FTF | FREESCALE TECHNOLOGY FORUM
POWERING INNOVATION

# How to Solve Multicore Partitioning Challenges Using Linux® KVM

FTF-NET-F0148

Stuart Yoder
Software Architect

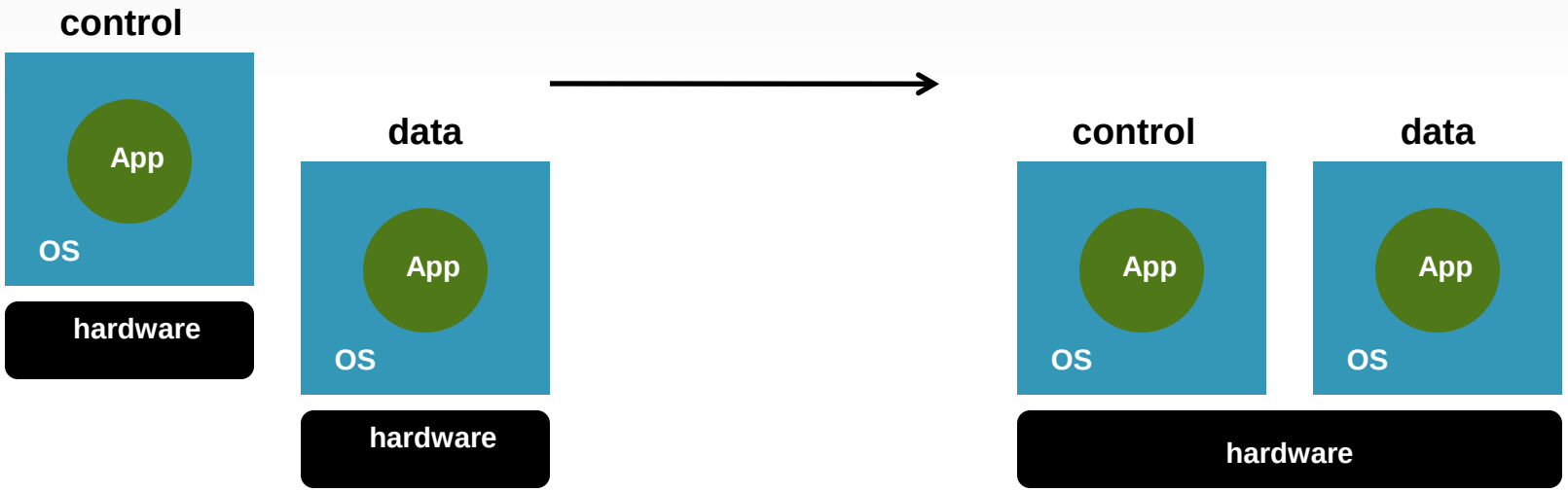
June 2012

# Agenda

- Embedded Consolidation
- Overview of KVM
- Performance Considerations
- Status/Roadmap

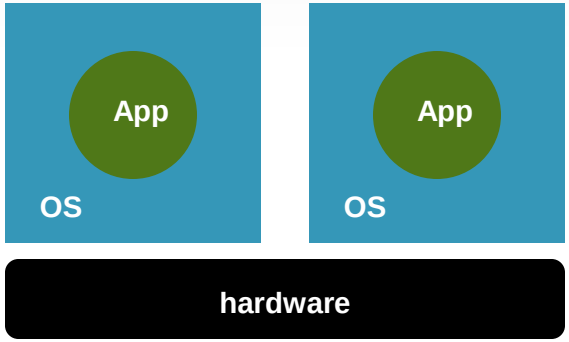
# Overview of Embedded Consolidation

# Consolidation



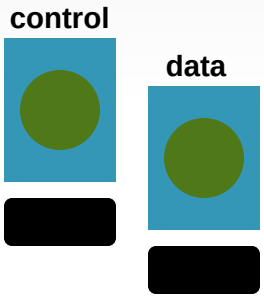

- Take multiple discrete systems/domains on separate processors and consolidate on a single multi-core processor.
- Benefits:
  - Cost savings (bill-of-material, power)
  - Flexibility

# Consolidation…Examples

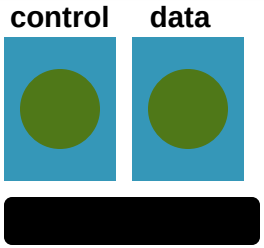



- Migration – move to new hardware, preserve investment in software
  - Run legacy software alongside Linux
- Provide an isolated environment where untrusted software can run
- Dynamic resource management
- High availability – active/standby configuration without additional hardware

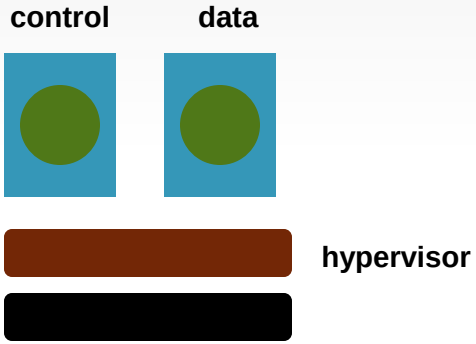
# Consolidation Overview



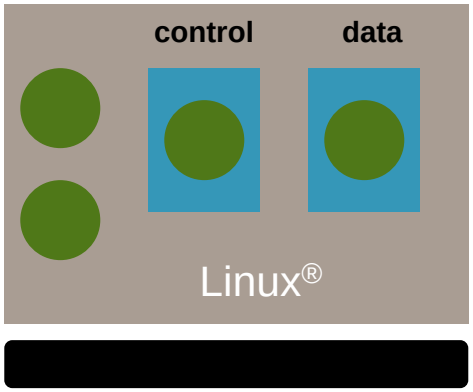

Multiple Processors / Boards



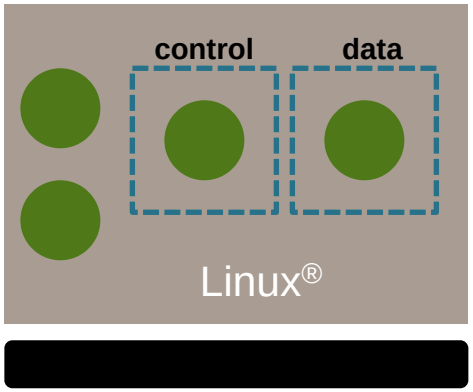

Unsupervised AMP



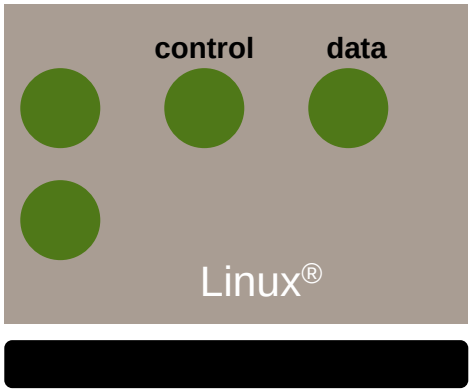

Topaz (Supervised AMP)




KVM




Linux Containers

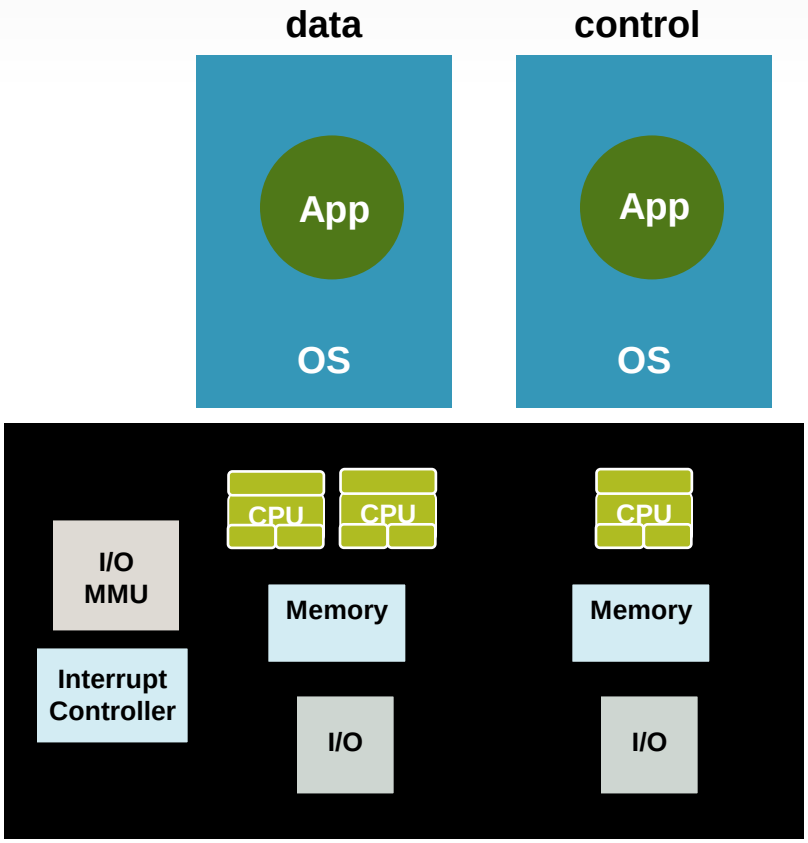



USDPAA

= hardware = OS = App

# Unsupervised AMP

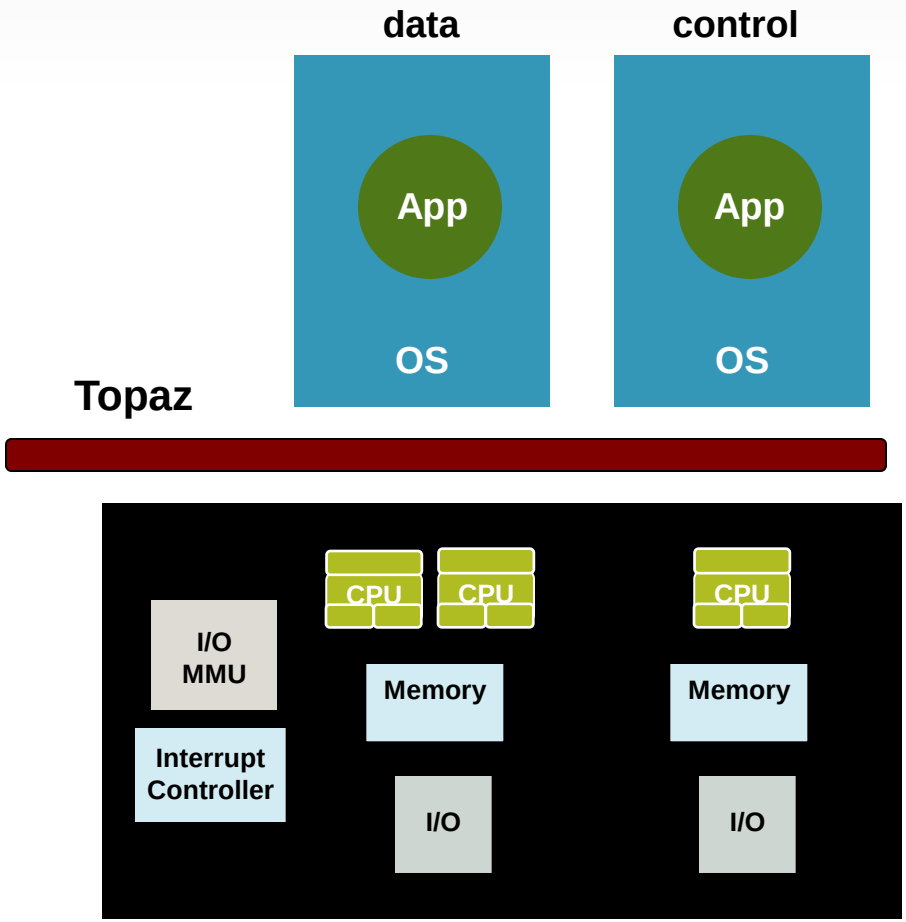
- Good performance, but at cost of fragility and complexity
- Agreement by all OSes required on how memory and I/O devices are partitioned
- Cooperation by all OSes required for initializing and managing global resources
- Complexities: boot sequence, OS reboot, error management, debugging




Unsupervised AMP

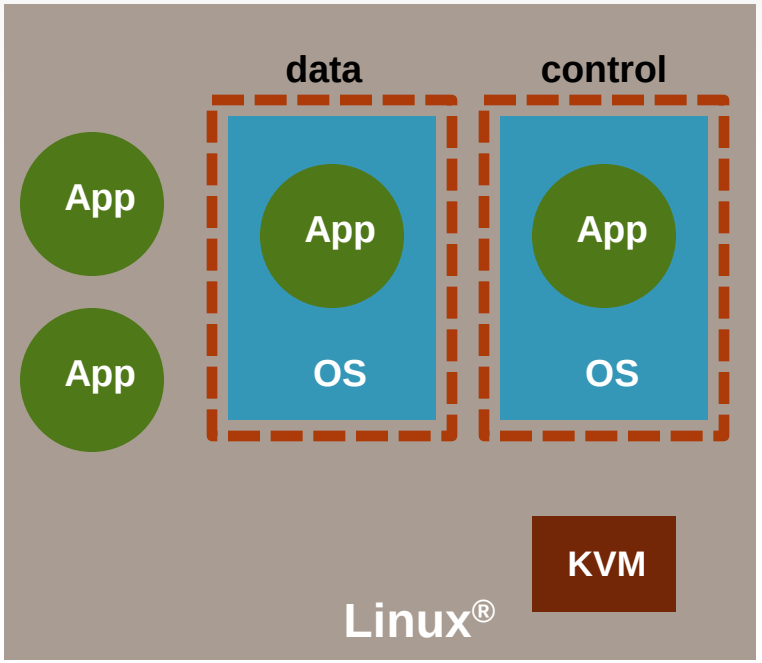
# Freescale Embedded Hypervisor (Topaz)

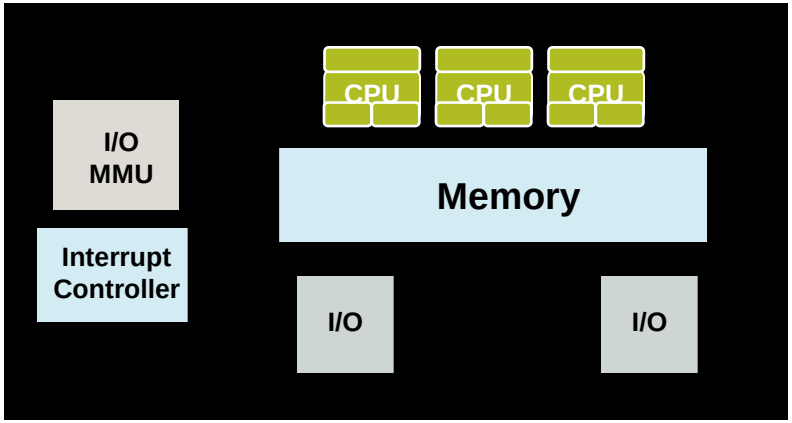
- A lightweight framework for partitioning an SoC
- Partition CPUs, memory, I/O devices (no scheduler)
- Gives you the best of both worlds–performance comparable with AMP with enforced partitioning
- Solves many of the headaches of running multiple unsupervised Oses
- Supported cores: e500mc, e5500



# KVM - Overview

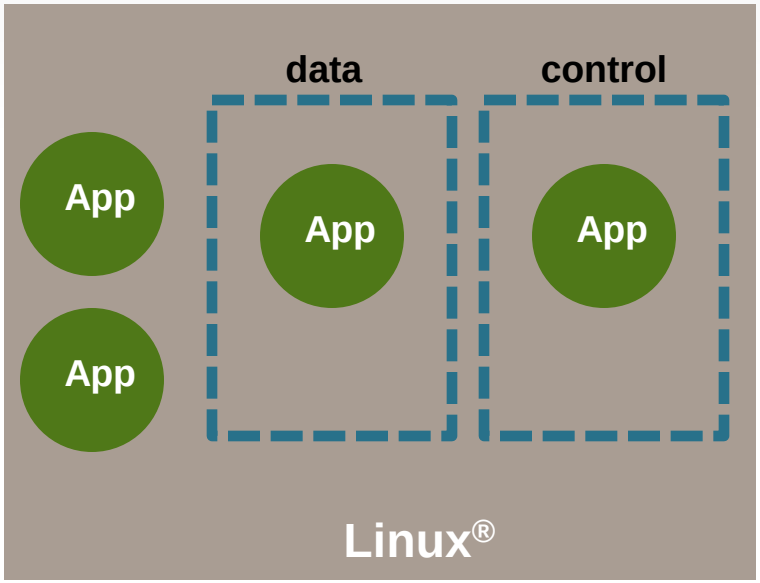
- KVM/QEMU – open source virtualization technology based on the Linux kernel
- Run virtual machines alongside Linux applications
- Virtual Machines (VMs) are fully isolated from rest of the system
- Number of VMs supported is limited only by available resources (CPU cycles, memory)
- Virtual I/O capabilities

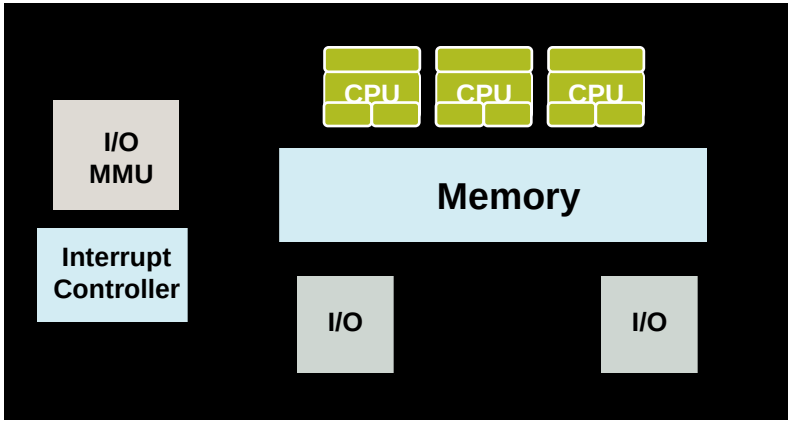


# Linux Containers Overview

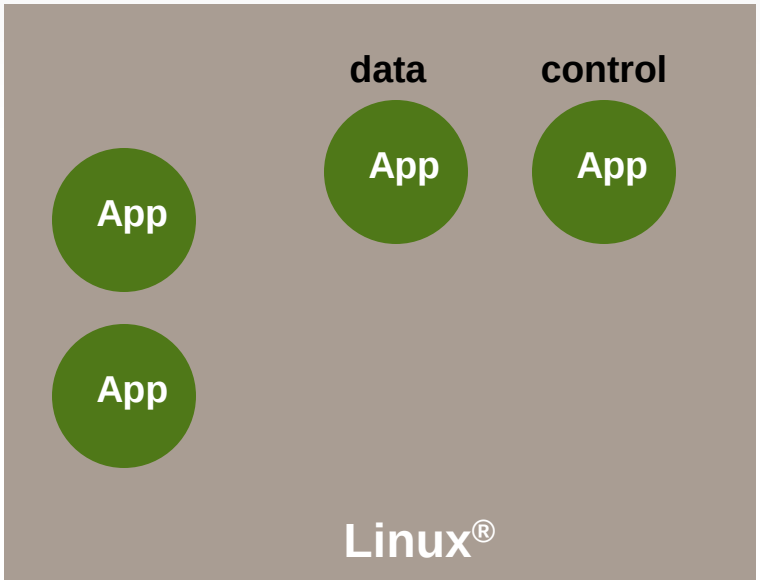
- Linux Containers provides OS-level virtualization
  - Provides low-overhead, lightweight, secure partitioning of Linux applications into different domains
  - Can control resource utilization of domains – CPU, I/O utilization
  - Linux Containers is based on a **collection of technologies** including kernel and user-space components (LXC).

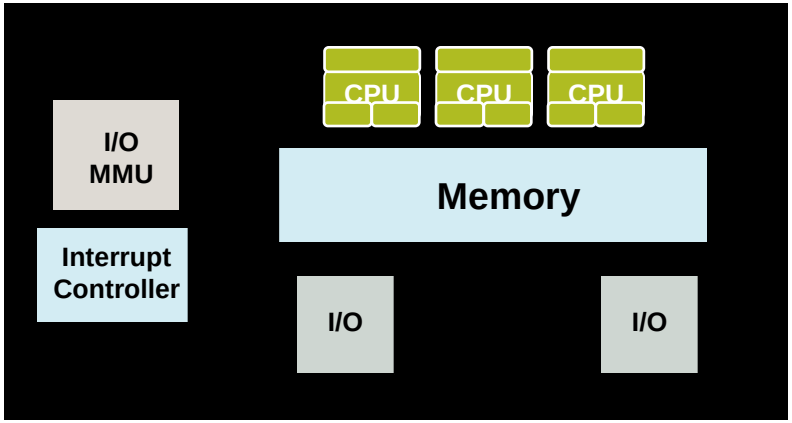


# User Space Data Path Acceleration Architecture (USDPAA)

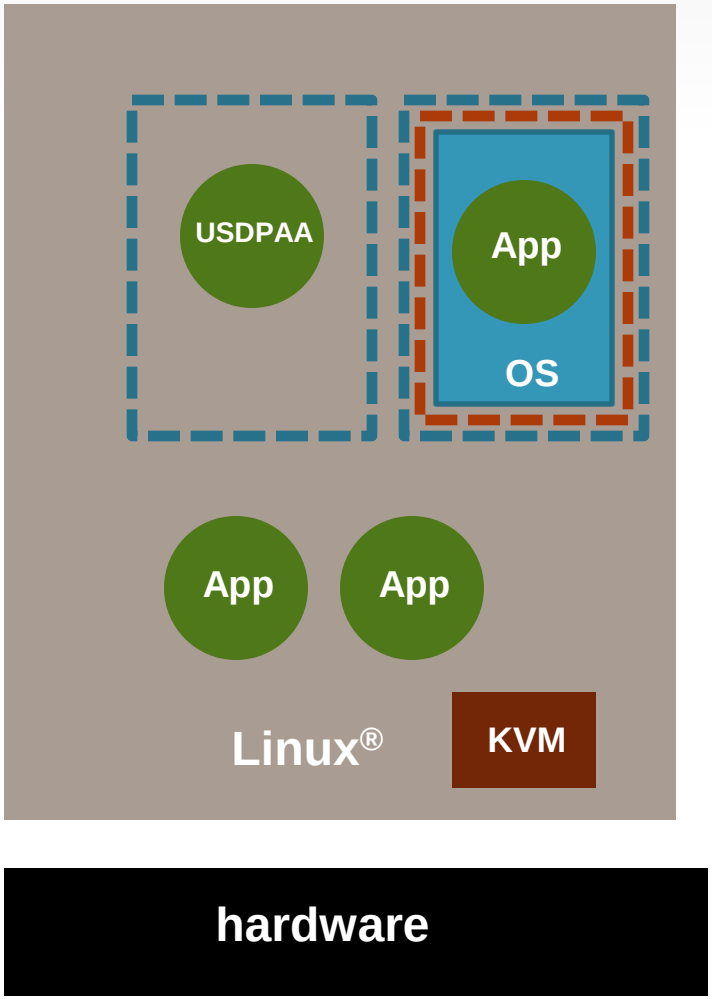
- Infrastructure to build Linux based networking applications
- Bare metal performance with the rich APIs available in Linux



# Combining Technologies

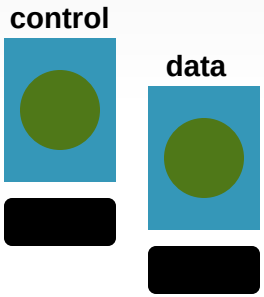
- These technologies are not mutually exclusive:
  - Run USDPAA on a Linux guest on Topaz
  - Run USDPAA in a Linux container
  - Run a KVM virtual machine in a Linux container



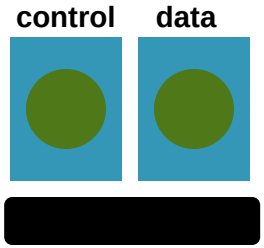
# Consolidation Technologies-- Review

- Unsupervised AMP
  - High-performance– at cost of fragility, complexity
- Topaz
  - Static partitioning of hardware
- Linux containers
  - Partitioning of Linux applications
- KVM
  - Virtual machines in a Linux environment

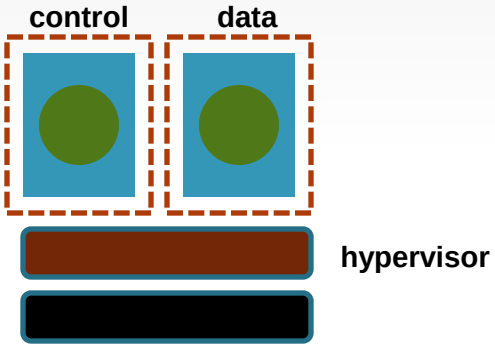
# Consolidation Overview



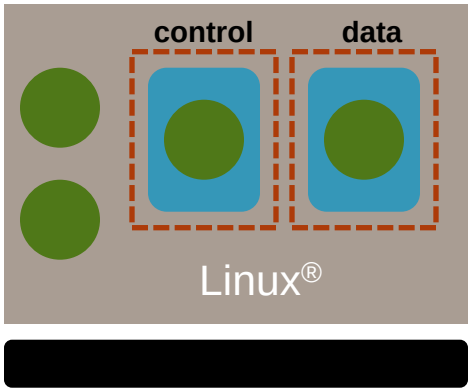

Multiple processors/boards



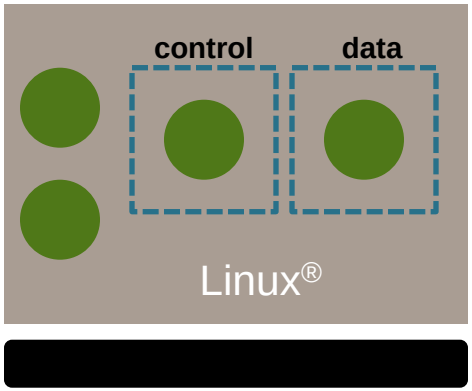

Unsupervised AMP



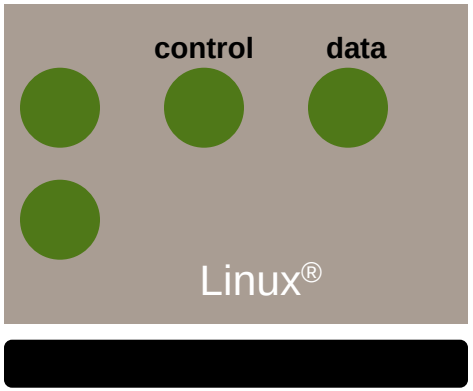

Topaz (Supervised AMP)




KVM




Linux Containers

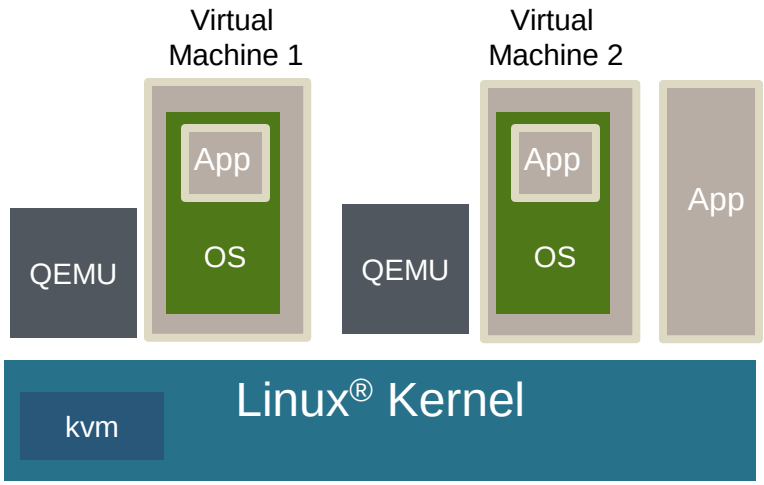



USDPAA

# KVM (Kernel-based Virtual Machine)

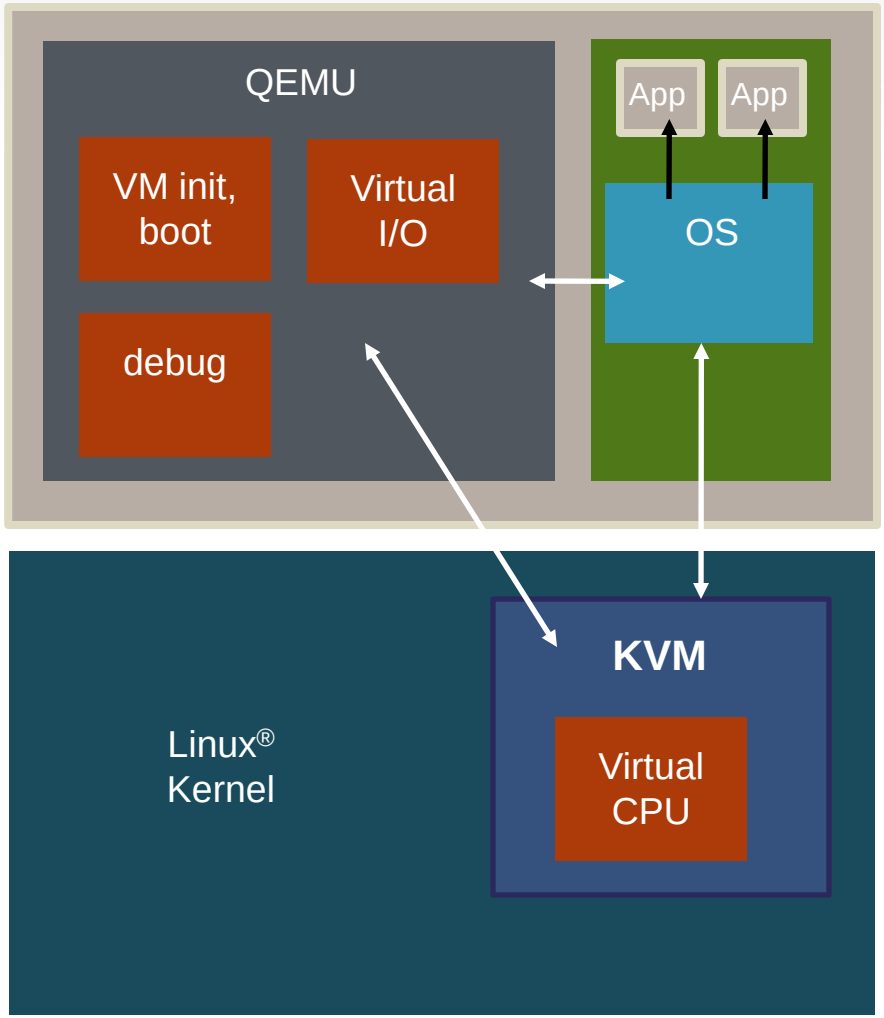
# KVM - Overview

- KVM/QEMU– open source virtualization technology based on the Linux kernel
- Supports e500v2, e500mc, e5500 CPUs
- No or minimal OS changes required
- Virtual I/O – virtual disk, network interfaces, serial
- Direct/pass thru I/O – assign SoC devices to partitions (some limitations)
- ePAPR compliant
- e500v2 uses paravirtualization (OS modifications) for improved performance



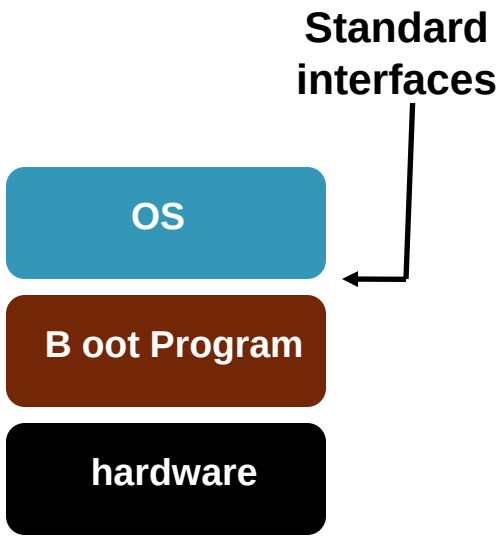
# KVM/QEMU – Overview

- QEMU provides
  - Virtual machine setup
  - Initialization
  - Memory allocation
  - Virtual I/O services
  - Debug stub
- KVM provides
  - Virtual CPU services
  - API used by QEMU (see Documentation/kvm/api.txt)
- Kernel schedules VMs



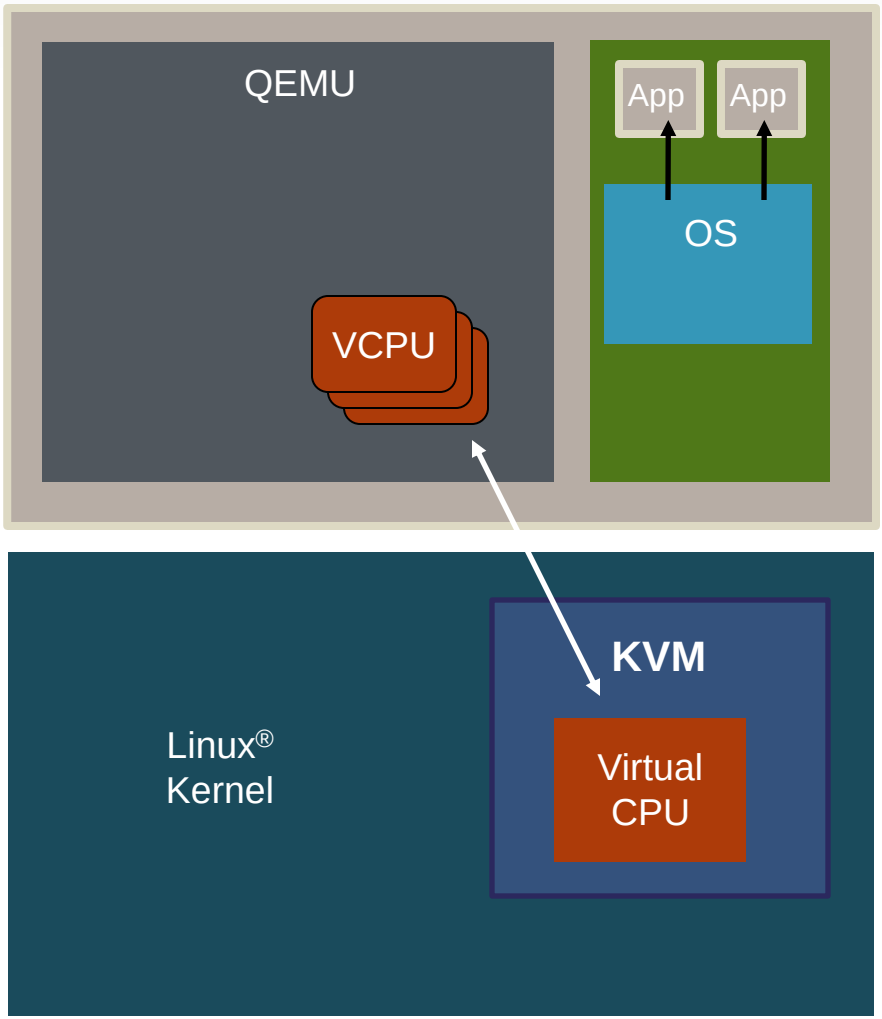
# ePAPR Standard

- Embedded Power Architecture® Platform Requirements (ePAPR)
- A platform standard from Power.org
- Standard for how to boot an embedded OS
  - Device tree standards
  - Multi-CPU boot
  - Definition of initial state of system
- Version 1.1 released in 2011
  - Virtualization extensions-- /hypervisor node, hcall ABI, set of hcall APIs
  - Hardware thread representation



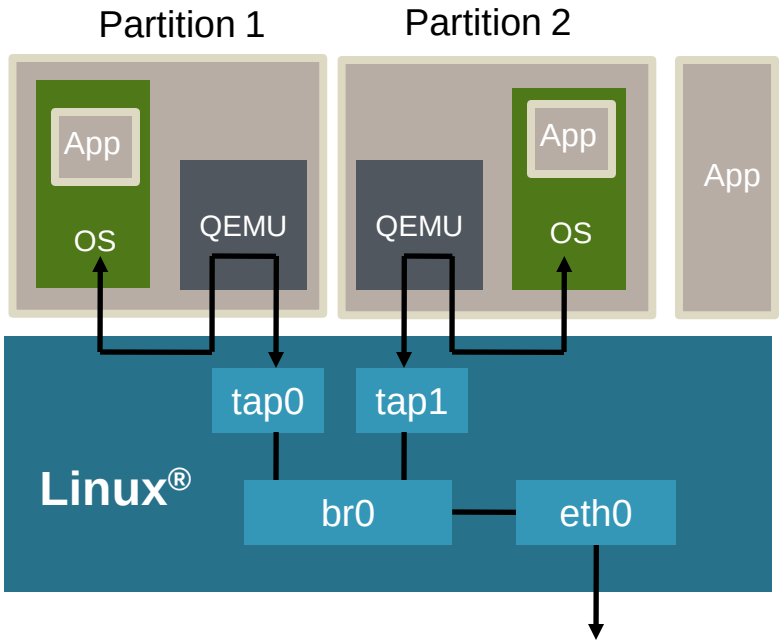
# Multiple Virtual CPUs (VCPUs)

- Each CPU in a virtual machine is Linux thread (created by QEMU)
- Full capabilities of the Linux scheduler can be used to manage VCPUs/threads
  - CPU affinity
  - priority
  - Isolcpus (Isolate CPUs)



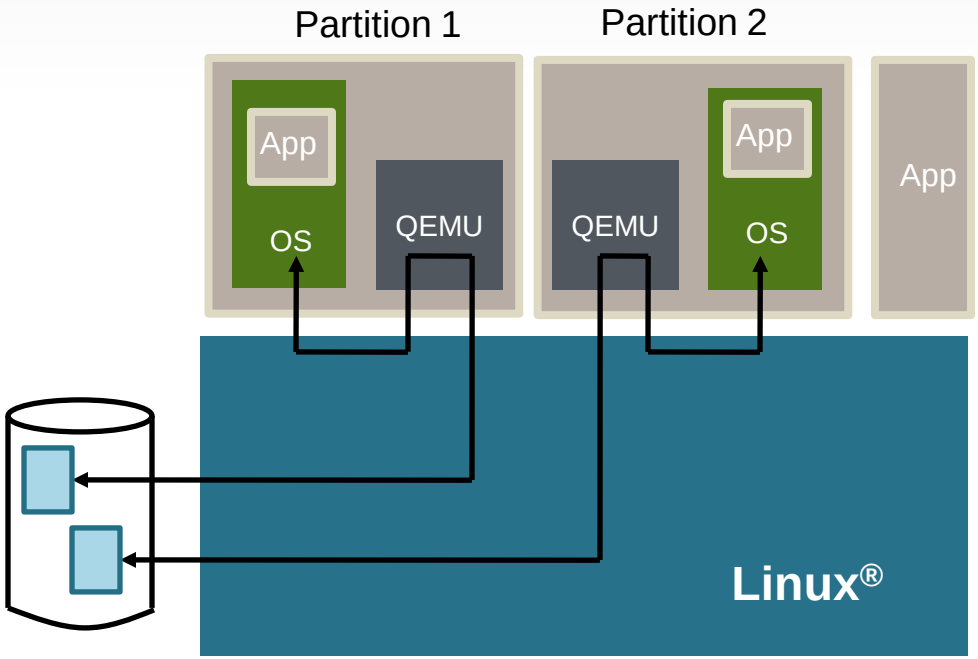
# Virtio Networking

- Enables sharing of host network interfaces
- Host
  - Bridge (virtual switch) is connected to physical host interface
  - QEMU uses tun/tap device connected to the bridge
- Guest
  - Each guest sees a private “virtio” network device on PCI bus
  - Virtio network driver is needed in guest



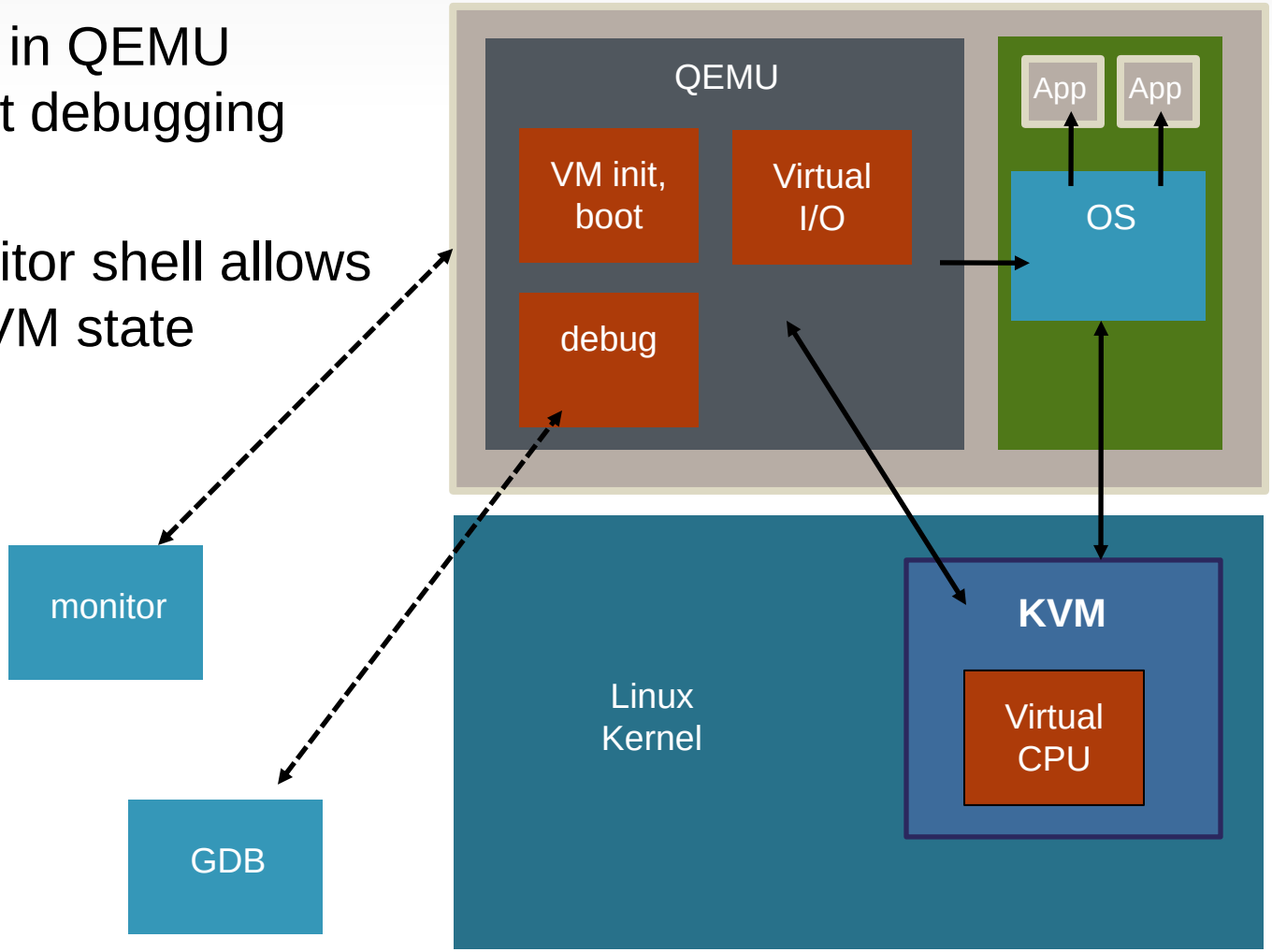
# Virtio Block

- Give each guest a private storage device
- Virtual disk could be single binary image on host file system or logical volume on the host's disk



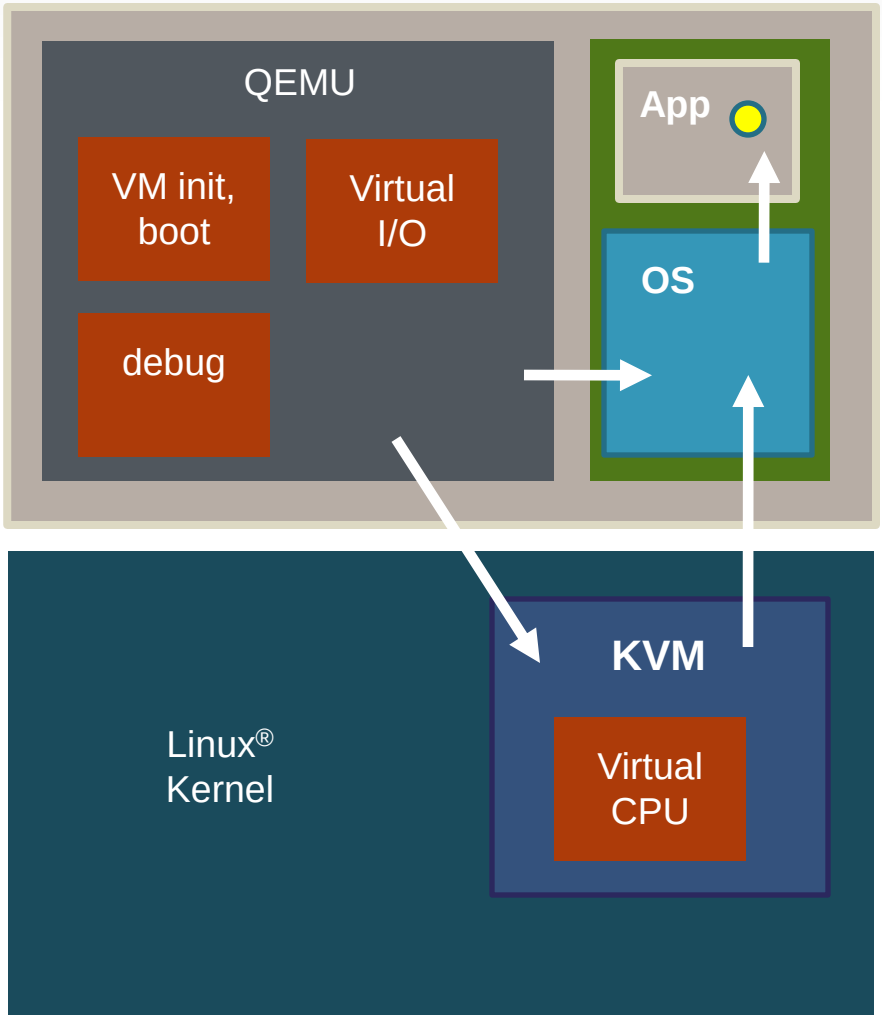
# Debugging

- Debug stub in QEMU allows guest debugging using GDB
- QEMU monitor shell allows examining VM state



# KVM Performance Considerations

# KVM Execution Flow



# CPU Performance Considerations

- The performance overhead when running on a hypervisor is workload dependent

- Sources of CPU overhead when running under a hypervisor
  - Privileged operations
    - Instructions– e.g. TLB operations (tlbwe, tlbilx, tlbsx)
    - Privileged SPRs– e.g. DEC, timer control registers
  - Exceptions – Decremeter, TLB misses, DSI/ISI, external interrupts, etc.
  - Emulated I/O accesses
  - Hypercalls
  - Scheduling / Context switches

# Privileged Instructions

**e500mc/e5500**

- ehpriv
- msgclr
- msgsnd
- rfci
- rfdi
- rfmci
- tlbivax
- tlbre
- tlbsx
- tlbsync
- tlbwe
- tlbilx

**e6500**

- ehpriv
- msgclr
- msgsnd
- rfci
- rfdi
- rfmci
- tlbivax
- tlbre
- tlbsx
- tlbsync

With LRAT **tlbwe** and **tlbilx** can be executed by guest OS

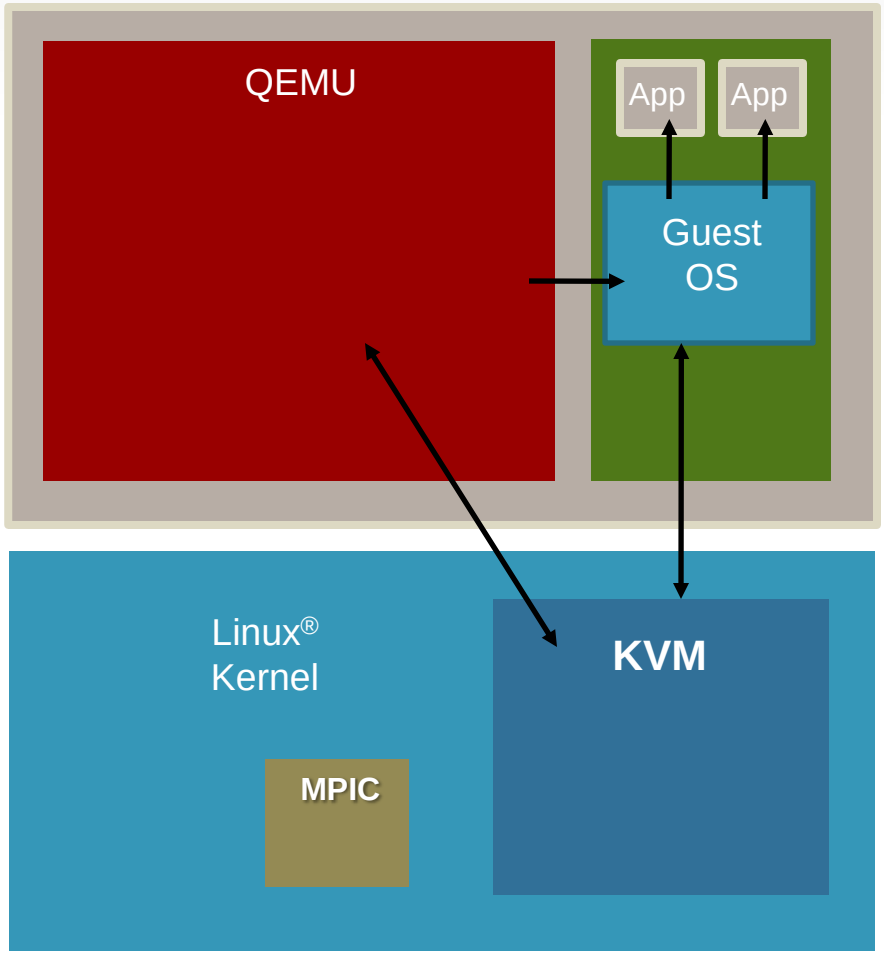
# Privileged SPRS

- BUCSR
- CDCSR0
- CSRR0
- CSRR1
- DAC1
- DAC2
- DBCR0
- DBCR1
- DBCR2
- DBCR4
- DBSR
- DBSRWR
- DEC
- DECAR
- DSRR0
- DSRR1
- HID0
- IAC1
- IAC2
- IVOR13
- IVOR14
- IVOR0
- IVOR1
- IVOR2
- IVOR3
- IVOR4
- IVOR5
- IVOR6
- IVOR7
- IVOR8
- IVOR9
- IVOR10
- IVOR11
- IVOR12
- IVOR13
- IVOR14
- IVOR15
- IVOR35
- IVOR36
- IVOR37
- IVOR38
- IVOR39
- IVOR40
- IVOR41
- IVPR
- L1CSR0
- L1CSR1
- L1CSR2
- L2CAPTDATAHI
- L2CAPTDATALO
- L2CAPTECC
- L2CSR0
- L2CSR1
- L2ERRADDR
- L2ERRATTR
- L2ERRCTL
- L2ERRDET
- L2ERRDIS
- L2ERREADDR
- L2ERRINJCTL
- L2ERRINJHI
- L2ERRINJLO
- L2ERRINTEN
- MCAR
- MCSR
- MCSRR0
- MCSRR1
- MMUCFG
- MMUCSR0
- SPRG9
- TBL(W)
- TBU (W)
- TCR
- TLB0CFG
- TLB1CFG
- TSR

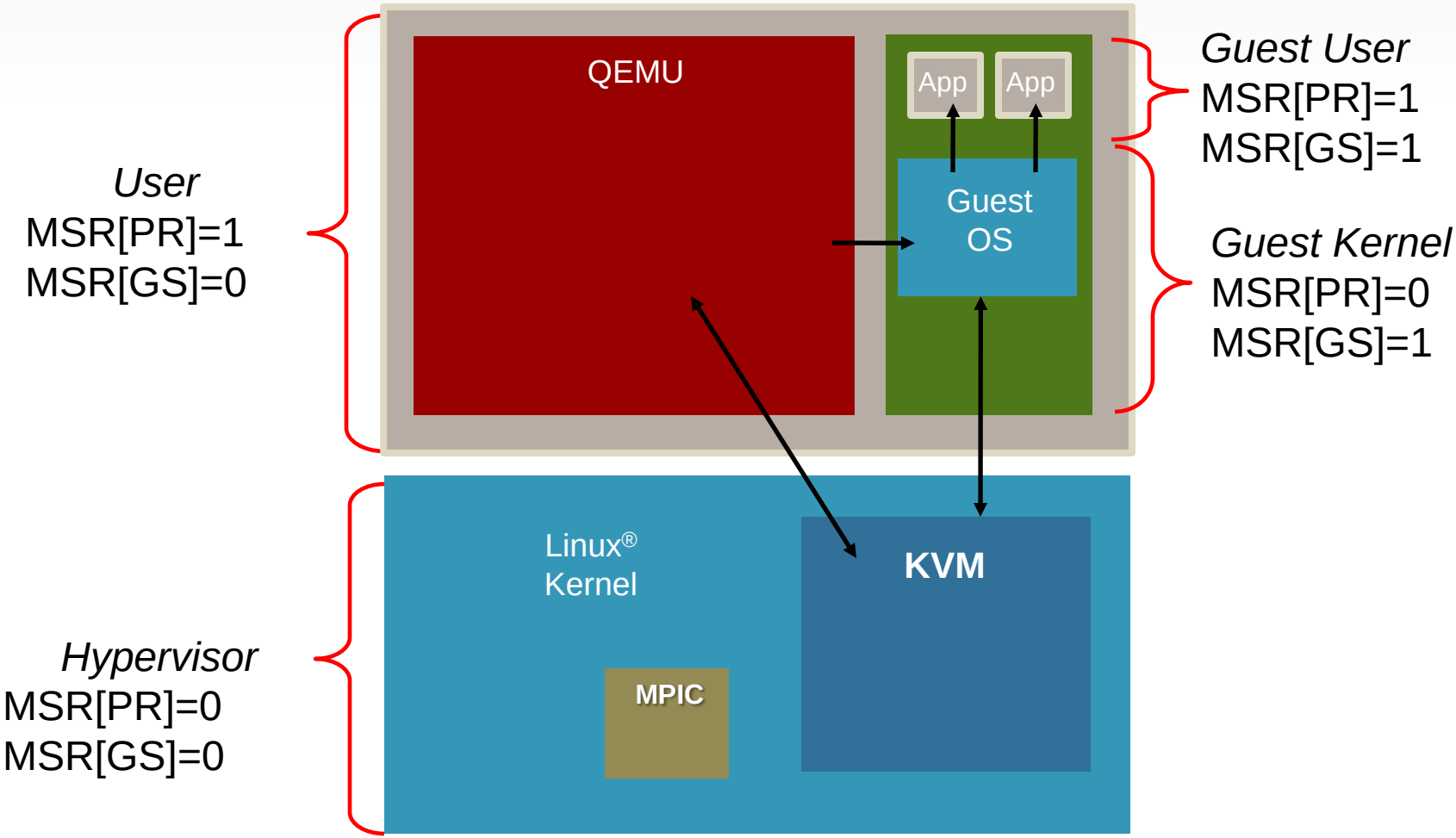
# KVM – e500v2

User mode
MSR[PR]=1

Kernel mode
MSR[PR]=0



# KVM – e500mc/e5500



# KVM Status/Roadmap

# KVM - Considerations

- Use cases
  - Need to run OS in addition to Linux on the same system
  - Need to run older version of Linux
  - Sandboxing untrusted software
  - Fault isolation – a guest kernel crash will not affect rest of system

- Other considerations
  - Linux is the hypervisor
  - Performance
  - Real time / determinism

# KVM Status

- KVM/QEMU are available in Freescale QorIQ SDKs
- Supports e500v2, e500mc, e5500
- New capabilities in SDK 1.2 (June 2012)
  - SMP (multiple CPUs in a VM)
  - 64-bit support
  - Support for memory allocation by hugetlbfs
  - Improved error management

# KVM Roadmap

- Upstream
  - All KVM and QEMU development will go upstream

- Plans
  - Current direct map support is preliminary-- no IOMMU support
    - Plan is to move to a new Linux infrastructure for doing user space I/O called “vfio”, which includes IOMMU support.
  - E6500 CPU Support
    - Including LRAT (Logical to Real Address Translation)
  - Performance improvements
  - Virtual Time in guests
  - Future:  Direct external interrupts, Real time,  virtual machine management


freescale™