GDP-ivi9

Introduction

GDP-ivi9 Hands on Session

14th AMM

April 26th

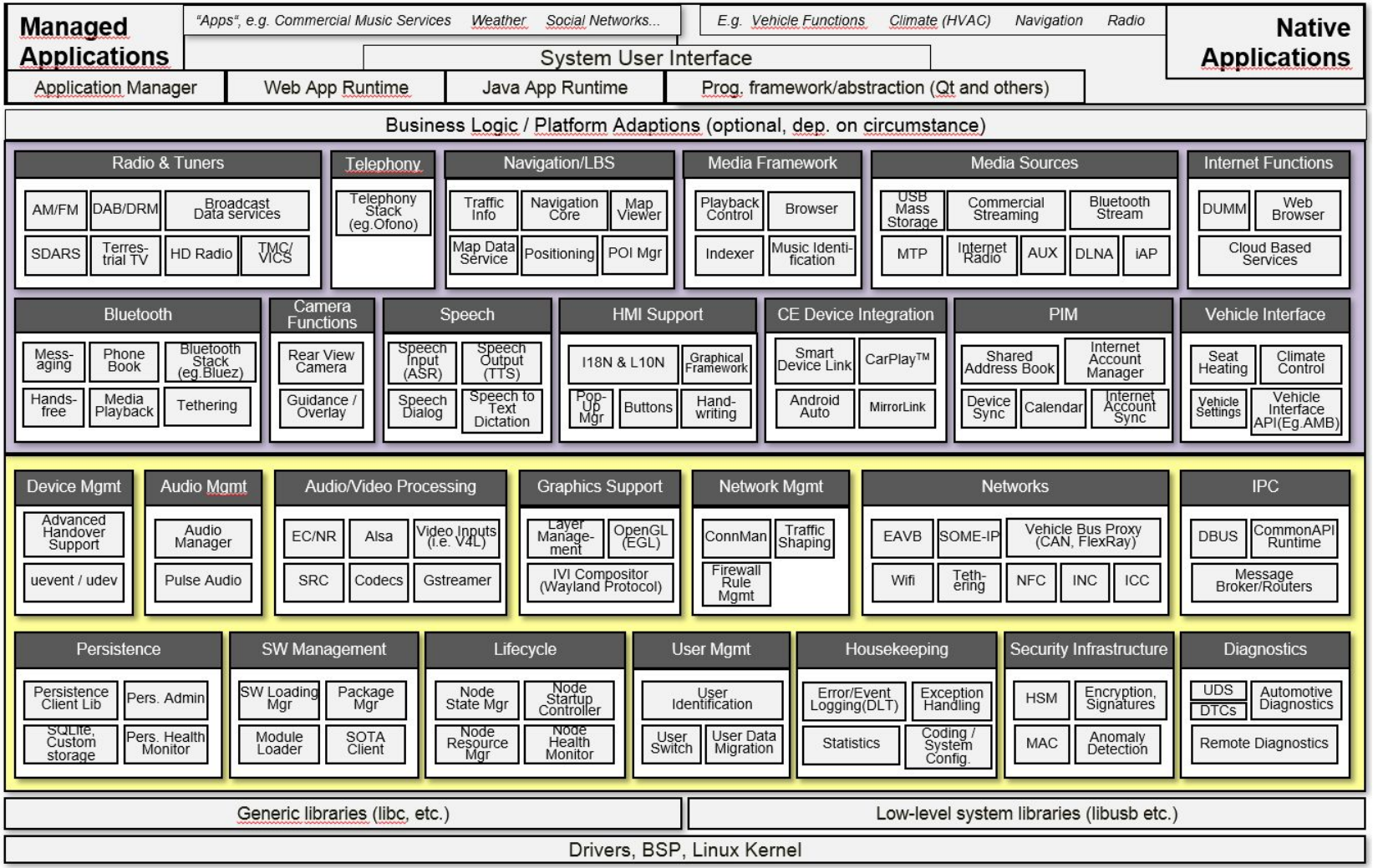
Agustin Benito Bethencourt

GDP delivery team

Codethink Ltd

GDP delivery team

- Maintainers
  - Changhyeok Bae, community member
  - Jonathan Maw, Codethink Ltd.
  - Tom Pollard, Codethink Ltd.
- Other key people
  - Stephen Lawrence: Renesas BSP maintainer. Renesas.
  - Agustín Benito Bethencourt: management. Codethink Ltd.
- Contributors: the most important ones. Thank you.


Managed Applications
"Apps", e.g. Commercial Music Services Weather Social Networks...
E.g. Vehicle Functions Climate (HVAC) Navigation Radio
Native Applications
System User Interface
Application Manager
Web App Runtime
Java App Runtime
Prog. framework/abstraction (Qt and others)
Business Logic / Platform Adaptions (optional, dep. on circumstance)
Radio & Tuners
AM/FM
DAB/DRM
Broadcast Data services
SDARS
Terres-trial TV
HD Radio
TMC/ VICS
Telephony
Telephony Stack (eg.Ofono)
Navigation/LBS
Traffic Info
Navigation Core
Map Viewer
Map Data Service
Positioning
POI Mgr
Media Framework
Playback Control
Browser
Indexer
Music Identi-fication
Media Sources
USB Mass Storage
Commercial Streaming
Bluetooth Stream
MTP
Internet Radio
AUX
DLNA
iAP
Internet Functions
DUMM
Web Browser
Cloud Based Services
Bluetooth
Mess-aging
Phone Book
Bluetooth Stack (eg.Bluez)
Hands-free
Media Playback
Tethering
Camera Functions
Rear View Camera
Guidance / Overlay
Speech
Speech Input (ASR)
Speech Output (TTS)
Speech Dialog
Speech to Text Dictation
HMI Support
I18N & L10N
Graphical Framework
Pop-Up Mgr
Buttons
Hand-writing
CE Device Integration
Smart Device Link
CarPlay™
Android Auto
MirrorLink
PIM
Shared Address Book
Internet Account Manager
Device Sync
Calendar
Internet Account Sync
Vehicle Interface
Seat Heating
Climate Control
Vehicle Settings
Vehicle Interface API(Eg.AMB)
Device Mgmt
Advanced Handover Support
uevent / udev
Audio Mgmt
Audio Manager
Pulse Audio
Audio/Video Processing
EC/NR
Alsa
Video Inputs (i.e. V4L)
SRC
Codecs
Gstreamer
Graphics Support
Layer Manage-ment
OpenGL (EGL)
IVI Compositor (Wayland Protocol)
Network Mgmt
ConnMan
Traffic Shaping
Firewall Rule Mgmt
Networks
EAVB
SOME-IP
Vehicle Bus Proxy (CAN, FlexRay)
Wifi
Teth-ering
NFC
INC
ICC
IPC
DBUS
CommonAPI Runtime
Message Broker/Routers
Persistence
Persistence Client Lib
Pers. Admin
SQLite, Custom storage
Pers. Health Monitor
SW Management
SW Loading Mgr
Package Mgr
Module Loader
SOTA Client
Lifecycle
Node State Mgr
Node Startup Controller
Node Resource Mgr
Node Health Monitor
User Mgmt
User Identification
User Switch
User Data Migration
Housekeeping
Error/Event Logging(DLT)
Exception Handling
Statistics
Coding / System Config.
Security Infrastructure
HSM
Encryption, Signatures
MAC
Anomaly Detection
Diagnostics
UDS
DTCs
Automotive Diagnostics
Remote Diagnostics
Generic libraries (libc, etc.)
Low-level system libraries (libusb etc.)
Drivers, BSP, Linux Kernel

Who is the GENIVI Alliance

The GENIVI Community is currently represented by 140 member companies…

… committed to driving the broad adoption of specified, Open Source, In-Vehicle Infotainment software.

What does GENIVI Alliance do?

- Deliver Open Source IVI middleware which scope is non-differentiating elements.
  a. Individual software components and standard interfaces/APIs
  b. A flexible technical architecture
  c. Pre-integrated, reusable IVI software platform.
- Best practices and tools to ensure deployments when using GENIVI's IVI middleware.

What is GDP?

GENIVI Demo Platform:

- Open Source project. Done “in the open”.
- Integration + delivery project.
- Includes meta-ivi (Baseline).
- YOCTO and Baserock (future) based.
- Targets QEMU and several boards.

Why GDP? (i)

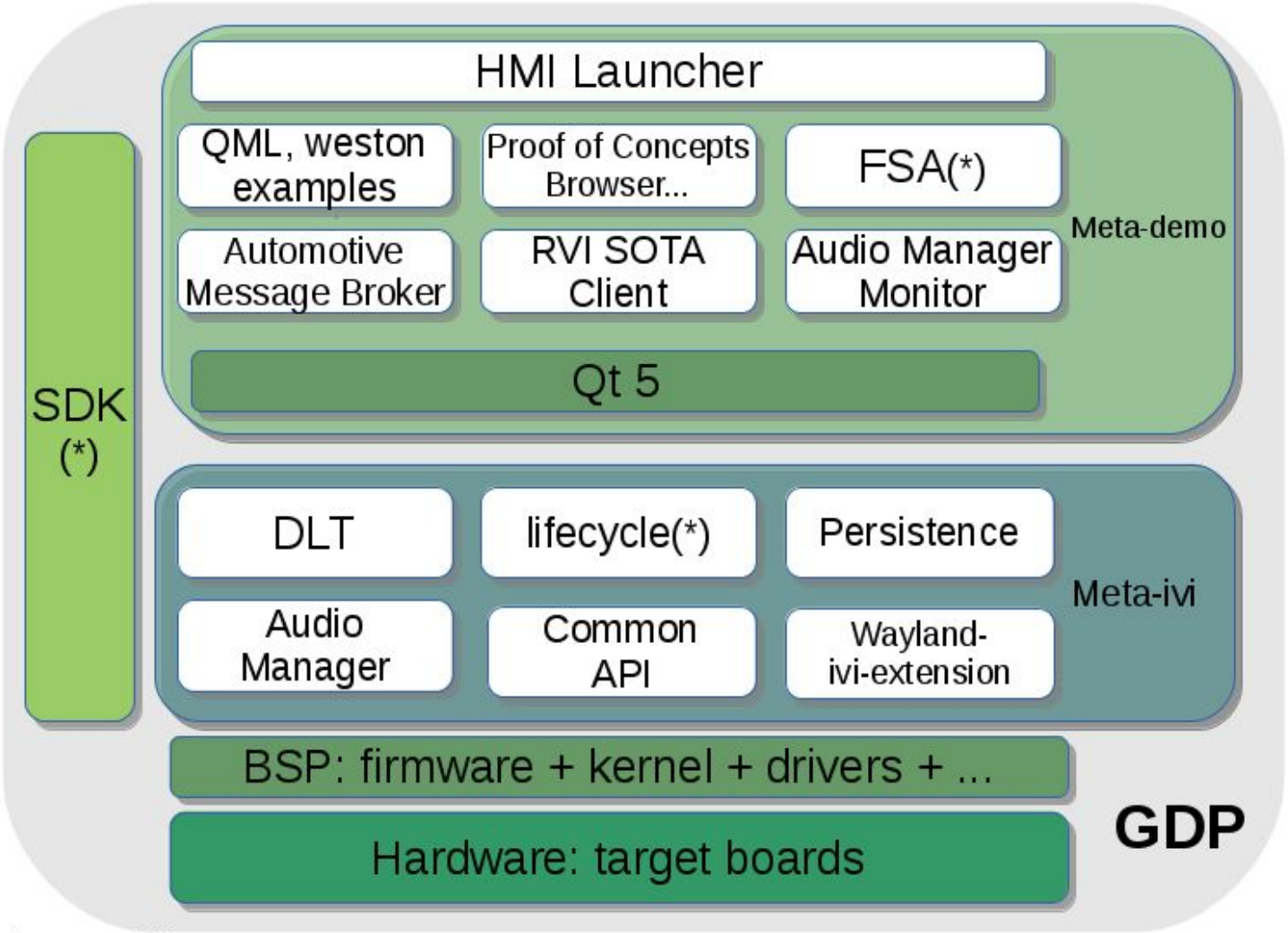
- Test middleware (components) developed by GENIVI.
- Rapid Prototyping to demonstrate integration with GENIVI components and specifications.
- Develop Open Source UI and apps for automotive.

Why GDP? (ii)

- Introduce yourself to Open Source for automotive.
- Promote your Open Source solutions or hardware within this industry.
- Become part of a disruptive change.

What is GDP-ivi9

- Latest GDP version, released on April 19th.
- Targets: QEMU, Renesas Porter and RPi2, so far.
- Base for demos across different events and tradeshows.
  - Used in every 14th AMM Hands On sessions.
- Check the feature page for highlights.



(*) Not available yet in GDP

Future of GDP

- GDP-ivi10: from demo platform to platform for innovation (development platform).
- Expectations:
  - More components integrated: lifecycle, RVI, FSA…
  - Updates of the existing ones + baseline.
  - Porting to further boards.
  - Participate in the GDP-ivi10 requests discussion.
- Towards quantitatively managed delivery practices.

14th AMM GDP-ivi9 Hands On session

- Check the pre-requisites before coming.
- Get support before the Hands On session to be ready.
  - One hour before the session starts at the same room.
- GDP-ivi9 Hands On sessions content.
- How was the session? Please provide feedback.

Hands On session goals

- Understand what is and what is not GDP
- Get familiar with GDP-ivi9
- Learn the basics to potentially become a GDP contributor

Expectations

- Please introduce yourself
  - Name, company, position…
  - How familiar are you with Linux / Yocto / GDP / GDP-ivi9 ?
- What do you expect from this session? Let us know.

Get GDP-ivi9

- Download GDP
- Cool but… show me the code.
- GDP wiki.
- Channels:
  - IRC: #automotive at freenode.net
  - Mailing list: https://lists.genivi.org/mailman/listinfo/genivi-projects
  - Weekly project meeting.

Interesting links

- www.genivi.org
  - GENIVI FAQ
  - projects.genivi.org
- GENIVI Demo Platform
  - GDP Roadmap
  - GDP-ivi9
  - Download GDP-ivi9
- Get involved:
  - Get the sources
  - Contribution policies
  - Report bugs
- Follow up
  - Delivery reports
  - Weekly status
  - GDP Out There

Thank you

Questions?

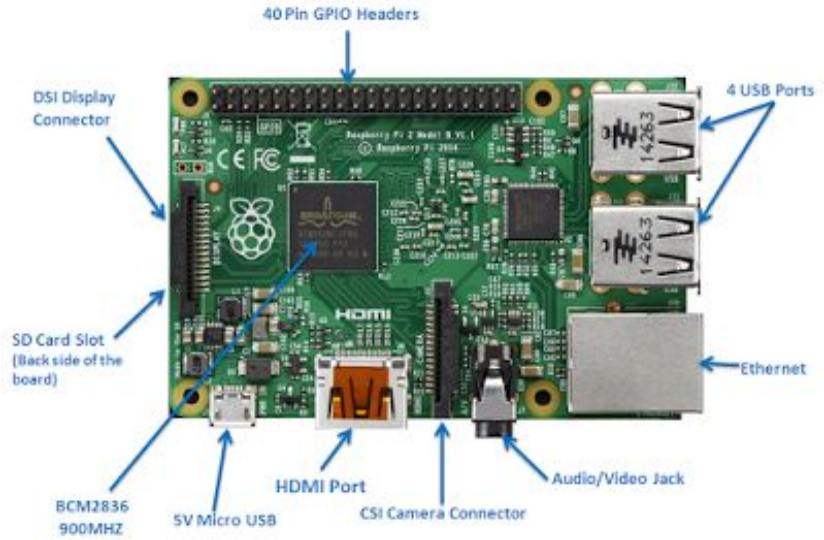
Happy hacking

Yocto / bitbake

Introduction

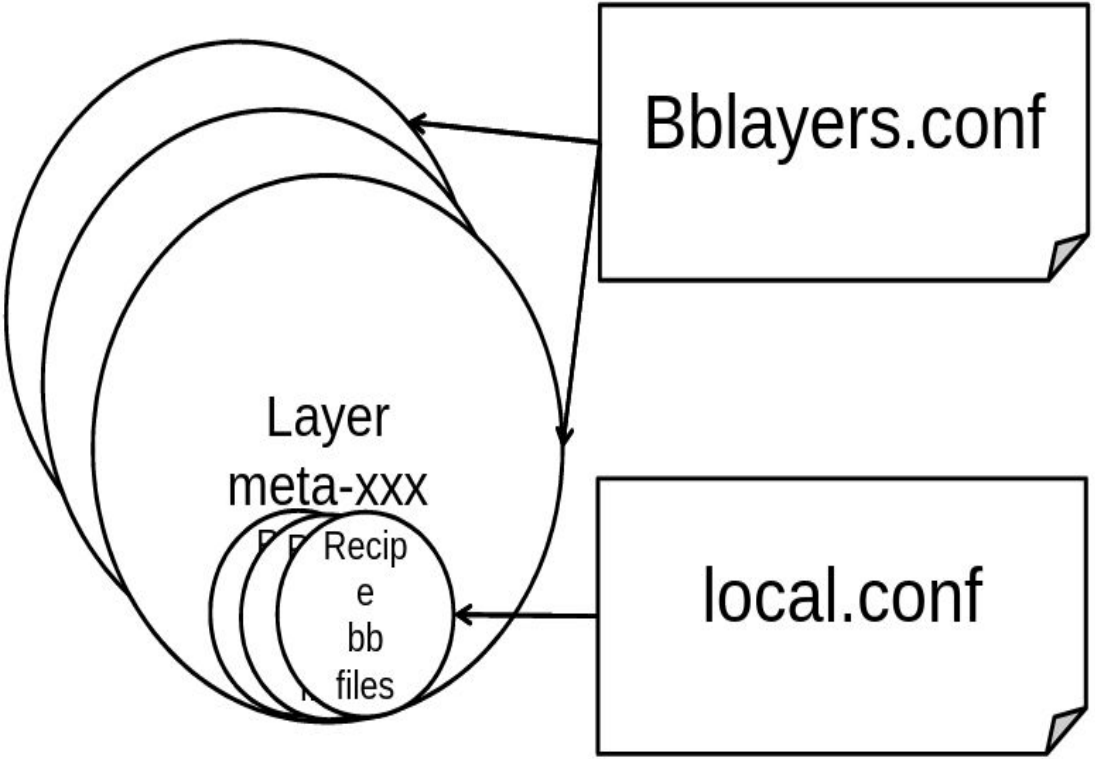
Raspberry Pi2 Model B

900MHz quad-core ARM Cortex-A7 CPU
1GB RAM
4 USB ports
40 GPIO pins
Full HDMI port
Ethernet port
Combined 3.5mm audio jack and composite video
Camera interface (CSI)
Display interface (DSI)
Micro SD card slot
VideoCore IV 3D graphics core



[1] https://www.raspberrypi.org/products/raspberry-pi-2-model-b/
[2] http://elinux.org/RPi_Hardware
[3] http://git.yoctoproject.org/cgit/cgit.cgi/meta-raspberrypi

Yocto Overview



- What layers to look into for recipes

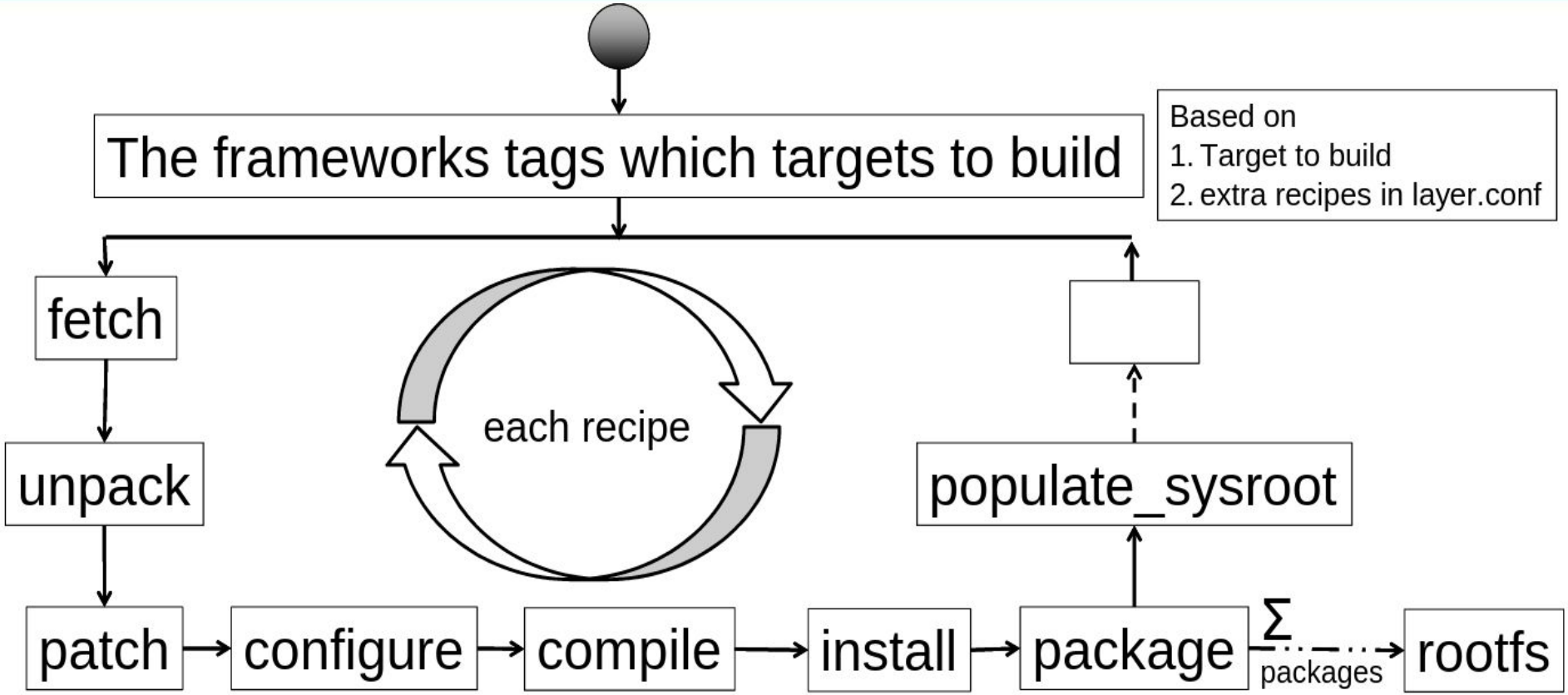
- Distribution: poky-ivi-systemd
- Machine: porter, intel-corei7-64, …
- Package Extra Configuration
- Add more Packages in the build image

Recipe Overview

- Where to find to source code (git, svn, tar.gz, .c)
- What version
- Apply patches on it ?
- Special commands.

```
SUMMARY = “aaa"
DESCRIPTION = “bbb"
HOMEPGAE = “ccc"
LICENSE = "LGPL-2.1"
LIC_FILES_CHKSUM = "file://ddd;md5=597c8d49137513c98683e1d73158292f"

inherit cmake

PV = “hhh+git${SRCPV}"

DEPENDS = “eee fff ggg"

SRC_URI = “iii.jjjj.kkk"
SRC_URI += "file://lll.patch"
SRCREV = "955972390d16ca275159891cad29c2166217094d"

S = "${WORKDIR}/mmm"
do_install_append() {
    mv ${D}/usr/include/nnn/* ${D}/usr/include
}
INSANE_SKIP_${PN} = "dev-deps"
```

Bitbake Build Process



Useful Bitbake Commands

Command	Description
bitbake <image>	**Bake an *image* (add *-k* to continue building even errors are found in the tasks execution)**
bitbake <package> -c <task>	**Execute a particular *package*'s *task*. Default Tasks names:** ***fetch, unpack,patch, configure, compile, install, package, package_write*, and *build.*** ***Example:* To (force) compile a kernel and then build, type:** ***$ bitbake linux-$target -f -c compile*** ***$ bitbake linux-$target***
bitbake <package> -c listtasks	**List all tasks for package**
bitbake-layers show-layers	**Show layers**
bitbake -s | grep $package	**Check recipe version**

Contribute Patches

Ensure you read over:

1. http://www.genivi.org/contribute
2. https://at.projects.genivi.org/wiki/display/PROJ/How+to+contribute+to+GENIVI
3. Relevant MAINTAINERS file, i.e: http://git.projects.genivi.org/?p=meta-genivi-demo.git;a=blob;f=MAINTAINERS;h=a159c3a1750f91e0b42730288e71c9ce692532aa;hb=HEAD

Generic steps to generate patchset to mailing list:

1. Ensure you've created your commits against the head of the $target branch following genivi submission guidelines.
2. git format-patch --cover-letter -M origin/$target -o outgoing/
   edit outgoing/0000-* (This is your cover letter)
   git send-email --to=maintainer@maintaineremployer.com --cc=genivi-projects@lists.genivi.org --cc=username2@personalaccount.com outgoing/*.patch

This will generate the patch(set), allowing you to edit the default cover letter to add a blurb etc & finally send it. Patches to genivi-projects@lists.genivi.org are public and will be reviewed publically. Please check https://at.projects.genivi.org/wiki/display/PROJ/Git+Email+Setup for further help and examples.

Review CI Pipeline

- Workflow & Infra still in development - http://go.genivi.org/go/pipelines
- Basic ‘CI’ pipelines setup for QEMU target
- Currently two pipelines, to handle typical patchsets
- GDP-Yocto-QEMU_x86-64-qemu-ci (qemu-ci branch of genivi-demo-platform.git) http://go.genivi.org/go/tab/pipeline/history/GDP-Yocto-QEMU_x86-64-qemu-ci
- GDP-Yocto-QEMU_x86-64-meta-genivi-demo-qemu-ci (qemu-ci branch of meta-genivi-demo.git submodule, qemux86-64 branch of genivi-demo-platform.git) - http://go.genivi.org/go/tab/pipeline/history/GDP-Yocto-QEMU_x86-64-meta-genivi-demo-qemu-ci
- This allows us to test patches to the qemux86-64 genivi-demo-platform branch, and changes to meta-genivi-demo against the head of the qemux86-64 branch.
- We would expect all incoming patchsets to pass builds in these pipelines before merging into the qemu target ‘master’ branch.
- Still in development, more target ‘post-merge’ CI branches need adding, and ultimately CD tooling for physical hardware tests.
- Quick demo if time permits

Thank you

Jonathan Maw: jonathan.maw@codethink.co.uk

Tom Pollard: tom.pollard@codethink.co.uk

www.codethink.co.uk