getting started with the Yocto Project

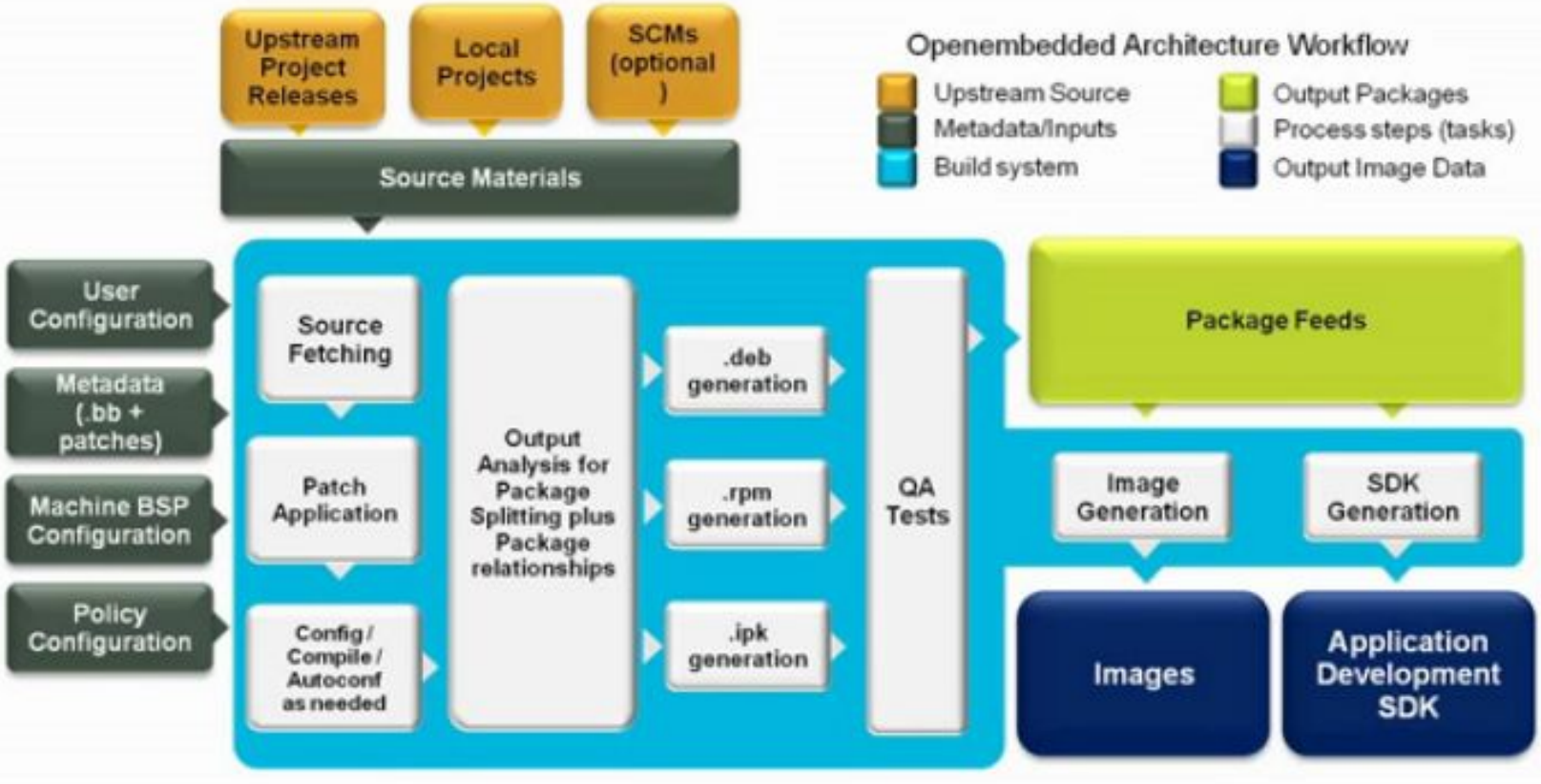
ESE3005

Yocto Project

- backed by a number of large tech corporations and individuals, in order to expedite getting embedded Linux systems working!
- hardware companies like IBM and TI have a vested interest in getting IoT hardware systems working (so they can sell their chips!)
- getting a Linux OS working for a newly designed embedded system is not a simple matter (there is no off-the-shelf distribution like Beagleboard Debian or Raspbian!)
- although there are several embedded OS “generic” operating systems available, notably: Android, Angstrom, OpenWrt;

Yocto Project Cont’d

- Yocto is not the only embedded Linux Build System:
  - Baserock
  - Buildroot
  - OpenEmbedded
- What is hard about a custom Linux distribution?
  - Bootloader
  - Kernel
  - Device Drivers
  - Life Cycle Management
  - Application Software Management


Openembedded Architecture Workflow
Upstream Source
Metadata/Inputs
Build system
Output Packages
Process steps (tasks)
Output Image Data
Upstream Project Releases
Local Projects
SCMs (optional)
Source Materials
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

Yocto: fetch, extract, patch, configure, build...

- essentially, you use various configuration files and special script files called “recipes” to direct the build system in how to construct your Linux image
- the image is then “flashed” on your embedded system
  - in this unit, we’ll be emulating the embedded system, so you won’t have to worry about flashing a physical device!
- the Yocto’s BitBake program then runs, based on the configuration files and recipes goes through a repetitive process involving the same steps:
    - fetch (from many different sources on the Internet)
    - extract (un-zipping, un-archiving)
    - patch (updating the source code)
    - configure (configuring the build)
    - build (compiling, linking)
    - install (placing the program in the right directories, etc.)
    - package (creating the archive or image)