Linaro Connect
MADRID 2024 | MAY 14-17 2024

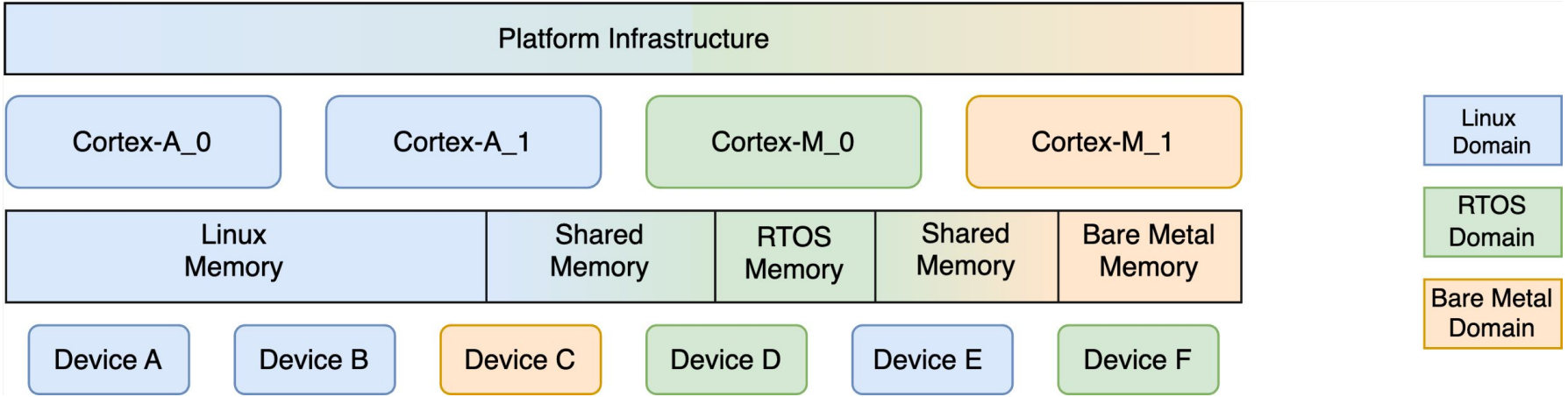
Developing and Deploying Software for Hybrid Systems

Chris Adeniyi-Jones, Principal Software Architect
Arm Ltd

Agenda

- What and why are hybrid systems?
- Investigation
- Options and Challenges
- End Game
- Questions

A Hybrid System



Evolving Edge

Cloud Native | **Cloud Edge/IoT** | **Embedded/IoT** | **Deeply Embedded**

High performance cloud native compute

Optimized compute

Subject to cyber regulations, needs maintaining

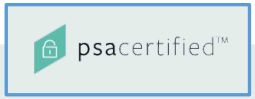
Server/hyperscaler: high performance, portable workflows

Use case optimized efficiency, targeted workflows

Deploy, maintain and enhance	Deploy, maintain and enhance	Deploy, maintain and enhance	Deploy and maintain e.g. SW updates	Deploy and forget

Cloud native workflow scales down to edge server, hardware abstracted and portable but still ‘in the cloud’

Embedded systems scale-up, becoming secure, connected, supporting software updates and continuously maintained



Evolving Edge

Cloud Native | **Cloud Edge/IoT** | **Embedded/IoT** | **Deeply Embedded**

High performance cloud native compute

Optimized compute

Subject to cyber regulations, needs maintaining

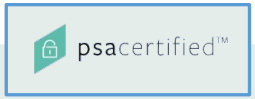
Server/hyperscaler: high performance, portable workflows

Use case optimized efficiency, targeted workflows

Deploy, maintain and enhance	Deploy, maintain and enhance	Deploy, maintain and enhance	Deploy and maintain e.g. SW updates	Deploy and forget

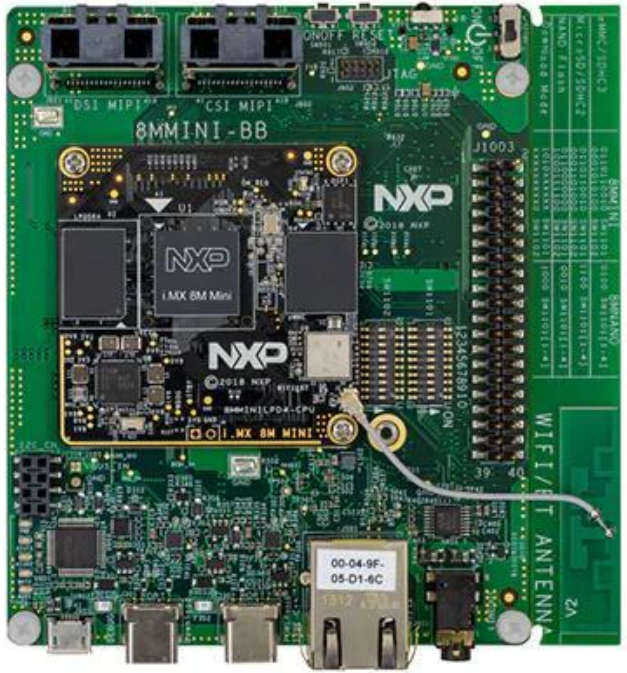
Cloud native workflow scales down to edge server, hardware abstracted and portable but still ‘in the cloud’

Embedded systems scale-up, becoming secure, connected, supporting software updates and continuously maintained



Investigation

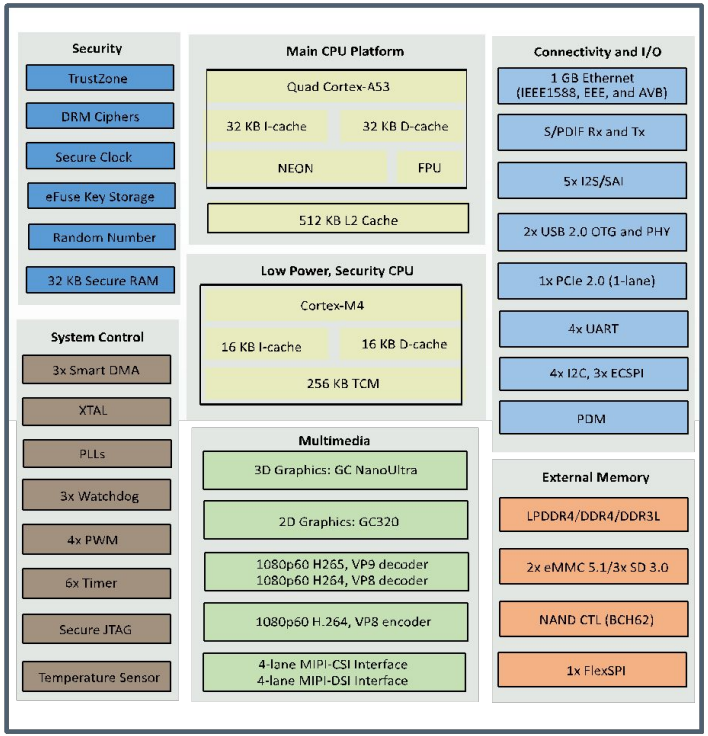
NXP iMX8M-MINI-EVK



NXP iMX8M-MINI-EVK

Quad Cortex-A53

Cortex-M4



1 Gigabit ethernet

UARTs (Console output)

2GB DRAM

System software

Cortex-A53

- Yocto built Linux
  - remoteproc
  - rpmsg
  - Sleep and resume functionality
- k3s
  - Lightweight Kubernetes distribution
  - Can be configured to be more edge-centric - SMARTER

Cortex-M4

- FreeRTOS
- NXP MCUXpresso SDK
- GNU Arm Embedded Toolchain
- rpmsg_lite

https://getsmarter.io

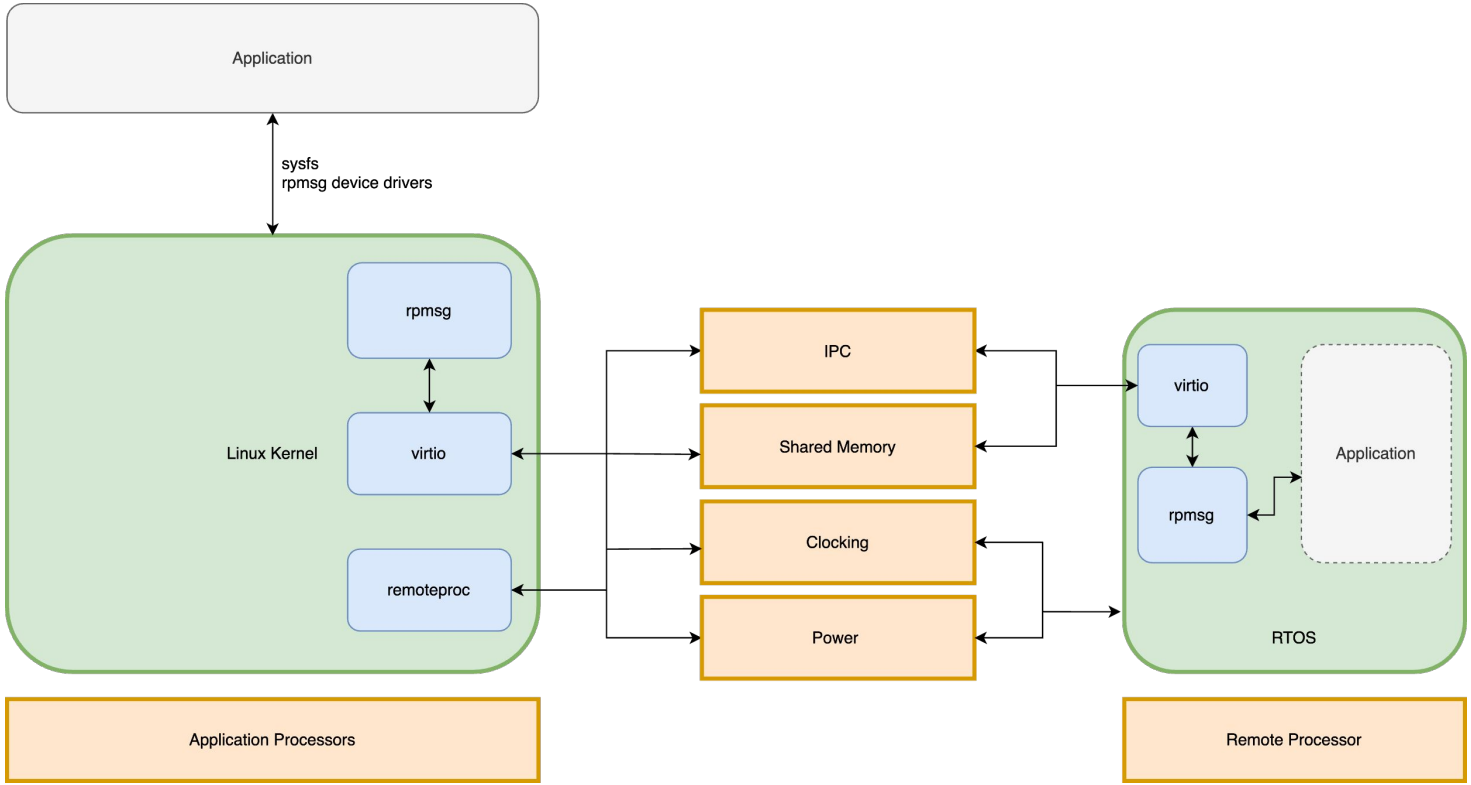
Yocto

- The Yocto Project helps developers build custom embedded Linux distributions, it contains the following key elements:
  - Tools for Linux development
  - Poky, a reference embedded distribution
  - OpenEmbedded build system
- Yocto is widely adopted, supported on many architectures
- Many ODMs, OSVs and chip vendors provide SDKs, BSPs for use with Yocto Project

remoteproc and rpmsg

- The remoteproc framework allows different platforms/architectures to control (power on, load firmware, power off) the remote processors.
- virtio: VirtIO framework that supports virtualization. It provides an efficient transport layer based on a shared ring buffer (vring).
- rpmsg: A virtio-based messaging bus that allows kernel drivers to communicate with remote processors available on the system.
- DTB file configuration required for Cortex-M4, shared memory

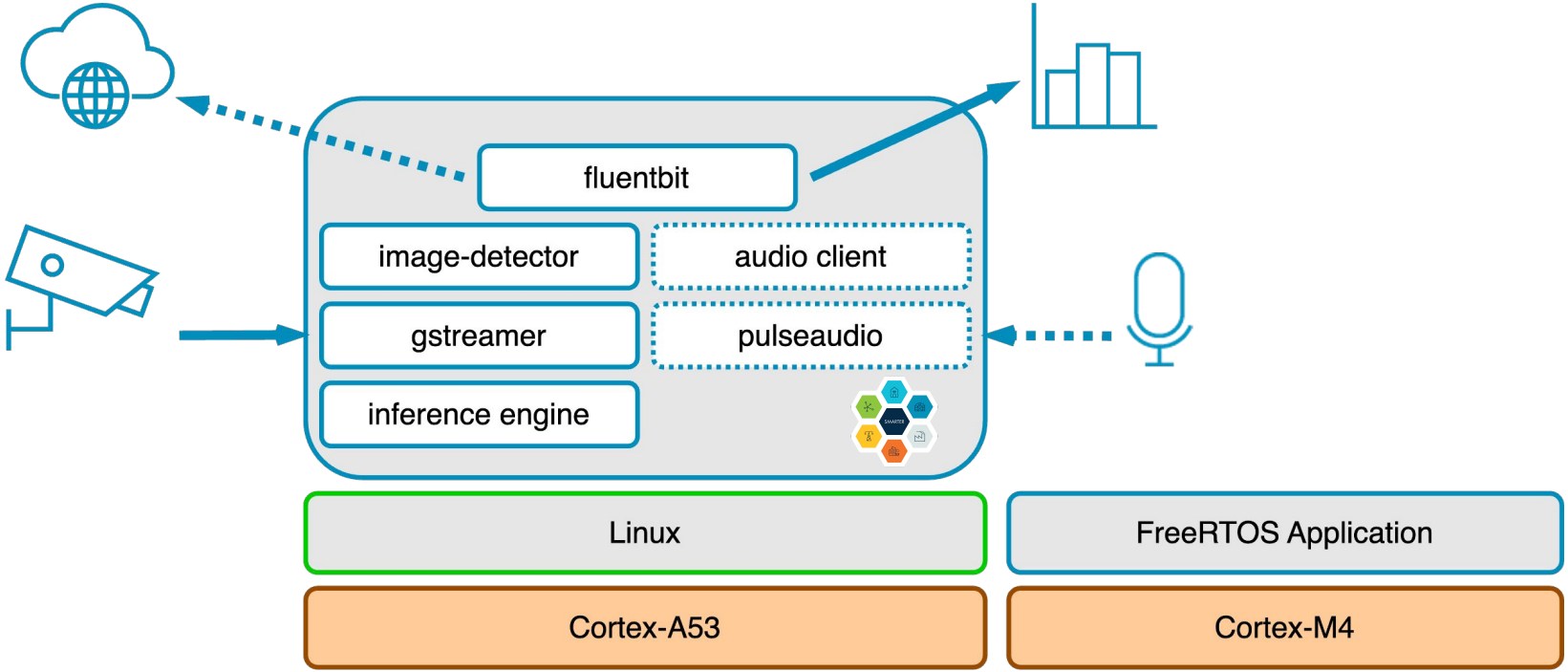
remoteproc and rpmsg



MCUXpresso-SDK

- FreeRTOS
- Peripheral drivers
- Board-support
  - i.MX8M-MINI-EVK
  - Configuration via various files
    - Memory map (TCM, DDR)
    - Peripheral access
  - Supports power-management
    - Sleep/Deep Sleep (which sub-systems remain awake)
    - Wake
  - Support for IPC
    - Read Power state
    - Wake other processors
    - NXP rpmsg_lite implmentation

Our Demo Application



Demo FreeRTOS Application

- Writes a regular timestamp to a console log via UART
- Periodically checks to see if the Cortex-A53 processors are in sleep mode
  - If so, then the application wakes them up
- Very simple proxy.
  - More realistically the application would perform some processing to decide whether to trigger a wake up event. e.g hearing a wake word.

Required patches: Arm Trusted Firmware and Linux kernel

Cortex-M4 would also sleep when Cortex-A53s suspended!

Cortex-M4 could not wake the Cortex-A53s

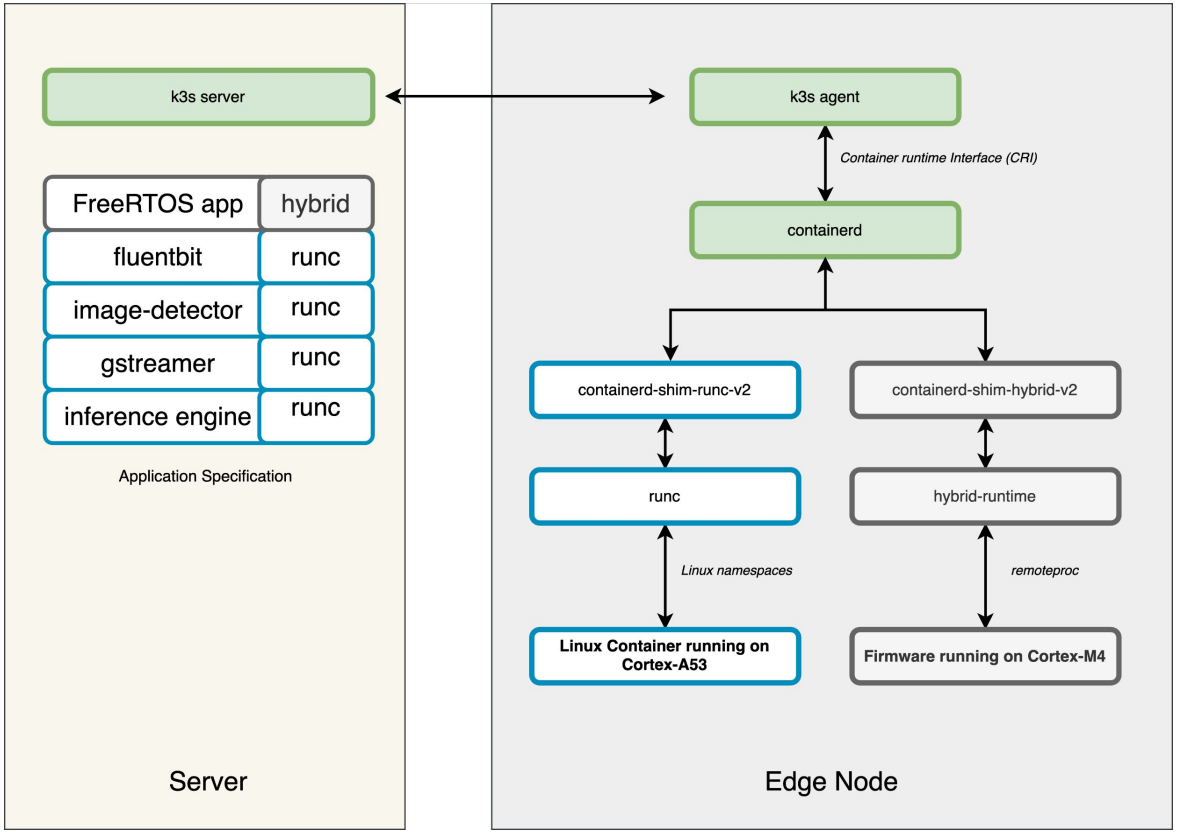
k3s

- k3s is a lightweight Kubernetes distribution
- Open-source
- Automating deployment, scaling, and management of containerized applications
- Compatible with k8s => access to ecosystem of tools
- Can be configured to be more edge-centric – SMARTER
- Applications deployed as Pods
  - Each pod consists of 1 or more containers that execute on the same node

https://getsmarter.io

Hybrid Container runtime

- k3s configured with custom RuntimeClass
- Runtime class attribute added to pod description
- containerd configured to use the container runtime specified by kubelet
- Container specification annotated with Board and MCU type
- Runtime can determine if Cortex-M4 is already executing firmware



Gaps

- Partitioning is split between Linux DTS, FreeRTOS/SDK configuration files
- Configuration is static (boot or compile-time)
- Names in Linux device-trees and device-drivers are not standardized
- General lack of discoverability

Options and Challenges

OpenAMP Project

OpenAMP (Open Asymmetric Multi-Processing) seeks to standardize the interactions between operating environments in a heterogeneous embedded system through open-source solutions for Asymmetric MultiProcessing (AMP).

- Runtime coexistence and collaboration
- Runtime hardware resource assignment
- Resource sharing and IPC between runtimes
- Control mechanisms to start and stop runtimes

A Linaro Community Project: https://www.openampproject.org

System Device Tree

- Extends the Devicetree Specification to handle heterogeneous SoCs with multiple CPUs, possibly of different architectures, as well as the execution domains running on the CPUs.
- Additional bindings for describing multiple distinct CPU clusters in a single heterogeneous SoC, as well as the memories and devices connected to them.
- Additional nodes which define the execution domains running on the SoC and assign hardware resources to them. This is done through a new node, /domains, and additional bindings related to it.
- New tools to convert SDT into DTS (for Linux) or vendor-specific configuration files

A description of the whole system **and** the desired partitioning

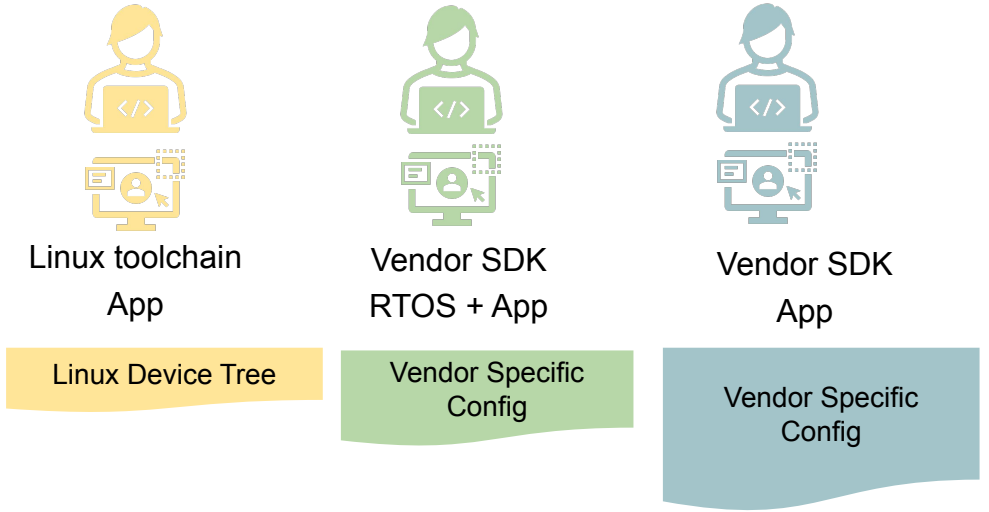
Part of the OpenAMP Project.

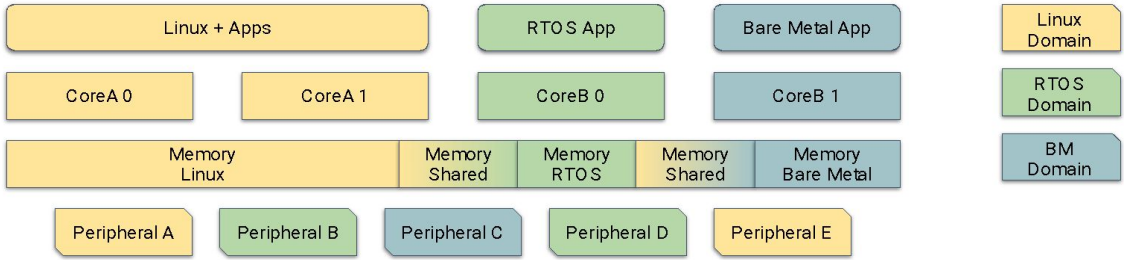
Challenges

- OpenAMP Reference Platforms and examples
- RTOS Support
- System Device Tree WIP
- Device Tree transformation paradigm

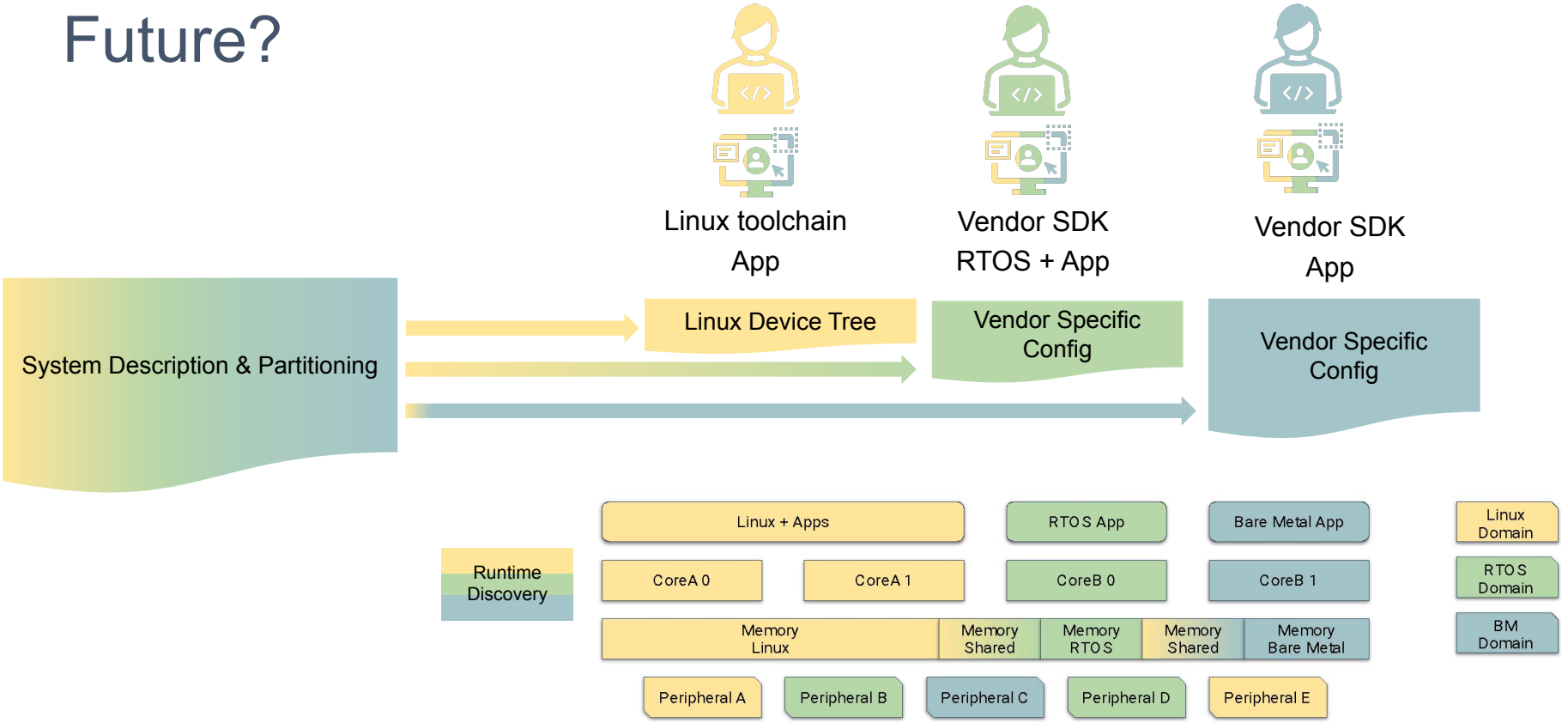
End Game

Now


Linux toolchain
App
Vendor SDK
RTOS + App
Vendor SDK
App
Linux Device Tree
Vendor Specific
Config
Vendor Specific
Config


Linux + Apps
RTOS App
Bare Metal App
Linux
Domain
CoreA 0
CoreA 1
CoreB 0
CoreB 1
RTOS
Domain
Memory
Linux
Memory
Shared
Memory
RTOS
Memory
Shared
Memory
Bare Metal
BM
Domain
Peripheral A
Peripheral B
Peripheral C
Peripheral D
Peripheral E

Future?


Linux toolchain
App
Vendor SDK
RTOS + App
Vendor SDK
App
System Description & Partitioning
Linux Device Tree
Vendor Specific
Config
Vendor Specific
Config
Runtime
Discovery
Linux + Apps
RTOS App
Bare Metal App
Linux
Domain
CoreA 0
CoreA 1
CoreB 0
CoreB 1
RTOS
Domain
Memory
Linux
Memory
Shared
Memory
RTOS
Memory
Shared
Memory
Bare Metal
BM
Domain
Peripheral A
Peripheral B
Peripheral C
Peripheral D
Peripheral E

Questions?

Linaro Connect
MADRID 2024 | MAY 12-17 2024

Thank you