Current activities and developments for future ARM core led by Linaro from mobile to server

Akira Tsukamoto, Member Services

2012

Linaro

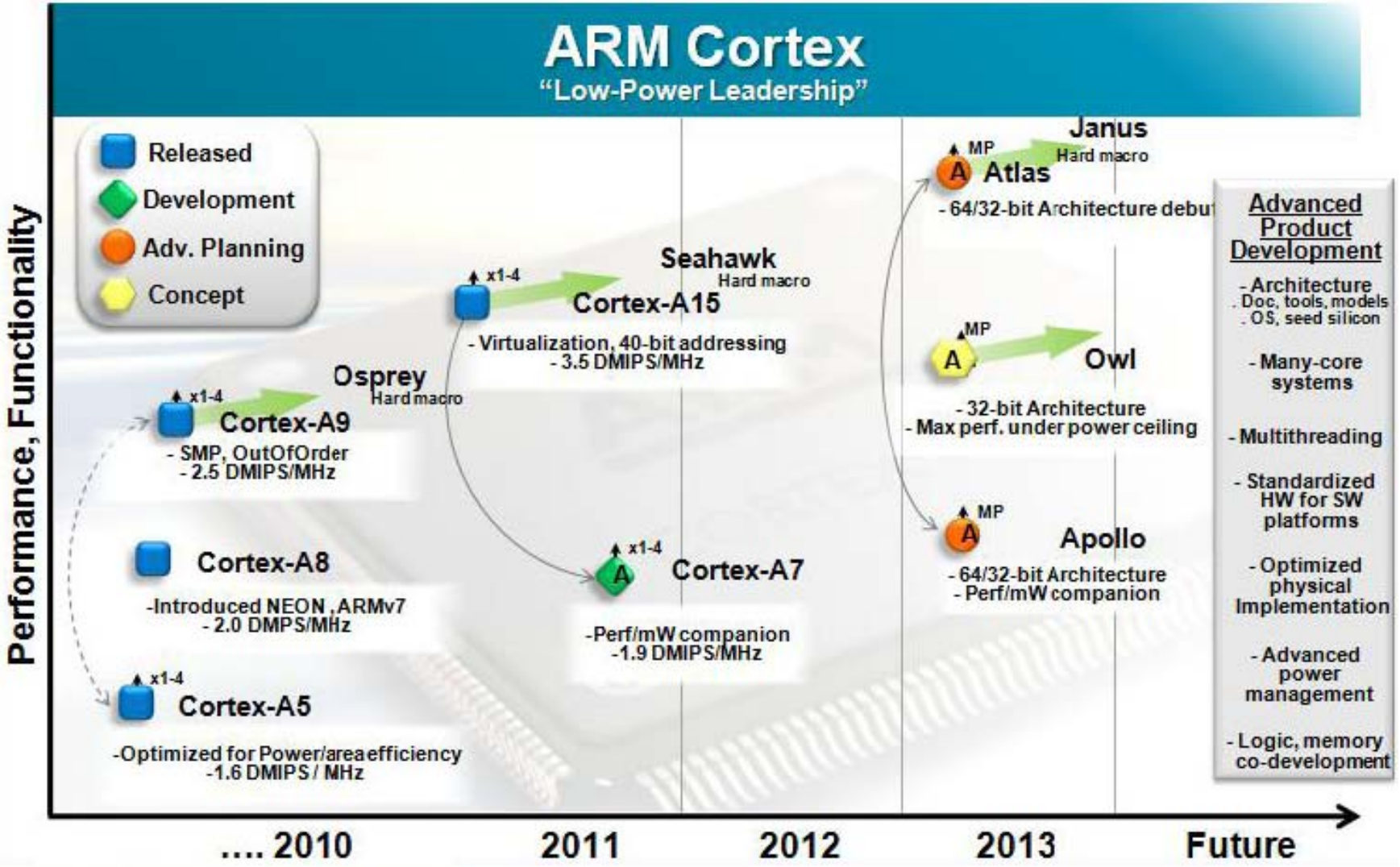
About Linaro

- Founded in June 2010 “to make it easier and faster for ARM partners to deliver product-quality Linux software platform”

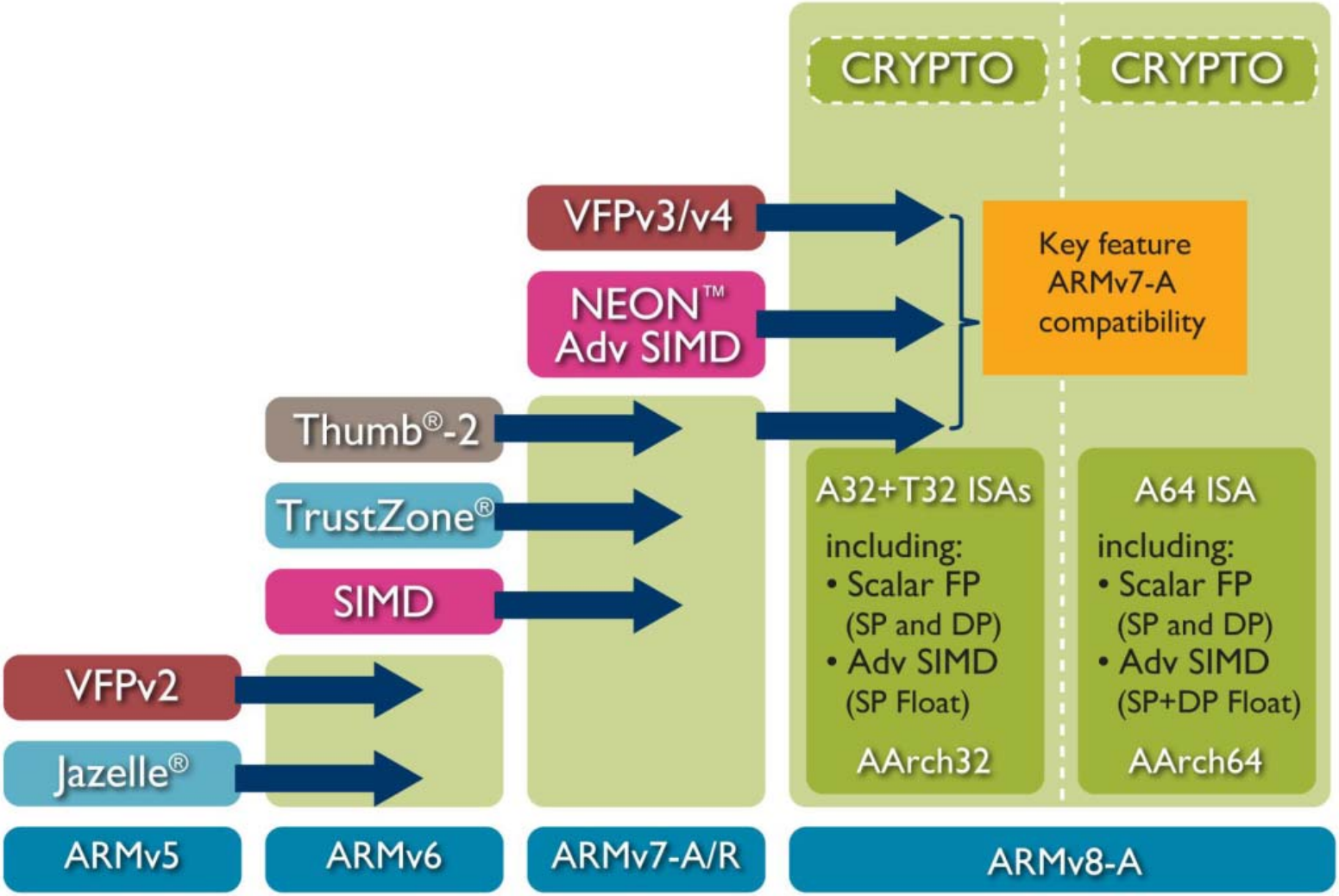
- Not-for-profit software engineering company
- Over 120 full time engineers distributed world-wide, including Linux kernel maintainers and subject experts

Before talking about Linaro
Introduction of ARM cores

ARM Cortex-A series roadmap



ARM Cortex-A series roadmap



What is Linaro and Why important

Why was Linaro Founded?

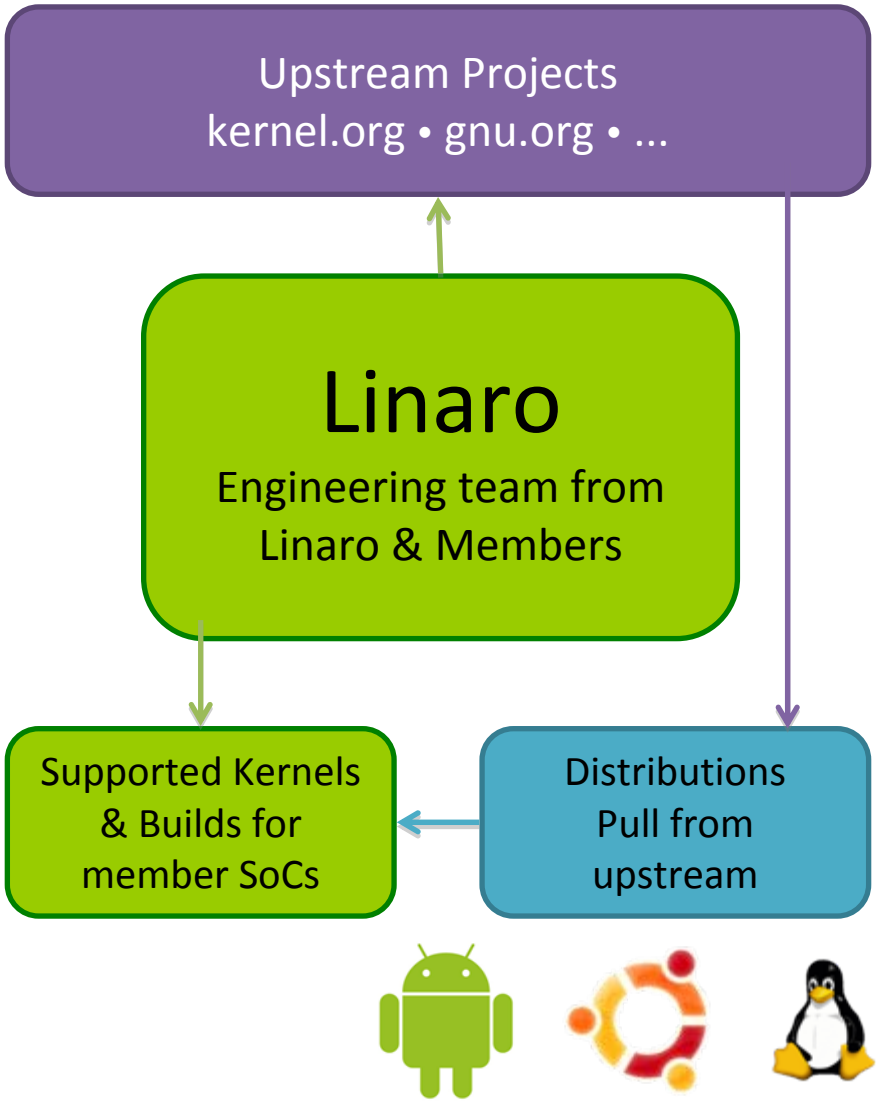
- To lead Open Source software development on ARM with shared cost among members
- To help members deliver high quality OSS-based products to market as quickly as possible
- Develop ahead on future Items, big.LITTLE, ARMv8, servers
- To solve common problems and enable members to focus their resources on differentiation

Why Linaro for ARM partners?

- Not enough ARM optimization in kernel and toolchain
  - Multicore, NEON, power management etc.
- ARM Kernel; Large and duplicated sources
  - Linus Torvalds complained about ARM on kernel mailing list, March 2011
  - Each SoC vendor needs their own kernel
- Quality assurance and Maintenance cost
  - Open Source Software has limited testing
  - Limited ARM Support in Linux distributions
- Time to market is compromised

What does Linaro do?

- **Linux kernel on ARM SoC**
  - Focus on Consolidation & Optimization
  - Latest ARM SoCs, Cortex A series
- **ARM gcc toolchain: best in class**
  - Monthly builds
- Delivers upstream
- LAVA, **Test, Validation**
  - Continuous Integration framework for Linux & Android on member SoCs
- Focus on member SoCs
  - Linux, Android & Ubuntu for members
  - Advance work on next Android kernel for member SoCs



Misunderstandings of Linaro

- Is Linaro Linux distribution?

Fact

- Open Source Engineering teams focusing on developing kernel and gcc toolchain, and providing automated testing for Linaro member companies

Objective

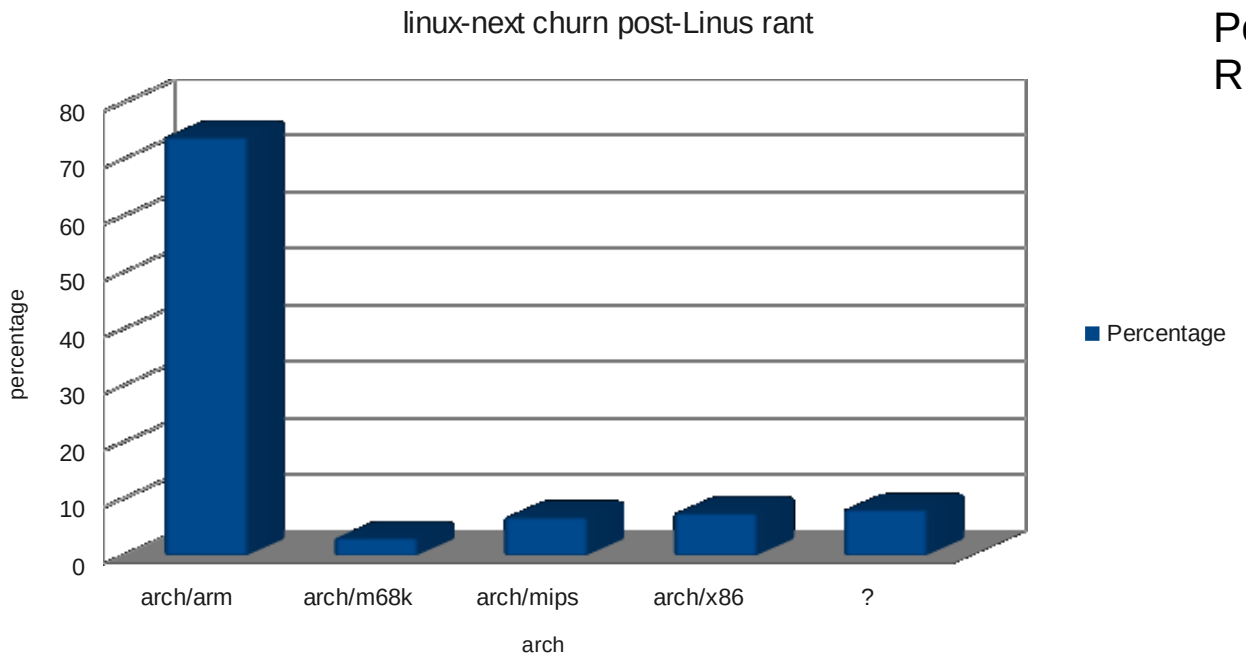
- Linaro provides faster time to market for member companies on Linux based products

Some examples of Linaro activities

Large kernel source (1/2)

- ARM source is 5x more than all other CPU before Linaro activity

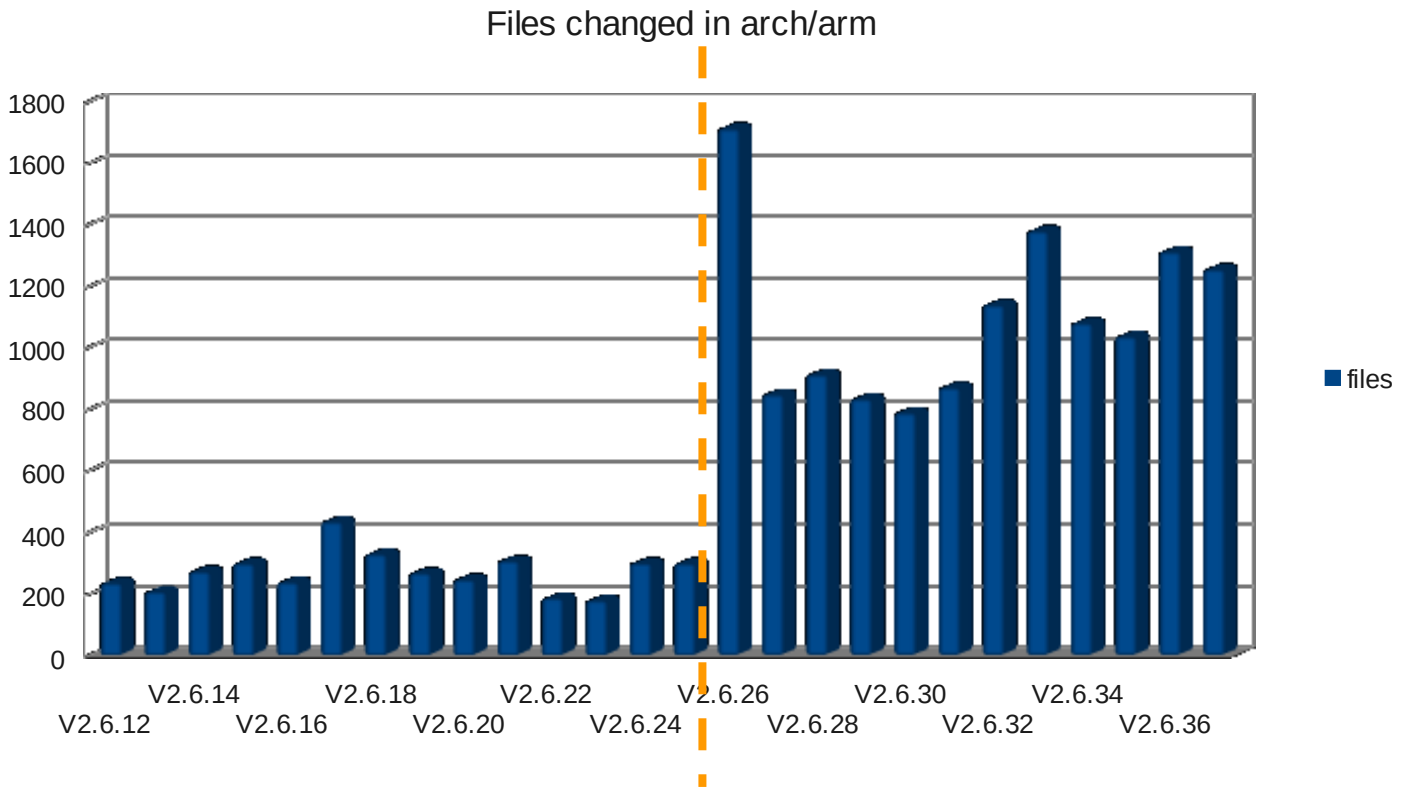
From dirstat
Posted by
RMK Apr 14



http://lists.infradead.org/pipermail/linux-arm-kernel/2011-April/048133.html

Large kernel source (2/2)

- History of files added to ARM arch



Started to increase after ARM11 released around 2.6.26

Linus Torvalds complained

From: **Linus Torvalds** <torvalds@linux-foundation.org>
Date: Thu, Mar 17, 2011 at 7:50 PM
Subject: Re: [GIT PULL] omap changes for v2.6.39 merge window

On Thu, Mar 17, 2011 at 11:30 AM, Tony Lindgren <tony@atomide.com> wrote:
>
> Please pull omap changes for this merge window from:

Gaah. Guys, this whole ARM thing is a ***** pain in the ass.**
...

From: **Linus Torvalds** <torvalds <at> linux-foundation.org>
Date: 2011-03-31 03:24:30 GMT (34 weeks, 4 hours and 23 minutes ago)
Subject: Re: [GIT PULL] omap changes for v2.6.39 merge window

So let's take a really simple example of this kind of crap.

Do this:

```
git ls-files arch/arm/ | grep gpio
```

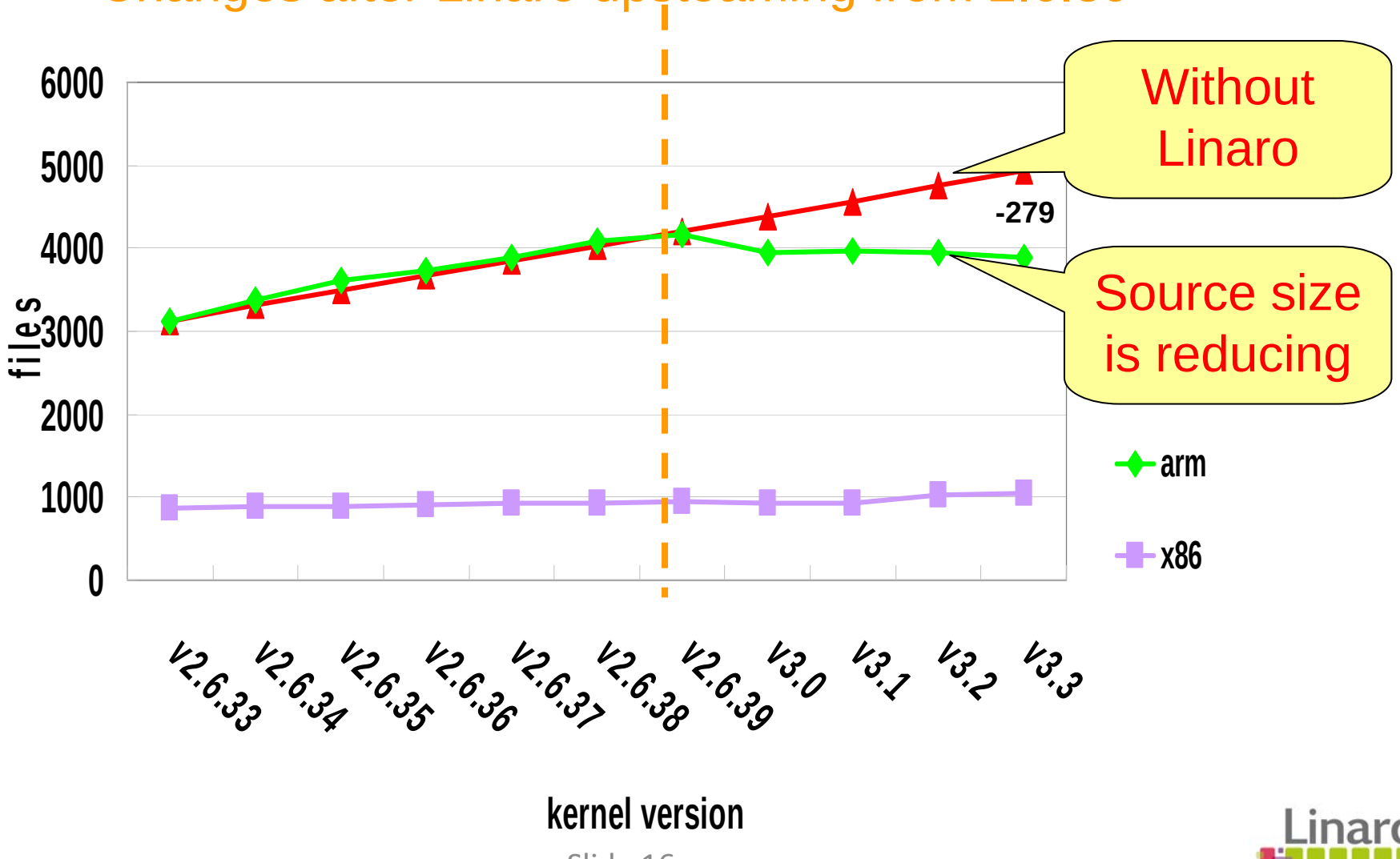
and cry. That's 145 files in the arm directory that are some kind of crazy gpio support.

Linaro to solve fragmentation

- Timeline
  - Mar 2011, Linus Torvalds complains publically
  - April 2011, Initial discussion at Embedded Linux Conference, SF
  - May 2011, Wide agreement at Linaro Developer Summit, Budapest
  - June 2011, Linus positive at LinuxCon Japan
- The proposed solution
  - Introduce **Device-Tree** for ARM
    - Rejecting separate sources for different boards “arch/arm/mach-xxxx”
  - create **arm-soc tree** maintained by Linaro
    - Consolidate ARM patches before sending to “linux-next tree”
  - Unification on “drivers/”

Linaro achievement on fragmentation

Changes after Linaro upsteaming from 2.6.39

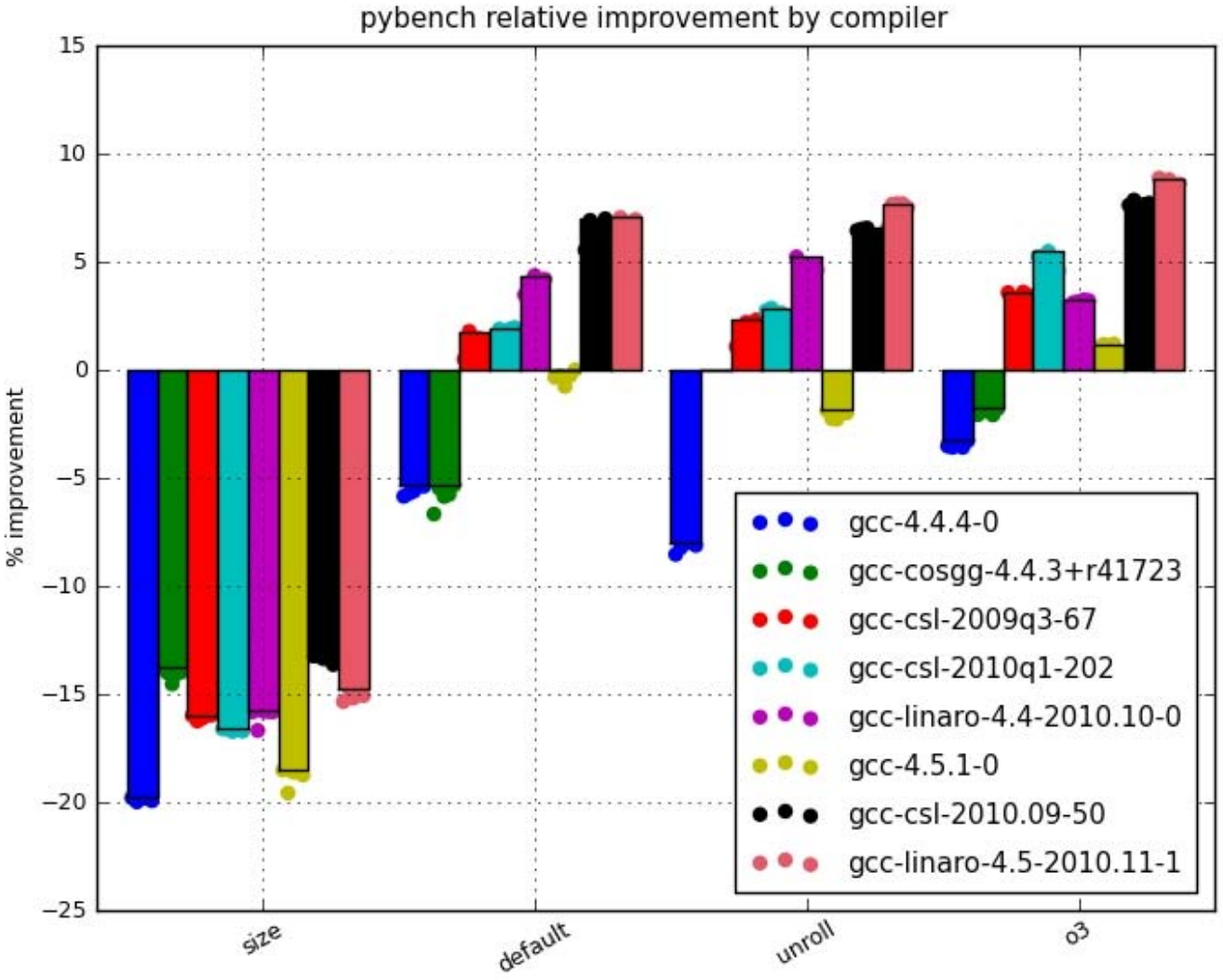


Linus Torvalds on LWN.net for Oct 27, 2011

ARM is clearly the most important architecture other than x86, he said, and some would argue that the order should be reversed. The good news is that **ARM Linux is getting better, and the ARM community seems to be making progress**, so he is **much happier with ARM today than** he was **six months ago**. It's not perfect, and he would like see more standardization, but things are much better. Torvalds said that he doesn't necessarily think that the PC platform is wonderful, but "supporting only a few ways to handle timers rather than hundreds is wonderful".

http://lwn.net/Articles/463908/

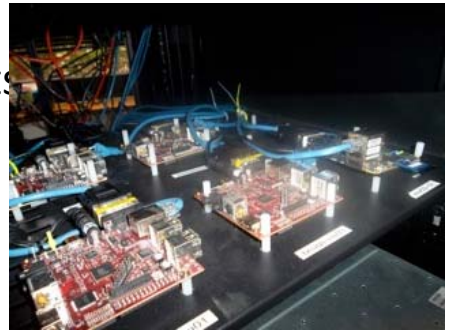
Good performance of Linaro toolchain



Linaro Open Source Testing & Validation

Open Source Software has limited testing

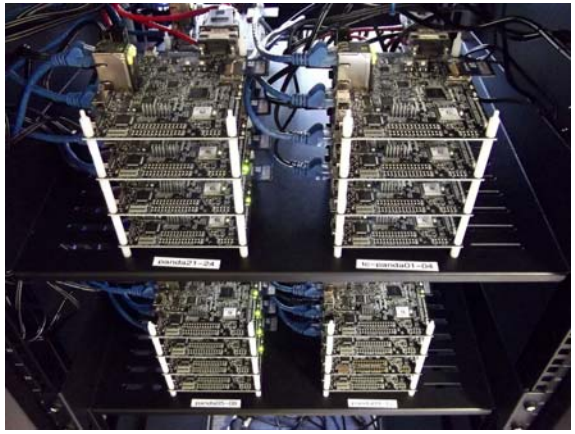
- Yet SoCs have commercial level software quality requirements



LAVA – Linaro Automated Validation Architecture

LAVA is an open source software test & validation platform

Populated only by Linaro members hardware



Provides Members

- Continuous Integration for daily build & testing
- Smoke, System and Customized testing
- Web dashboard for results and trends
- Measures distribution quality & trends

Framework is open source

Linaro maintaining large and expanding farm of latest Member SoC boards

Testing -> faster time to market

- Continuous Linaro Android kernel build loop
  - Rebasing Linaro kernel with Linux HEAD
  - Integrating with AOSP from Google
  - Testing

- Android 4.0 Ice Cream Sandwich (ICS)
  - Nov. 14, Source code was released to public from Google
- Linaro success to build and run ICS on Nov. 15 (Next day)
  - Video uploaded
    - http://www.youtube.com/watch?v=eaVszdsZ8aY
  - Success to port and build ICS on all Samsung, TI, STE, Freescale in 10 days after ICS release (Linaro does not have early access from Google)

- Current development for future Android release
  - Building successfully Android kernel version 3.3
  - Building successfully Android 4.0.4

Strong Linaro leadership on Linux

(None)	1111	13.1%
Red Hat	882	10.4%
(Unknown)	749	8.8%
Intel	616	7.3%
Broadcom	428	5.1%
Novell	380	4.5%
IBM	301	3.6%
Texas Instruments	276	3.3%
(Consultant)	223	2.6%
Freescale	182	2.2%
Linaro	170	2.0%
Samsung	162	1.9%
Google	150	1.8%
Wolfson Microelectronics	142	1.7%
Fujitsu	131	1.5%
Renesas Technology	100	1.2%
Oracle	82	1.0%
MiTAC	80	0.9%
Nokia	79	0.9%
(Academia)	73	0.9%

Linaro members are #7, #8, #10 and #12 most active contributors to Linux 3.0 by changesets

Linaro is already #11

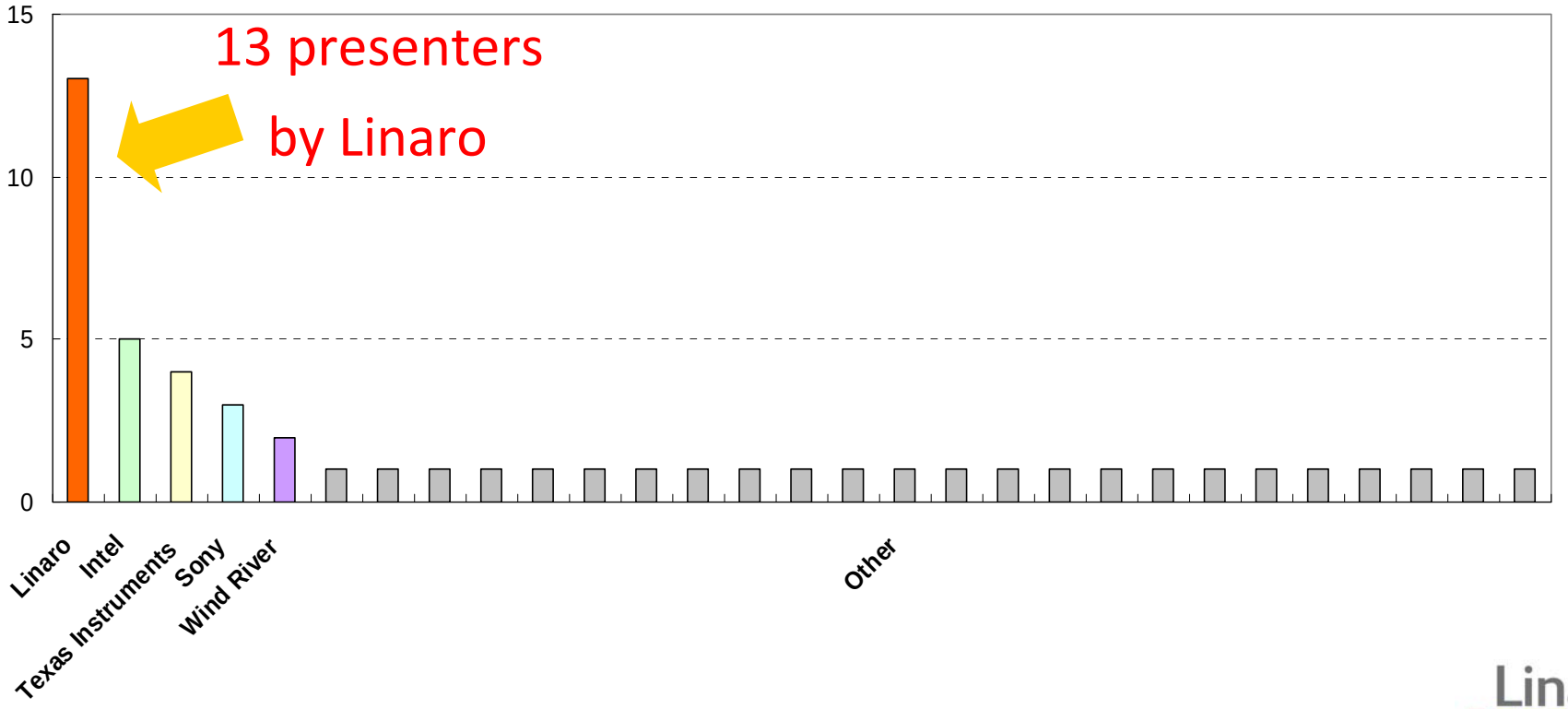
Also notable is the continued slow climb by companies like Texas Instruments and Samsung; Nokia, instead, appears to be about to fall out of the top 20. The handling of Linaro deserves an explanation: contributions by Linaro assignees is normally credited back to their home companies. Nonetheless, Linaro makes an appearance on its own here as the result of the work of an increasing number of engineers employed by the organization itself.

Source: LWN, Sept 28, 2011

Linaro

Strong Linaro leadership on Linux

- Embedded Linux Conference 2012 (Feb 15 – Feb 17)
- Linaro is dominating the share of all presenters

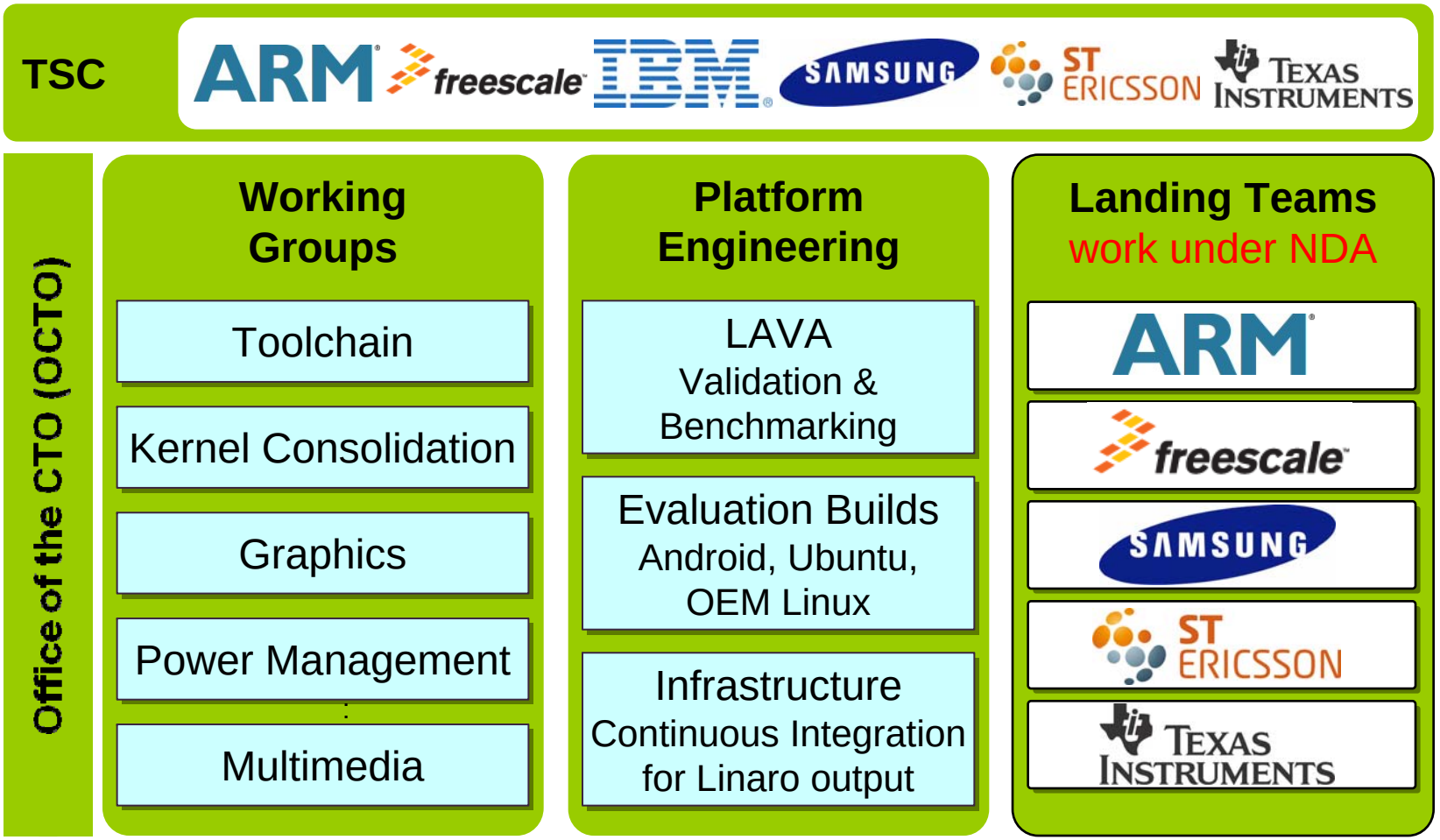


Inside Linaro

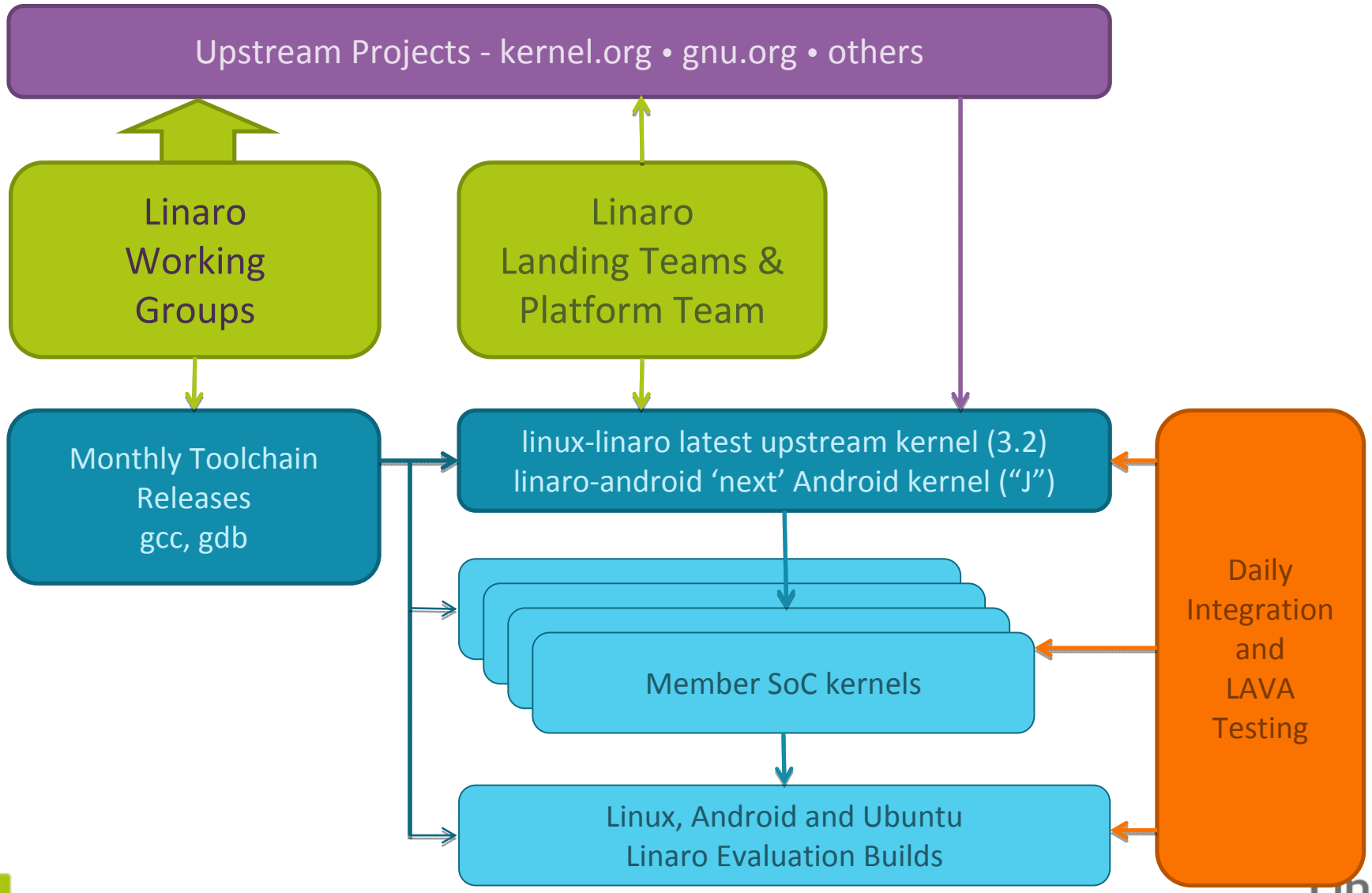
How Does Linaro Operate?

- Over 120 Linux kernel, middleware and toolchain engineers are located around the world
- The Linaro Technical Steering Committee (TSC)
  - Has 1 representative from each Linaro Member
  - The TSC determines the activities and priorities for the Linaro Working Groups through bi-weekly calls and quarterly face to face meetings at Linaro Connect
- Linaro Working Groups operate in the open
  - See every level of work item detail at status.linaro.org

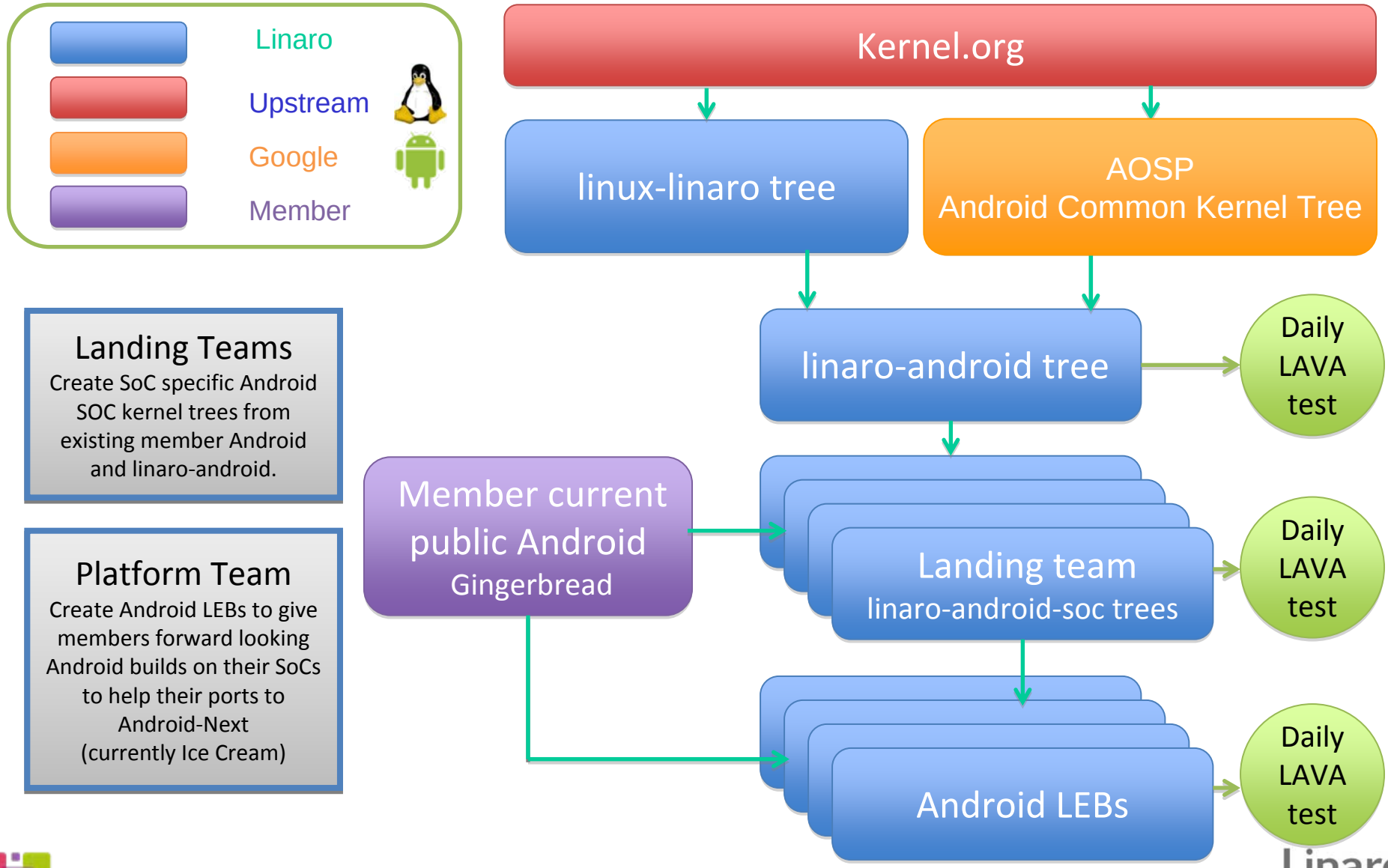
Linaro organization



Linaro Upstreaming and build



Linaro Android kernel build



Linaro Connect

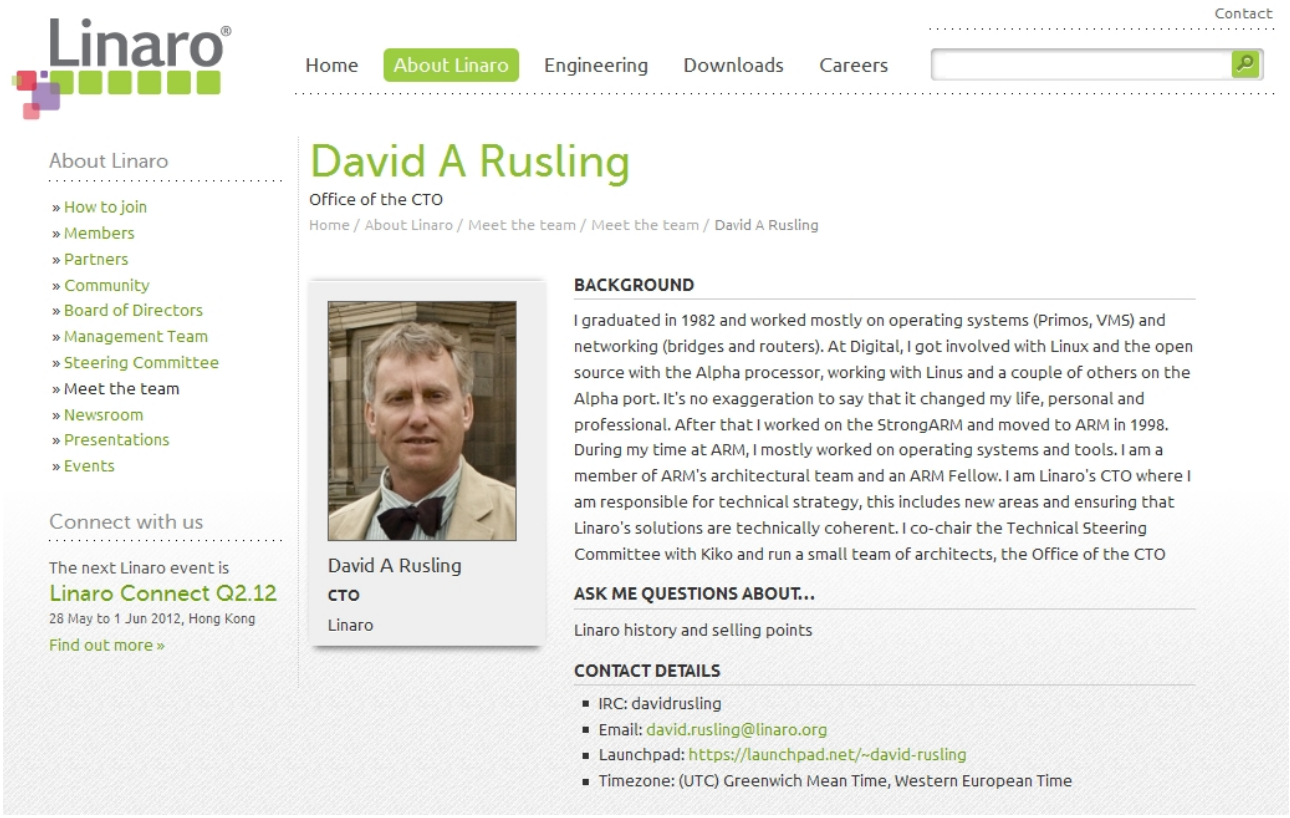
- Linaro Community Technical Conference held 3x per year
- Agree technical priorities & deliver on roadmap for ARM open source
  - Focus on member requirements
- **300 attendees from 50 companies attended in February**
- Future Connects will co-locate:
  - major industry conferences, to maximise external participation
  - or very near Member premises, to benefit specific Member involvement

Date	Location	Co-located Events
6-10 Feb 2012	San Francisco, California	ELC & Android Builders
28 May – 1 Jun	Hong Kong	
Q4 2012	Barcelona, Spain	UDS
Q1 2013	San Francisco	ELC

Linaro Wiki

- wiki.linaro.org
  - Example: contact infor of all the people of Linaro

http://www.linaro.org/about/meet-the-team



Globally distributed organization

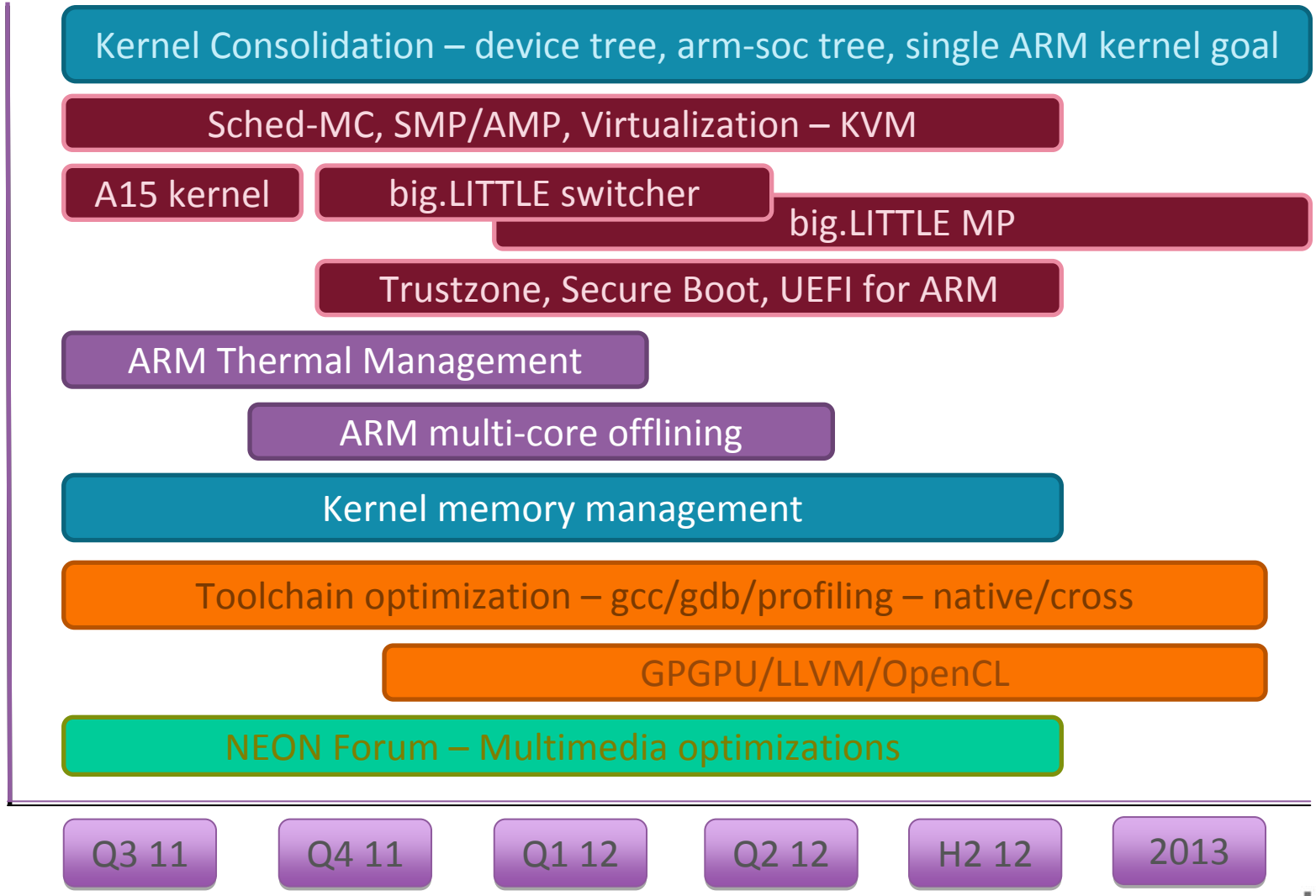
- People make up Linaro
  - ARM Management
  - Assignees:
    - ARM, Freescale, IBM, Samsung, ST-E, Texas Instrument
  - Canonical
  - Linaro
  - Partner Consultants
- People by Country (over 20 countries)
  - USA, India, China, UK, Sweden, Canada, France, Finland, Germany, Brazil, Russia, Australia, New Zealand, Poland, Korea, Greece, Japan, Pakistan, Switzerland, Spain, Serbia, Ukraine

Globally successful management

- Highly depended on systems over the Internet
- Communication
  - Mailing lists
  - IRC (Chart system)
  - Google Hangouts, Calendar
- Tracking Action Items
  - Blueprints
  - Launchpad
- Documentation
  - Wiki, Etherpad
- Individual performance evaluation
  - Web based goals and performance review

Linux projects led by Linaro

Linaro Roadmap - Partial



Device Tree on ARM

- Problems without device tree
  - ARM platforms rely on static list of platform devices for all non-discoverable devices
  - Each board has separate hardware configuration files under arch/arm
  - Causing fragmentation of the kernel
- Device-Tree
  - Device tree is a simple tree like data structure that can pass hardware configuration to the kernel from boot loader
  - Easier to add support for newer platforms
  - Reduces amount of board specific code

Unified Memory Management (UMM)

- Current issues
  - Memory management was implemented differently by every ARM-SoC vendors
  - Very difficult for porting drivers across between SoCs even for the same graphic chip
- dma_buf API:
  - A uniform mechanism to share DMA buffers across different devices and sub-systems
  - Provides uniform APIs that allow various operations related to buffer sharing
- Adopted by Android ION memory manager as well as vanilla kernel

Multiarch

- 3 major issues of faking autoconf causes and cross building
  - Installing build dependencies: native tools, cross libs/headers
  - Finding/linking libraries
  - Running build-time tools
- Multiarch

arch	GNU triplet	Multiarch path
amd64	x86_64-linux-gnu	/usr/lib/x86_64-linux-gnu
i386	i586-linux-gnu	/usr/lib/i386-linux-gnu
armel	arm-linux-gnueabi	/usr/lib/arm-linux-gnueabi

  - Cross-compilation is no longer special
  - Better support for binary-only software

Binary Blobs

- Current issues
  - If the audio blob is locked to the 3.0 kernel and the graphics blob is locked to 3.1, then dead end
  - hardfp and other toolchain improvements are not applicable because blobs are not able to rebuild by yourself
  - Unfortunately, binary blobs won't be going away soon
- To solve the problem
  - Create official binary box interfaces
  - Split devices from the kernel completely

Future Items: big.LITTLE, v8, servers

- Currently, platforms in Linaro are Cortex-A7/A8/A9/A15
  - big.LITTLE models in house
  - ARMv8 in planning
- Much still to do around consolidation
  - UMM upstreaming begun, plenty left to do in all the working groups
  - Building blocks for ARMv8
- Working on ARM server architecture for Linux
  - Hard Float, Grub2, UEFI, PXE, SMP, LAMP, LTS kernel etc.
  - Single ARM kernel zImage binary goal
- Involving community and Linux server distributions
  - Ubuntu, Fedora, Red Hat, Debian, OpenSUSE etc.
  - Server study in Office of CTO (OCTO) https://wiki.linaro.org/OfficeofCTO/Servers

ARM Server - Overview

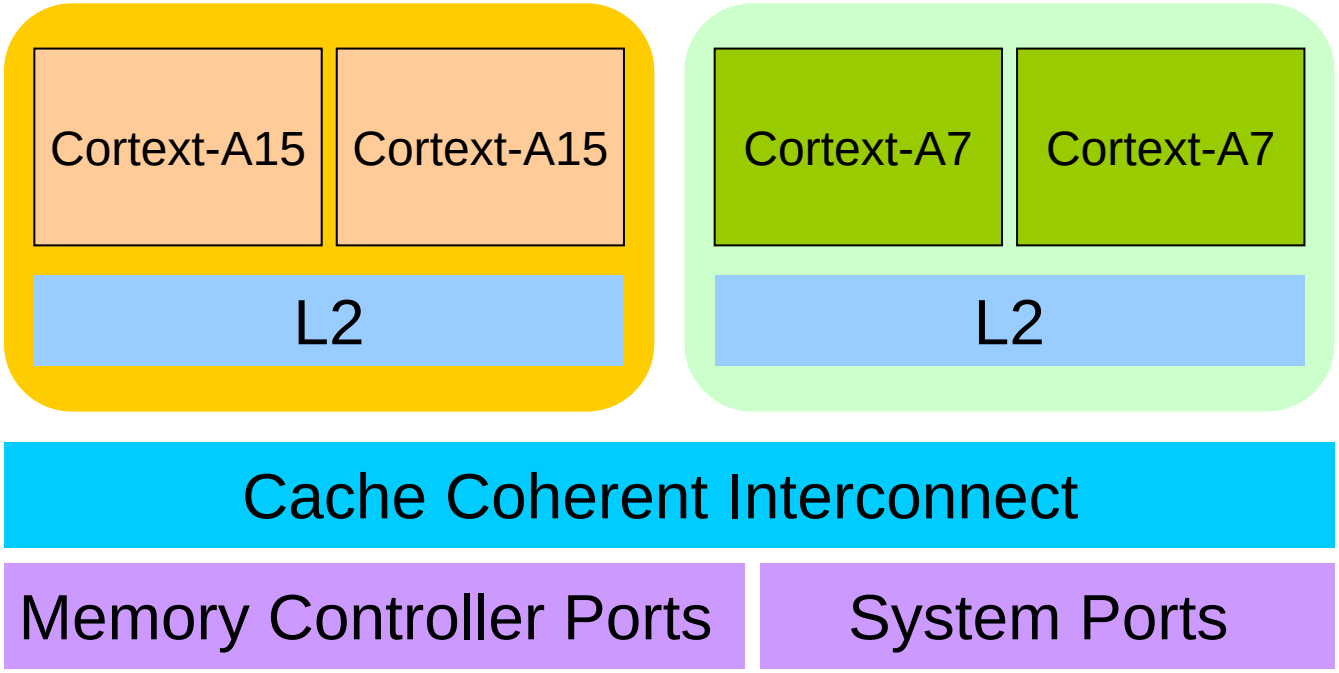
- 32 bit, ARMv7 architecture extensions
  - Large Physical Address Extensions (LPAE)
    - Page table format common with ARMv8
  - Virtualization extensions
  - A15 based systems ‘in the pipeline’
- 64 bit, ARMv8 architecture
  - Establish 64 bit arm-soc maintainer’s tree

ARM Server - Engineering

- Boot architecture
  - boot – security, standards (UEFI, ACPI)
  - Remote update
- Kernel
  - Virtualization (for reliability and security)
    - KVM plus commercial hypervisors
  - Neon acceleration (raid control etc)
- Distribution
  - Enterprise specific application footprint
  - Remote management
  - Single zImage

big.LITTLE overview

- ARM core including both Cortex-A15 and Cortex-A7
- Cortex-A15 and Cortex-A7 are software compatible



big.LITTLE support

- Stage 1: Cluster Switching (Task Migration)
  - Hypervisor resident code switches OS between A15 and A7 clusters
  - ARM's BSD proof of concept code is available
    (git://git.linaro.org/arm/big.LITTLE/switcher.git)
- Stage 2: Integrated Kernel Switching
  (Disclosed to Linaro member companies only from this stage)
  - Performance advantage over the hypervisor based solution
    - Task switching between A15 and A7 is done in kernel without hypervisor
  - Kernel switches between individual Cortex A7 and Cortex A15 pairs
  - Relies on existing kernel mechanisms, such as cpu_freq
- Stage 3: MP
  - Kernel executes on any or all cores using kernel scheduler
  - Asymmetric MP adds significant changes to the mainline kernel scheduler and power management code
  - Offers maximum configuration flexibility

Trying Linaro builds on evaluation boards

Pre-requisite

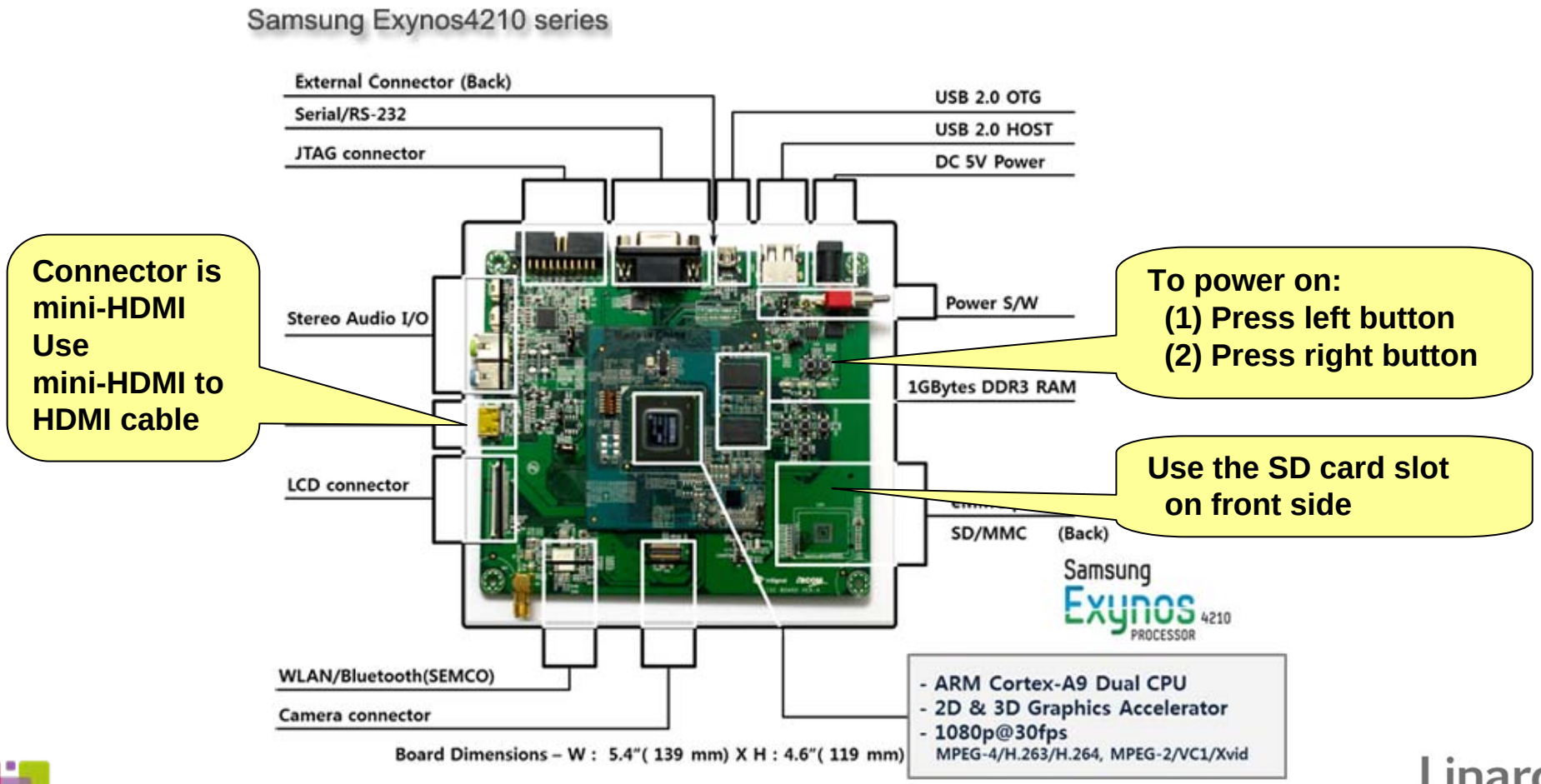
- SD card or micro-SD card
- HDMI display
- HDMI cable or mini-HDMI cable
  - Origen uses mini-HDMI connector
- Serial cable or mini-B USB cable
  - For serial console, Snowball has mini-B USB serial
- USB hub which does not use USB bus power
- USB keyboard
- USB mouse

Member Evaluation Boards

	Board Name	CPU	Memory	GPU
Samsung	Origen	Cortex A9 Dual	1GB DDR3	Mali-400
Texas Instrument	Panda board	Cortex A9 Dual	1GB DDR2	PowerVR SGX540
ST Ericsson	Snowball	Cortex A9 Dual	1GB DDR2	Mali 400
Freescale	Quick Start	Cortex A8 Single	1GB DDR3	PowerVR
	Cortex A9 board based on i.MX 6 series is coming…			

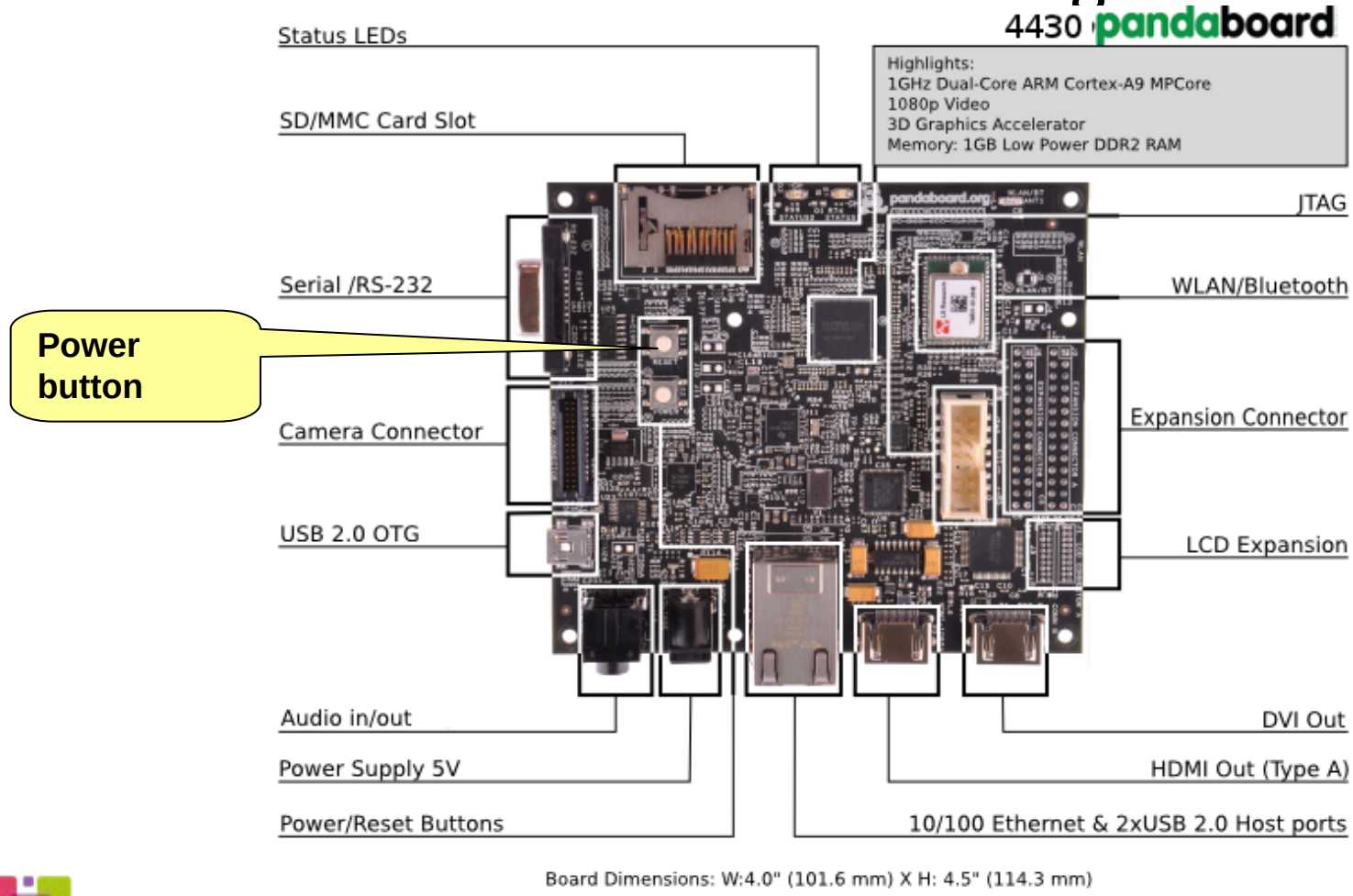
Origen board

- Uses SD card to boot the image



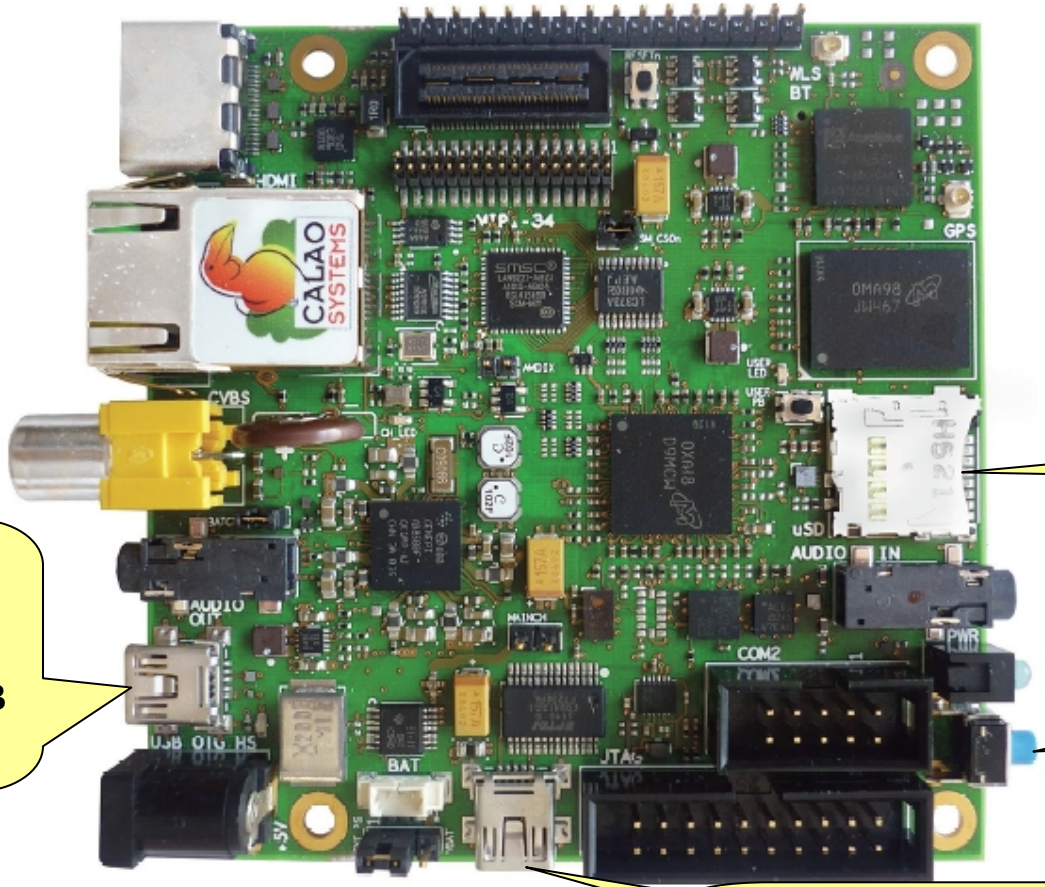
Panda board

- Uses SD card to boot the image



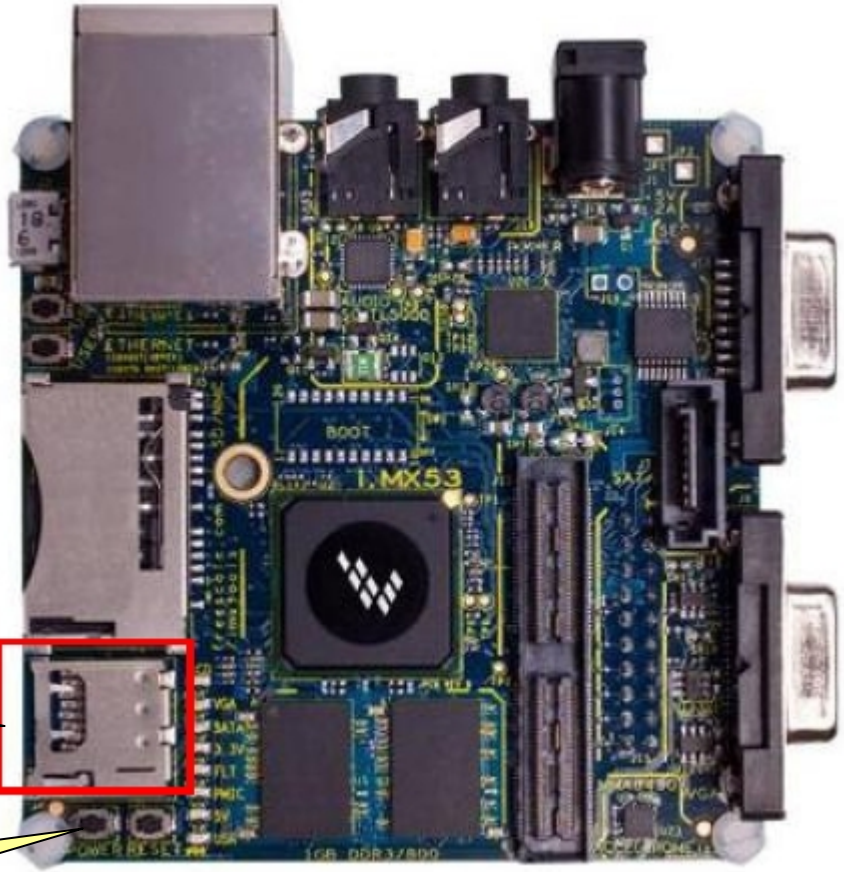
Snowball board

- Uses micro-SD card to boot the image



Quick Start board

- Uses micro-SD card to boot the image
- Recommended to buy HDMI Daughter Card (MCIMXHDMICARD)



serial port

micro-SD slot

Power button

Boot with pre-built image

- Download a image from:
  - http://www.linaro.org/downloads
- Image file name will be similar to bellow for the each board
  - origen-ics-gcc46-samsunglt-stable-blob.img.gz
  - panda-ics-gcc46-tilt-tracking-blob.img.gz
  - snowball-ics-gcc46-igloo-stable-blob.img.gz
- Write image to SD card
  - $gunzip -c <image-file> | sudo dd bs=64k of=</dev/mmcblk0 or /dev/sd*X*>
- Insert the SD card to the board and power on

Creating Android boot image (1/2)

- Download files from:
  - https://android-build.linaro.org/
- Download these files match your board:
  - boot.tar.bz2
  - userdata.tar.bz2
  - system.tar.bz2
- Get the latest linaro-image-tools:
  - $bzr branch lp:linaro-image-tools

- If you do not have ‘bzr’ in your ubuntu, then just:
  - $sudo apt-get install bzr

Creating Android boot image (2/2)

- Create boot image as following command line:
  - $./linaro-image-tools/linaro-android-media-create --mmc /dev/mmcblk0 --dev <board_type> --system system.tar.bz2 --userdata userdata.tar.bz2 --boot boot.tar.bz2
    - My PC has SD slot at /dev/mmcblk0 but change it if it is different for you like /dev/sd*X*
    - <board_type> could be
      - **origen, panda, snowball_sd, mx53loco** (Quick Start)
- Example of installing binary files (not all boards required)
  - wget http://people.linaro.org/~vishalbhoj/install-binaries-4.0.3.sh
  - chmod a+x install-binaries-4.0.3.sh
  - ./install-binaries-4.0.3.sh /dev/mmcblk0p2 <- system partition
- Insert the SD card to the board then boot it

Naming convention of build image

- e.g.

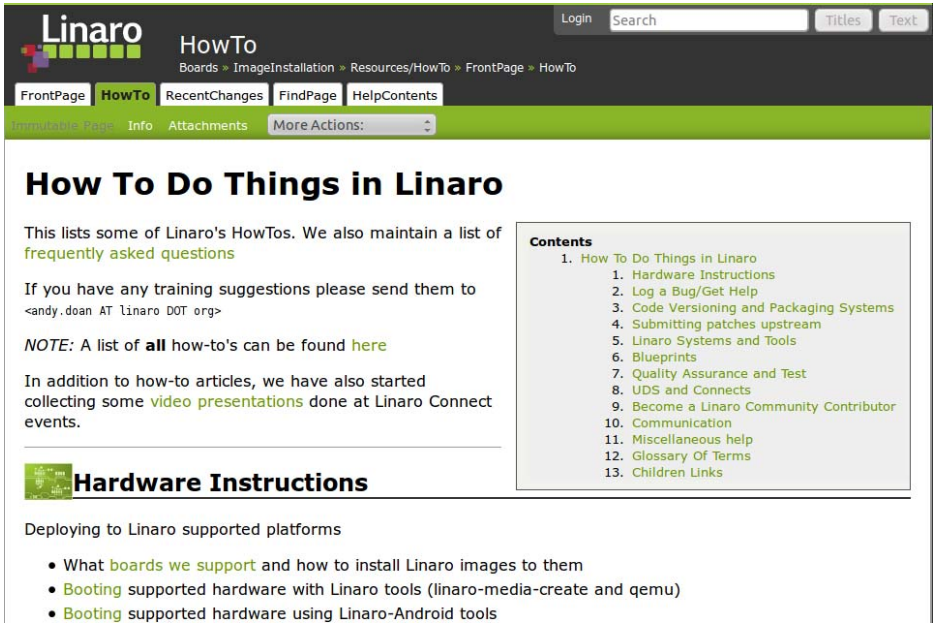
origen-ics-gcc46-samsunglt-stable-blob.img.gz

<target>-<android_ver>-<toolchain>-<kernel_origin>-<kernel_type>-<enablement_type>.img.gz

- Parameters
  - <kernel_origin>
    - 'lt' means landing team
    - kwg is kernel working group
    - aosp is from Google AOSP kernel
  - <enablement_type>
    - with closed binaries -> blob
    - without closed binaries -> open

HowTo Wiki (Technical information)

- wiki.linaro.org
  - https://wiki.linaro.org/HowTo



  - Use Google search
    - site:wiki.linaro.org “keyword”

Resources

- http://lists.linaro.org/mailman/listinfo
  - linaro-dev (greatest amount of internal involvement)
- irc.freenode.net
  - #linaro, #linaro-android (biggest public involvement)
- http://www.linaro.org/linaro-blog/ (social – videos & photos)
- http://planet.linaro.org/ (technical)
- http://connect.linaro.org/
- http://ask.linaro.org/ (technical questions)

www.linaro.org
wiki.linaro.org