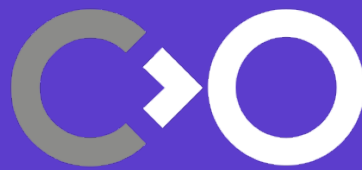


# Yocto and OpenEmbedded at Collabora

Andrew Shadura

LVEE Winter 2016



COLLABORA

**What is Yocto?**

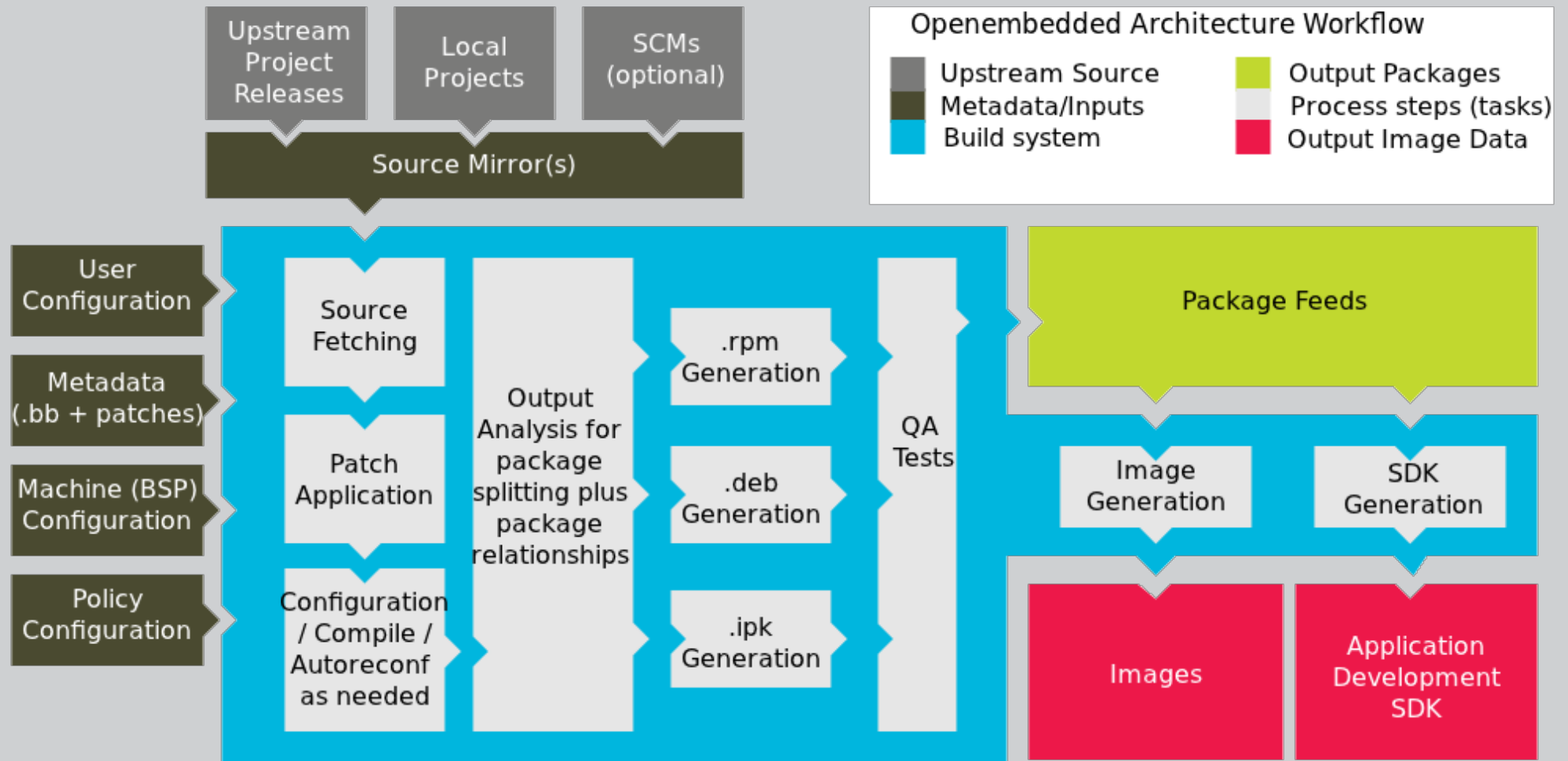
# Yocto Project

- Umbrella project for a collection of embedded Linux technologies
- Based on OpenEmbedded technology
- Allows for easy creation of heavily customised Linux distributions
- Layer architecture for easy code reuse
- Supports x86, ARM, MIPS...

# Yocto Project Components

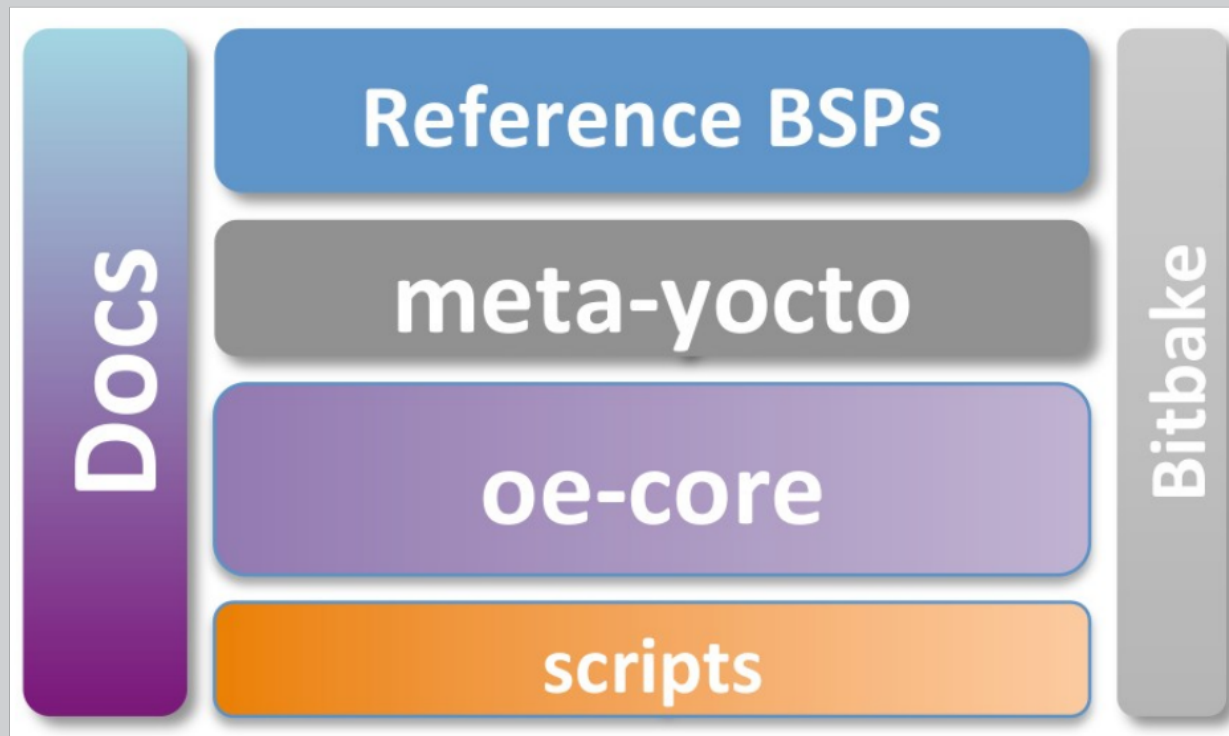
- OpenEmbedded Core:
  - BitBake and other tools
  - package metadata and configuration
- Poky reference distribution
- Opkg
- Pseudo
- ...and more!

# Yocto Project Development Environment



# Poky

- BitBake build engine
- meta-yocto (contains distribution policy)
- build scripts (infrastructure)
- foundation package recipes (oe-core)
- hardware support layers (BSPs)
- documentation



## BitBake recipes

- language similar to Make and shell
- similar to Gentoo ebuilds
- used to build binary packages
- supports including, inheritance overrides
- provides classes to simplify common tasks (build system support, image generation, init systemd)

Examples



```
SUMMARY = "Gives a fake root environment"
HOMEPAGE = "http://fakeroot.alioth.debian.org"
SECTION = "base"
LICENSE = "GPLv3"
LIC_FILES_CHKSUM = "file://COPYING;md5=f27defe1e96c2e1ecd4e0c9be8967949"

SRC_URI = "
    ${DEBIAN_MIRROR}/main/f/fakeroot/fakeroot_${PV}.orig.tar.bz2
"

inherit autotools

do_configure_prepend() {
    mkdir -p ${S}/build-aux
}

do_install_append() {
    install -d ${D}${includedir}/fakeroot
    install -m 644 *.h ${D}${includedir}/fakeroot
}

# fakeroot needs getopt which is provided by the util-linux package
RDEPENDS_${PN} = "util-linux"

SRC_URI[md5sum] = "706171d8d520b1ca1576ac73f2ceb4f3"
SRC_URI[sha256sum] = "0a359efa3e9496c33234b3e9c89306a09bb4da9d33de43c261f1d8447e6ebea2"
```

```
FILESEXTRAPATHS_prepend := "${THISDIR}/${PN}:"
```

```
SRC_URI += " file://systemconf.diff"
```

```
SRC_URI += " file://ugly-hack.target"
```

```
SRC_URI_remove_wandboard = "file://systemconf.diff"
```

```
rootprefix = "${exec_prefix}"
```

```
DEPENDS_remove = "dbus"
```

```
PACKAGECONFIG[dbus] = "--enable-dbus,--disable-dbus,dbus"
```

```
EXTRA_OECONF_append = "\n\n    --enable-compat-libs \n    --disable-split-usr \n"
```

```
do_install_append() {
```

```
    install -m 0644 ${WORKDIR}/ugly-hack.target ${D}${systemd_system_unitdir}/
```

```
    install -d ${D}${sysconfdir}/systemd/system/multi-user.target.wants
```

```
    ln -sf ${systemd_systemd_unitdir}/ugly-hack.target
```

```
        ${D}${sysconfdir}/systemd/system/multi-user.target.wants/ugly-hack.target
```

```
    rm -f ${D}${sysconfdir}/udev/rules.d/touchscreen.rules
```

```
}
```

# How do we use Yocto at Collabora

# Collabora: open source consultancy

- GStreamer
- Wayland and Weston
- Linux kernel
- D-Bus and systemd
- LibreOffice
  - core LibreOffice team
- LibreOffice Online

## Project: Yocto-based distro

- Customer: big corporation, medical equipment manufacturer  
strict rules
- Buildroot-based custom OS  
with Linux 3.x
- Lots of custom and proprietary  
dæmons and APIs

# Customer wanted

- Yocto
- revamp the build system
- use upstream mainline code whenever possible
- drop custom abstractions
- migrate to systemd
- adopt a better development methodology

## Distro from scratch

- start with a minimal bootable image
- based on stable Poky (jethro)
- systemd with user sessions
- Wayland + Weston + XWeston
- A new layer with just three files:
  - layer definition
  - distribution configuration
  - machine definition

```
# We have a conf and classes directory, add to BBPATH
BBPATH .= ":{LAYERDIR}"
```

```
# We have recipes-* directories, add to BBFILES
BBFILES += "${LAYERDIR}/recipes-*/*/*.bb \
    ${LAYERDIR}/recipes-*/*/*.bbappend"
```

```
BBFILE_COLLECTIONS += "distro-core"
BBFILE_PATTERN_distro-core = "^${LAYERDIR}/"
BBFILE_PRIORITY_distro-core = "10"
```



```
require conf/distro/poky.conf
DISTRO = "distro-core"
```

```
DISTRO_NAME = "Core Distro Platform"
DISTRO_VERSION = "2.0"
DISTRO_CODENAME = "badger"
DISTRO_FEATURES_append = " systemd wayland xwayland xattr pam apparmor"
DISTRO_FEATURES_remove = "x11"
# Enable systemd as init
VIRTUAL-RUNTIME_init_manager = "systemd"
# Disable sysv init and prevent any init scripts in the images
DISTRO_FEATURES_BACKFILL_CONSIDERED = "sysvinit"
VIRTUAL-RUNTIME_initscripts = ""
PREFERRED_PROVIDER_jpeg = "jpeg"
PREFERRED_PROVIDER_jpeg-native = "jpeg-native"
```

```
#@NAME: vexpress-a9 machine
#@DESCRIPTION: Machine configuration for the vexpress a9 board

PREFERRED_PROVIDER_virtual/xserver = "xserver-xorg"

# Ship all kernel modules by default
MACHINE_EXTRA_RRECOMMENDS = " kernel-modules"

# Allow for MMC booting (required by the NAND-less)
EXTRA_IMAGEDEPENDS += ""

# Uncomment the following line to enable the hard floating point abi. Note that
# this breaks some binary libraries and 3D (neither of which ship with
# meta-yocto). For maximum compatibility, leave this disabled.
#DEFAULTTUNE ?= "cortexa8hf-neon"
include conf/machine/include/tune-cortexa9.inc

#IMAGE_CLASSES += "sdcard_image"

IMAGE_FSTYPES += "tar.bz2 ext3"
#EXTRA_IMAGECMD_jffs2 = "-lnp "

# 2.6.37 and later kernels use OMAP_SERIAL, ttyO2
# earlier kernels use ttyS2
SERIAL_CONSOLE = "115200 ttyO2"

PREFERRED_PROVIDER_virtual/kernel ?= "linux-yocto"

KERNEL_IMAGETYPE = "zImage"

UBOOT_MACHINE = "ca9x4_ct_vxp_config"
UBOOT_ENTRYPOINT = "0x80008000"
UBOOT_LOADADDRESS = "0x80008000"
KERNEL_EXTRA_ARGS += "LOADADDR=${UBOOT_ENTRYPOINT}"

MACHINE_FEATURES = "kernel26 apm usb gadget usbhost vfat alsa"
```

- i.MX6 and i.MX5 hardware using Freescale's meta-fsl-{arm,arm-extra} layers saved a lot of time initially
- later, we've got rid of those layers, replacing them with their simplified versions.
- initial single layer has later been split into Core and Applications layers

## **Core layer:**

updates, hotfixes, configuration,  
HW support dæmons, not-release-ready SW

## **Application layer:**

proprietary applications only



Enable the user-sessions support with a PACKAGECONFIG flag.

```
diff --git a/meta/recipes-core/dbus/dbus_1.10.6.bb b/meta/recipes-core/dbus/dbus_1.10.6.bb
index fec68a4..245798a 100644
--- a/meta/recipes-core/dbus/dbus_1.10.6.bb
+++ b/meta/recipes-core/dbus/dbus_1.10.6.bb
@@ -69,7 +69,9 @@ FILES_${PN} = "${bindir}/dbus-daemon*
        ${datadir}/dbus-1/session.conf
        ${datadir}/dbus-1/system.d
        ${datadir}/dbus-1/system.conf
-       ${systemd_system_unitdir}"
+       ${systemd_system_unitdir}
+       ${systemd_user_unitdir}
+"
FILES_${PN}-lib = "${libdir}/lib*.so.*"
RRECOMMENDS_${PN}-lib = "${PN}"
FILES_${PN}-dev += "${libdir}/dbus-1.0/include ${bindir}/dbus-test-tool"
@@ -106,5 +108,6 @@ PACKAGECONFIG_class-nativesdk = ""
PACKAGECONFIG[x11] = "--with-x --enable-x11-autolaunch,--without-x --disable-x11-autolaunch, virtual/libx11 libsm"
PACKAGECONFIG[largefile] = "--enable-largefile,--disable-largefile,,"
+PACKAGECONFIG[user-session] = "--enable-user-session --with-systemduserunitdir=${systemd_user_unitdir},
--disable-user-session"

do_install() {
    autotools_do_install
```

dbus: enable user sessions

Use a bbappend to enable user session and user bus support for use when deploying systemd user sessions.

```
diff --git a/recipes-core/dbus/dbus_%.bbappend b/recipes-core/dbus/dbus_%.bbappend
new file mode 100644
index 0000000..afb225a
--- /dev/null
+++ b/recipes-core/dbus/dbus_%.bbappend
@@ -0,0 +1 @@
+PACKAGECONFIG_append = "user-session"
```

## Contributing back

- Customer **insists** on upstreaming
- Patches to OE, Weston, Linux, ifupdown



Thanks!