Linaro

Embedded Linux collaboration

An open source development consortium for ARM and the embedded community

www.linaro.org

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 four of the 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 forks & versions (Distribution fragmentation problem)

3. Lack of efficient SoC integration e.g. power management, graphics and multimedia. Most SoC vendors have different approaches (SoC fragmentation problem)

4. Not enough optimization. Lots of features in the latest processors not being used (Non-optimized problem)

Linaro

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

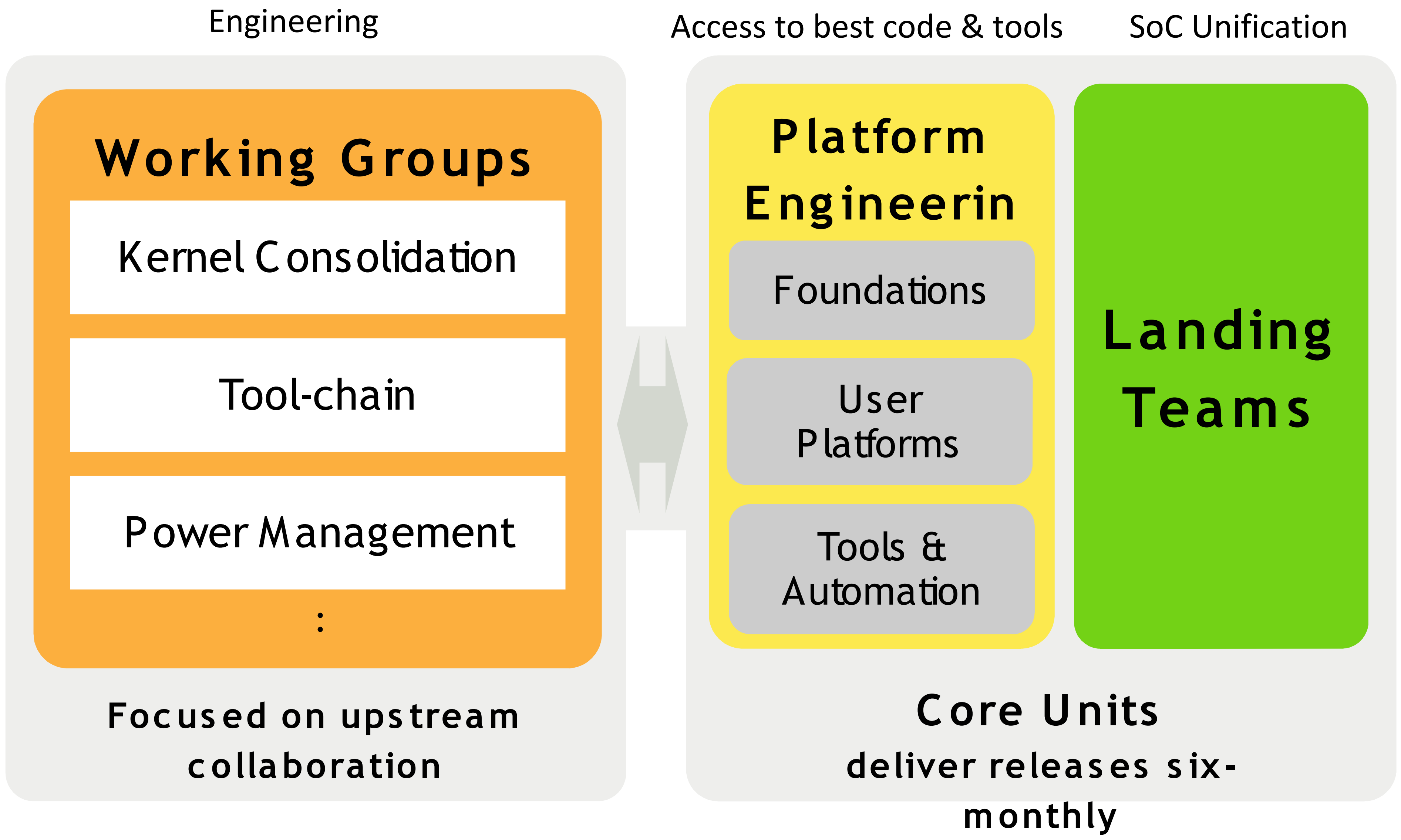
Do this by...

Providing aligned investment upstream, work with SoC vendors and IP providers to align and deliver the best Linux on ARM

What is Linaro?

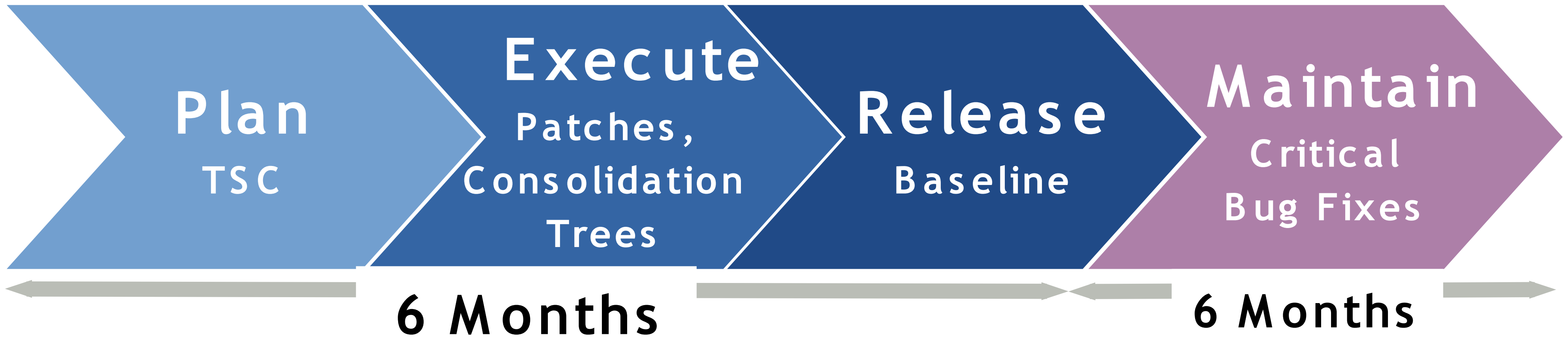
- A not-for-profit collaboration (started June ‘10)
  - Sponsored by ARM, IBM, TI, STE, Samsung, Freescale
  - https://wiki.linaro.org/EngineeringTeam
  - ~70 engineers
- Engineering in upstream projects, providing stable releases
  - A place for everyone to get optimized open source tools and code
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

Linaro Engineering Groups



Linaro

Linaro Development Cycle



- Release cadence of 6 months
- Planning driven by Technical Steering Committee
- Engineering starts at the end of the Developer summit

10.11 Themes (cycle 1)

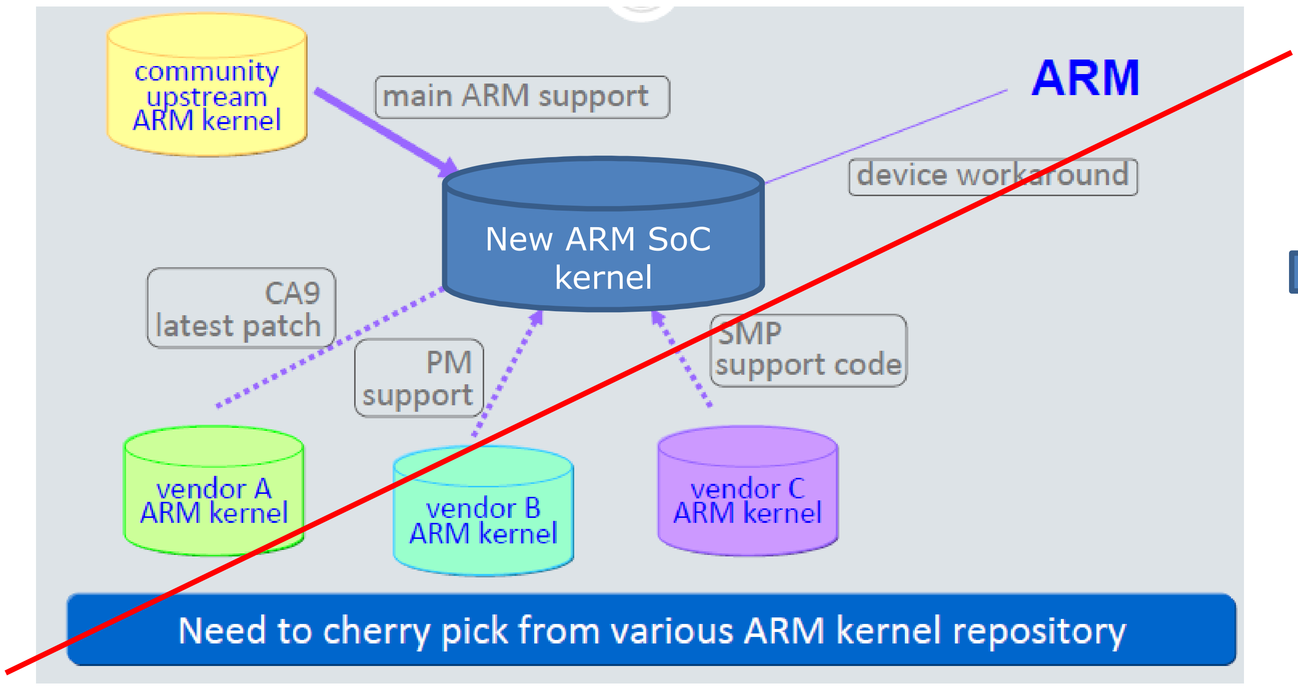
- GCC Tool-chain
  - Quick staff ramp-up
  - Back ported state of the art Thumb-2 tuning into 4.4.4 and 4.5
  - Fixed missing /broken profiling and debug features
- Linux Kernel
  - Mostly consolidation work (kernel, U-Boot)
  - See Flattened Device Tree (FDT) as important
- Power Management
  - Consolidation work, initially

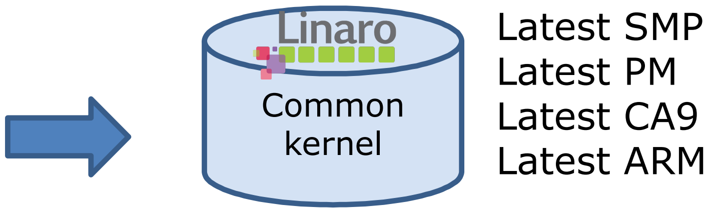
Linux Kernel

- Kernel baseline set at 2.6.35 and moves forward
- Consolidation
  - Developing Flat Device Tree for multiple SoCs
  - Clocks infrastructure
  - U-Boot
- <arm-next> source tree used to actively merging WG code
  - Tracking profiling, ftrace, architectural work
  - Includes hardware enablement patches
  - Consolidation for upstream contribution
- Monthly releases

Kernel consolidation

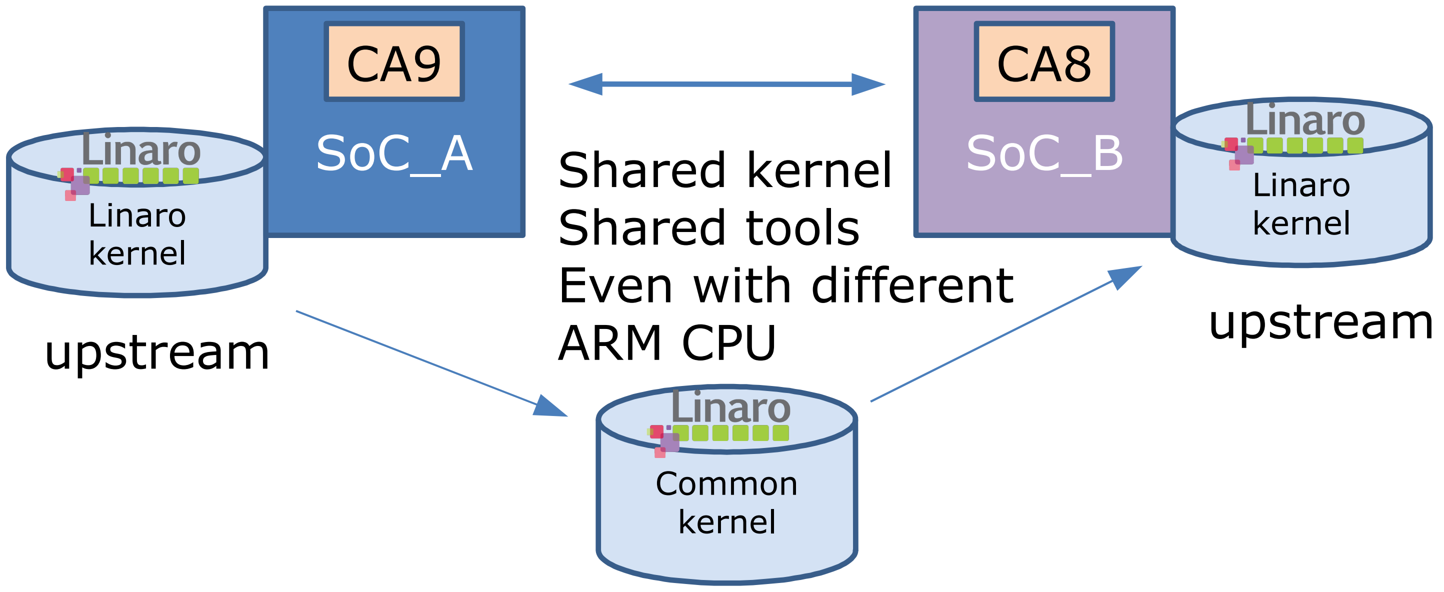
- No need to cherry pick anymore between multiple trees
- Get Latest features at – www.linaro.org





User benefits from common kernel

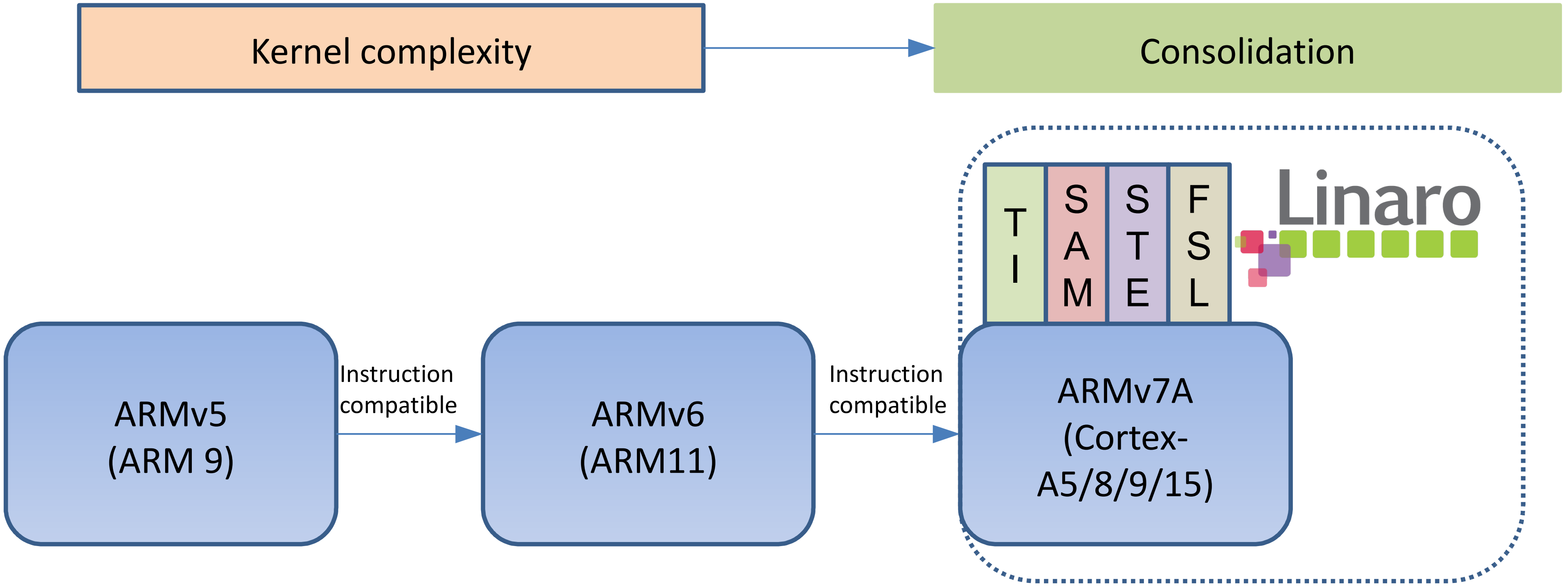
- Linaro works with silicon partners to upstream SoC support
- Make it easier to use single kernel across multiple devices



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
- ARMv7A is backwards compatible with ARMv6 & ARMv5



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 Codesourcery 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

Linaro

Platform Engineering

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

11.05 Outline (cycle 2)

- Recently added two new working groups
    - Graphics
    - Multimedia
- Continue to consolidate where needed
    - Looking at the platform as a whole
- Consolidated kernel support for latest SoCs
- Start to take a leadership position
    - SMP
    - Power management

Cycle 2: Toolchain

- Aiming for best-in-class development tools
  - Compiler targeting 10% incremental improvement for Thumb-2 code speed and size
  - Debug and visibility
  - Extend tools support beyond core tools
    - Ltrace, ftrace, LTTng, SystemTap, OpenOCD, Valgrind, Qemu
  - Include tools for self-hosting and cross-development

Cycle 2: Power Management

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

Cycle 2: Graphics and Multimedia

- Graphics
  - Consolidating and restructuring the 'plumbing'
  - OpenGLES 2.0 graphics a priority
  - Run time switchable use of ARM, Neon /VFP
  - Rendering optimizations
- Multimedia
  - Consolidation
  - OpenMax and GStreamer work
  - Optimize tageted codecs

Cycle 2: Distributions

- Enable more distributions to use GNU tool-chain
    - Benefit from consolidation and testing done in Linaro
- Consolidated Linux kernel tree may provide a quicker route for some developers or companies
    - Still best if everything is upstream! (depends on kernel features needed)
- Help to direct upstream graphics and multimedia work

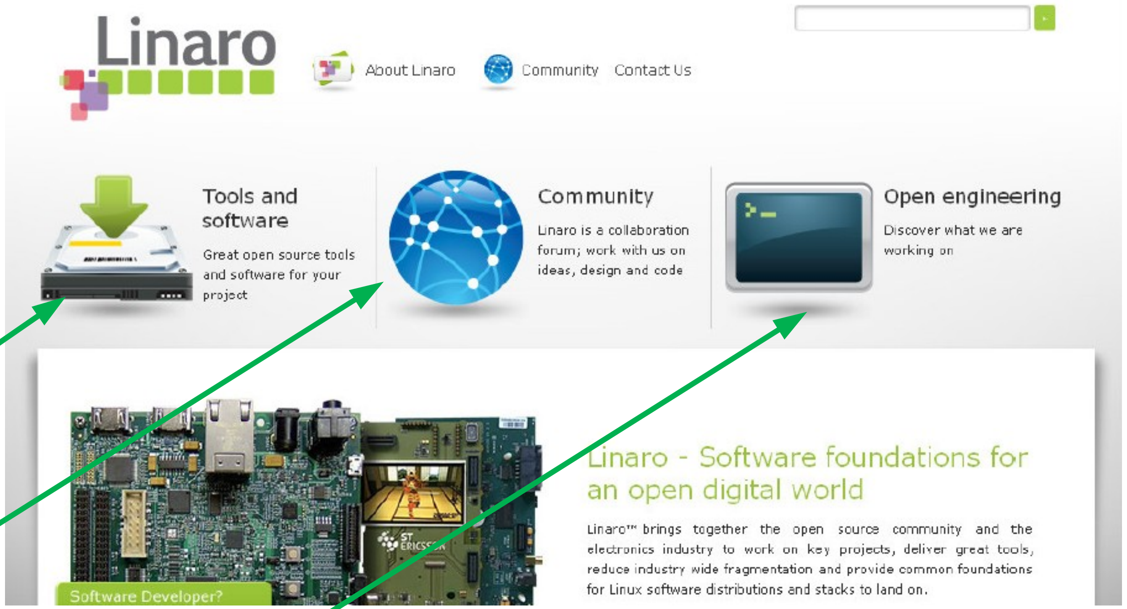
Getting Involved

- Everyone is welcome
- Register your project ideas on Launchpad
- Everything is open
  - wiki, specifications, discussion reports, developer summit.
- Community manager - Michael Opdenacker
  - E-mail: community-manager@linaro.org
- Mailing list: coming soon.
- Forums: http://linaro.org/forums/ (coming soon)

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?


Linaro
About Linaro
Community
Contact Us
Tools and software
Great open source tools and software for your project
Community
Linaro is a collaboration forum; work with us on ideas, design and code
Open engineering
Discover what we are working on
Software Developer?

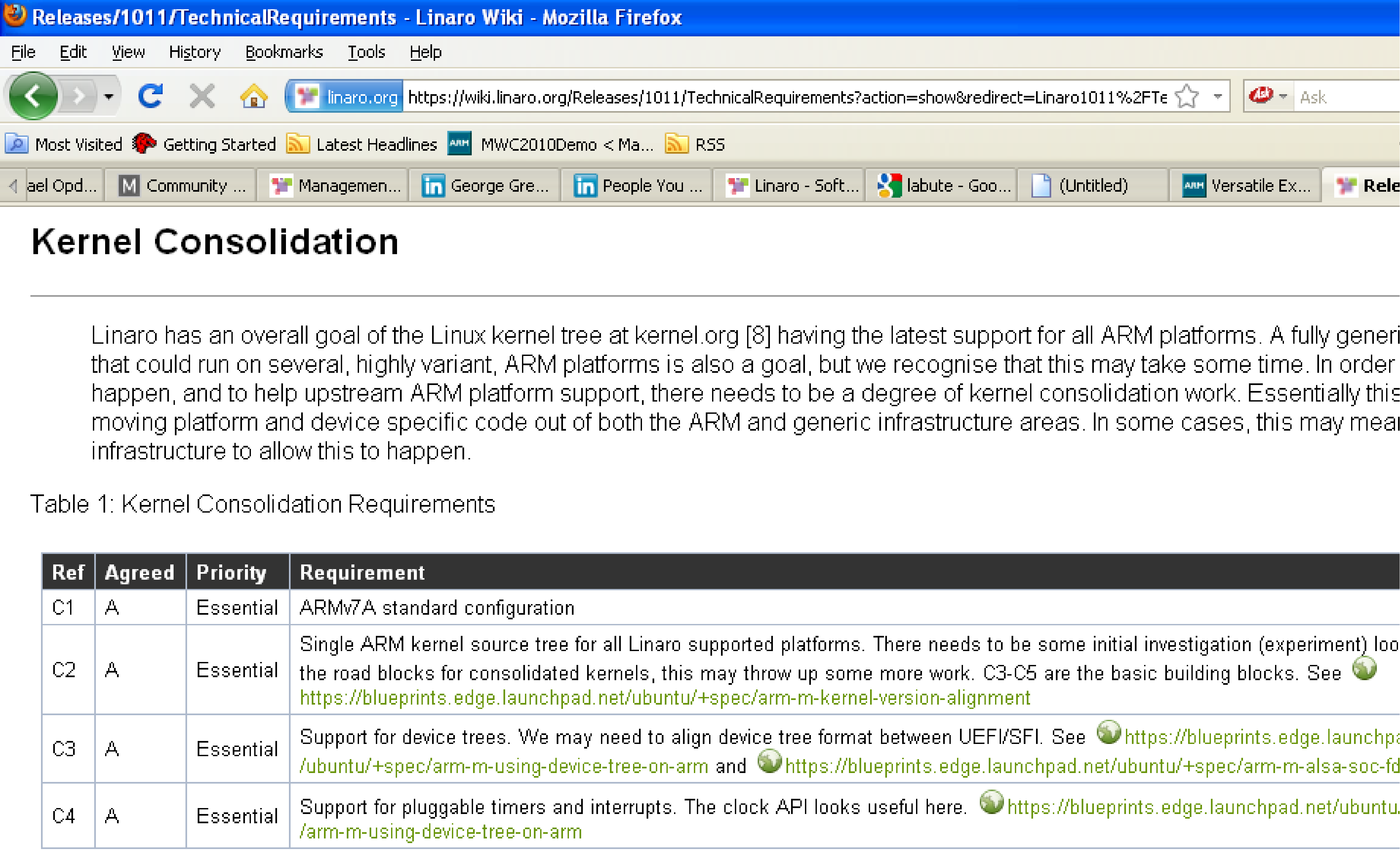
Linaro - Software foundations for an open digital world

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.

If you want to download...
www.linaro.org

If you want to get involved...
www.linaro.org/community

If you want to see the engineering...
www.linaro.org


Releases/1011/TechnicalRequirements - Linaro Wiki - Mozilla Firefox
File Edit View History Bookmarks Tools Help
linaro.org https://wiki.linaro.org/Releases/1011/TechnicalRequirements?action=show&redirect=Linaro1011%2FTe
Ask
Most Visited Getting Started Latest Headlines MWC2010Demo < Ma... RSS
ael Opd... Community ... Managemen... George Gre... People You ... Linaro - Soft... labute - Goo... (Untitled) Versatile Ex... Rele

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 generi
that could run on several, highly variant, ARM platforms is also a goal, but we recognise that this may take some time. In order
happen, and to help upstream ARM platform support, there needs to be a degree of kernel consolidation work. Essentially this
moving platform and device specific code out of both the ARM and generic infrastructure areas. In some cases, this may mear
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) loo the road blocks for consolidated kernels, this may throw up some more work. C3-C5 are the basic building blocks. See https://blueprints.edge.launchpad.net/ubuntu/+spec/arm-m-kernel-version-alignment
C3	A	Essential	Support for device trees. We may need to align device tree format between UEFI/SFI. See https://blueprints.edge.launchpa /ubuntu/+spec/arm-m-using-device-tree-on-arm and https://blueprints.edge.launchpad.net/ubuntu/+spec/arm-m-alsa-soc-fd
C4	A	Essential	Support for pluggable timers and interrupts. The clock API looks useful here. https://blueprints.edge.launchpad.net/ubuntu /arm-m-using-device-tree-on-arm

Linaro