
PCI
SIG®

I/O Device Virtualization Overview

Richard Solomon

IC Design Engineer

LSI Logic

Outline

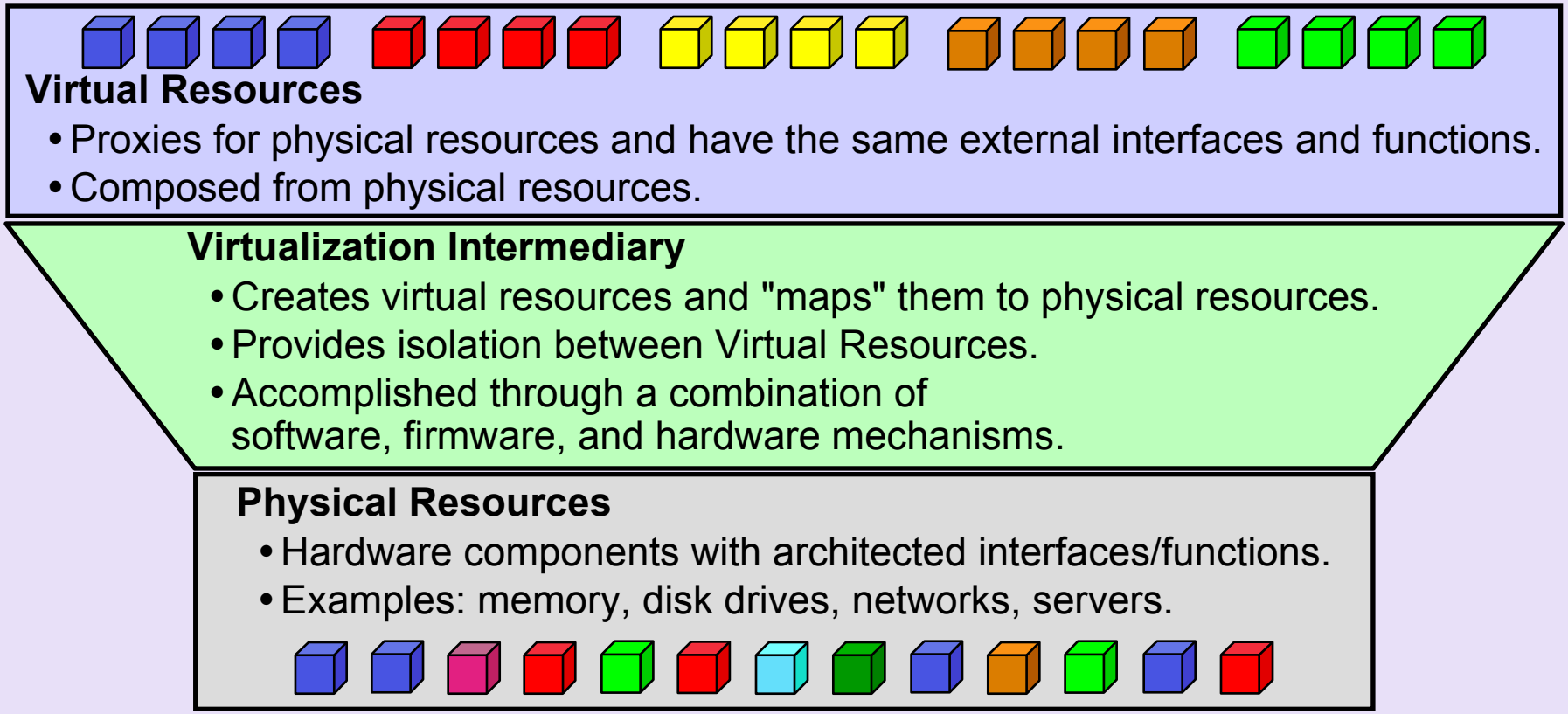
- Overview
- Scope of Work
- Examples of Key Requirements Documented in the Spec Template
- Current Status

Workgroup Members

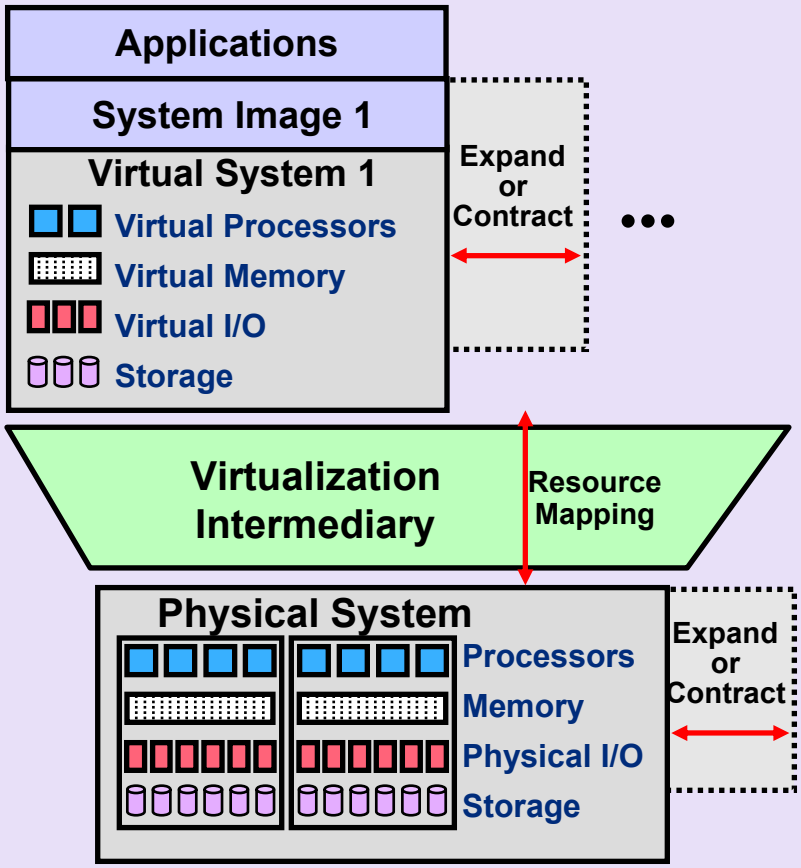
- Adaptec
- AMD
- ATI
- Broadcom
- Emulex
- HP (co-chair)
- IBM (co-chair)
- IDT
- Intel
- LSI
- Microsoft
- Neterion
- NextIO
- NVidia
- PLX Technology
- Qlogic
- Sun
- StarGen
- vmWare (EMC)

What is Virtualization?

- ***Virtualization*** *- The division of a physical system's processors, memory, I/O, and storage, where each such set of resources operates independently with its own System Image instance and applications.*

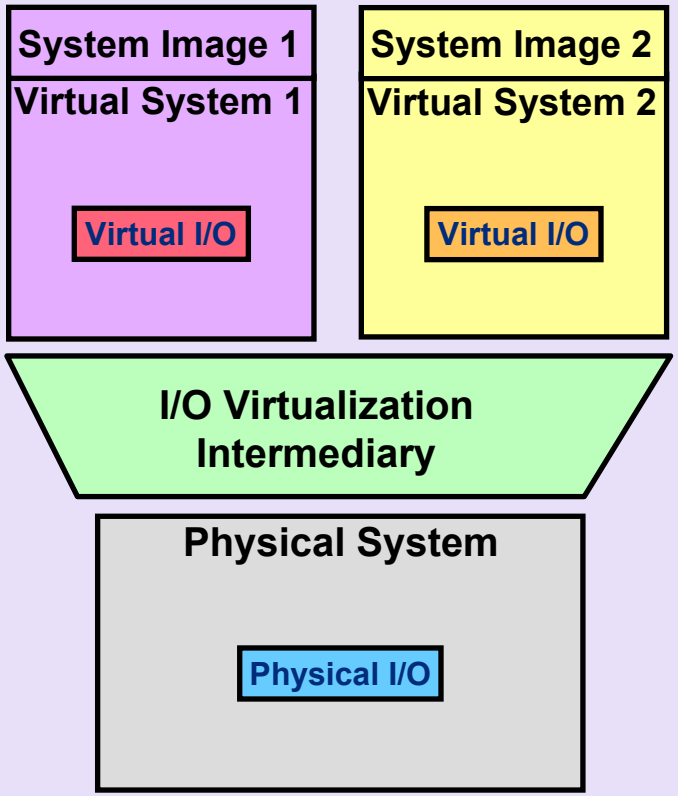


Terminology



- ***System Image (SI)*** *- A software component, such as a general or special purpose Operating System, to which specific virtual and physical devices can be assigned.*
- ***Virtualization Intermediary (VI) -*** *A component that manages the allocation of resources to a System Image and isolates resources assigned to a System Image from access by other System Images.*
- ***Virtual System (VS)*** *- The physical or virtualized resources necessary to run a single System Image instance. The virtual resources typically consist of: processors, memory, I/O, and storage.*

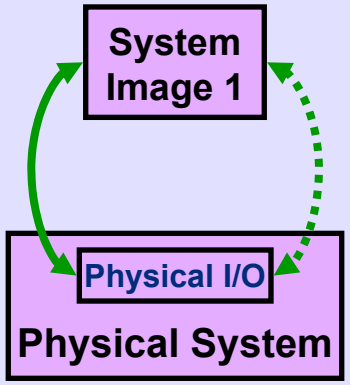
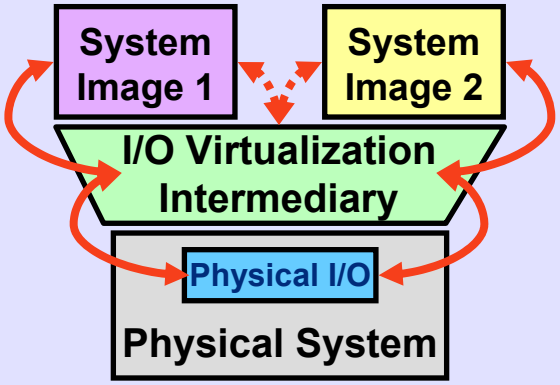
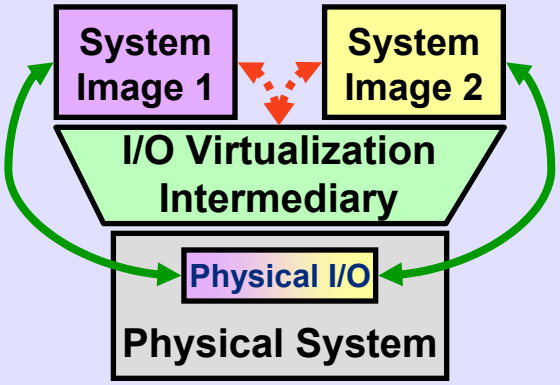
Terminology (cont.)



- ***I/O Virtualization (IOV)*** *- The capability for a single physical I/O unit to be shared by more than one System Image.*
- ***I/O Virtualization Intermediary (IOVI)*** *- Software or firmware that is used to support IOV by intervening on one or more of the following: Configuration, I/O, and Memory operations from a System Image; and DMA, completion, and interrupt operations to a System Image.*

Several mechanisms have evolved in the industry, each with varying performance and function characteristics.
Strictly for background, some of the key ones will be covered next.

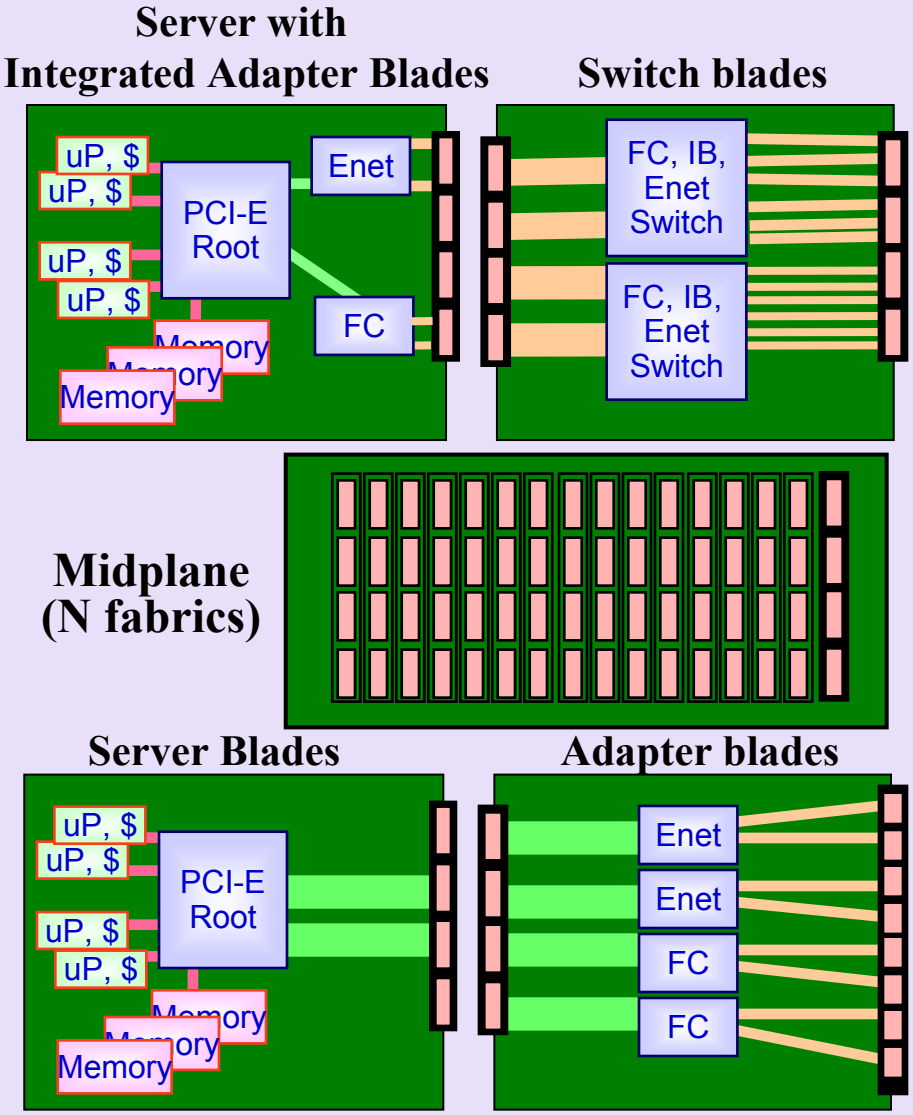
Mechanisms within a Single Physical System

	Dedicated Adapter (No Virtualization)	Adapter Shared Through Intermediary	Natively Shared Adapter
Graphic Depiction			
Intermediary Role	None	Virtualizes physical I/O by intervening on configuration and data transfer operations	Manages assignment of Virtual Resources by intervening on configuration operations
Configuration Operation Path	SI direct to Adapter	VI serves as proxy (SI to VI; VI to Adapter)	VI serves as proxy (SI to VI; VI to Adapter)
Data Transfer Operation Path	SI direct to Adapter	VI serves as proxy (SI to VI; VI to Adapter)	SI direct to Adapter

Terminology (cont.)

- **Configuration Time VI Involvement** – Refers to the operations performed by the VI to configure a VE and associate it with an SI.
- **Run-Time VI Involvement** – Refers to the intervention by a VI on data movement and interrupt operations performed between an SI and a physical PCI Endpoint (versus a Virtual Endpoint).
- **Native based PCI IOV** – An IOV mechanism where:
  - ✓ A physical PCI Endpoint supports one or more Virtual Endpoints.
  - ✓ Each VE is associated with an SI.
  - ✓ Each VE is created through IOVI issued configuration operations.
  - ✓ The Virtual System may support mechanisms that allow data movement and interrupt operations to be performed directly between an SI and the VE, without VI involvement.

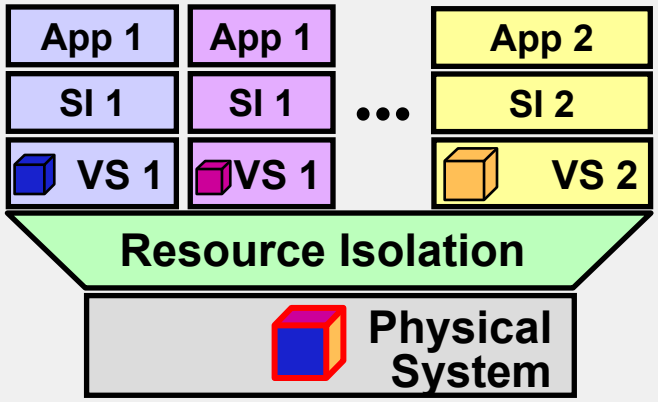
Multi-Host Mechanisms



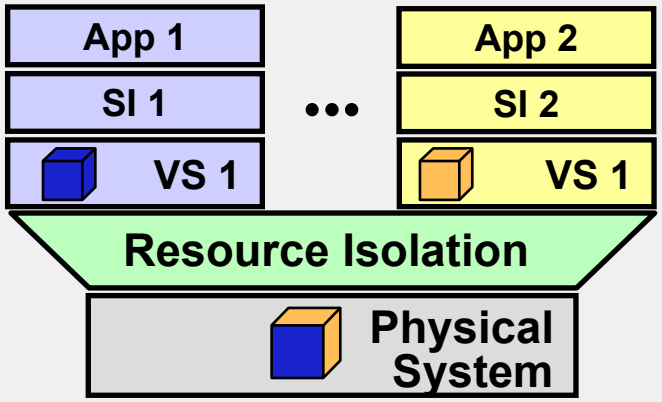
- **Today:** PCI adapters used to access external networks are integrated into blades.
  - ✓Pros/Cons:
    - + Uses well established network management infrastructures.
    - – Adapters cannot be shared.
    - – SAN/LANs link rates may not scale with CPU performance over time.
- **Multi-host PCI Topologies**: PCI fabric used to share adapters.
  - ✓Pros/Cons:
    - + Adapters can be shared.
    - + PCI family generations have higher link rate than SAN/LAN links.
    - – New network management infrastructure.

Scope of Work

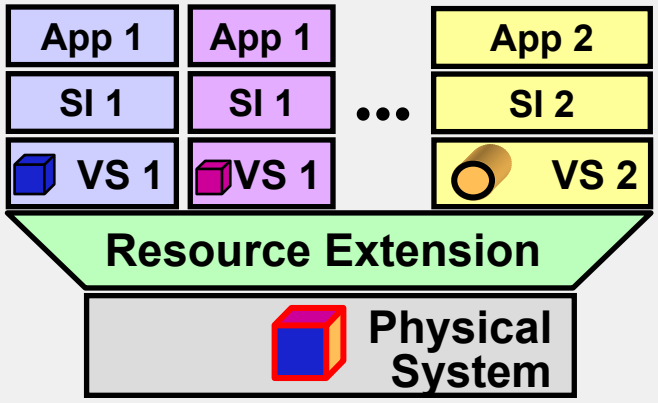
Problem Statement



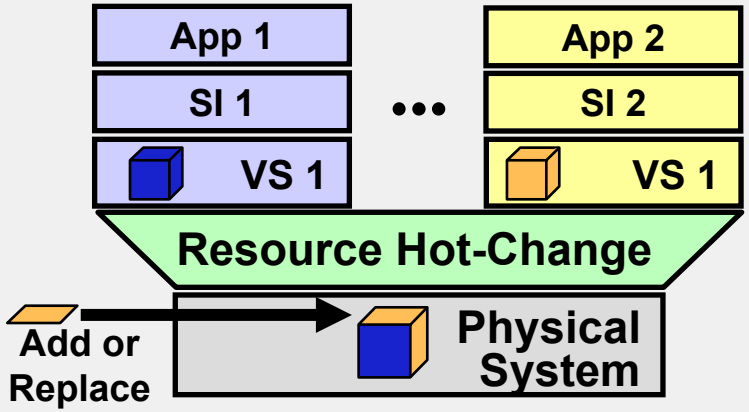
Isolation allows application co-location within a shared physical system.



Resource sharing allows lower TCO by utilizing resources more efficiently.



Extension of functions available to applications without physical system changes.



Resources can be transparently added or replaced beneath running applications.

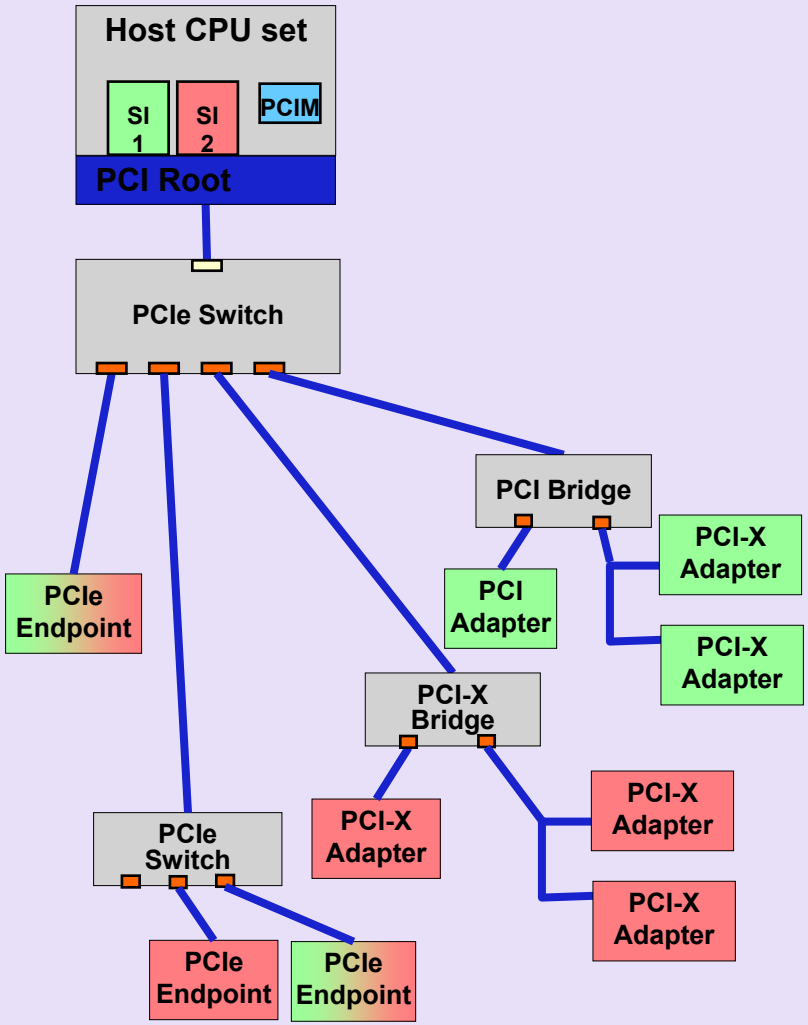
Problem Statement (Cont.)

- Problem:
  - ✓ The industry has made significant strides in developing processor and memory virtualization technology.
  - ✓ Due to I/O's dependencies on industry standard, such as PCI, Memory Mapped I/O Virtualization has remained largely undefined or proprietary.
- Workgroup Consensus
  - ✓ I/O virtualization and I/O sharing specifications are needed for point-to-point and switch-based configurations.
  - ✓ These specifications will enable the industry to create interoperable components – chipsets, switches, endpoints, and bridges.
- I/O Sharing – Sharing of an I/O component by multiple SIs. Sharing types:
  - ✓ **Serial Sharing** – ability to transfer exclusive I/O component usage from one SI to another.
  - ✓ **Simultaneous Sharing** – ability to share an I/O component by multiple SIs in parallel.

High-Level Requirements

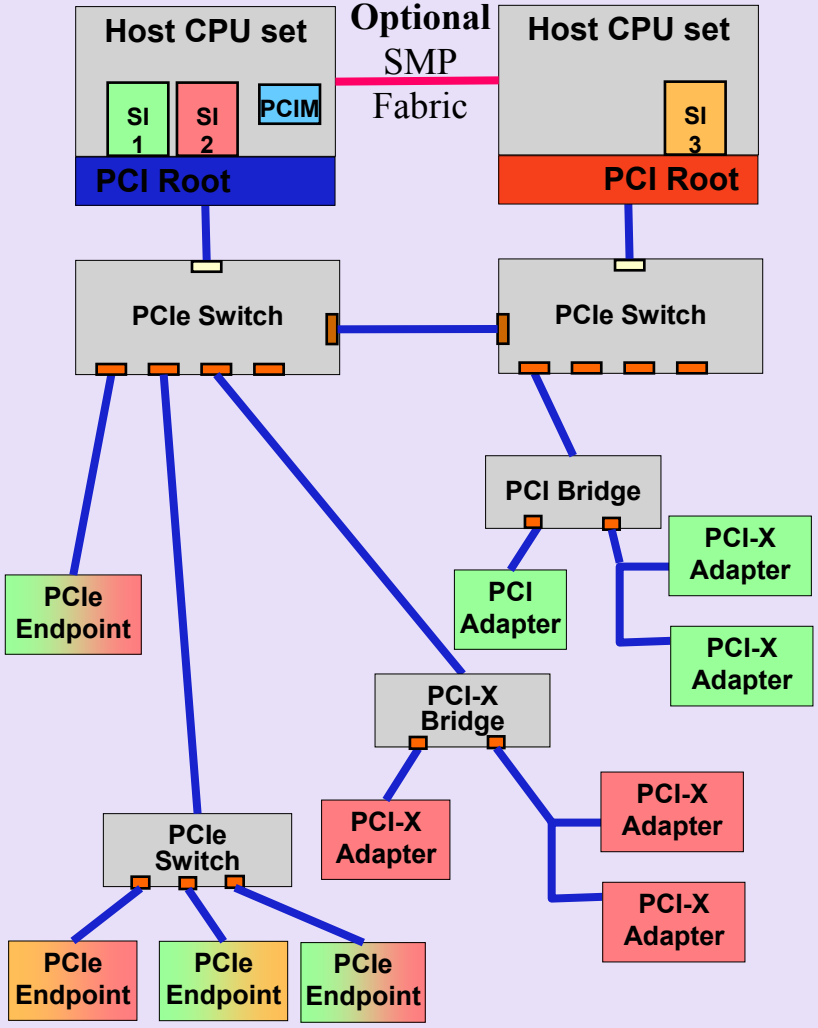
- The scope **will** be limited to the PCIe™ specification suite.
  - ✓ No materials generated will require changes to conventional PCI or PCI-X® specification suites.
- The scope **may** include changes to the PCI Firmware Spec based on subsequent investigation by the workgroup.

High-Level Requirements (cont.)



- The scope **will** enable an I/O Endpoint to be serially shared or simultaneously shared by multiple SIs (e.g. OS guests) in a single hierarchy domain (single RC).
  - ✓ **IOV Endpoint** – An Endpoint that supports the required IOV extensions defined in this specification.
- The scope **will** investigate whether new mechanisms are required to prevent, or selectively prevent, peer-to-peer operations within a single RC hierarchy.

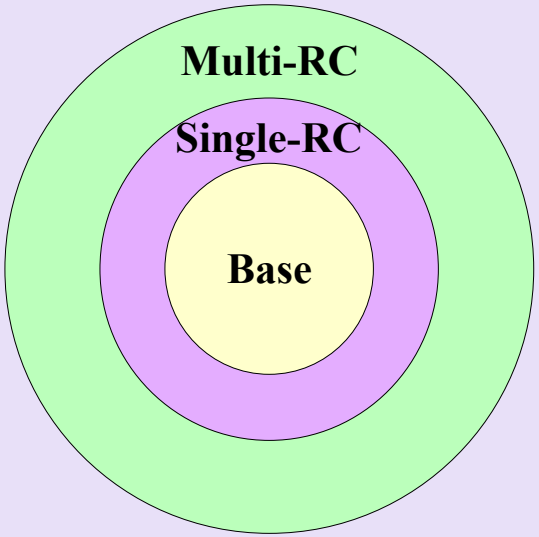
High-Level Requirements (cont.)



- The scope **will** enable an I/O Endpoint to be serially shared or simultaneously shared by multiple SIs (e.g. OS guests) in a single hierarchy domain (single RC) or in a multi-hierarchy domain (multiple RC connected to a common PCIe switch fabric).

High-Level Requirements (cont.)

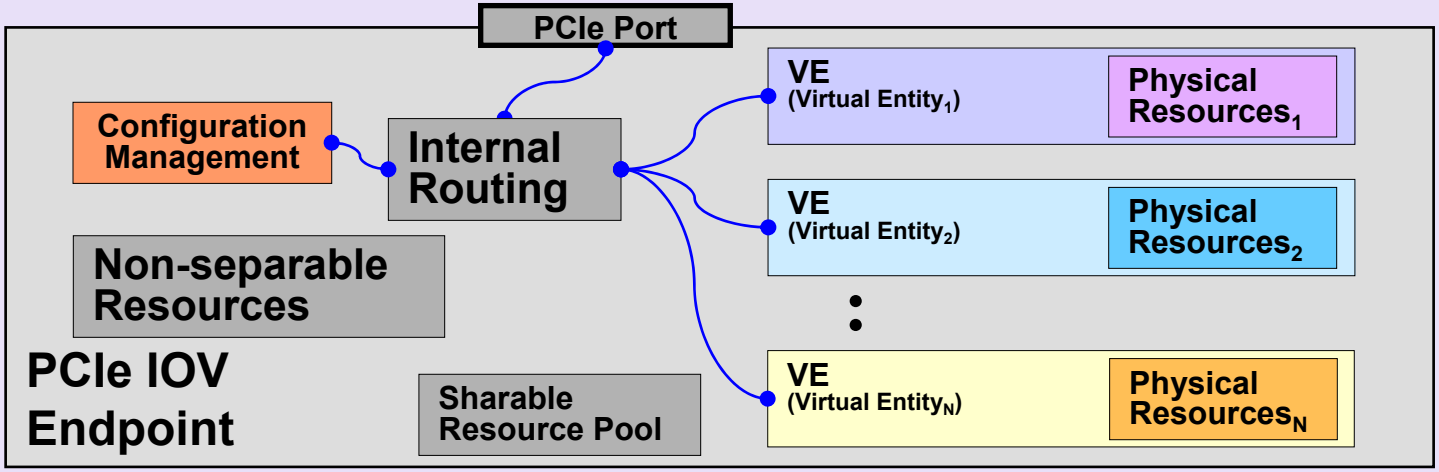
The scope **will** enable:



- Attachment of existing PCIe 1.x Base components – RC, Switches, Endpoints, and Bridges.
- A solution to use a combination of existing base and IOV-aware components:
  - ✓ Single-RC capabilities **shall** be a superset of the PCIe 1.x Base specification.
  - ✓ Multi-RC capabilities **shall** be a superset of the single-RC capabilities.
  - ✓ Objective is to maximize component interoperability across multiple topologies, e.g.:
    - An I/O component designed to support multi-RC **shall** support the single-RC capabilities.
- IOV-capable components to be backwards compatible with existing software.
  - ✓ Although some or all of the new IOV capabilities may not be supported in these circumstances.

High-Level Requirements (cont.)

- The scope **will** include support for the creation, management, and destruction of resources associated with a shared, Virtual Endpoint.
  - ✓ A VE may be serially or simultaneously shared by one or more RC or SIs.

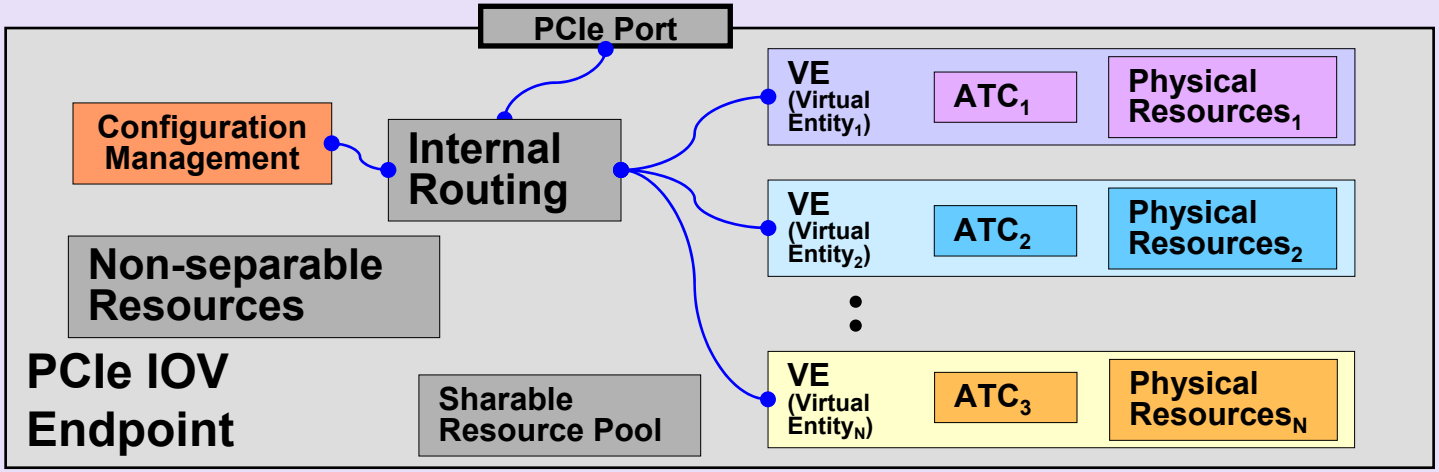


- **Virtual Endpoint (VE)** - An Endpoint, within an IOV enabled Endpoint, that shares one or more physical Endpoint resources, such as a link, with another and can, without run-time intervention by a VI, directly:
  - ✓ sink I/O and Memory operations from an SI; and
  - ✓ source DMA, completion and interrupt operations to an SI.

  An IOVI is still required for configuration operations.

High-Level Requirements (cont.)

- The scope **will** include support for an Endpoint to request translated addresses prior to injecting a request to a RC, e.g. a DMA Read or DMA Write operation.
- The minimum expected operations are:
  1. Request a translation,
  2. Return a translated address, and
  3. Invalidate a prior translation.



- **Address Translation Caching (ATC)** – The caching of information used by the RC to translate a PCI bus address in order to access host memory.

Goals

The scope of this work **will** enable:

- The development of interoperable components – RC, switches, endpoints, and bridges – that provide I/O virtualization and sharing capabilities.
  - ✓ These components may be jointly deployed with existing PCIe 1.x based components to provide varying degrees of functionality .
- Multiple SIs to:
  - ✓ Be co-located onto a shared physical domain; and
  - ✓ Provide attestable trust and fault isolation, when co-located.
- Support for a variety of physical topologies enabling the technology to be deployed across multiple market segments and usage models.

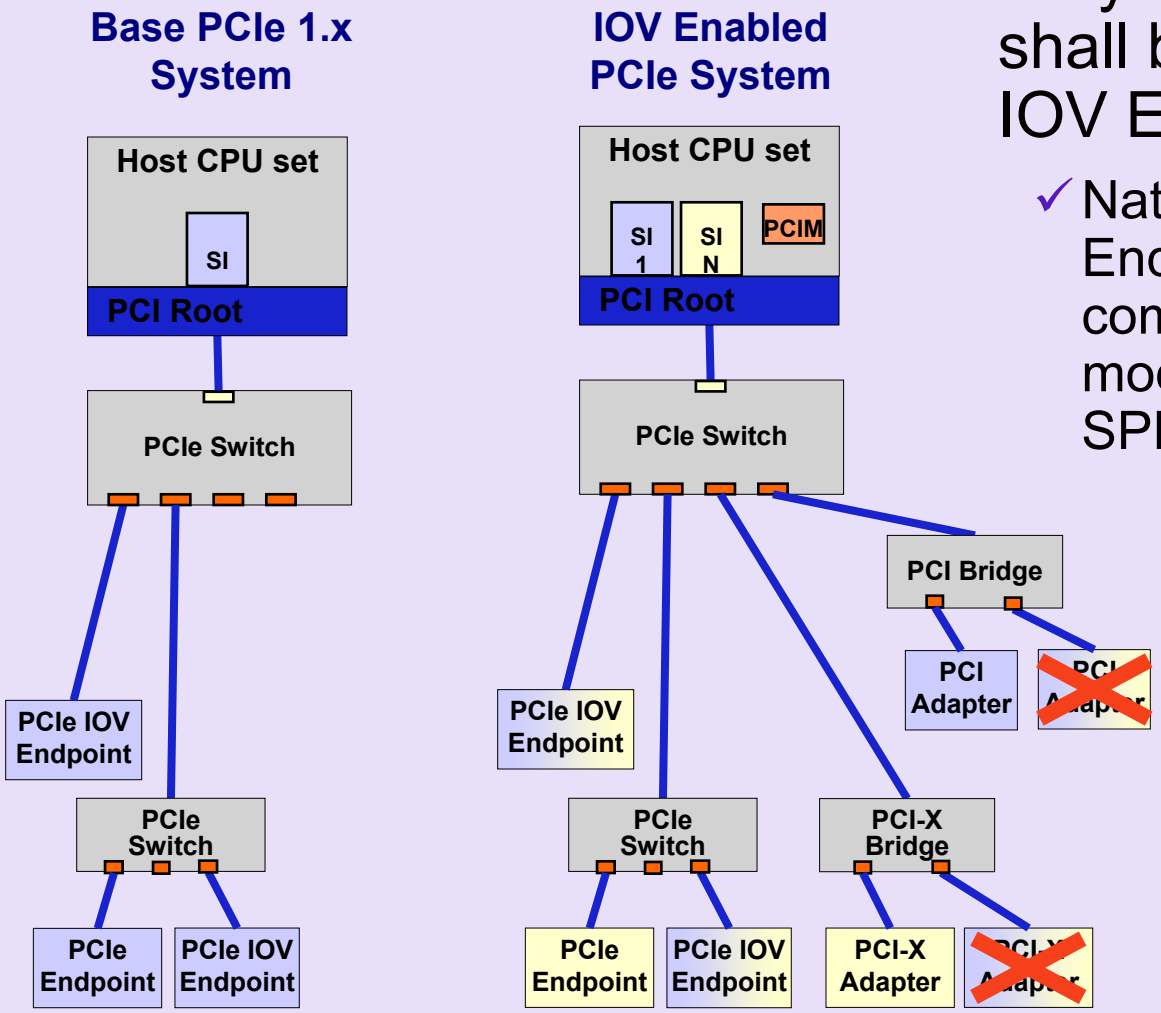
Non-Goals

Based on the votes taken to date, the following areas were deemed as not in scope:

- The scope of this work **does not, in principle**, touch the physical layer.
  - ✓ Subsequent investigation (e.g. Cross-linking requirement analysis) could result in a change to the physical layer, but our objective is to avoid any impact if at all possible.
- The scope of this work **will not include:**
  - ✓ Host address translation or protection mechanisms, i.e. an ATPT;
  - ✓ The ability of multiple SIs to share a device attached to a PCIe Bridge; and
  - ✓ Support for peer-to-peer communication (e.g., IPC) between different RC.
- The scope of this work **will not define** policy management or other types of management functions used to provide higher-level management of the components specified by this workgroup. That is:
  - ✓ The scope will be conceptually focused on PCIe operation types, such as: configuration operations, event notification operations, etc…

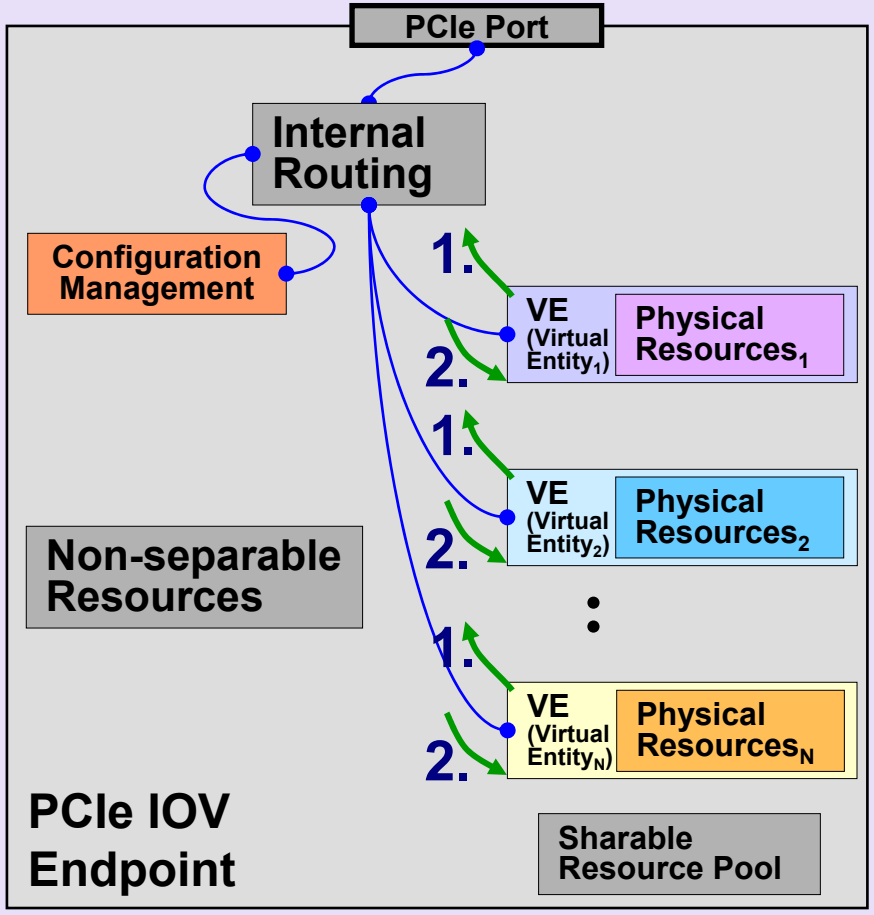
Examples of Key Requirements Documented in the Spec Template

Single RC PCIe IOV Endpoint Requirements



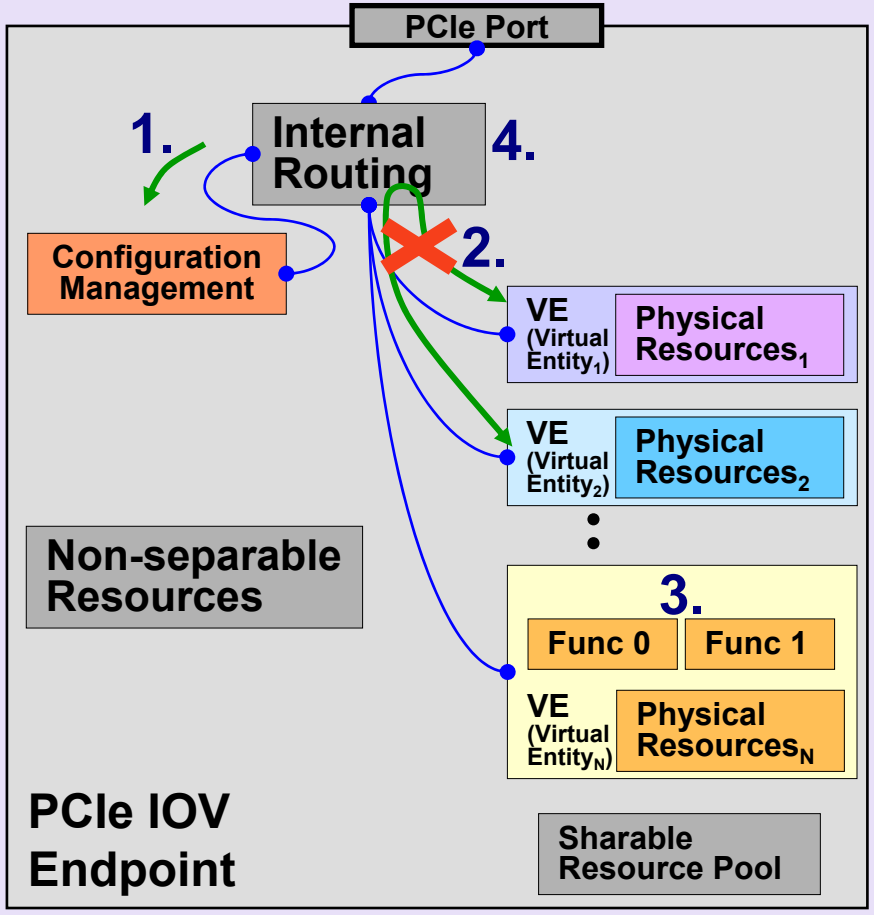
- Only PCIe endpoints shall be specified for IOV Endpoints.
  - ✓ Native based PCI IOV Endpoints shall be backwards compatible, in a non virtualized mode, with PCIe base 1.x SPEC.

Single Root PCIe IOV Endpoint Req's …Continued



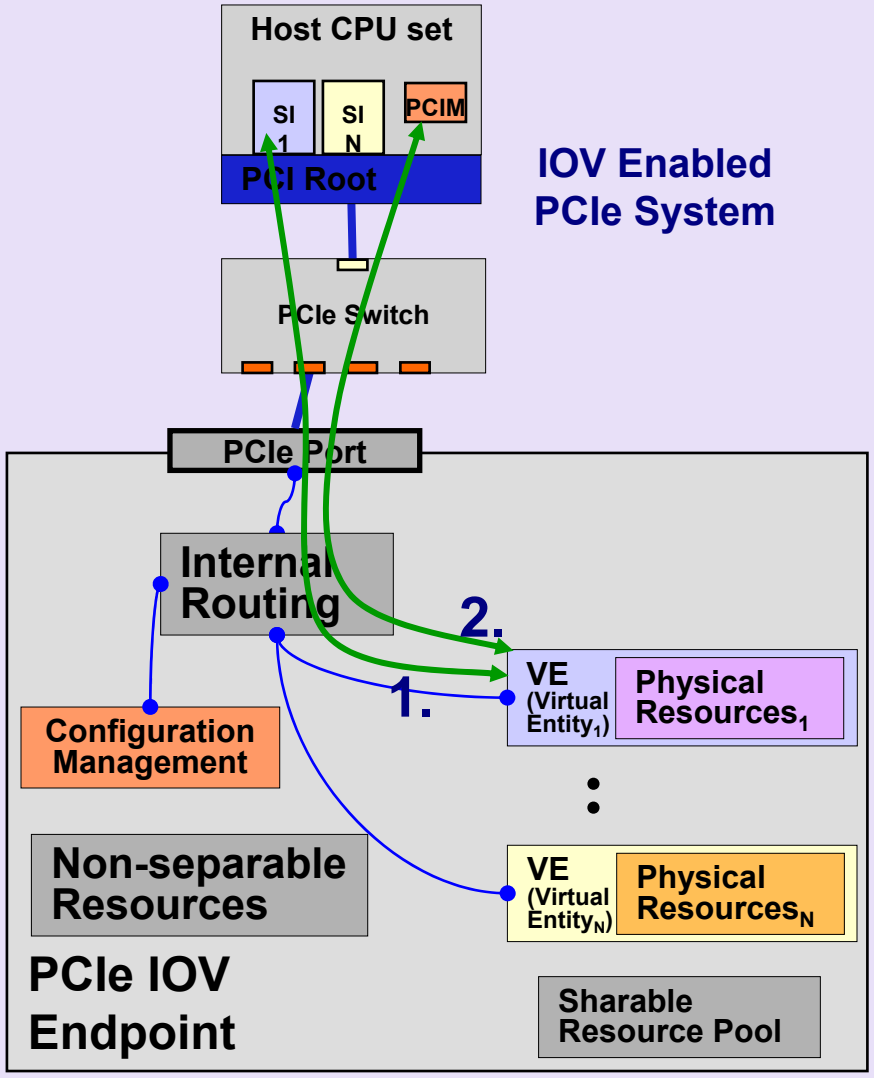
- Each VE shall have the capability to independently:
  1. Source DMA, completion, and interrupt operations within the system as an initiator;
  2. Sink configuration, I/O, and memory operations within the system as a target.
- Each VE shall appear in the fabric as one or more PCIe function(s).
- Except for rare errors conditions (spec will provide examples):
  - ✓ Each transaction issued from the PCIe IOV Endpoint shall be uniquely identifiable with the VE that sourced the transaction.
  - ✓ The PCIe IOV Endpoint shall be able to associate each transaction it sinks with the appropriate VE.

Single Root PCIe IOV Endpoint Req's …Continued



- IOV endpoints shall:
  1. Support a standard, in-band, mechanism for managing the IOV endpoint;
  2. include functionality to disable their peer to peer traffic to other functions within the IOV Endpoint.
  3. Be allowed to support multiple functions assigned to the same SI with the same level of functionality given to a current multi-function device.
  4. Support defined arbitration mechanisms between the VE's within an IOV enabled endpoint
  5. Support the capability to have each VE independently managed, including:
     - ✓ Functional reset (i.e. full device function state),
     - ✓ Enable/Disable (a.k.a. Create/Destroy)
     - ✓ Having a VE's PCIe state and resources modified, for example:
       - ✓ TC assignment,
       - ✓ Bus Master Enable, …
  6. Support the specification of the minimum and maximum number of resources available to each VE.

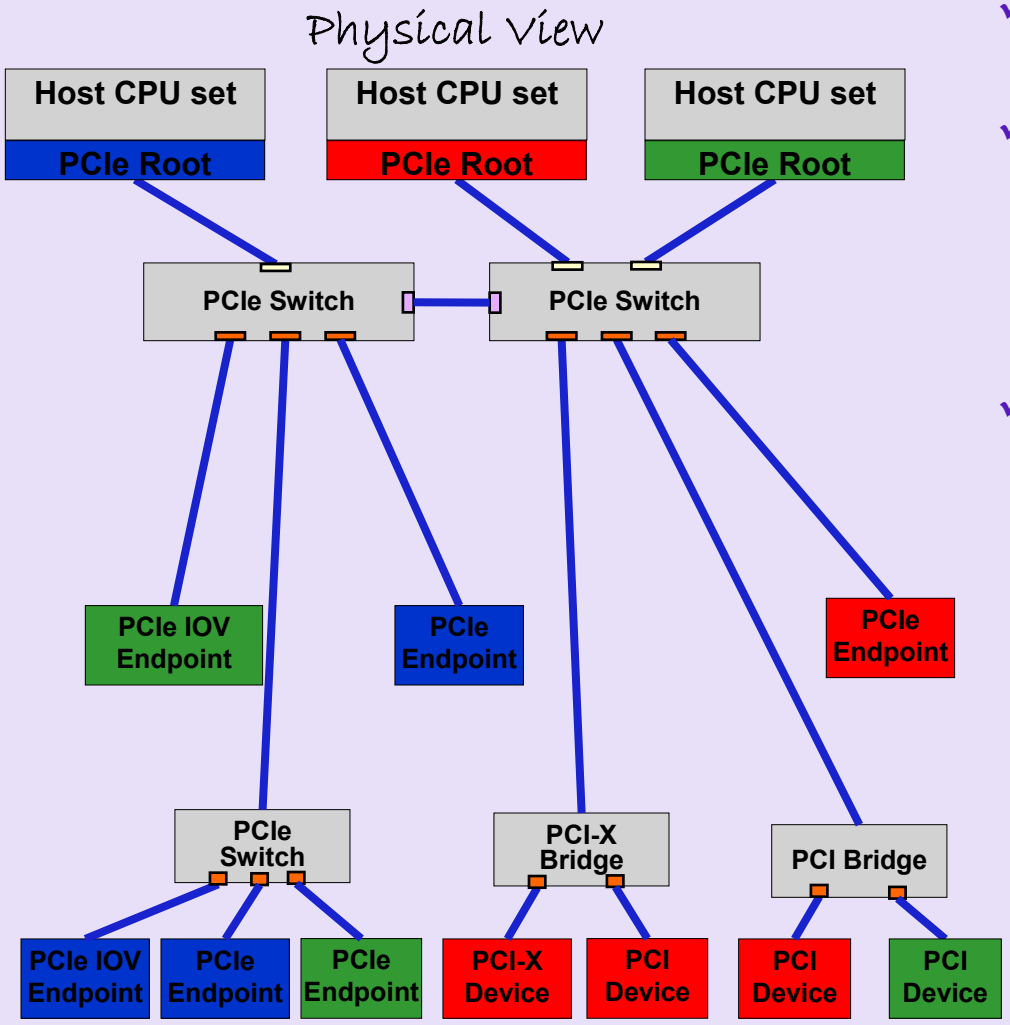
Single Root PCIe IOV Endpoint Req's …Continued

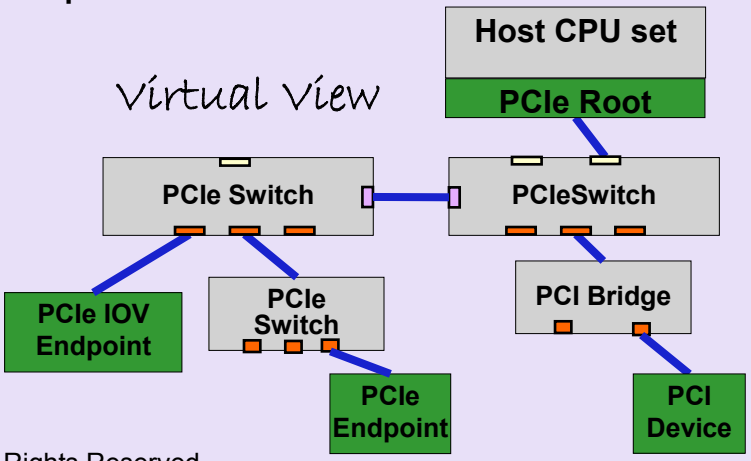


1. A mechanism shall be provided to allow VE to be associated with an SI, such that data movement operations are enabled and can be performed directly between the SI and its associated VE, without VI involvement.
2. The virtualization mechanisms defined in this specification may require a VI (such as a PCI Configuration Manager) to be involved for configuration operations performed on a VE.

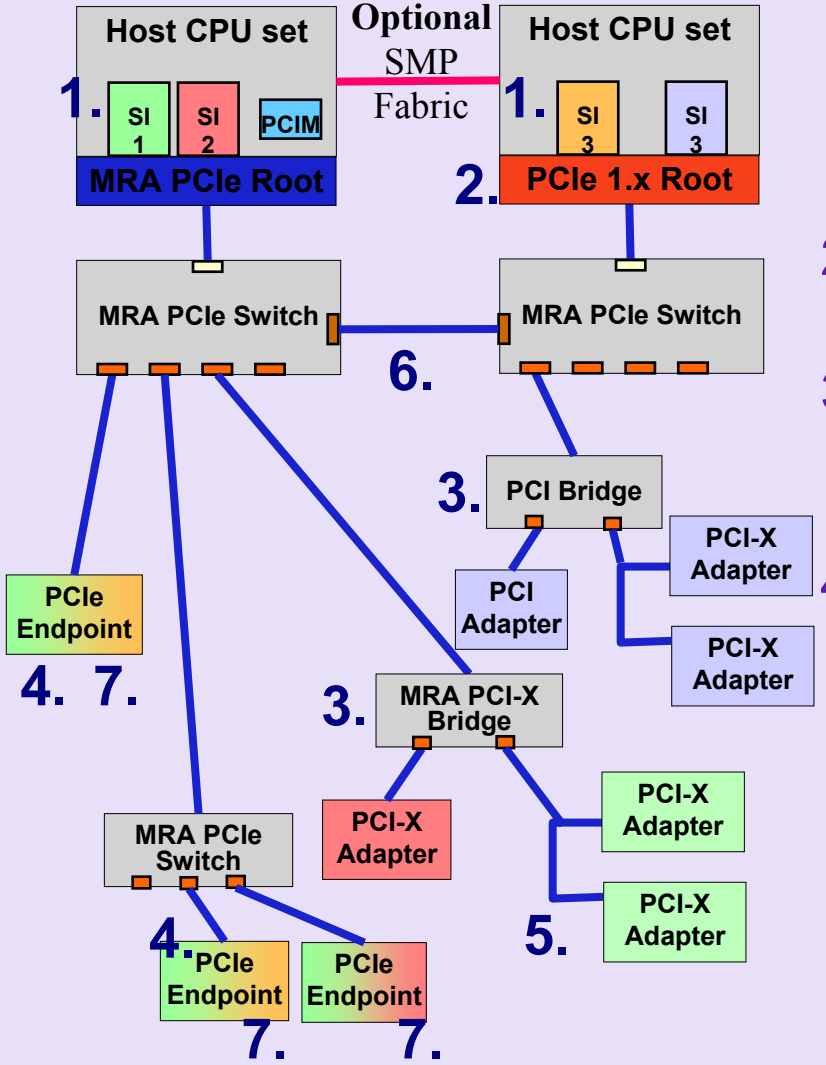
Multi-Root PCIe IOV Endpoint Requirements

- The multi-root solution
  - ✓ Shall give each RC its own Virtual Hierarchy.
  - ✓ Should enable SW written for PCI, PCI-X, or PCIe enumeration, configuration, and management to run, unchanged on each RC within the multi-root system.
  - ✓ Shall enable each switch, bridge, function, and VE to be uniquely represented in the configuration space of each RC.



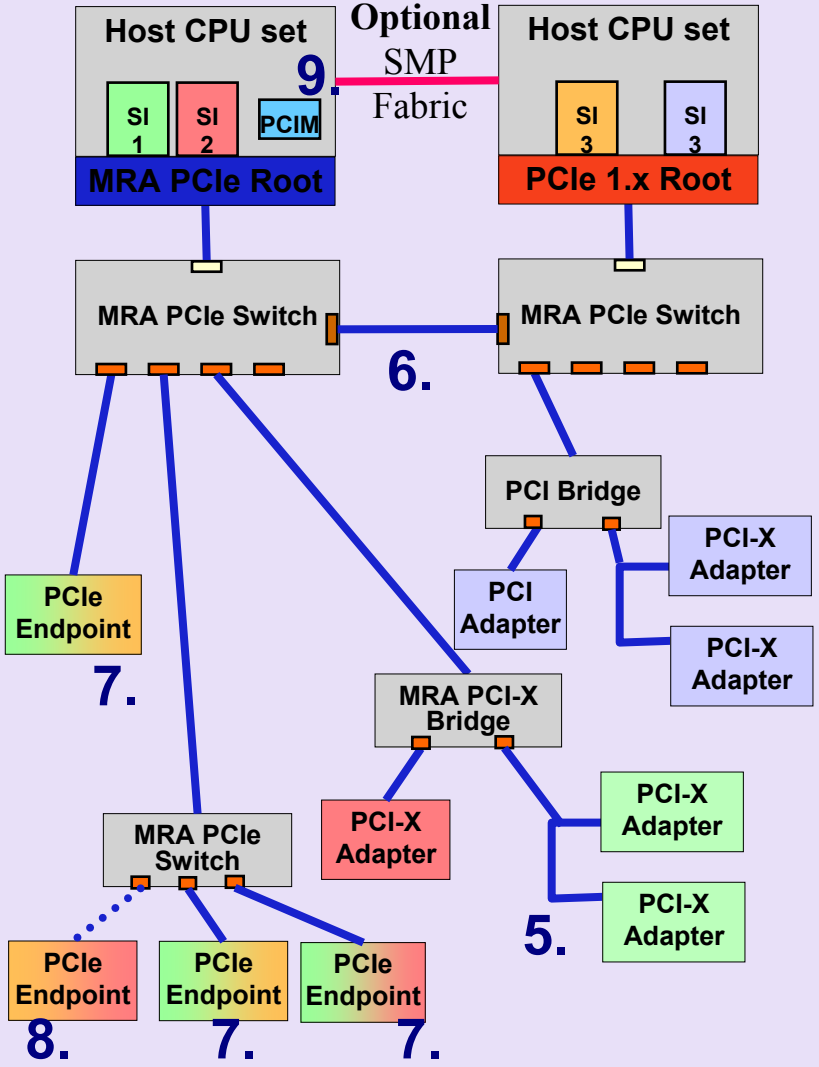


Multi-Root PCIe IOV Endpoint Req's …Continued



- The multi-root solution shall:
  1. Provide the same characteristics to its IOV Enabled endpoints as the single-root solution relative to separate SIs.
  2. Enable use of existing PCIe 1.x or later RC.
  3. Enable existing PCIe 1.x Switches, Endpoints, and PCIe to PCI/PCI-X Bridges to each be bound to a single RC.
  4. Enable an IOV enabled endpoint to be shared amongst multiple RC's using a **Multi-Root Aware** (MRA) PCIe switch.

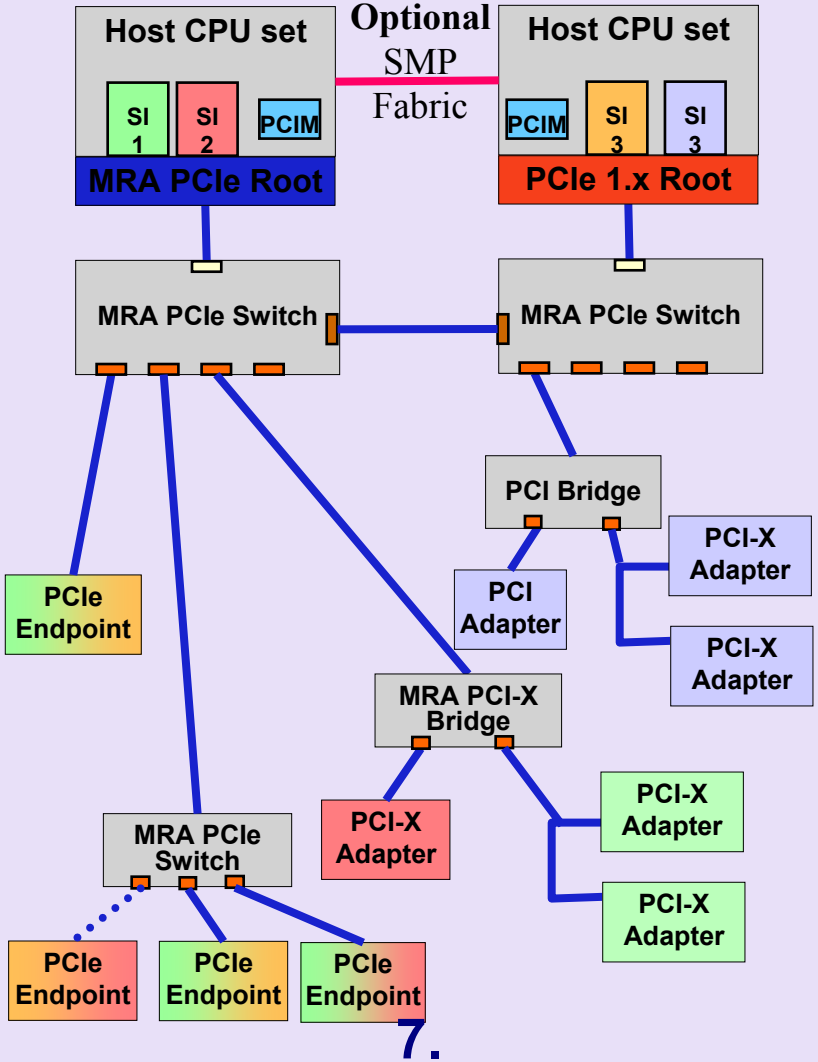
Multi-Root PCIe IOV Endpoint Req's …Continued



- The multi-root solution shall:

5. Enable a PCIe 1.x MRA bridge to PCI/PCI-X to assign each of its independent busses to a separate RC.
6. Enable multiple MRA switches to be connected together.
7. Enable a VE to be shared by multiple SI's within one or more RC's.
8. Enable hot-plug of physical EPs and VEs.
9. Enable a management entity (PCI Configuration Manger, PCI-M) to manage the fabric.

PCIM Req's



The PCIM shall be able to:

- ✓ Discover and configure IOV enabled RC's, switches, bridges, and endpoints using a defined PCIM interface.
- ✓ Discover and configure non-IOV components using existing interfaces.
- ✓ Control and manage errors within the fabric.

The solution supports one or more PCIM entities (with coordination between them).

A method will be defined:

- ✓ To establish a trust relationship between a PCIM and a given PCIe component.
- ✓ For the PCIM to control and limit access to IOV management functions within PCIe IOV enabled Endpoints by entities other than the primary PCIM.

The PCI Express® Multi-Root Configuration provides multiple configuration spaces for each switch, bridge and endpoint.

ATS Requirements

- Address Translation Services will consist of a set of transactions to enable a downstream component to:
  - ✓ Request a translation of a given address.
  - ✓ Indicate that a subsequent transaction, e.g. a DMA Read or DMA Write, has been translated.
  - ✓ Invalidation response to indicate that a translation invalidation request has been processed.
- ATS will consist of a set of transactions to enable a host Address Translation and Protection Table (ATPT) via an RC to:
  - ✓ Issue a translation completion of a given address.
  - ✓ Issue a translation invalidation request for a given address.
  - ✓ Ascertain whether an invalidation has been processed by a downstream component.

Current Status

- Address Translation Services
  - ✓ Specification Draft 0.5 is posted on PCI-SIG Website
    - http://www.pcisig.com/members/downloads/specifications/pciexpress/specification/draft/ats-spec-0_5_draft-051111.pdf
    - Review period just ended
  - ✓ Specification Draft 0.7 is in progress within the workgroup, look for an update on the web soon

- I/O Virtualization
  - ✓ Specification Draft 0.3 is posted on PCI-SIG Website
    - http://www.pcisig.com/members/downloads/specifications/pciexpress/specification/draft/IOV_spec_draft_0.3-051013.pdf
  - ✓ Workgroup has decided to separate Single Root and Multi Root into individual documents, expect an updated draft within the next 6 weeks or so

Questions

Thank you for attending the PCI-SIG Developers Conference Europe 2006.

For more information please go to www.pcisig.com

I/O Device Virtualization Overview

Richard Solomon

IC Design Engineer

LSI Logic


PCI
SIG®