



HKG2018-411: OpenAMP Introduction

Wendy Liang





Linaro
connect
Hong Kong 2018

Agenda

- OpenAMP Projects Overview
- OpenAMP Libraries
- Changes in Progress
- Future Improvements





Linaro
connect
Hong Kong 2018

OpenAMP Projects Overview



Introduction

With today's sophisticated SoCs there is often a need to integrate multiple runtime environments with multiple operating systems. This raises a lot of issues, such as:

- Lifecycle issues
 - Boot order, start one OS from another, tear down OS, reboot, ...
- Communication
 - Message passing
 - Data sharing
- Resource handling
 - Memory, devices, interrupts,
 - Power management
 - ...

OpenAMP aims to address these and other issues in a standardized way, both through an open source project and through standardization by MCA.



What is Needed to Be Able to Mix OSes?

- A standard “protocol” so OSes can interact without dependencies
 - On a given HW, any OS can interact with any other OS
 - Without special adaptation
 - On a shared memory system this is a set of data structures and conventions
 - e.g. the ring buffers in virtio
- A standard set of APIs for OS interactions
 - A low level API that abstracts underlying OS and HW
 - A set of lifecycle APIs
 - Messaging APIs
 - Other potential features
 - Proxy capabilities to make remote OS look like Linux process
 - Remote procedure calls, power management, device configuration, debug...
- Upstream Linux support for protocol and APIs
 - Linux is increasingly becoming the main OS in a multi-OS system
- Open Source implementation
 - Quickest way to adoption
 - Standardization by reference implementation



What is OpenAMP?

- OpenAMP standardizes how Operating Systems interact
 - In particular between Linux and RTOS/bare-metal
 - In particular in a multicore heterogeneous systems
 - Includes:
 - Shared memory protocol for OS interactions (virtio)
 - Lifecycle APIs to start/stop/? other OSES (rproc)
 - Communication APIs to share data (rpmsg)
 - More to come
- Both a standardization effort and an open source project
 - MCA OpenAMP working group
 - Linaro LITE Open source project focuses on implementation and testing new ideas
- 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





Linaro
connect

Hong Kong 2018

OpenAMP Libraries



OpenAMP Code Base

- The open source implementation source code: <https://github.com/OpenAMP>
- OpenAMP library:
 - <https://github.com/OpenAMP/open-amp>
 - rpmmsg for message passing
 - virtio for memory sharing
 - remoteproc for remote life cycle management and resource assignment
- Libmetal library
 - <https://github.com/OpenAMP/libmetal>
 - libmetal for device management, I/O and shared memory access



RPMmsg Overview

In asymmetric multiprocessor systems, the most common way for different cores to cooperate is to use a shared memory-based communication. Rpmmsg is a shared memory based messaging bus that allows communication between processors.

- RPMmsg Protocol Layer in OpenAMP

Transport Layer

RPMmsg

Media Access
Control Layer

VirtIO / VirtQueue /
Vring

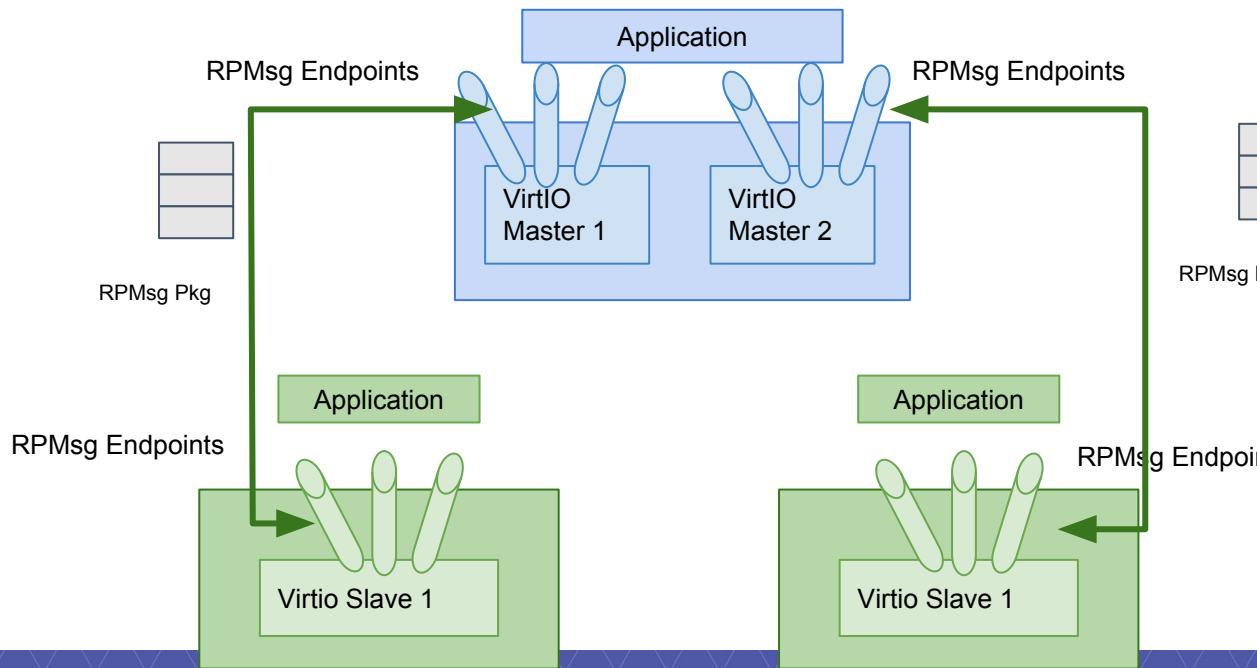
Physical Layer

Shared Memory,
inter-core
notification

- RPMmsg Header Definition

Source ep address (32bit)	
destination ep address (32bit)	
reserved (32bit)	
Length (16bit)	flags (16bit)
User payload	

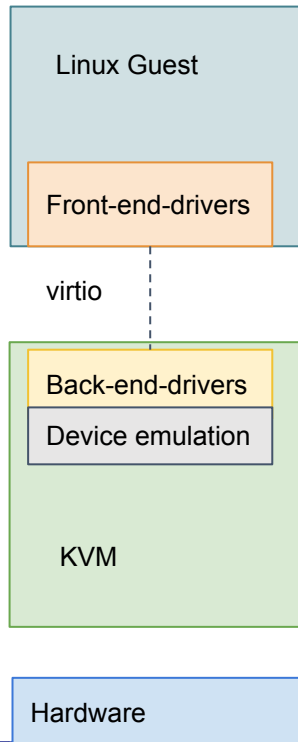
- RPMmsg Protocol Layer in OpenAMP



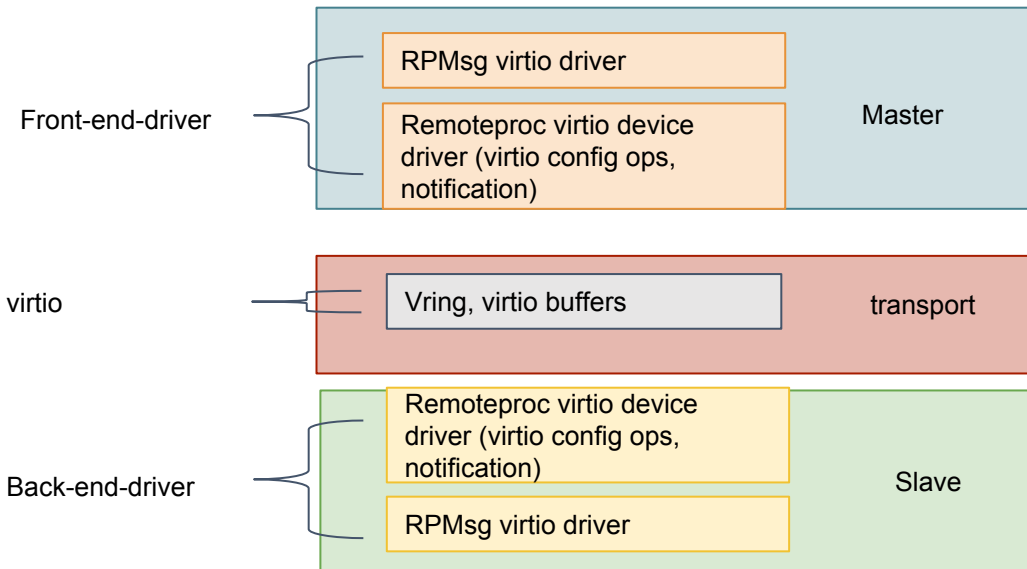
Virtio Overview

Virtio is an I/O virtualization framework, it is used for paravirtualization. OpenAMP uses virtio for manage shared memory.

- Virtio Architecture

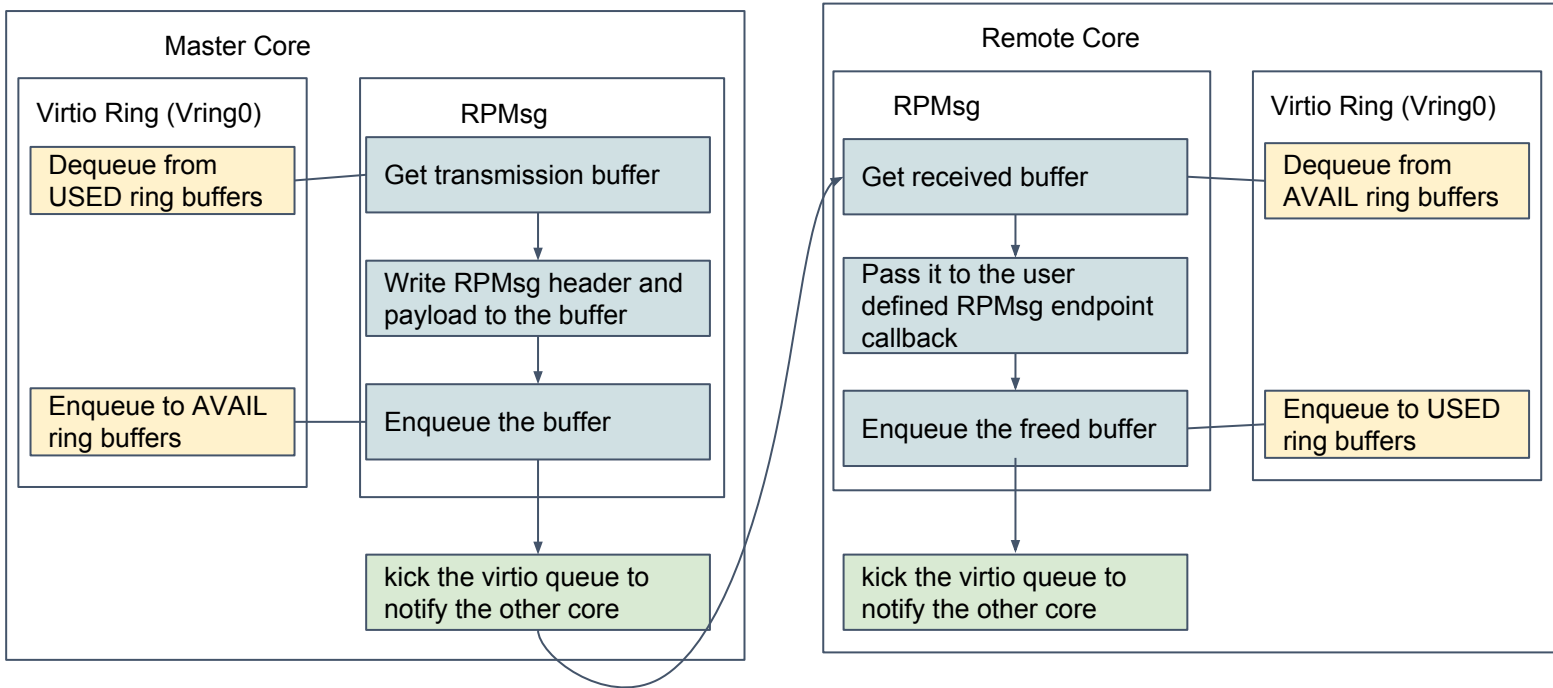


- Virtio in OpenAMP



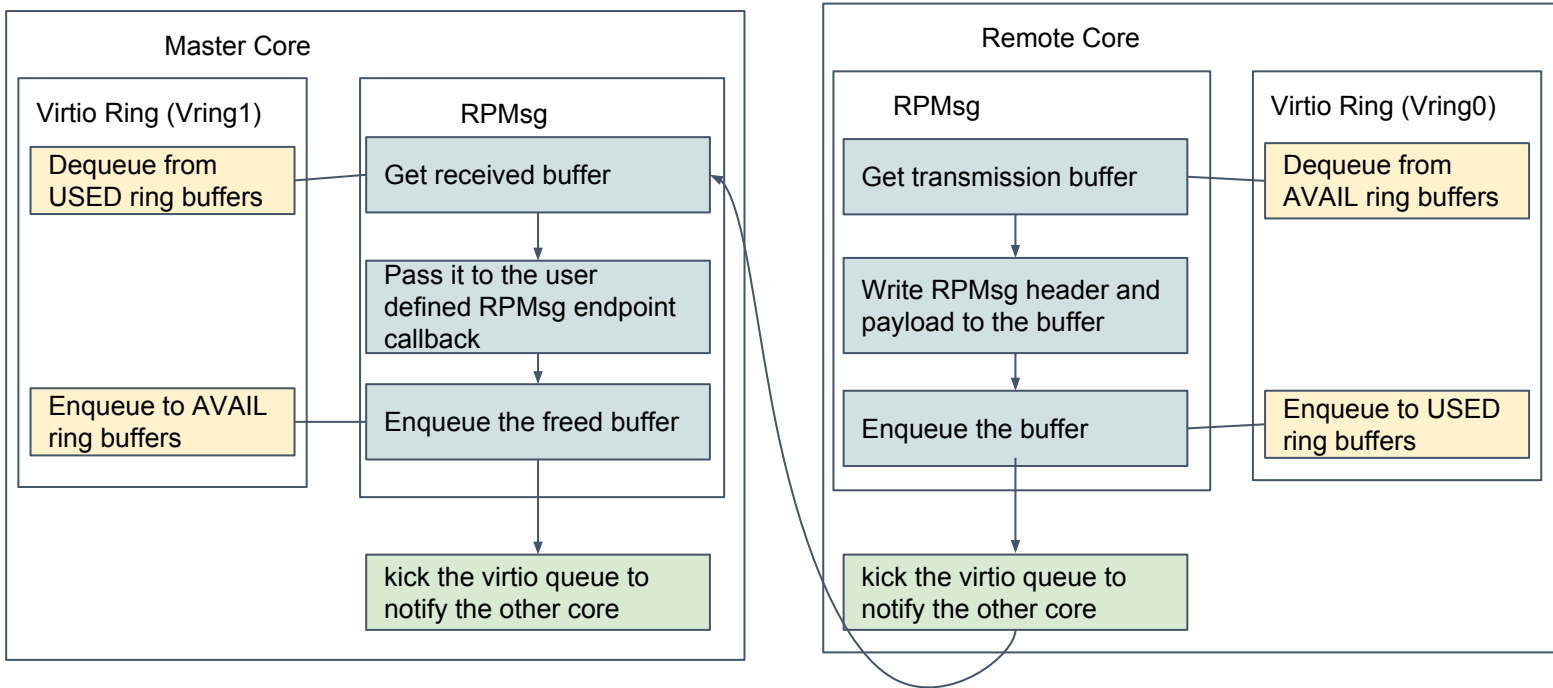
RPMmsg Virtio Flow - Master to Remote

Virtio Master to Remote



RPMmsg Virtio Flow - Remote to Master

Virtio Remote to Master



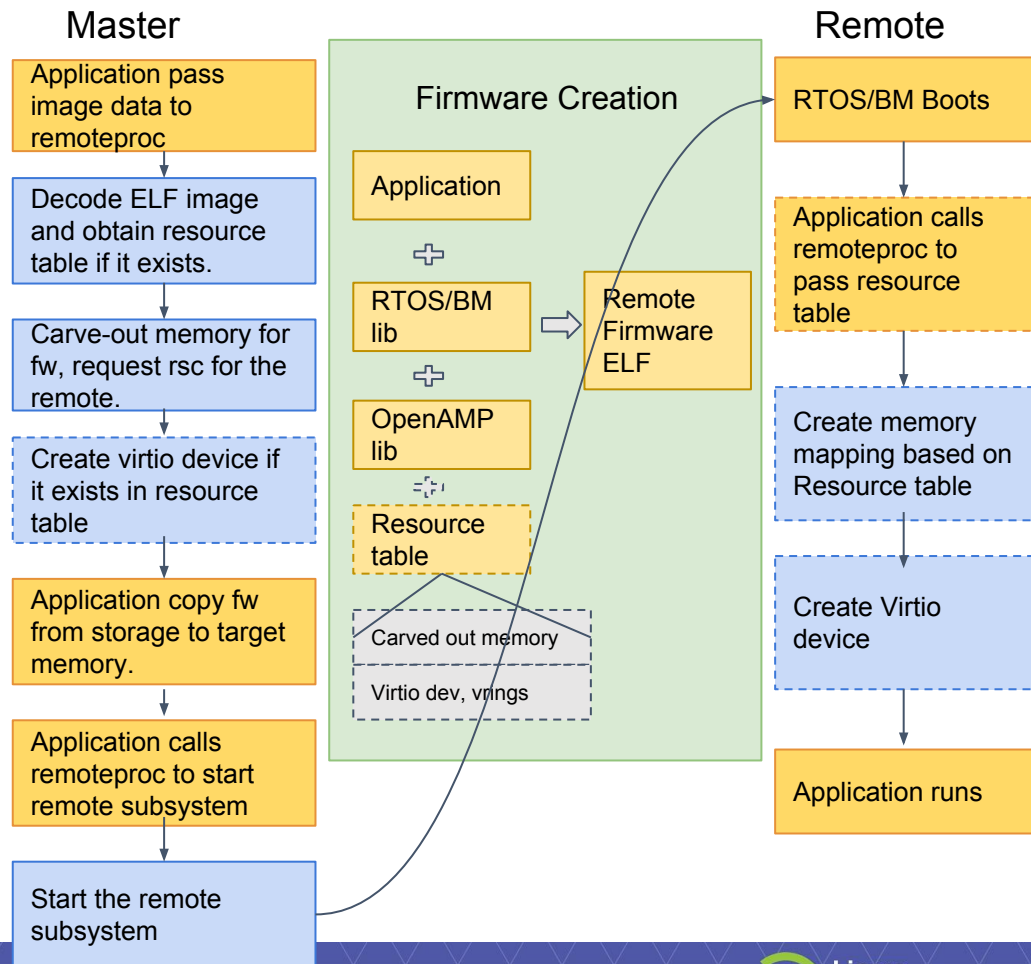
Virtio Based RPMsg Implementation Limitation

- RPMsg communication relies on the virtio master side to start the communication, not peer to peer
 - If one side is Linux, Linux has to be the virtio master, as Linux as virtio backend is not supported yet.
- It is mainly used to pass messages but not for big data sharing.
 - default rpmsg APIs requires copying data from application to the shared buffers
 - shared buffer size is fixed at initialization
 - How to mix control services and data services in the same pairs of vrings



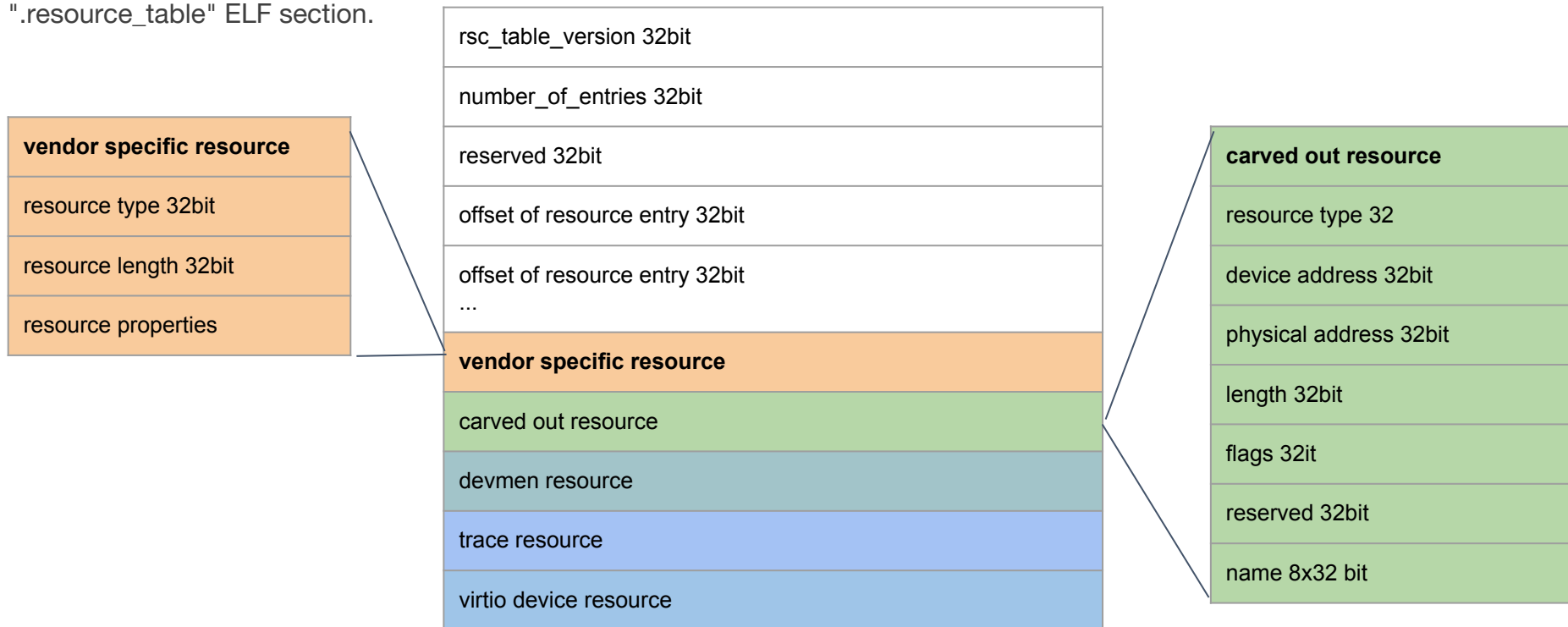
Remoteproc Overview

- Remoteproc provides user APIs to do life cycle management of the remote system and manage the resources of the remote system.
- It provides the following functions:
 - load remote system image
 - **setup resources for the remote system**
 - start the remote system
 - **manage the resource of the remote system**
 - **suspend the remote system**
 - **restore the remote system**
 - **stop the remote system**
 - **release the resource of the remote system**
 - shutdown the remote system and release its source



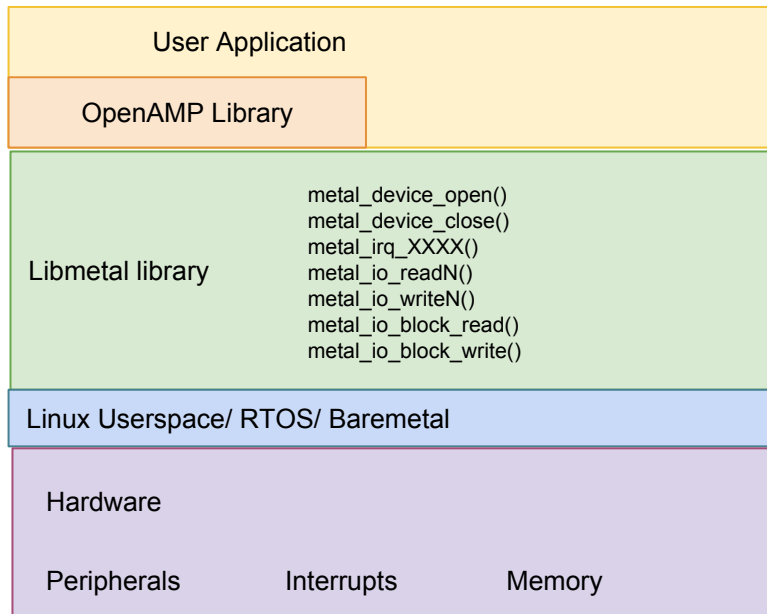
Remoteproc Resource Table

A resource table is essentially a list of system resources required by the remote system. It may also include configuration entries. e.g. virtio configuration space. If needed, the remote firmware should contain this table as a dedicated ".resource_table" ELF section.



Libmetal Overview

- Libmetal is an abstraction layer across different OSes and hardware platforms environments to manage devices, handle devices interrupts and request memory access.
- It was initially derived from the OpenAMP “porting layer” to enable OpenAMP library to be used across different OSes and hardware platforms. It can be used independently to the OpenAMP library.





Linaro
connect

Hong Kong 2018

Changes in Progress



What's OpenAMP 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-R5, MicroBlaze, x86, **MIPS32**)
 - **Secure** and non-secure worlds
 - Threads and processes (on Linux and RTOS)
 - **Virtualized guests and containers** (with hypervisors)
- Leverage hardware architecture
 - Processor ISA(A9, A53 64bit and R5), **coherency**, **exclusive monitors**, **IOMMU**

Not yet supported



Changes in Progress

- Enable OpenAMP / libmetal Zephyr support
 - Libmetal unit tests can run on Zephyr QEMU Cortex M3 platform
 - Zephyr is added to OpenAMP library cmake build
- Enable OpenAMP on Microcontroller
 - Standardize RPMsg, Remoteproc, virtio config ops APIs
 - restruct OpenAMP code base
 - decouple components features
 - remoteproc life cycle management implementation
 - remoteproc virtio config ops implementation
 - virtio based RPMsg implementation
 - libmetal shared memory operation implementation



Future Improvements

- Testing
 - CI for OpenAMP libraries
 - Unit tests
 - Travis CI plugin in github
 - OpenAMP tests in Lava lab
- Extend remoteproc resource management APIs
 - resource table
 - extend carved out memory to cover shared memory owned by remote and host
 - extend to cover vendor resource
 - line up with the virtio config operations
- Extend virtio drivers support
 - Besides virtio rpmsg, support virtio net, virtio block, virtio console, and virtio balloon to enable software stacks such as network stack to build on top
- Facility to launch a remote application and improve the proxy service in Linux userspace
 - e.g. binfmt_misc daemon to attach/dettach a remote application, send/receive data to the remote





Thank You

Questions/Comments:

jiaying.liang@linaro.org, arnaud.pouliquen@linaro.org, kumar.gala@linaro.org

#HKG18

HKG18 keynotes and videos on: connect.linaro.org

For further information: www.linaro.org

