OpenAMP Webinar
2022-12-13

Welcome & Agenda

Introduction	
Upstream Remoteproc & RPMsg	
Presentation	Bill Mills, Linaro
ST Demo	Arnaud Pouliquen, STMicroelectronics
AMD Xilinx Demo	Tanmay Shah, AMD Xilinx
Hypervisorless Virtio	
Presentation & Demo	Dan Milea, Wind River
System Devicetree	
Presentation	Tomas Evensen, AMD Xilinx
Demo	Bruce Ashfield, AMD Xilinx
Wrap up, Q&A	All above and more

OpenAMP: “**Open** **A**symmetric **M**ulti-**P**rocessing” Project

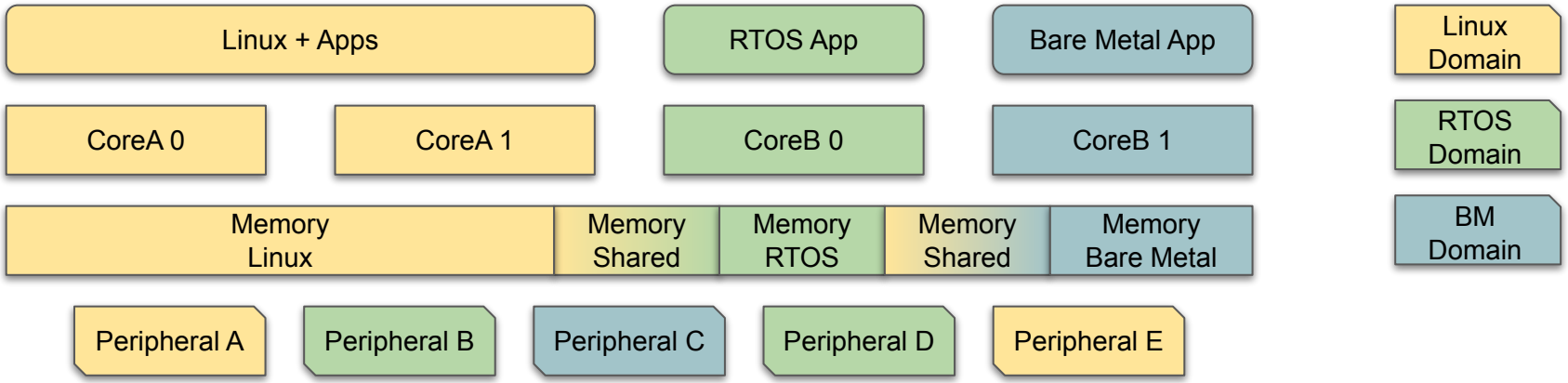
arm Linaro NORDIC SEMICONDUCTOR NXP SIEMENS

life.augmented Texas Instruments WNDRVR AMD XILINX

Runtime coexistence and collaboration
Runtime hardware resource assignment
Resource sharing and IPC between runtimes
Control mechanisms to start and stop runtimes
Typical system: Linux + RTOS on one system-on-chip

www.openampproject.org.

OpenAMP Mission



OpenAMP provides standards, runtime libraries and tooling built on top of existing open source projects to simplify runtime collaboration

In OpenAMP, we work on …

- Technologies, independent of implementation
  - Remoteproc & RPMsg
  - Virtio devices
  - System Devicetree
  - anything else that fits the mission
- Implementations
  - Linux Kernel
  - OpenAMP libraries: open-amp and libmetal
    - works on MCUs, SOCs,
    - Multiple RTOS & bare-metal
    - Linux userspace
  - meta-openamp for Yocto/OE
  - Lopper for System Devicetree

OpenAMP: A short history

- **2012-2015:** OpenAMP starts
  - Remoteproc & RPMsg entered the Linux kernel in v3.4 (2012)
  - The open-amp and libmetal libraries have existed since 2013/2014 respectively
  - OpenAMP project first founded in 2015
- **2018:** OpenAMP libraries are included in Zephyr
  - Skinny down the libraries
- **2019/2020:** OpenAMP relaunched as a Linaro community project
  - More formal organization and governance
  - New kernel maintainer came on board from Linaro
- **2022:** Linaro HPP
  - Heterogeneous Processing Project
  - Linaro & members add engineering resources to OpenAMP
  - ST and AMD Xilinx are current sponsors of HPP

Upstream Remoteproc and RPMsg

What are Remoteproc & RPMsg?

- Remoteproc:
  - Load, start, stop firmware (New since 2019)
  - Crash detect, dump, & recover
  - Optionally parses firmware resident table to coordinate resource usage
  - Can alternatively just attach to a remoteproc that is already running and provide RPMsg
  - Optional /dev/remoteprocNN device node for control beyond sysfs
- RPMsg:
  - IPC messages with multiple channels, name based port resolution and late binding
  - Multiple in kernel providers:
    - OpenAMP focuses on virtio based RPMsg
    - Qualcomm uses glink RPMsg to dedicated core
  - Can be used in-kernel or from user space (/dev/rpmsgNN)
  - Can export a named endpoint from user space
  - Can provide a tty device (real or virtual) from the remoteproc (/dev/ttyRPMSGNN)

In kernel users of RPMsg


```
drivers/bluetooth/btqcomsmd.c
drivers/media/platform/st/sti/delta/*
drivers/misc/fastrpc.c
drivers/net/wireless/ath/wcn36xx/*
drivers/net/wwan/rpmsg_wwan_ctrl.c
drivers/platform/chrome/cros_ec_rpmsg.c
drivers/soc/qcom/apr.c
drivers/soc/qcom/smd-rpm.c
drivers/soc/qcom/wcnss_ctrl.c
drivers/tty/rpmsg_tty.c
net/qrtr/smd.c
samples/rpmsg/rpmsg_client_sample.c
sound/soc/fsl/*
```

Why Upstream?

- OpenAMP technologies have existed for years but was always tied to vendor SDKs
  - Needed patches in vendor kernel
  - Only built as part of large vendor SDKs
  - No instructions to use it w/o vendor SDKs
- OpenAMP CI Builds
  - Use the latest kernel, old kernels, or even -rcN kernels
  - **future:** build & test patch series from remoteproc mail list
  - Build with Upstream Yocto/OE: minimal layers needed poky + meta-openamp
    - meta-arm is used today to get generic-arm64 machine def
    - has own meta-openamp-bsp to supply generic-armv7a machine def
  - Build one OS image for arm64 and one for arm32 (v7a)
    - Boot firmware from board as SystemReady defines
    - Should work *everywhere, but …*
      - *Only have one arm32 platform tested today*
      - *Only have one arm64 platform tested today*

Why Zephyr?

- OpenAMP libraries are portable
  - support multiple RTOS, bare-metal, and even Linux user space
- For CI, demo, and reference we have to start somewhere
- So why Zephyr?
  - Complete, non-trivial system, with batteries included
  - Same clone, configure, build setup for all SOC vendors
  - Same build tools for all vendors
- OpenAMP CI Build
  - Provide OpenAMP demos & reference apps for use with Upstream Zephyr
  - Allow use of latest OpenAMP libraries as an option
  - Test experimental branches of Zephyr & OpenAMP libs
  - **Future:** Build and test library PRs as they come in
  - **Future:** Automate release testing as much as possible

A lot has been happening in the last 3 years

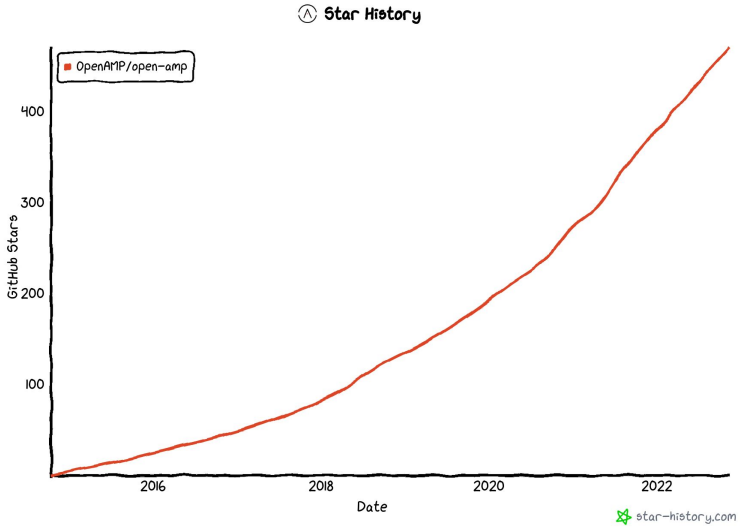
	v5.3 to v6.1-rc6 (3+ years)	v3.4 to v5.3 (7+ years)
kernel Remoteproc	442 patches	399 patches
kernel RPMsg	104 patches	174 patches
	2019 to 2022	2014 to 2018
open-amp library	115 PRs	114 PRs
libmetal	95 PRs	69 PRs

OpenAMP library usage

The project has a good growth momentum extending to multiple OSes and silicon architectures
Adopt it and join OpenAMP Community!

- Multi-OS support
  - FreeRTOS
  - NuttX (integrated)
  - Linux (Userland)
  - Zephyr (integrated)
  - baremetal solutions
- Multi-architecture support:
  - ARM & ARCH64
  - CEVA
  - RISCV
  - X86 & X86_64
  - XTENSA
  - …

Library usage history based on GitHub stars:



Demos

STMicroelectronics demonstration:
Use of upstream Linux and Zephyr for inter-processor communication

life.augmented

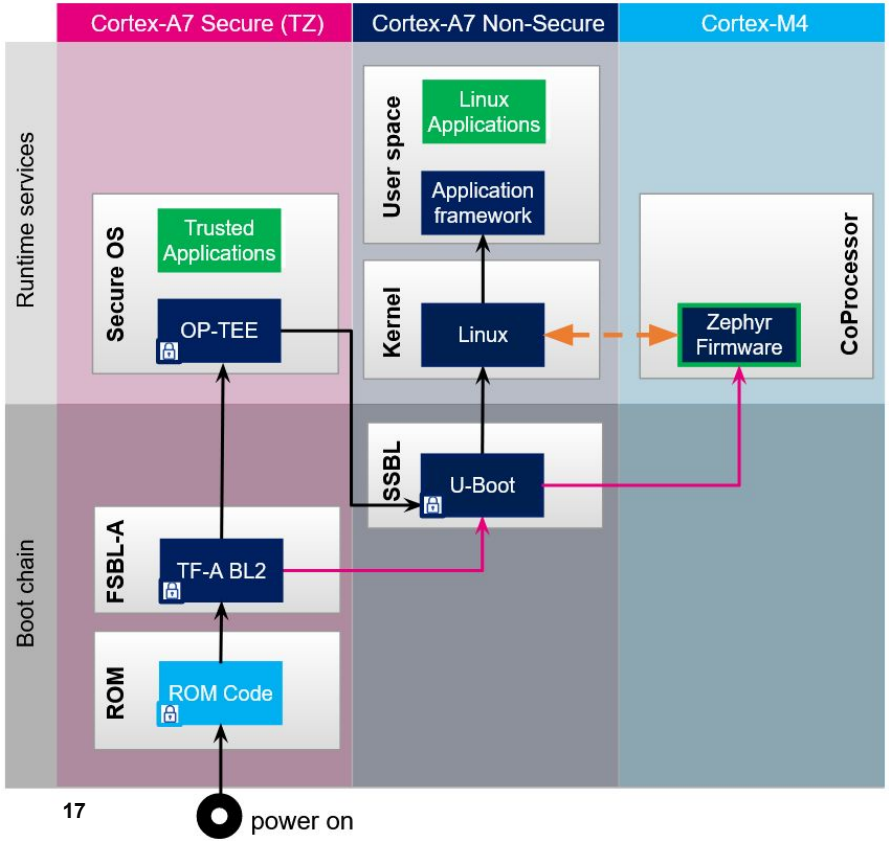
STM32MP157 platform introduction

- **Hardware**:
  The STM32MP157F SoC embeds:
  - Arm® Cortex®-A7 dual core
  - Arm® Cortex®-M4

- **Software**:
  - Cortex®-A7: ST OSS Yocto distribution based on upstream repositories:
    - Linux kernel 6.0
    - Arm Trusted Firmware 2.7.0
    - U-Boot 2022.10
    - OP-TEE 3.19.0-rc1

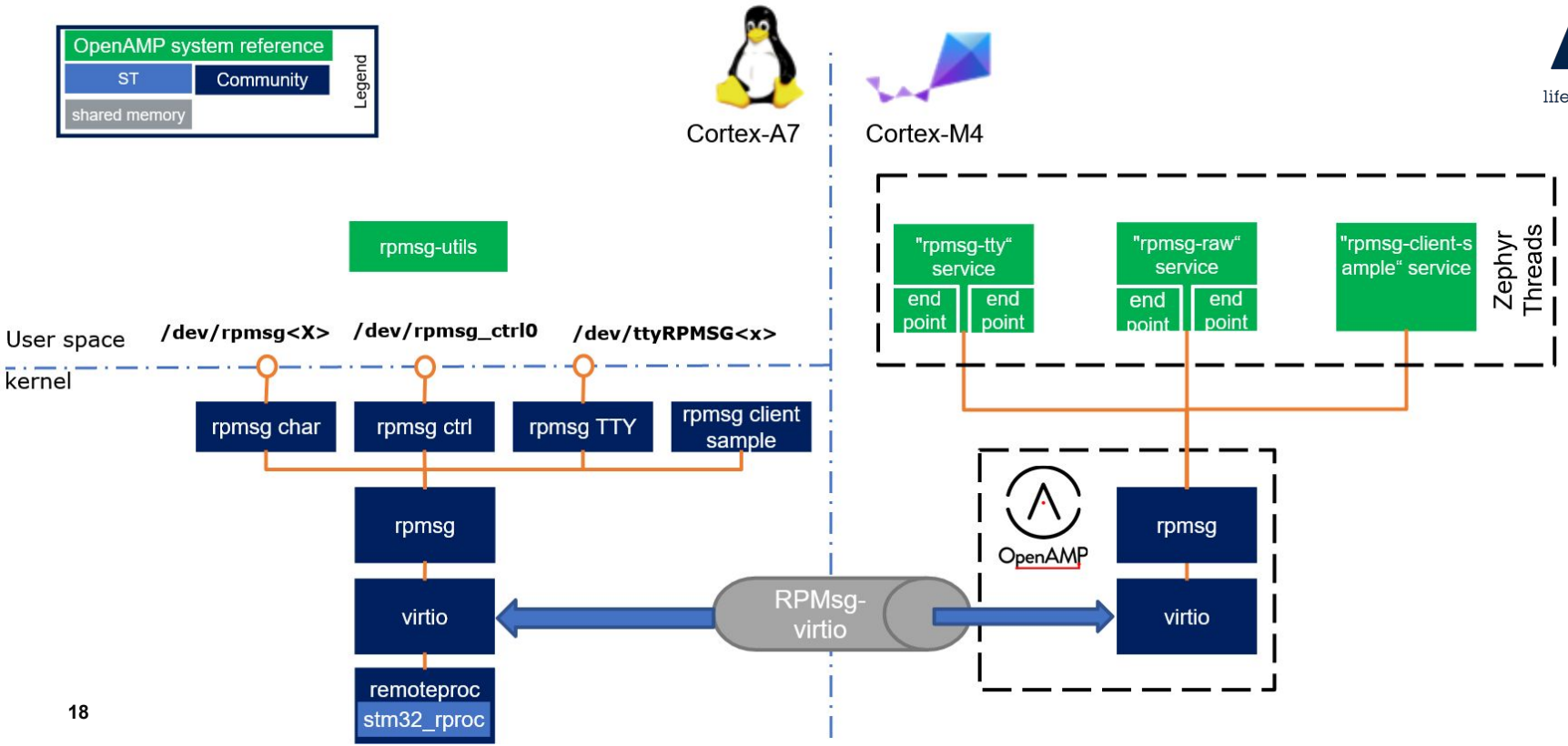
  - Cortex®-M4: Zephyr 3.2 (integrating OpenAMP v2022.04) with openamp system reference example

Demo 1: Preloading of a firmware by U-Boot and Linux attachment for RPMsg communication



- The Zephyr Firmware is loaded and started by the U-Boot before booting Linux
- The Linux “attaches” to the coprocessor. It configures the VirtIO to initiate the RPMsg protocol

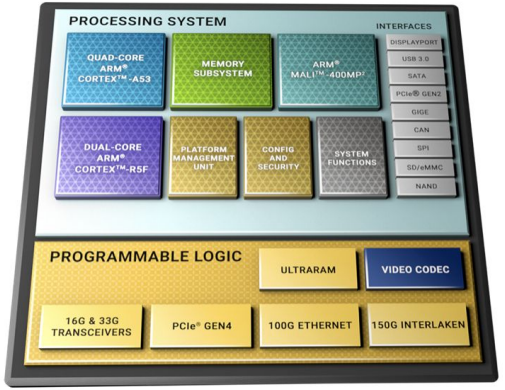
Demo 2 : upstreamed RPMsg services usage



OpenAMP on the Xilinx ZynqMP heterogeneous platform

Xilinx ZynqMP Platform Introduction

- Xilinx Zynq® UltraScale+™ MPSoC contains
  - Quad-core Arm® Cortex®-A53
  - Dual-core Arm Cortex-R5F
  - Microblaze controller
  - and many other HW blocks
  - SOC TRM: https://docs.xilinx.com/r/en-US/ug1085-zynq-ultrascale-trm

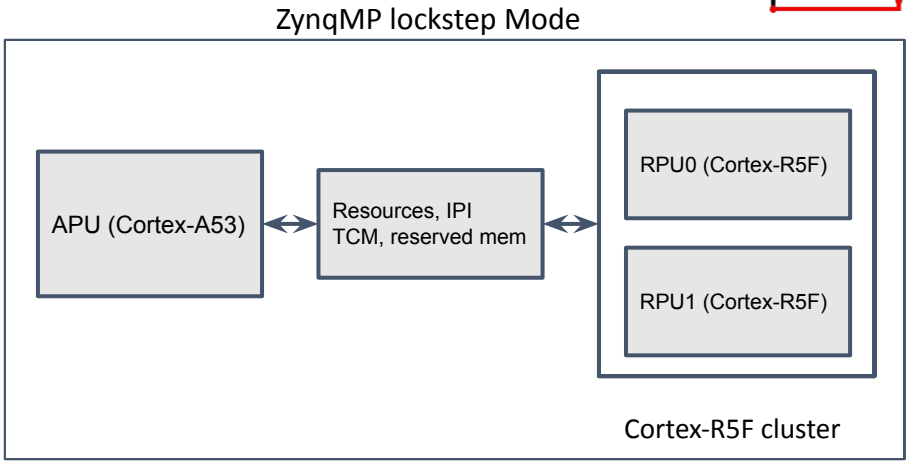


- OpenAMP and Linux kernel remoteproc and rpmsg framework is used for different operations and communication between Cortex-A core and Cortex-R5F cores

- As of now Cortex-R5F cores Can be configured in two modes:
  - Lockstep Mode (Also known as Safe Mode)
  - Split Mode (Also known as performance mode)

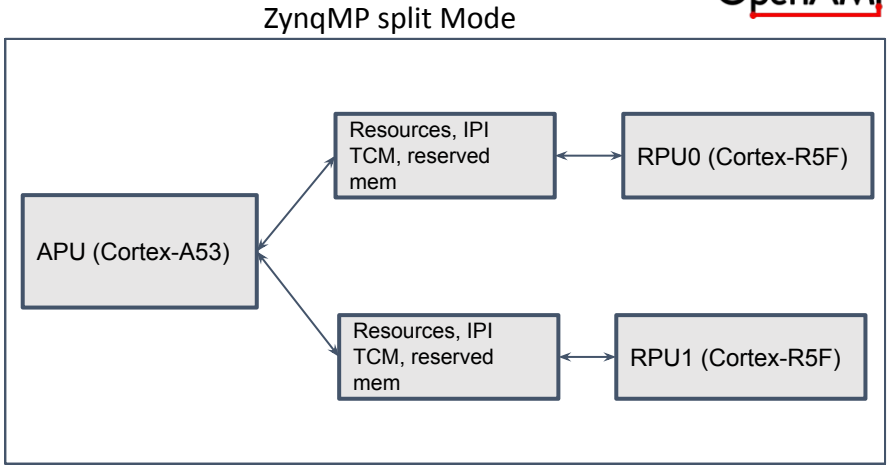
Xilinx ZynqMP Platform Introduction

- Lockstep Mode (Also known as Safe Mode)
  - Both RPU cores uses same set of resources (IPI, TCM, etc…)
  - Both RPU cores execute the same code in lock-step, clock-for-clock
  - The outputs from the lead RPU, core 0, are checked by core 1
  - Discrepancies are flagged as a system error



Xilinx ZynqMP Platform Introduction

- Split Mode (Also known as performance mode)
  - Each RPU core operates independently
  - Each RPU core has its own caches and TCMs

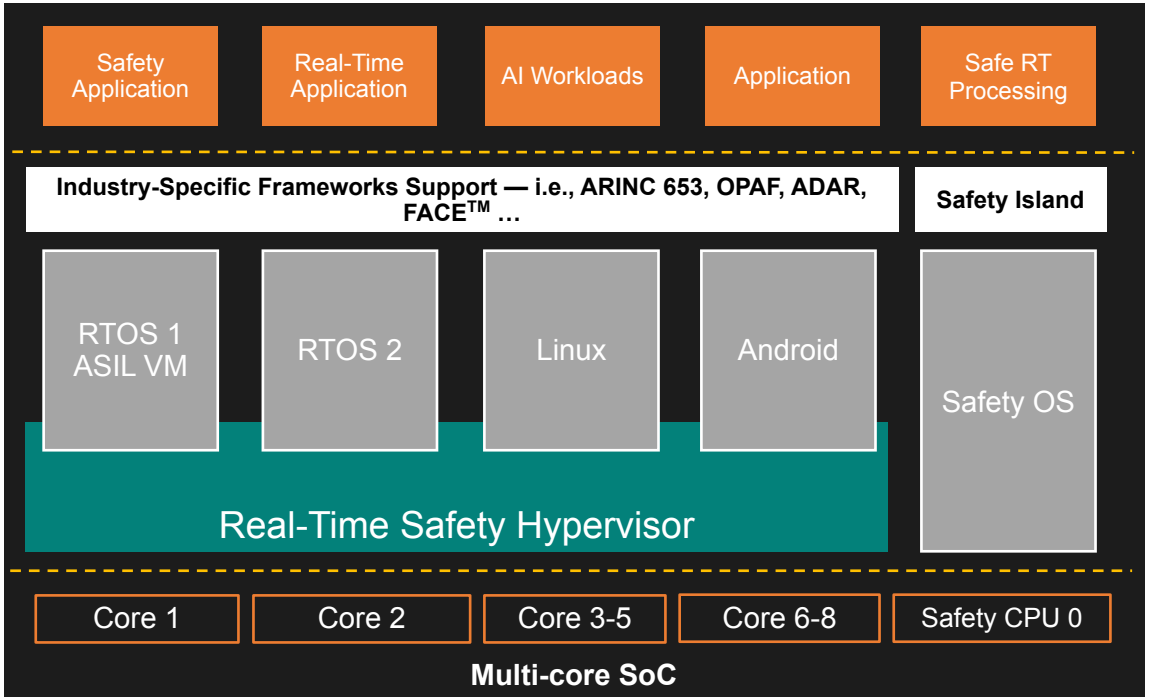


Hypervisor-less virtio

Intra-SoC connectivity and device sharing

WNDRVR

Intelligent Edge / Partitioned systems



Complex SoCs - flexible integration platforms:

- Many levels of safety, real-time, and manageability
- Decoupled multi-vendor software development
- Improved lifecycle management
- Increased testability

VIRTIO for intra-SoC workload integration

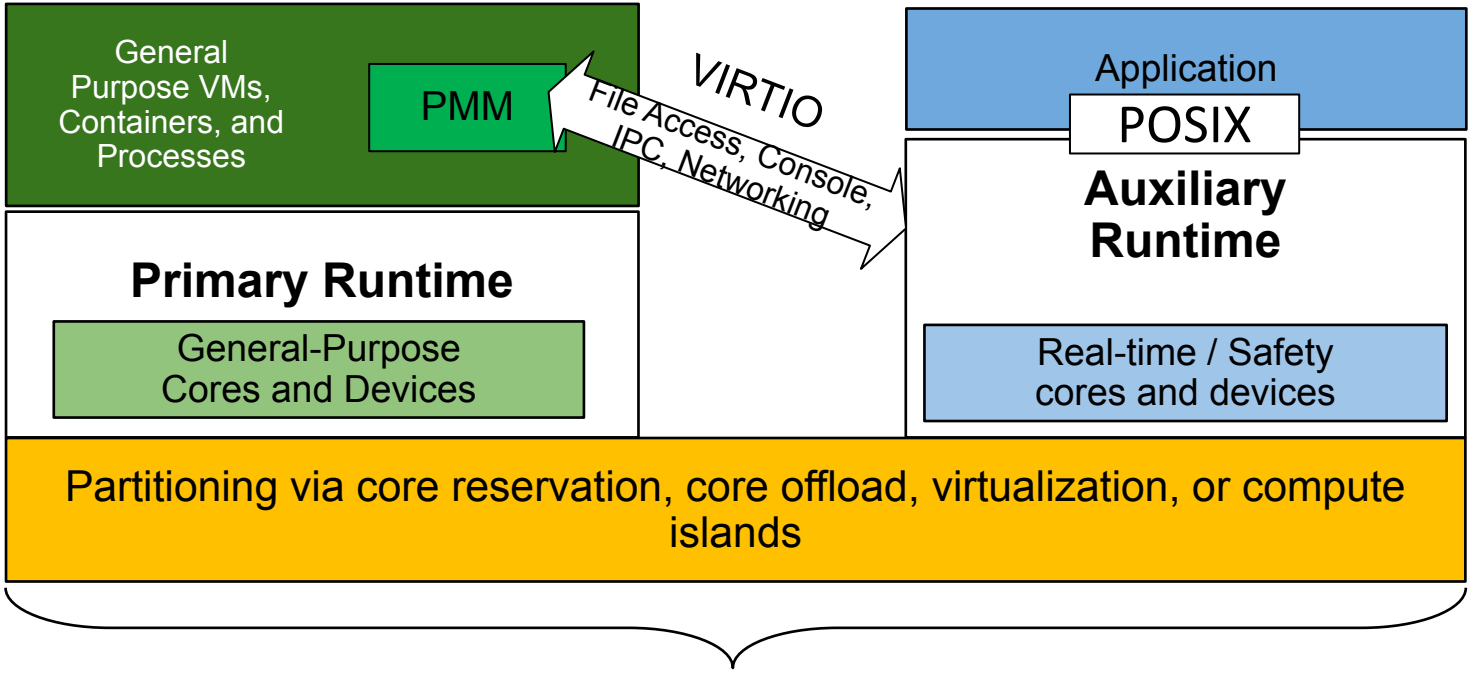
- Open-source standard
- Enables decoupling the software from the hardware and supports modular system integration
- Solves the fragmentation problems caused by hypervisor specific interfaces and device drivers implementations
- Allows the creation of software defined architectures
- Increases efficiency
- Enables code reuse

Hypervisor-less VIRTIO

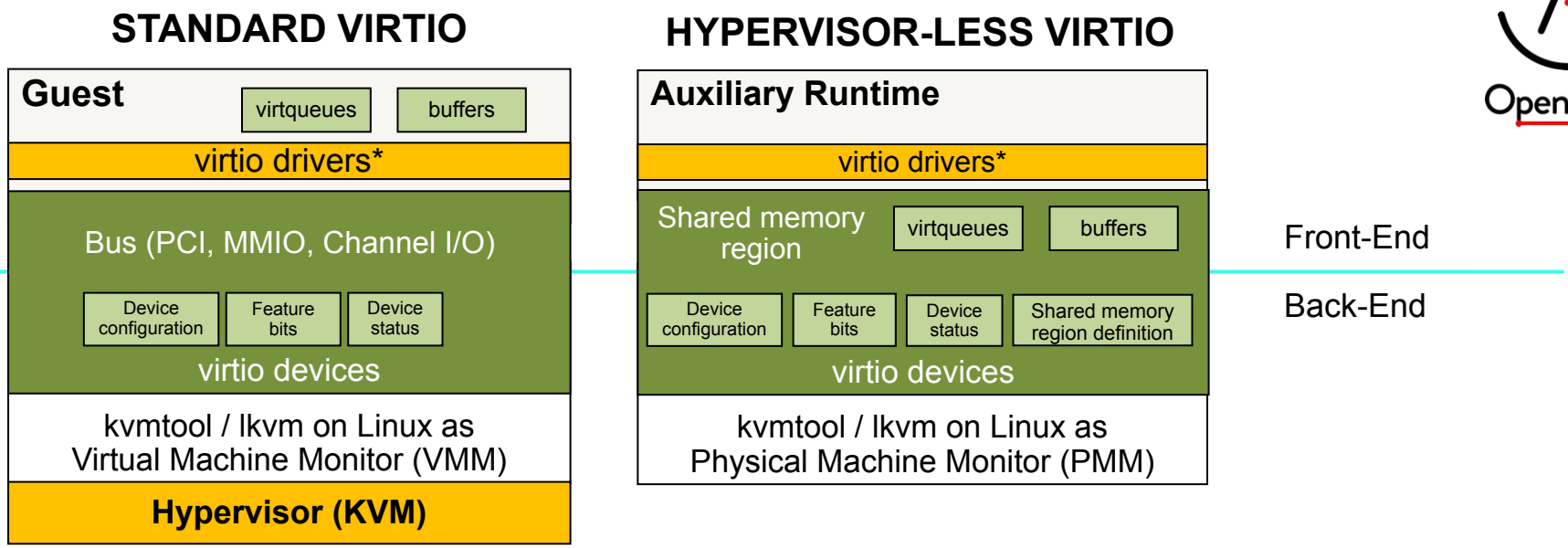
Define and prototype a framework for using VIRTIO as a communication infrastructure, while removing the constraints usually associated with the presence of a hypervisor.

- Intel and ARM support
- MMIO transport over shared memory
- Unsupervised AMP support
- Hardware notifications
- Static configuration (features, queues)
- Linux KVM tool used as a Physical Machine Monitor
- VIRTIO device support: console, 9P virtual file system, vsock, virtio-net.

Hypervisor-less VIRTIO



Independent Life Cycle Dependent upon Partitioning Technique



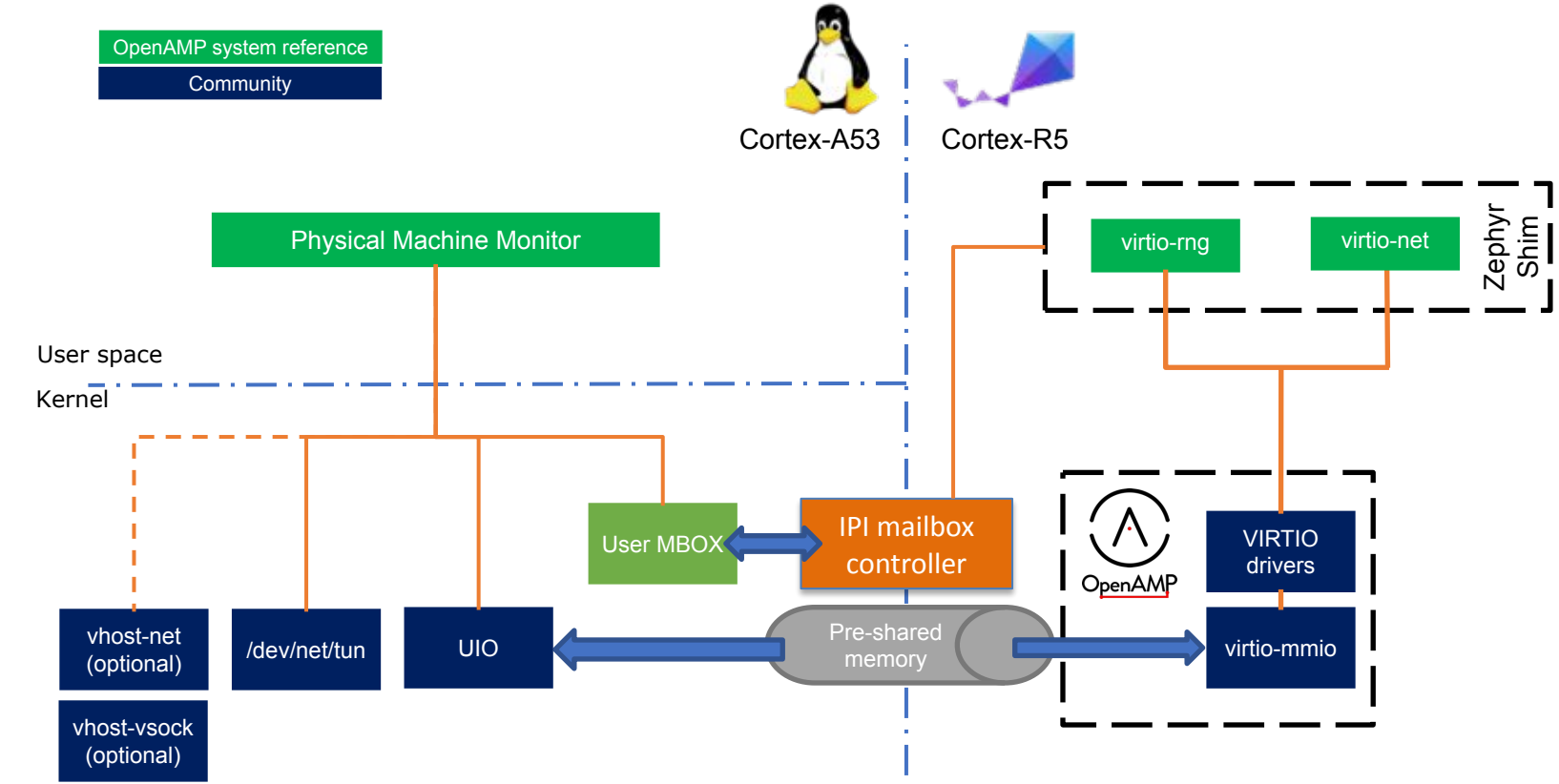
* File system (9P), Console (serial), Network (virtual ethernet), IPC (vsock)

OpenAMP App Services WG

VIRTIO in lib OpenAMP

- VIRTIO MMIO front-end support: network, console, entropy
- Hypervisor-less mode for network and entropy
- Zephyr reference using OpenAMP library for virtio device support (OA System Reference)
  - qemu_cortex_a53 for pure virtio
  - qemu_cortex_r5 for hypervisor-less virtio
- PMM derived from kvmtool (OA System Reference): virtio net, console, entropy, 9P FS, virtual sockets

Demo 4: Hypervisor-less VIRTIO

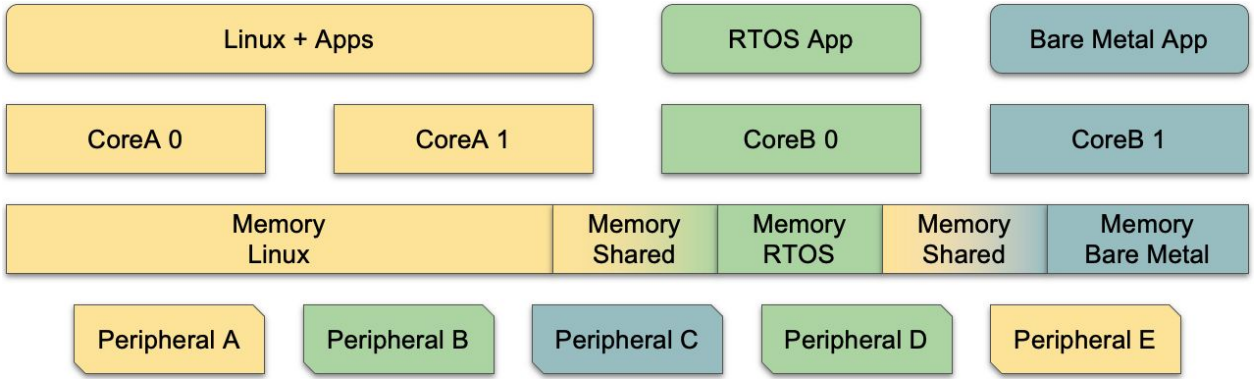


WNDRVR

OpenAMP includes the building blocks for assembling multi-OS systems using standards-based protocols with rich connectivity and device sharing capabilities.

System Devicetrees
Lopper Demo

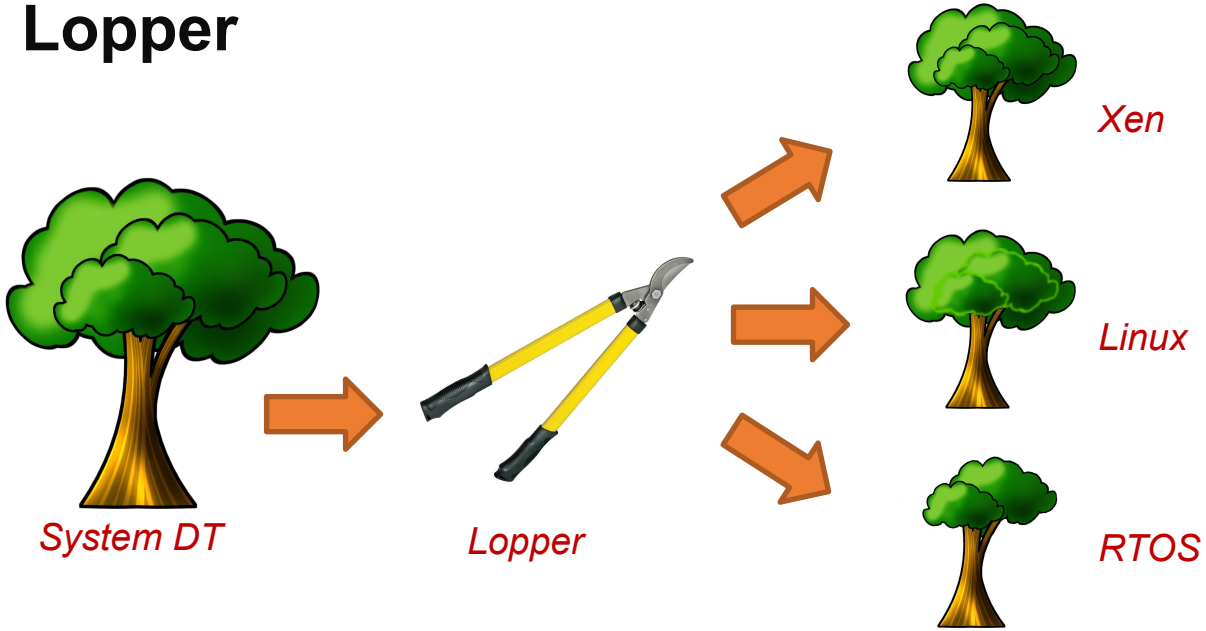
System Devicetree



One true source for all HW resources in a SoC

- *Specification of available resources*
- *Allocation of resources into domains*

Lopper

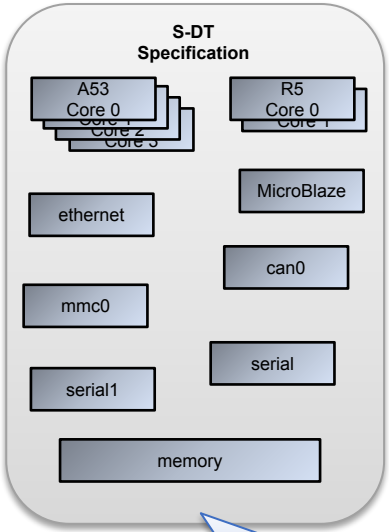


Open source framework to manipulate devicetrees

- *Prune System devicetree into domain-specific devicetrees*
- *Easy to configure to convert, filter and modify devicetrees*
  - *lops: data driven rules using DT or YAML format*
  - *assists: Python code to do more complex transformations*

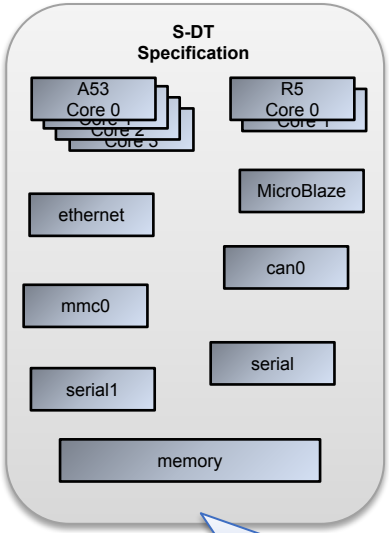
OpenAMP Demo

System DT - Specification



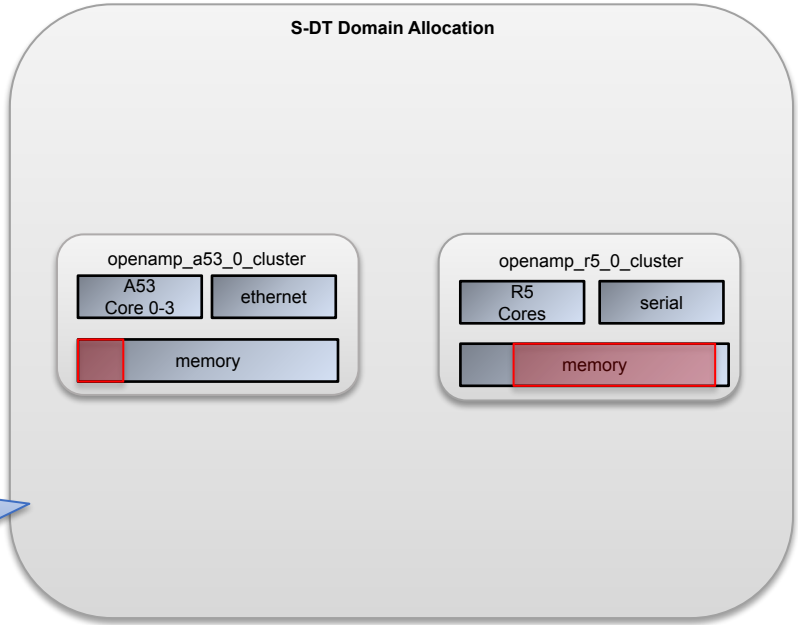
S-DT specification
- HW description before any allocations
- CPU clusters, devices and memory
- Addresses, topologies, …

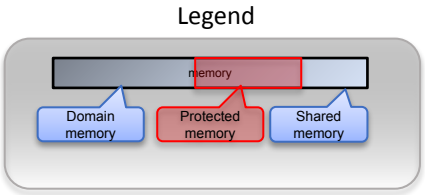
System DT - Allocation



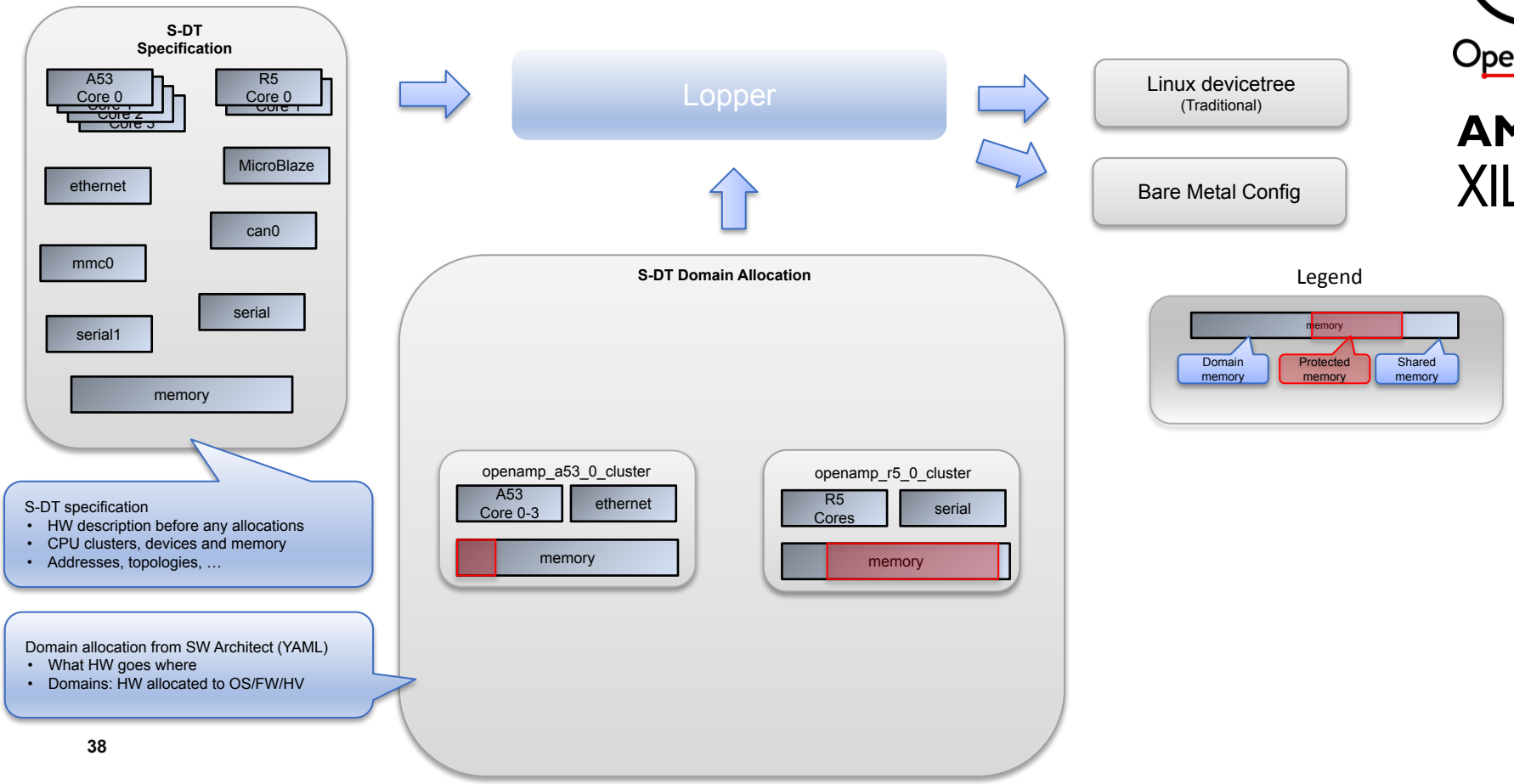
S-DT specification
- HW description before any allocations
- CPU clusters, devices and memory
- Addresses, topologies, …

Domain allocation from SW Architect (YAML)
- What HW goes where
- Domains: HW allocated to OS/FW/HV





System DT - Lopper



OpenAMP Demo

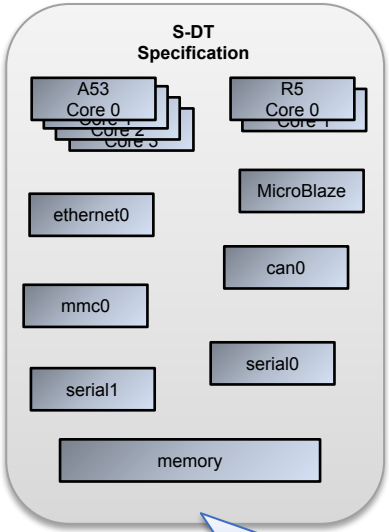
- Self contained docker image to try out
  - See end of webinar for instructions
- Simple yaml file with two domains
  - openamp-overlay-zynqmp.yaml
  - Defines shared virtIO memory for OpenAMP
- Lopper commands that generates output for each domain
  - Traditional devicetree for Linux
  - Config information used by bare metal build system

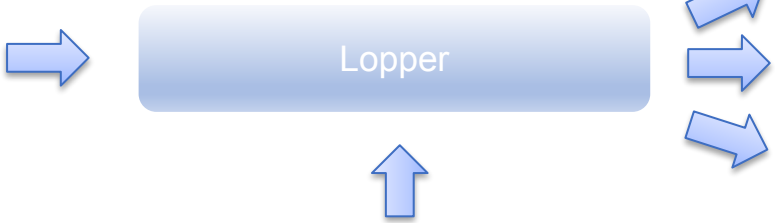
Change in yaml file flows into all domains

AMD
XILINX

Xen Devicetree Demo

System DT – Complex Example

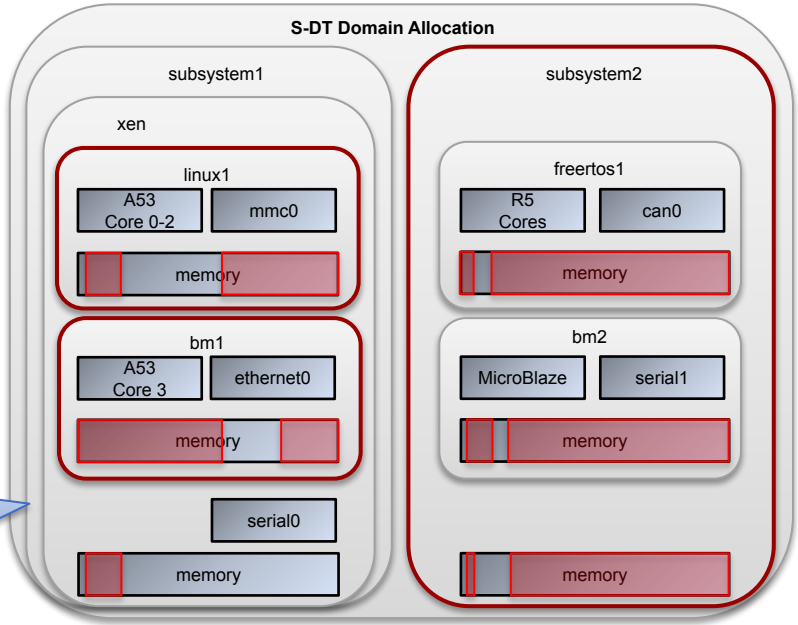




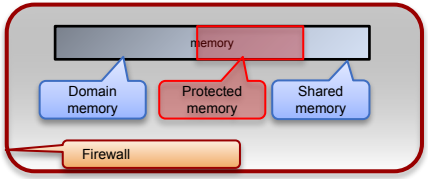
Xen devicetrees & configuration

Linux devicetree (Traditional)

Bare Metal config



Legend



S-DT specification
- HW description before any allocations
- CPU clusters, devices and memory
- Addresses, topologies, …

Domain allocation from SW Architect (YAML)
- What HW goes where
- Subsystems: HW with same lifecycle
- Domains: HW allocated to OS/FW/HV
- Firewalls: Protection between domains

OpenAMP

AMD XILINX

Xen Demo

AMD
XILINX

- Xen is an embedded hypervisor
  - The virtual machines (VMs) / guests are called *domains*
  - Primary use case is to isolate domains from each other
- Devicetrees are used for Xen itself as well as for each domain
  - Creating devicetrees for guest has been done manually
  - Very error prone due to splitting of clocks, etc.
- Lopper automates the creation of Xen devicetrees
  - Takes care of rewriting necessary nodes

Configuring a hypervisor is similar to AMP

Wrap-up

What is Next? 1/2

- Remoteproc & RPMsg
  - Get ZynqMP Remoteproc truly upstream
  - Support of multi VirtIO devices declared in the devicetree
  - RPMsg flow control and other control channel services
  - **possible:** Coprocessor firmware authentication upstream

- Other VirtIO devices
  - Finish the normal and hvl-virtIO services in the OpenAMP library
    - Get into main of open-amp, libmetal, and Zephyr
    - Build all Linux components OE/Yocto, include in CI
    - port to other platforms
  - **possible:** virtio backends on the remoteproc
  - **possible:** In kernel support for remoteproc based virtio
    eg: vsock

What is Next? 2/2

- System devicetree (S-DT)
  - Further Yocto integration
  - Zephyr integration
  - Productize S-DT usage in Xen
  - Added Lopper functionality including DT comparison and verification

- CI
  - Build and test open-amp & libmetal PRs on QEMU & HW
  - Build and test patch series from remoteproc on QEMU & HW
  - Better test coverage

- Documentation:
  - Good start already source (new as of 2022-10)
  - Clean-up and synchronize with 2023.04 release
  - Actual wire protocol specification, independent of kernel or open-amp
  - New resource table proposal
  - Control channel proposal

Key messages

OpenAMP Upstream

Remoteproc and RPMsg are working today from upstream.
Use it for your application or add support for your platform today

Hypervisorless Virtio

Virtio is a powerful standard.
This work enables you to leverage that
standard for AMP systems

System Devicetree

With one single source, you are able to configure
multiple runtimes: Linux, RTOS, Xen and more

Trying it on your own

Try it on your own

- We have a docker container with everything you need to run in QEMU
  - **`docker run -it openamp/demo-lite`**
  - All the details are on the document page

- If you would rather try it manually
  - CI-Build README
  - Zephyr Openamp-system-reference readme
  - HVL-Virtio readme
  - SystemDT demo readme

- Have questions about running the demos AFTER the webinar?
  - Join our OpenAMP community discord
  - Once your in, join the #openamp-community channel
- Have other questions?
  - Join our mailing lists, see openampproject.org

Q&A

Bill Mills,
Linaro

Arnaud Pouliquen,
STMicroelectronics

Tanmay Shah,
AMD Xilinx

Dan Milea,
Wind River

Tomas Evensen,
AMD Xilinx

Bruce Ashfield,
AMD Xilinx

Tammy Leino,
Siemens

Mathieu Poirier,
Linaro

OpenAMP

Video editing: Nathalie Chan King Choy, AMD Xilinx

Thank You

OpenAMP: “**Open A**symmetric **M**ulti-**P**rocessing” Project

Runtime coexistence and collaboration
Runtime hardware resource assignment
Resource sharing and IPC between runtimes
Control mechanisms to start and stop runtimes
Typical system: Linux + RTOS on one system-on-chip

www.openampproject.org.