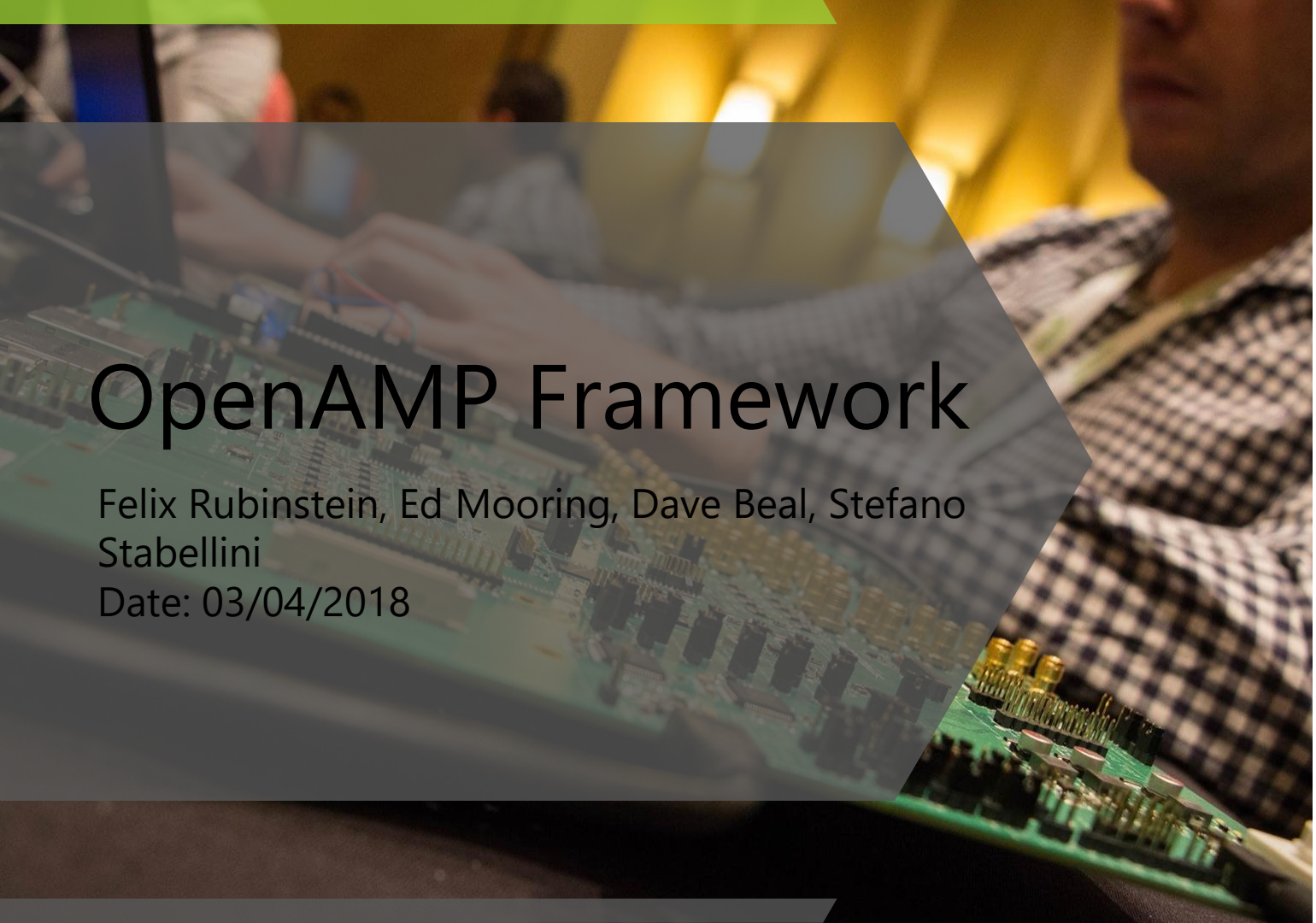


OpenAMP Framework

Felix Rubinstein, Ed Mooring, Dave Beal, Stefano Stabellini

Date: 03/04/2018

Linaro connect
Bangkok 2019

Executive Summary

- OpenAMP an open-source framework to interact with heterogeneous SoCs
  - Facilitates use of processing resources for complex designs
- Standardization effort and open-source project
  - Multicore Association (MCA) OpenAMP working group
  - Linaro LITE open-source project
- Evolving AMP/OpenAMP Roll-out
  - From foundation to advanced capabilities
    - APU as master
    - RPU as master
    - Authentication, Decryption of executables
    - Multiple memory types and coherency, zero copy, etc.
  - Arbitrary executable management
  - OpenAMP executable management

Agenda

- Glossary
- SMP vs. AMP
- OpenAMP Goals
- Life-Cycle Management
- Inter-Process Communication
- Startup Process
- Vendor Support
- Status
- Conclusion

Agenda

- Glossary
- SMP vs. AMP
- OpenAMP Goals
- Life-Cycle Management
- Inter-Process Communication
- Startup Process
- Vendor Support
- Status
- Conclusion

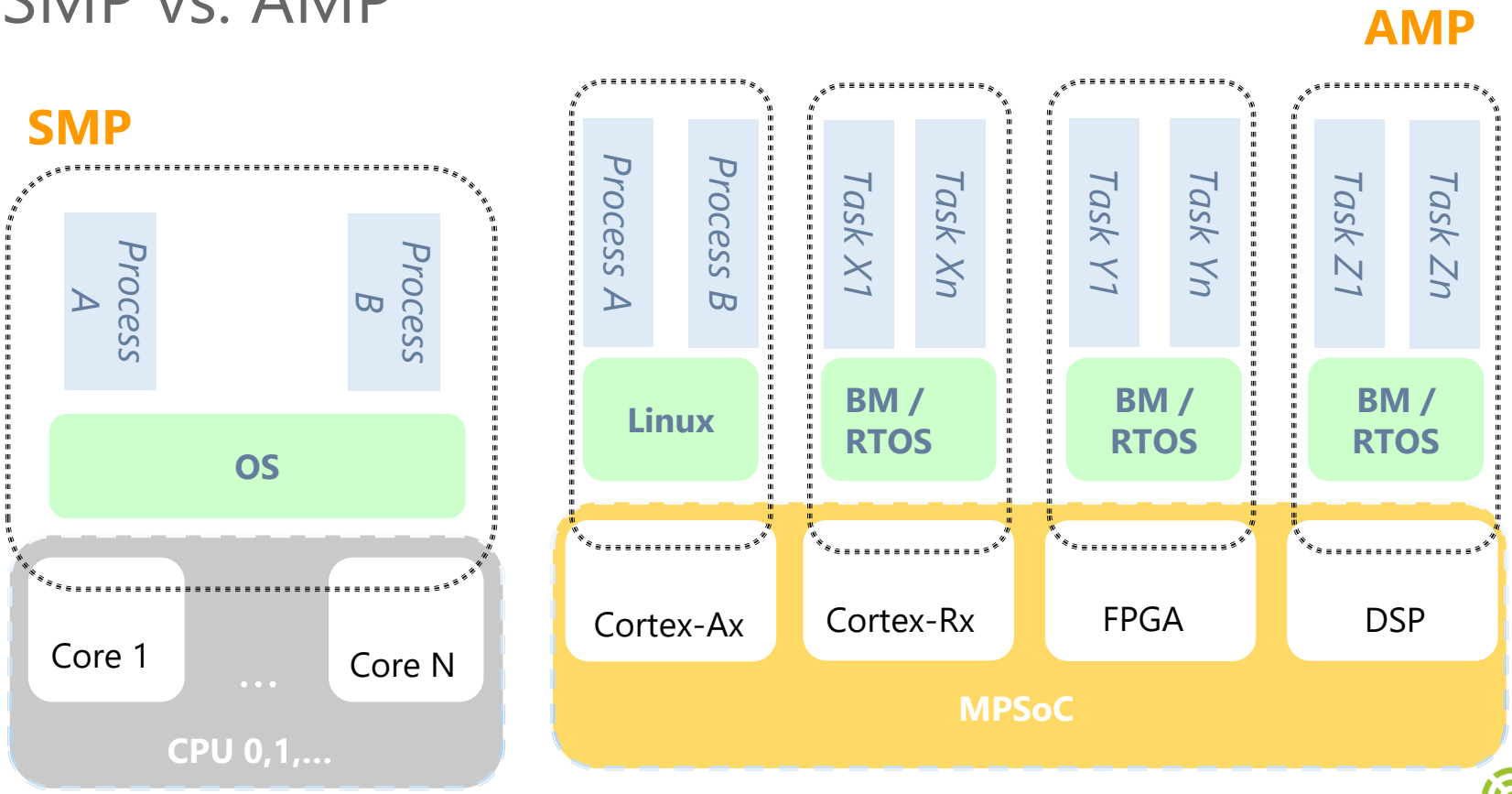
Agenda

- Glossary
- SMP vs. AMP
- OpenAMP Goals
- Life-Cycle Management
- Inter-Process Communication
- Startup Process
- Vendor Support
- Status
- Conclusion

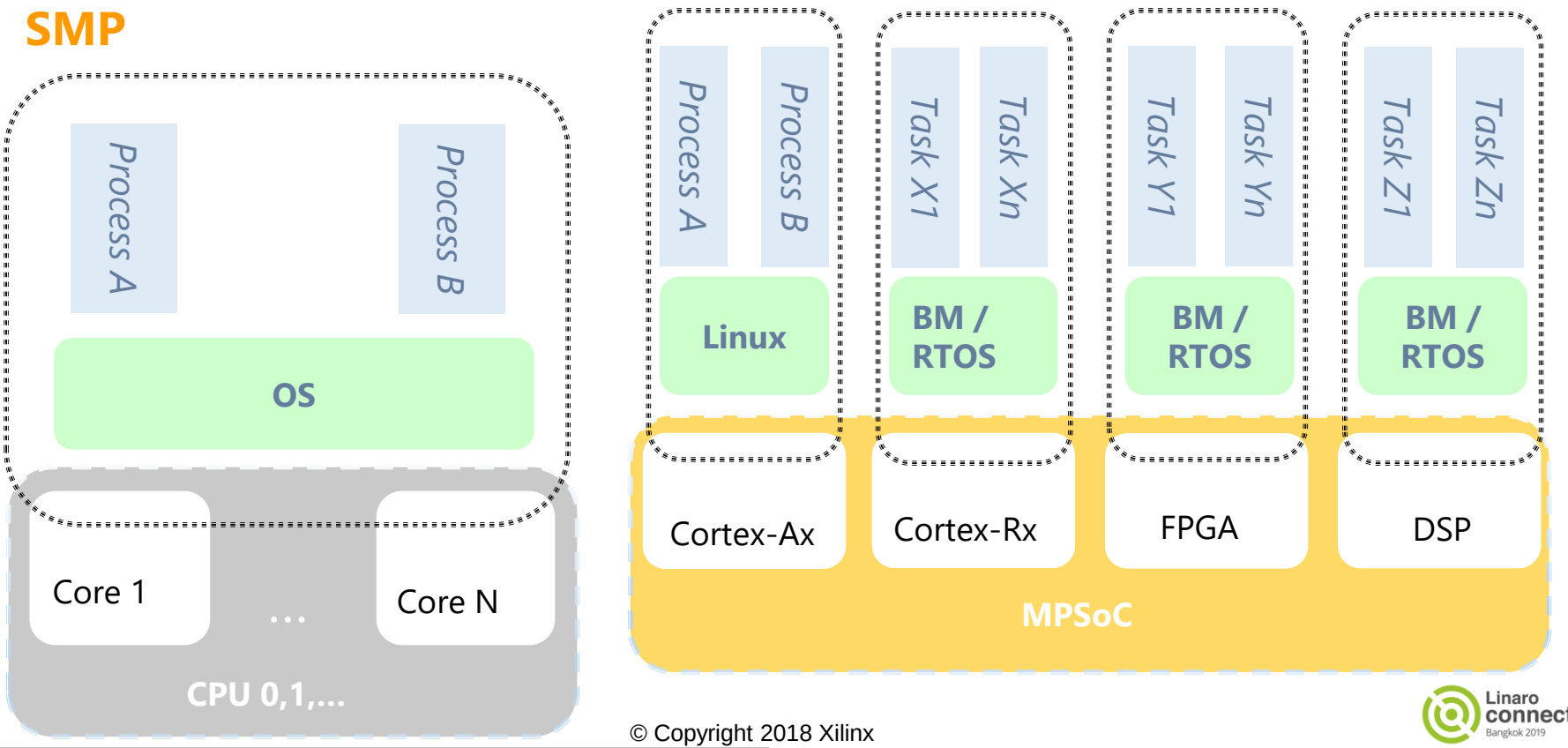
Glossary

- **SMP: Symmetric Multi-Processing**
- **AMP: Asymmetric Multi-Processing**
- **APU: Application Processor Unit**
- **RPU: Realtime Processor Unit**
- **LCM: Life Cycle Management**
- **IPI: Inter-Processor Interrupt**

SMP vs. AMP



SMP vs. AMP



What is OpenAMP?

- OpenAMP standardizes how Operating Systems interact:
  - In particular between Linux and RTOS/bare-metal
  - In particular in a multicore heterogenous systems
  - Includes:
    - Lifecycle APIs to start/stop/restart other OSes (RemoteProc)
    - Inter-Process Communication APIs to share data (RPMsg)
    - Shared memory protocol for OS interactions (VirtIO)
- Guiding principles
  - open-source implementations for Linux and RTOSes
  - Prototype and prove in open-source before standardizing
  - Business friendly APIs and implementations to allow proprietary solutions

OpenAMP libraries

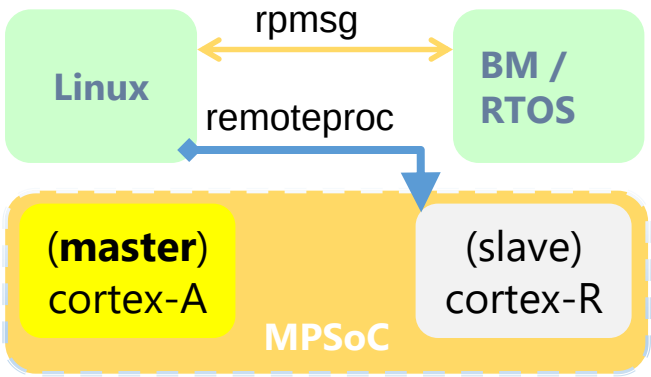
- Lifecycle Management (LCM) – allows a master to control/manage remote processors: power on/off, reset, load firmware
- Inter-Processor Communication (IPC) for shared memory management when sending/receiving data from/to master/remote
- Proxy operations – Remote access to systems services. A transparent interface to remote contexts from Linux user space applications running on the master processor
- Depends on libmetal acting as an OS environment and hardware abstraction layer
- Ongoing work to decouple RemoteProc and RPMsg so that they can be used independently

OpenAMP Topology

- Both star and daisy chain topologies are supported, or any combination thereof. The CPU with dual-responsibility (remote & master) provides chaining
- Linux is not required to be on any CPU
- Linux is not required to be the master
    - Linux as remote is currently only supported between Cortex-A cores

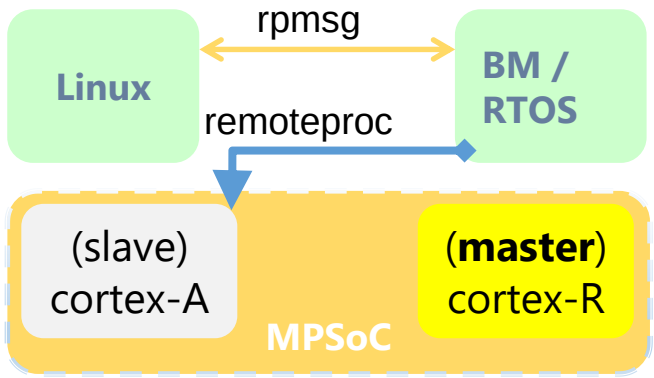
Linux AMP vs OpenAMP

- **Linux AMP**
  - Kernel modules. No support for firmware on remote processors. Apps on a remote must understand rpmsg/remoteproc.
  - Masters must run Linux
  - Low level device-specific code is not supported



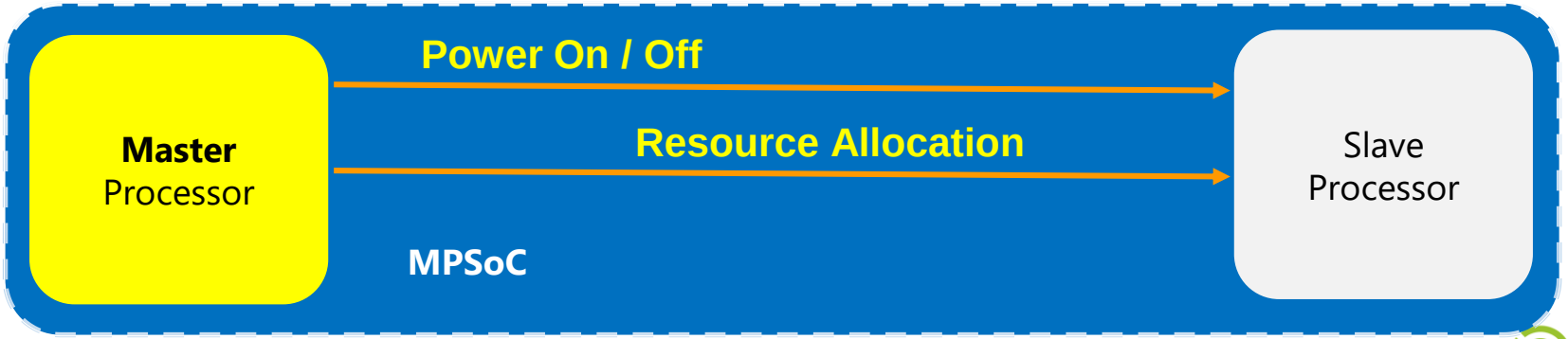
Linux AMP vs OpenAMP

- **OpenAMP**
  - User libraries. Adds support for the RemoteProc and RPMsg to RTOS and bare metal
  - Master no longer needs to be Linux-based
  - Abstracts the device-specific behavior



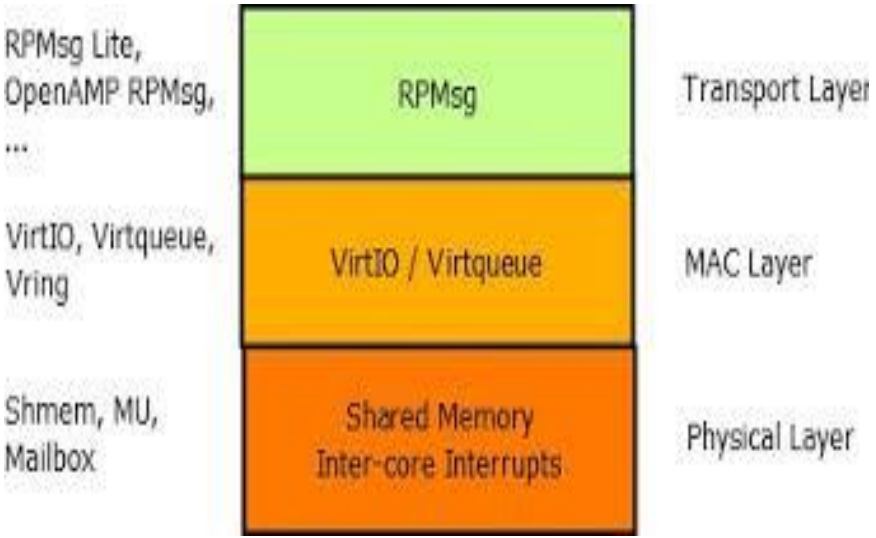
Remote LCM with RemoteProc

- **RemoteProc** – **Remote Processor,** provides support for a master to run firmware on a remote processor.
- **RemoteProc** is a framework that allows a master to control/manage remote processors (power on/off, reset, and load firmware). A RemoteProc driver is used for lifecycle management of remote firmware.
- **RemoteProc**
  - Provides API to control remote processor
  - Abstracts hardware differences between involved processors
  - Establishes communication channels between master and remote processors using the RPMsg framework
  - Declares a minimal set of device-specific low-level handlers



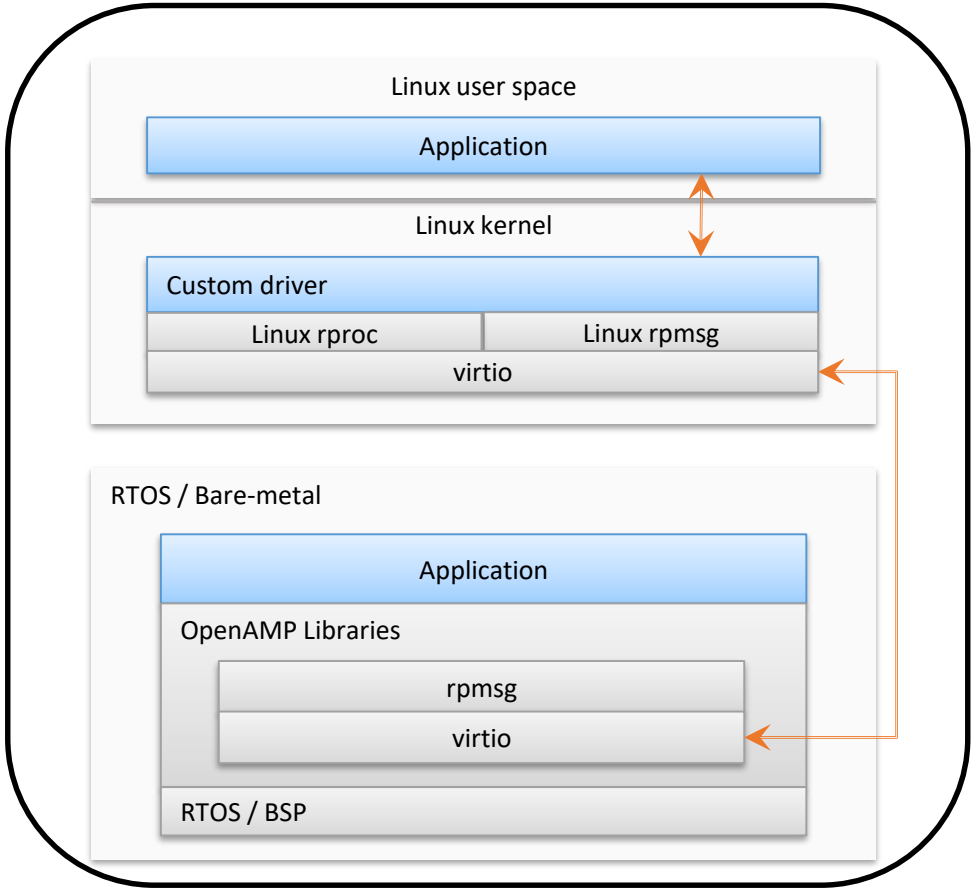
Master/Remote IPC with RPMsg

- RPMsg (Remote Processor Messaging)
  - Provides inter-processor communication (IPC) between master and remote processors.
    - An RPMsg represents a communication channel between the master and a specific remote processor
  - Defines only vendor agnostic aspects of communication
    - E.g. API and the format of messages.
    - Relies on RemoteProc for device-specific handlers



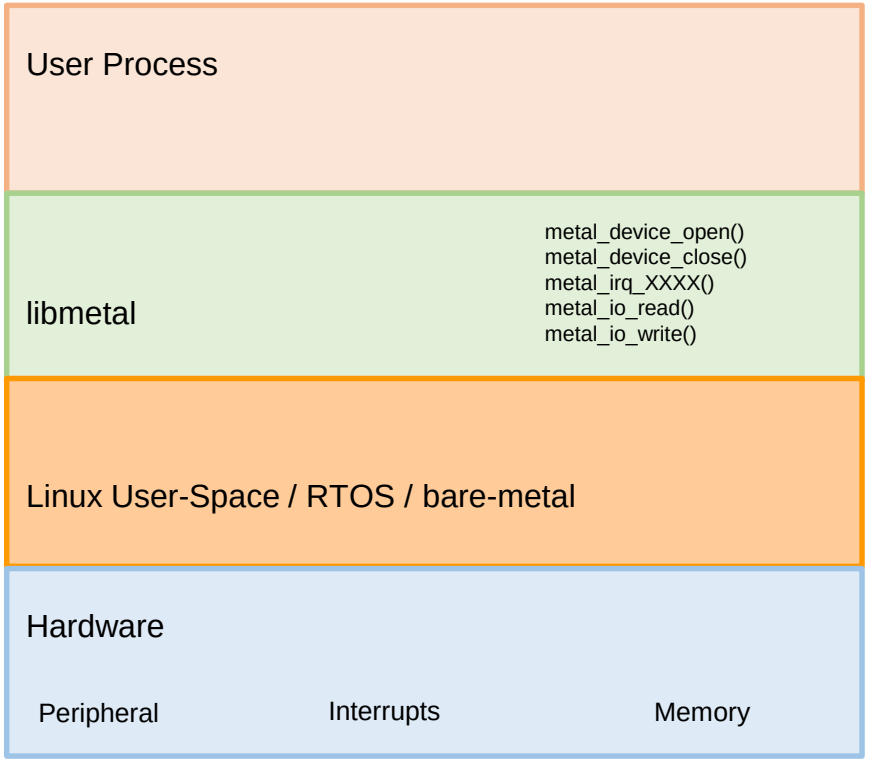
VirtIO

- VirtIO library
  - An abstraction layer over devices in a para-virtualized hypervisor
  - Implements the OASIS virtIO standard for shared memory management
  - A virtualization standard for network, disk device (etc.) drivers
    - Applicable to OpenAMP configurations



Libmetal Overview

- libmetal
  - Provides common APIs for:
    - Device access
    - Interrupt handling
    - Memory management
    - Synchronization promitives
  - Across:
    - Linux user space (based on UIO and VFIO support in the kernel)
    - RTOS (with and without virtual memory)
    - Bare-metal environments
- Fundamental to OpenAMP architecture
  - RemoteProc, RPMsg and VirtIO use libmetal



OpenAMP Remote Startup Process

- OpenAMP architecture
  - Assumes the master is already running and remote processor is in standby or powered down
- Remote Processor Firmware Loading
  - OpenAMP master loads firmware into the memory location
- Remote Processor Start
  - OpenAMP master starts remote processor
    - Example: wake-up remote, release remote from reset, power-on remote, etc.
  - Master waits for remote
  - Master establishes a communication channel to the remote processor

Vendor Support

- Vendor handles the low-level porting for their specific platform(s)
- Vendor supplies example applications for their platform(s)
  - Application includes demonstration of resource table
  - Example application demonstrating basic IPC (e.g., echo)
  - Example application demonstrating master off-loading
- Vendor supplies Linux RPMsg driver for their platform(s)
- Vendor supplies example kernel module and user-space application for interacting with remote device

Status

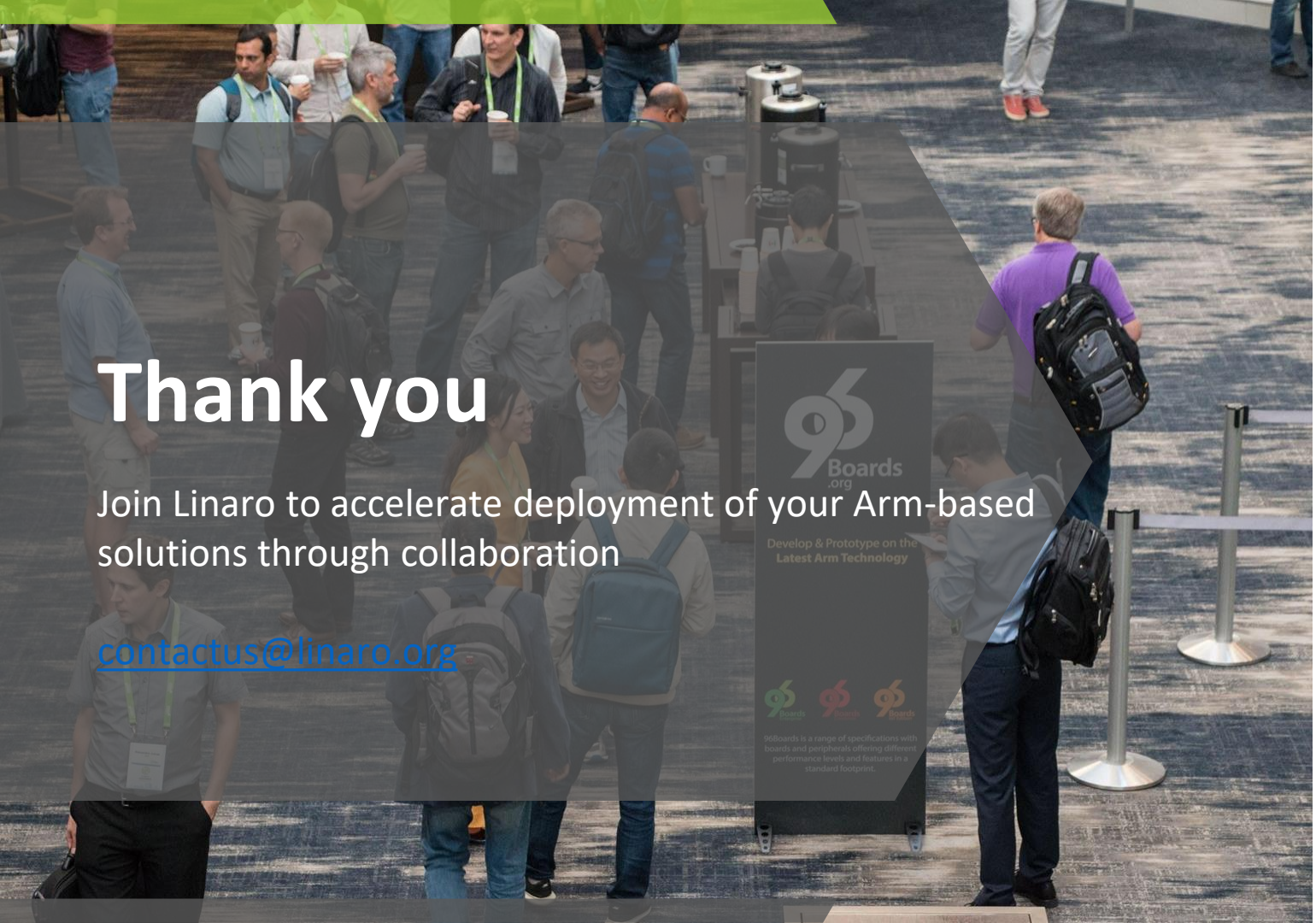
- OpenAMP is an active, evolving community project
  - Project home: github.com/OpenAMP
  - Source structure is fluctuating and standardization is a work in progress
  - Roadmap for advanced use-cases and features
    - IPC performance needs improvement (WIP)
- Availability
  - Commercial
    - uC/OS, Thread-X, Enea OSE, Mentor Nucleus
  - Open Source
    - Zephyr
    - Linux
    - FreeRTOS
  - Porting of the framework still necessary for many commonly used platforms

What's Supported Today

- Range of use cases:
  - Topologies: **peer-to-peer**, master-slave and **hierarchical**
  - Interfaces: message passing, file-system, block, graphics, network
- Provide consistent and portable application interfaces across:
  - Environments: **Linux kernel and user-space**, **FreeRTOS**, **Zephyr**, **bare-metal**
  - Processor architectures: **Cortex-A53,** Cortex-A72, **Cortex-R5, MicroBlaze,** x86, **MIPS32**
  - Secure and **Non-Secure** modes
  - Threads and Processes (on Linux and RTOSes)
  - Virtualized guests and containers (with hypervisors)

Conclusion

- OpenAMP provides a software framework for developers to
  - Enable MPSoC Life Cycle Management (LCM)
  - Load firmware across a multi-processor system
  - Establish communication between the processors
- OpenAMP provides these features in a platform agnostic manner



Thank you

Join Linaro to accelerate deployment of your Arm-based solutions through collaboration

contactus@linaro.org

Linaro connect
Bangkok 2019