



---

# Linaro

## Embedded Linux collaboration

An open source development consortium  
for ARM and the embedded community

[www.linaro.org](http://www.linaro.org)

Non-confidential





# Why Linaro?

- Our world is being transformed by Billions of Linux and open source based connected devices



- But developing open source consumer products is tough...
- Linaro was formed to tackle the 4 main problems of embedded Linux and make it easier & quicker to develop open source products



# The 4 problems

---

- 1. Under investment in the many open source projects that make up a Linux platform (Underinvestment problem)
- 2. Distribution fragmentation – different tools, versions, different graphics and multimedia plumbing, kernel forking & versions (Distribution fragmentation problem)
- 3. Lack of SoC integration e.g. kernel, power management, graphics and multimedia Different SoC vendors with different approaches (SoC fragmentation problem)
- 4. Not enough optimization. Lots of features in the latest processors not being used (Non-optimized problem)



# The vision

---

- . 1. Different connected devices built with different distributions on different SoCs all built with a common foundation of open software and tools
- . 2. Making it easier and quicker for device manufacturers to get products to market and easier for chip vendors to support multiple distributions
- . 3. Providing excellent performance – optimized tools and code making best use of latest processor features

Do this by...

- . Providing aligned investment upstream, work with SoC vendors and IP providers on unification and deliver the best Linux on ARM



# What is Linaro?

---

- A not-for-profit collaboration (started June '10)
  - Sponsored by ARM, IBM, TI, STE, Samsung, Freescale
  - <https://wiki.linaro.org/EngineeringTeam>
  - ~70 engineers
- Engineering in upstream projects, providing stable releases
  - A place for everyone to get the best open source tools and code
  - Reducing low level fragmentation
- Building a community – please get involved!

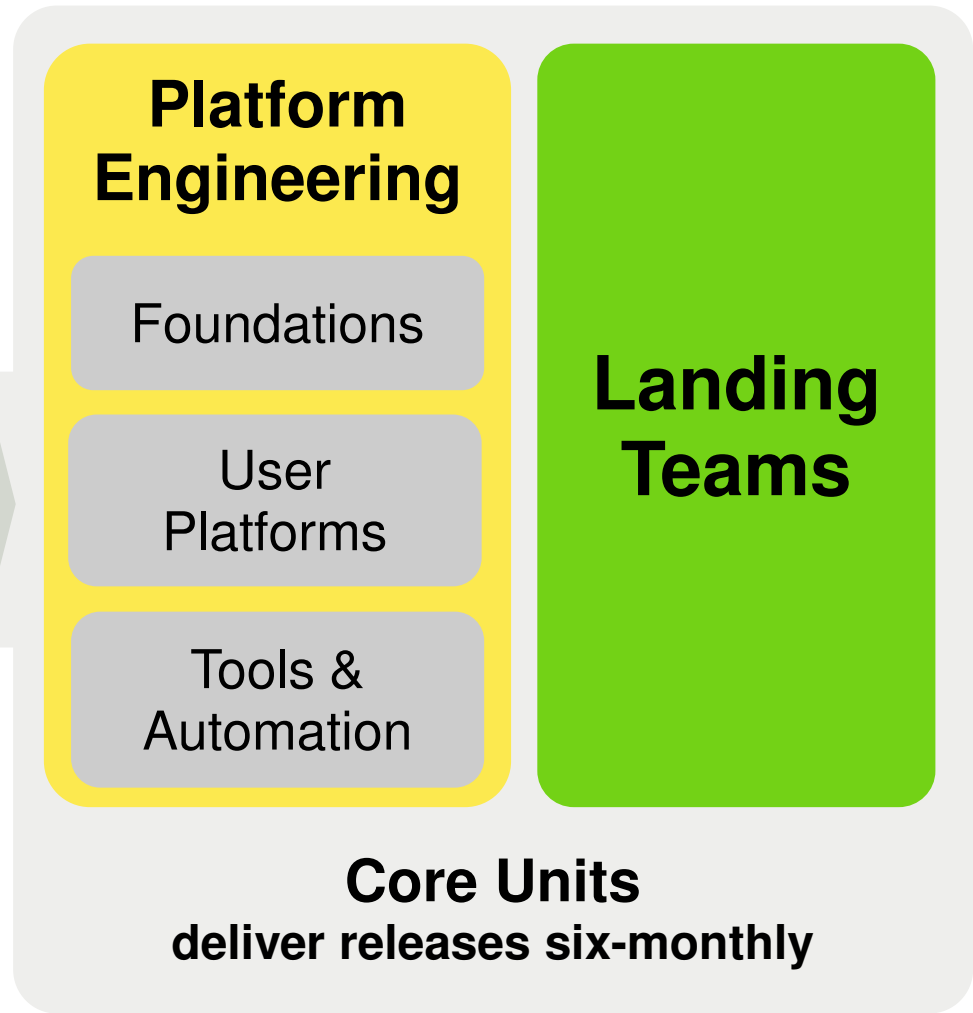
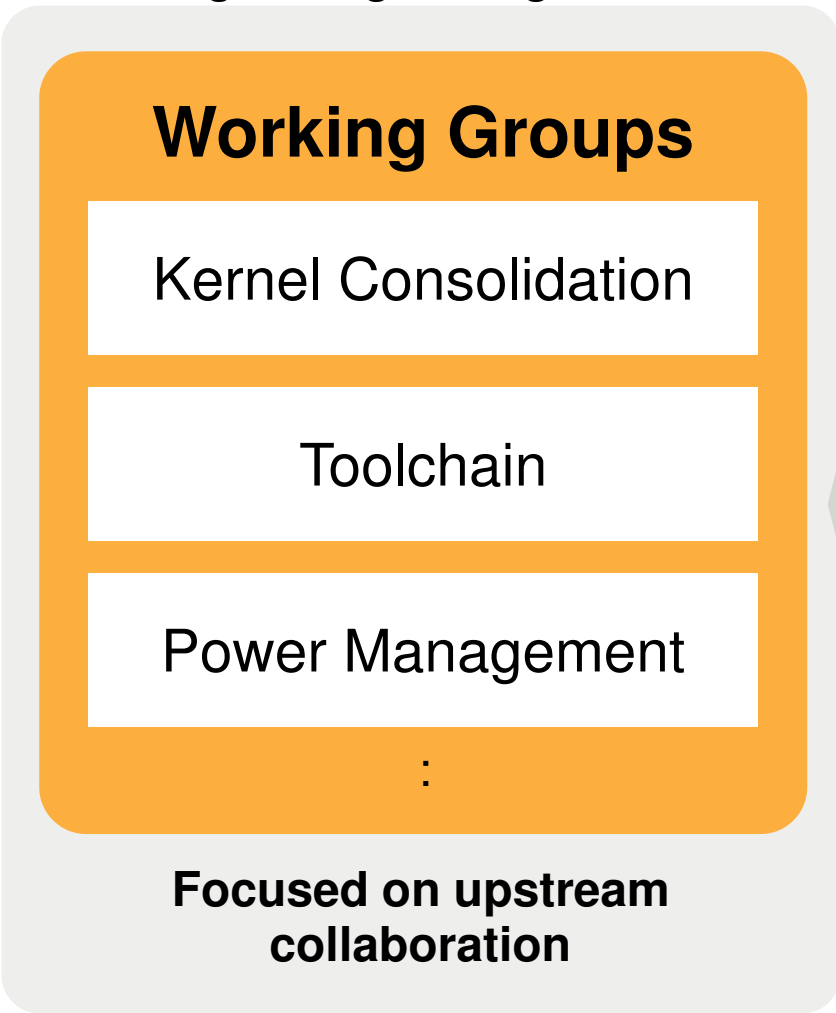


# Linaro Engineering Units

Aligned engineering

Access to best code & tools

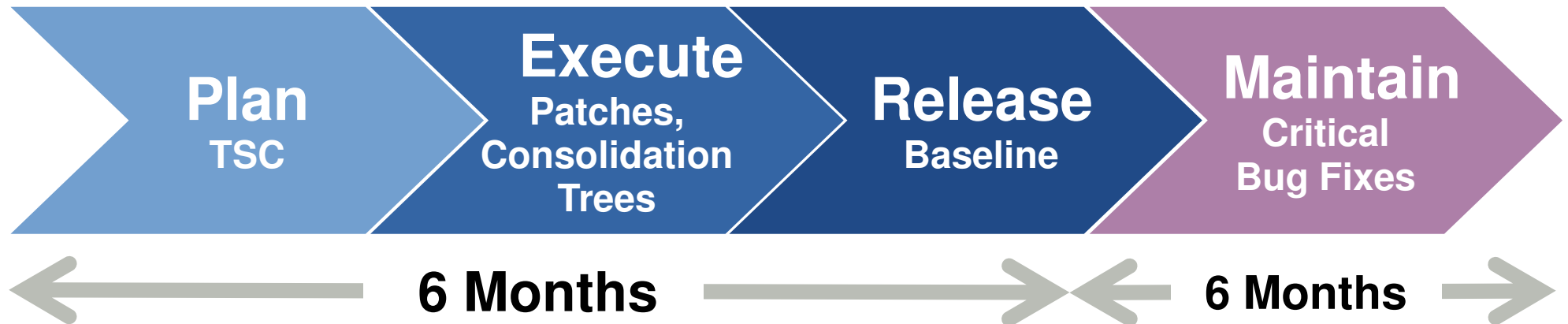
SoC Unification





# Linaro Development Cycle

---



- Release cadence of 6 months
- Planning is done with Technical Steering Committee
- Engineering starts at the end of the Developer summit



# 10.11 Themes (cycle 1)

---

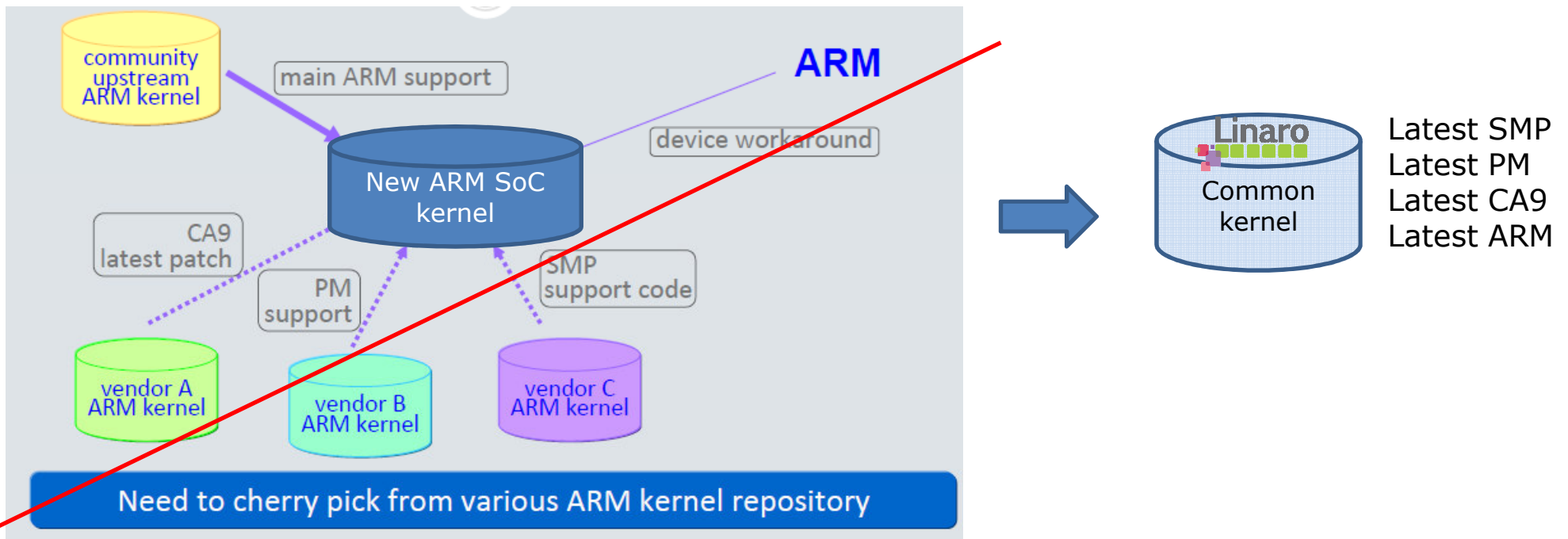
- Toolchain
  - Staffed up quickly
  - Back ported state of the art T2 tuning into 4.4.4 and 4.5
  - Fixed missing / broken profiling and debug features
- Kernel
  - Mostly consolidation work (kernel, uboot)
  - See Flattened Device Tree (FDT) as important
- Power Management
  - Consolidation work, initially



- Kernel baseline set at 2.6.35 and moves forward
- Consolidation work
  - Device tree work items underway
  - clocks infrastructure
  - uboot
- arm-next tree actively merging WG code
  - Tracking profiling, ftrace, architectural work
  - Includes hardware enablement patches en route to upstream
- Monthly releases

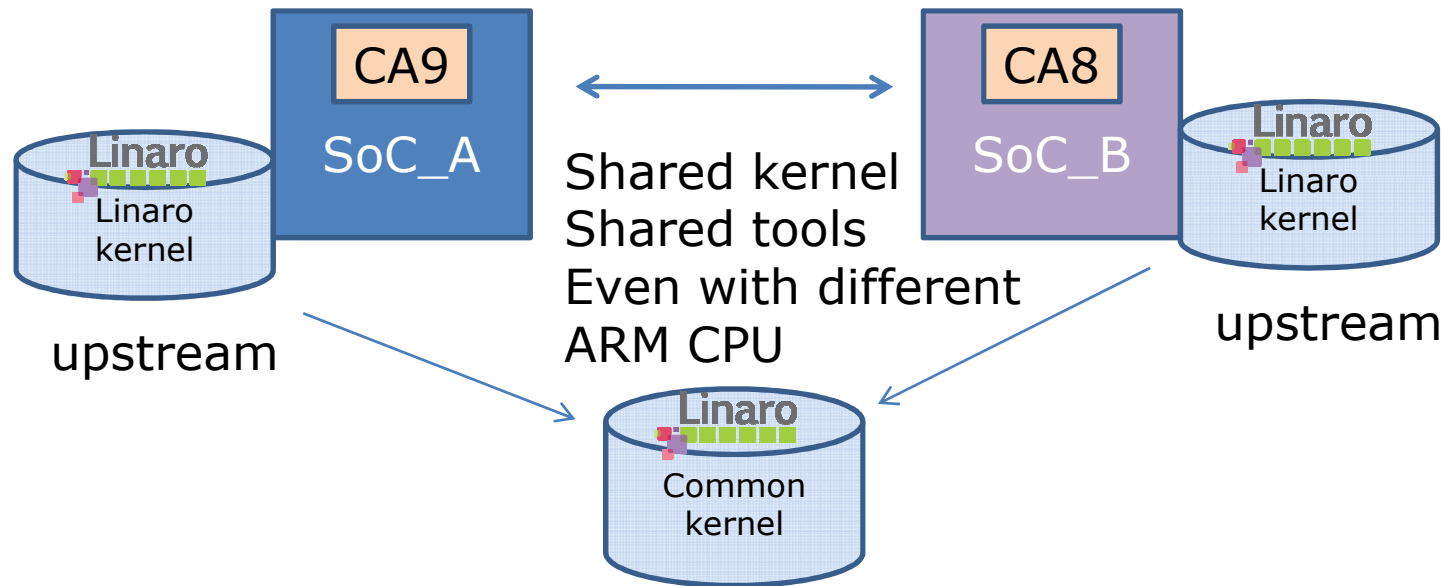
# Kernel consolidation

- No need to cherry pick anymore
- Get Latest features at – [www.linaro.org](http://www.linaro.org)



# User benefits from common kernel

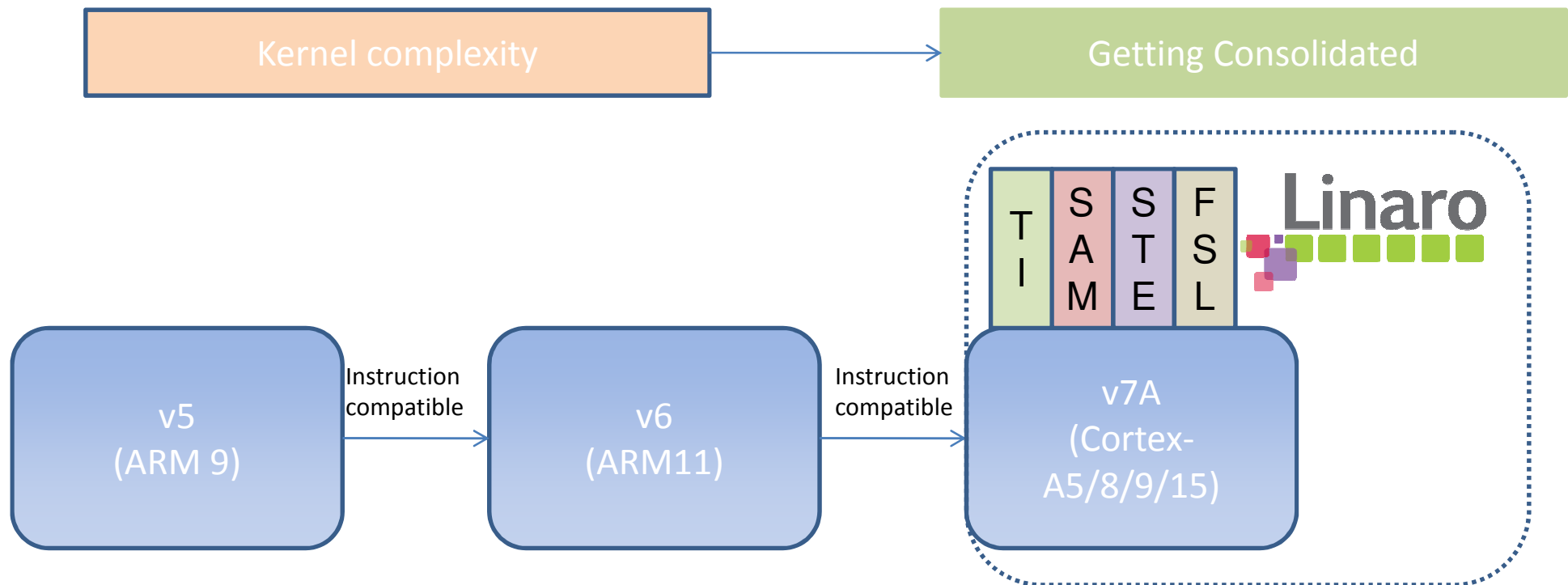
- Linaro works with silicon partners to upstream SoC support
- Easier to share kernel across devices



User can have common kernel experience across different SoC vendor and different ARM core e.g. Cortex-A8 or Cortex-A9

# Linaro focus is v7A (Cortex-A class)

- Linaro is helping global consistency for Linux kernel
- Goal is a single source tree that integrates support for multiple modern ARM SoCs
- v7A is backwards compatible with v6 & v5



Slide 12

Additional optimizations possible:  
Thumb-2, SMP, NEON





# Toolchain

---

- Scope is core tools plus visibility
  - Instrumentation trace, profiling etc.
  - Monthly releases
- Focus is on ARMv7A Thumb 2 and VFP / Neon
  - Code size and performance
- gcc-4.4.4 plus CS merge into initial release
  - 10% performance and code size improvement
  - All patches analyzed and submitted upstream
  - Test rebuild of Ubuntu main (1200+ packages)
- gcc-4.5 consolidation branch
  - ~5% better performance than stock 4.5 (and improving)



# Power Management

---

- Initial focus: basic power management support
- Consolidating across several platforms
  - \_Cpufreq
  - \_Cpuidle
  - \_Common clock API
- Tools
  - \_PowerTop



# Platform Engineering

---

- Consolidation
  - Profiling and debug
  - Uboot
  - Multiarch
- Build system
- Footprint reduction
- Memory profiling tool analysis and packaging underway
- Platforms
  - Standalone minimal head (compiler, debugger, profiling)
  - Cross compilation



# 11.05 Outline (cycle 2)

---

- We have added two new working groups
  - Graphics
  - Multimedia
- Continue to consolidate where needed
- Support the whole system
- Consolidated kernel support for latest SoCs
- Start to take a leadership position
  - SMP
  - Power management





# Cycle 2: Toolchain

---

- Best in class development tools
  - Compiler target 10% T2 code speed and size per release
  - Debug and visibility
  - Extend tools support beyond core tools
    - Ltrace, ftrace, LTTng, SystemTap, OpenOCD, Valgrind, Qemu
  - Self hosting and cross development tools



# Cycle 2: Power Management

---

- Continue restructuring and consolidating power management support
- Power analysis and tuning tools
  - PowerDebug
- Create a leadership position in energy management
  - Investigating the best power management framework for ARM Linux systems



# Cycle 2: Graphics and Multimedia

---

- Graphics
  - Consolidating and restructuring the 'plumbing'
  - OpenGL ES 2.0 graphics a priority
  - Run time switchable use of ARM, Neon / VFP
  - Rendering optimizations
- Multimedia
  - Consolidation
  - OpenMax and gstreamer work
  - Tune prioritized list of codecs



# Cycle 2: Distributions

---

- More distributions will take the toolchain
  - Linaro acts as an upstream open source project
- Some will start to consider the consolidated kernel tree
  - Although, may be better to take from upstream (depends on kernel features needed)
- Are helping to direct upstream graphics and multimedia work



# Conclusions

---

- We are providing aligned investment for embedded Linux in upstream projects
  - Continuous improvement of code and tools
- Reducing fragmentation e.g. kernel and graphics consolidation
- Making it easy to get the latest code & tools
- Acting as a Focal point for embedded Linux community
- Providing technical leadership
- Aligned with ARM partnership's engineering
- Encouraging Community involvement

# QUESTIONS?

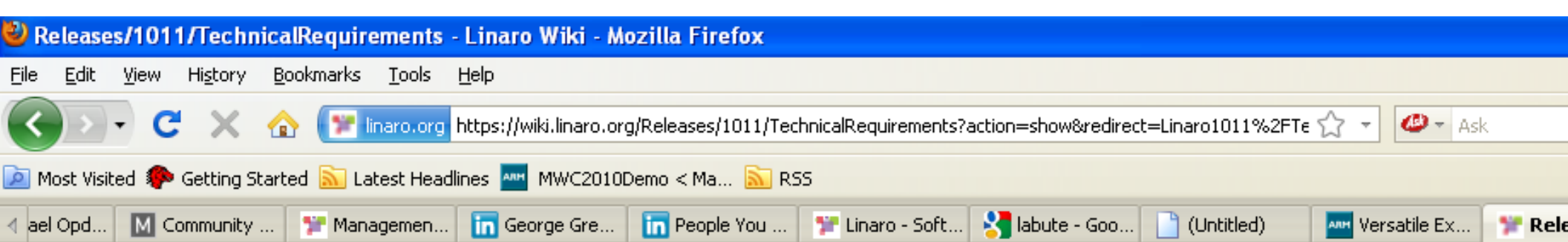
If you want to download...  
[www.linaro.org](http://www.linaro.org)

If you want to get involved...  
[www.linaro.org/community](http://www.linaro.org/community)

If you want to see the engineering...  
[www.linaro.org](http://www.linaro.org)

Slide 22

The screenshot shows the Linaro website homepage. At the top is the Linaro logo and navigation links for 'About Linaro', 'Community', and 'Contact Us'. Below the navigation are three main sections: 'Tools and software' (with a download icon and a hard drive image), 'Community' (with a network icon), and 'Open engineering' (with a monitor icon). A large image of a circuit board is at the bottom, with a 'Software Developer?' button. A green box at the bottom right contains the text: 'Linaro - Software foundations for an open digital world' and 'Linaro™ brings together the open source community and the electronics industry to work on key projects, deliver great tools, reduce industry wide fragmentation and provide common foundations for Linux software distributions and stacks to land on.' Three green arrows point from the text on the left to the 'Tools and software', 'Community', and 'Open engineering' sections respectively.



## Kernel Consolidation

Linaro has an overall goal of the Linux kernel tree at kernel.org [8] having the latest support for all ARM platforms. A fully generic kernel that could run on several, highly variant, ARM platforms is also a goal, but we recognise that this may take some time. In order for this to happen, and to help upstream ARM platform support, there needs to be a degree of kernel consolidation work. Essentially this means moving platform and device specific code out of both the ARM and generic infrastructure areas. In some cases, this may mean adding infrastructure to allow this to happen.

Table 1: Kernel Consolidation Requirements

Ref	Agreed	Priority	Requirement
C1	A	Essential	ARMv7A standard configuration
C2	A	Essential	Single ARM kernel source tree for all Linaro supported platforms. There needs to be some initial investigation (experiment) looking at the road blocks for consolidated kernels, this may throw up some more work. C3-C5 are the basic building blocks. See <a href="https://blueprints.edge.launchpad.net/ubuntu/+spec/arm-m-kernel-version-alignment">https://blueprints.edge.launchpad.net/ubuntu/+spec/arm-m-kernel-version-alignment</a>
C3	A	Essential	Support for device trees. We may need to align device tree format between UEFI/SFI. See <a href="https://blueprints.edge.launchpad.net/ubuntu/+spec/arm-m-using-device-tree-on-arm">https://blueprints.edge.launchpad.net/ubuntu/+spec/arm-m-using-device-tree-on-arm</a> and <a href="https://blueprints.edge.launchpad.net/ubuntu/+spec/arm-m-alsa-soc-f">https://blueprints.edge.launchpad.net/ubuntu/+spec/arm-m-alsa-soc-f</a>
C4	A	Essential	Support for pluggable timers and interrupts. The clock API looks useful here. <a href="https://blueprints.edge.launchpad.net/ubuntu/+spec/arm-m-using-device-tree-on-arm">https://blueprints.edge.launchpad.net/ubuntu/+spec/arm-m-using-device-tree-on-arm</a>

