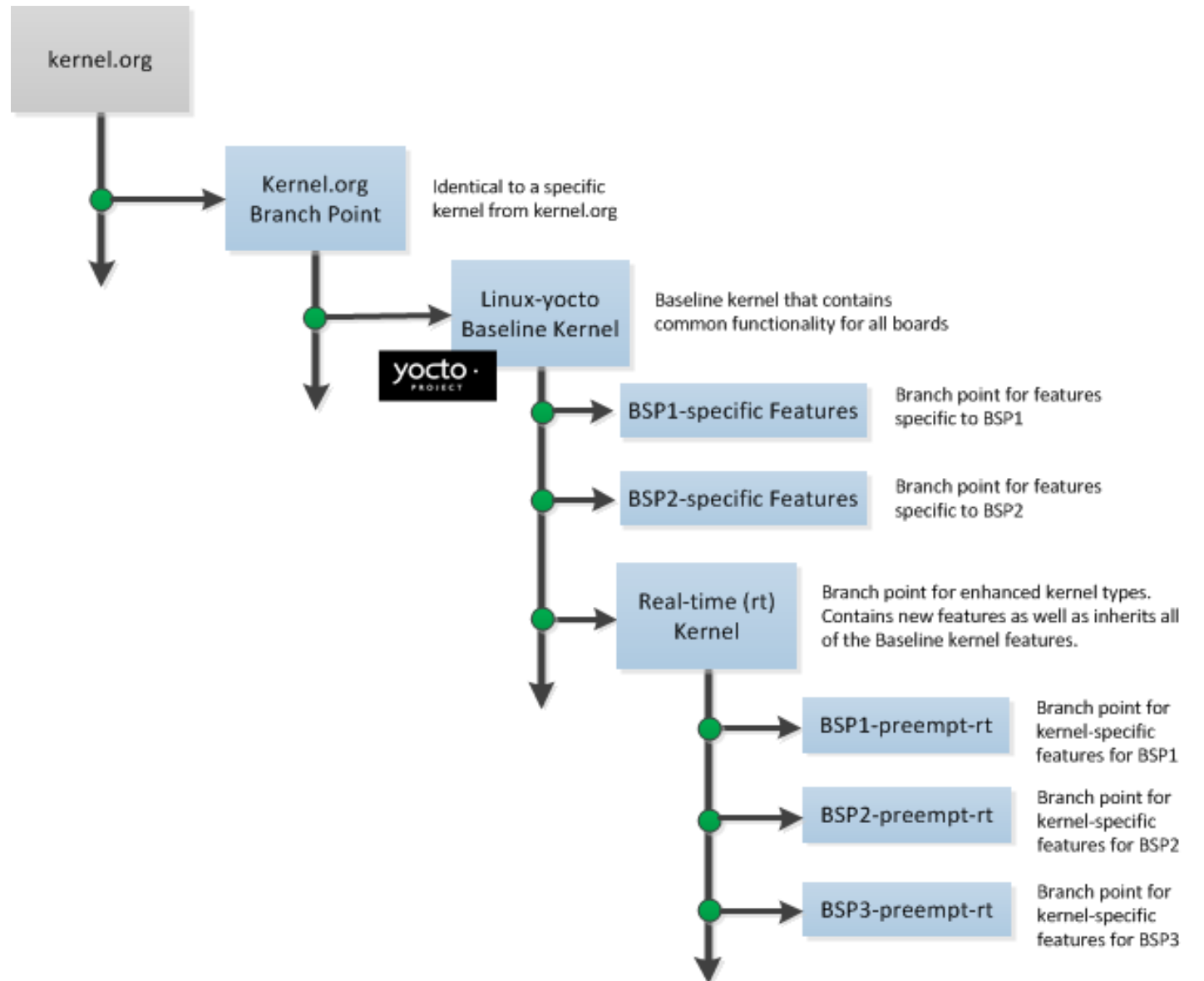
The BSP layers contain extensions and customizations to the base system such as:

- image customizations
- adding extra configuration
- additional recipes
- modifying recipes

Machine settings are added inside: *conf/machine/*.conf*

New layers should be added to the *BBLAYERS* variable inside *build directory conf/bblayers.conf* file:

Modifying the kernel



Application development

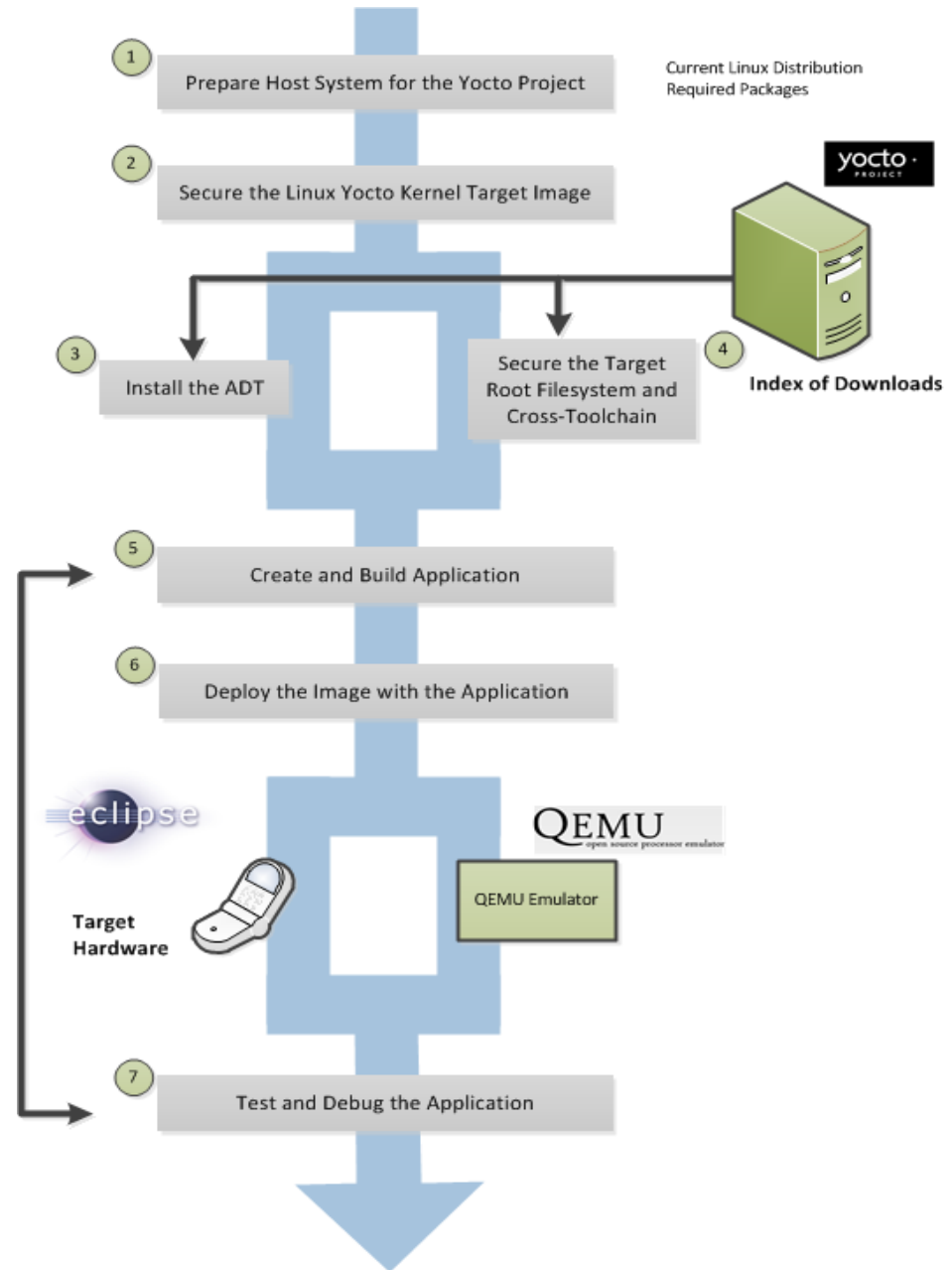
Create applications for the target hardware.

The target hardware runs an operating system created using the Yocto Project build system.

The Yocto Project provides:

- Application Development Toolkit (ADT)
- A stand-alone cross-development toolchains
- Eclipse Yocto Plugin to develop, deploy and test applications from within Eclipse.

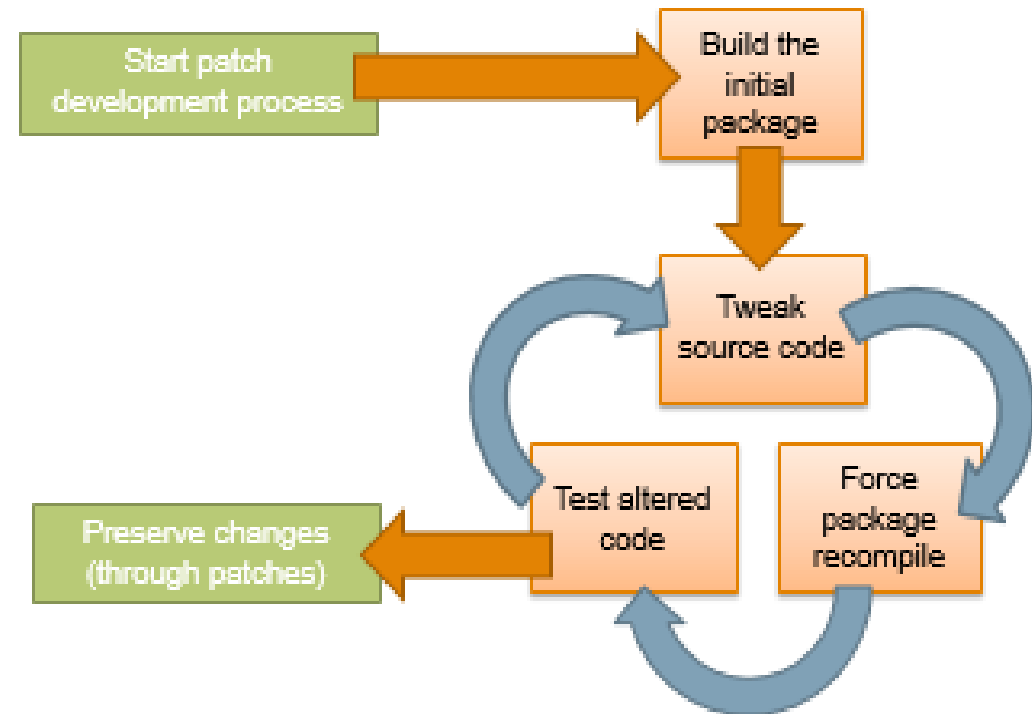
ADT workflow



Modifying temporary source code

Patches

This means modifying the temporary source code used by recipes to build packages during development.



Hob development

The Hob is a GUI for the Yocto Project build system. It is not used anymore.

You can use the Hob to build custom OS images.

Hob actions

- Select the *MACHINE* for which you are building the image.
- Modify policy settings.
- Manage layers.
- Select an image and add extra packages.
- Launch and monitor the build.

Toaster development

Web interface for the Yocto Project. More up to date the Hob.

Allows users to configure and run builds:

- Browse listed layers
- Browse images, recipes and machines
- Add, remove and import layers
- Set configuration variables
- Set target(s) to build
- Start a build

Provides extensive information about the build process:

- See what was built
- Browse through the directory structure
- See variables value
- Examine errors, warnings and other messages
- Bitbake task execution information
- Dependency relationships
- Performance information

Switch between command line and web interface at any time

Devshell

Devshell is useful to debug certain commands or even edit packages.

Devshell opens a terminal with a shell prompt that allows executions inside Yocto Project environment.

When using devshell the full name of the compiler should be used.

The same applies to other applications such as binutils, libtool and others. BitBake sets up environment variables such as CC to assist applications, such as make to find the correct tools.

It is also worth noting that devshell still works over X11 forwarding and similar situations .

Questions?