
PCI
SIG®

I/O Virtualization and Sharing

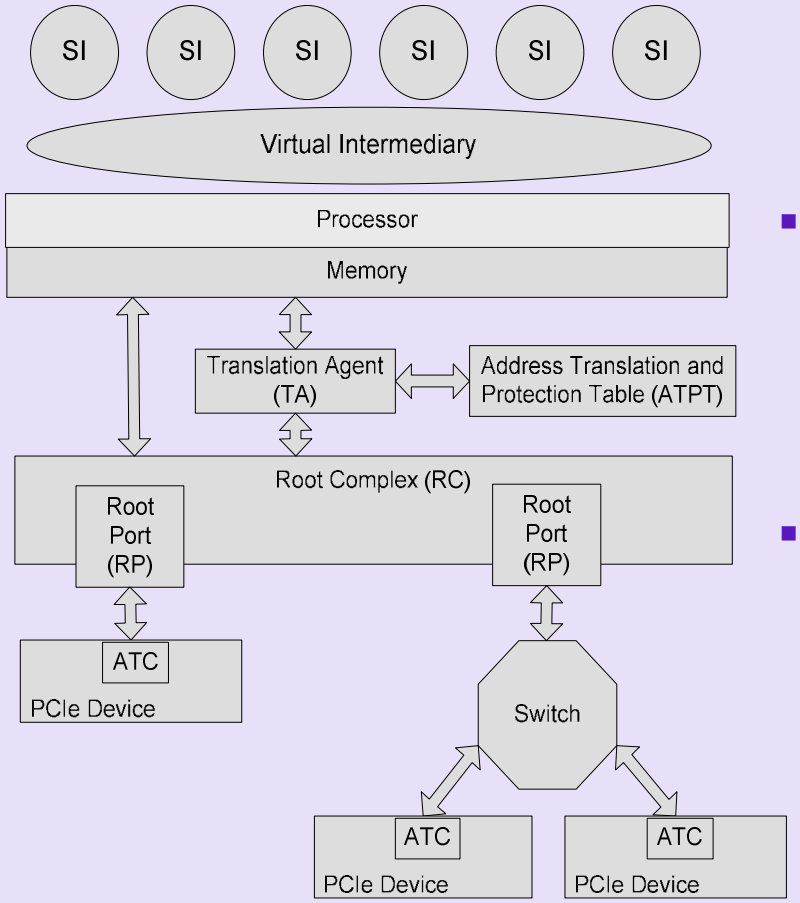
Michael Krause (HP, co-chair)
Renato Recio (IBM, co-chair)

Outline

- Virtualization Overview
  - ✓ Terminology
  - ✓ Single-Root (SR) IOV
  - ✓ Multi-Root (MR) IOV
  - ✓ Address Translation Services
- Specification Status and Working Schedule to Completion

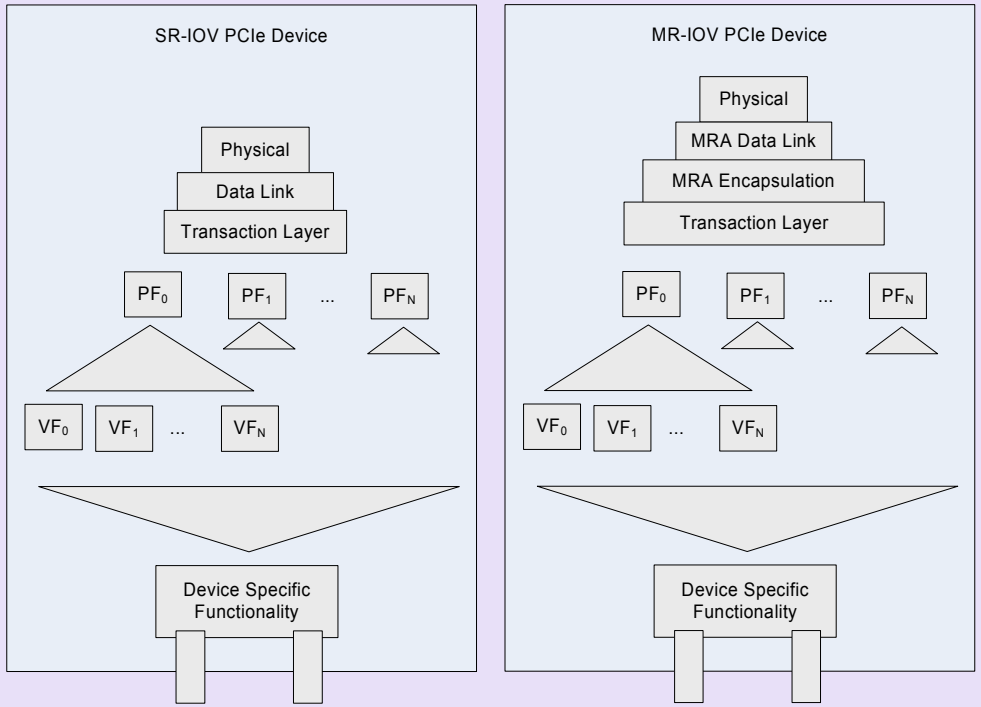
Terminology

Terminology



- System Image (SI)
  - ✓ *S/W, e.g. a guest OS, to which virtual and physical devices can be assigned*
- Virtual Intermediary (VI)
  - ✓ *Resource management and event handling component*
  - ✓ *Allocates resources and isolates resources to each SI*
- Translation Agent (TA)
  - ✓ Translates PCI Addresses to platform physical addresses
    - – May also provide interrupt remapping for MSI / MSI-X interrupts
  - ✓ ATPT provides translations and access rights typically on a per Function identifier basis

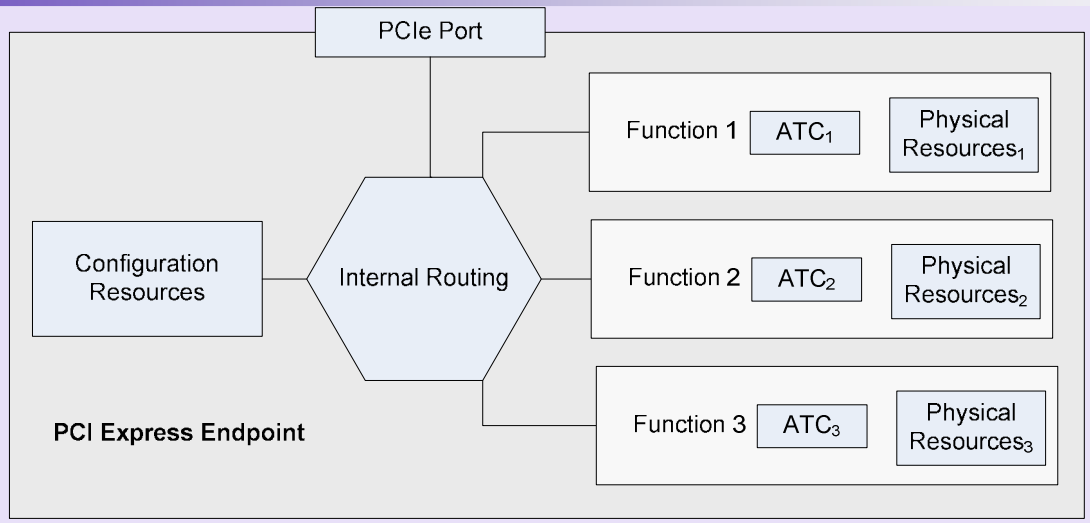
Terminology (cont.)



- Physical Function (PF)
  - ✓ Function that supports SR-IOV
  - ✓ Has full configuration space / BAR
- Virtual Function (VF)
  - ✓ A Function associated with a PF.
  - ✓ Shares PF resources
  - ✓ Supports SR-IOV capability
- Multi-Root Aware (MRA)
  - ✓ Supports MR-IOV capability
  - ✓ MR provides each Virtual Hierarchy (VH) with its own PCI 32 / 64-bit Memory, I/O, and Configuration Space.

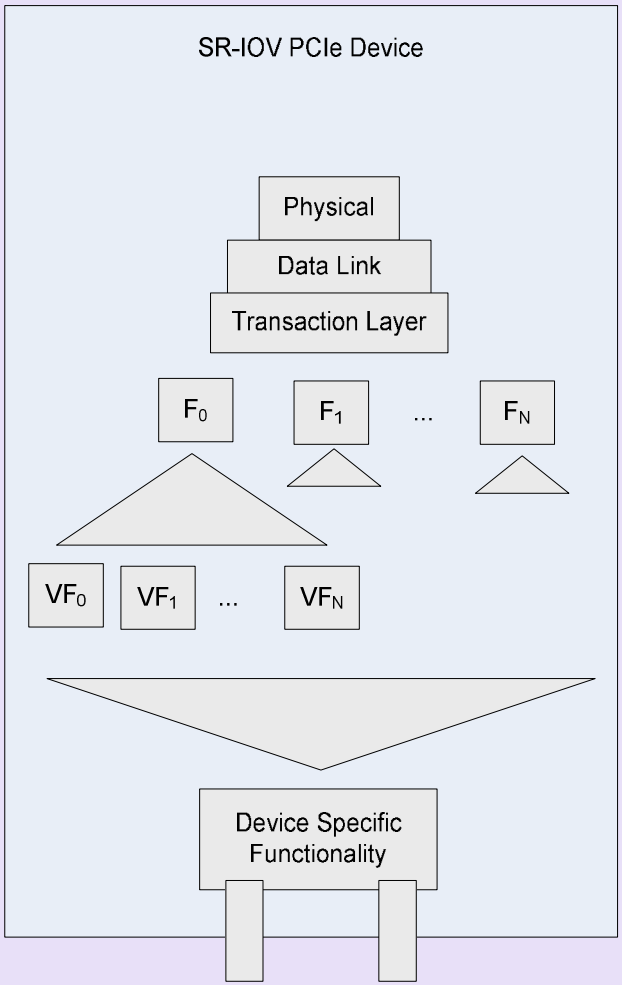
Single-Root (SR) IOV

Example I/O Device Today



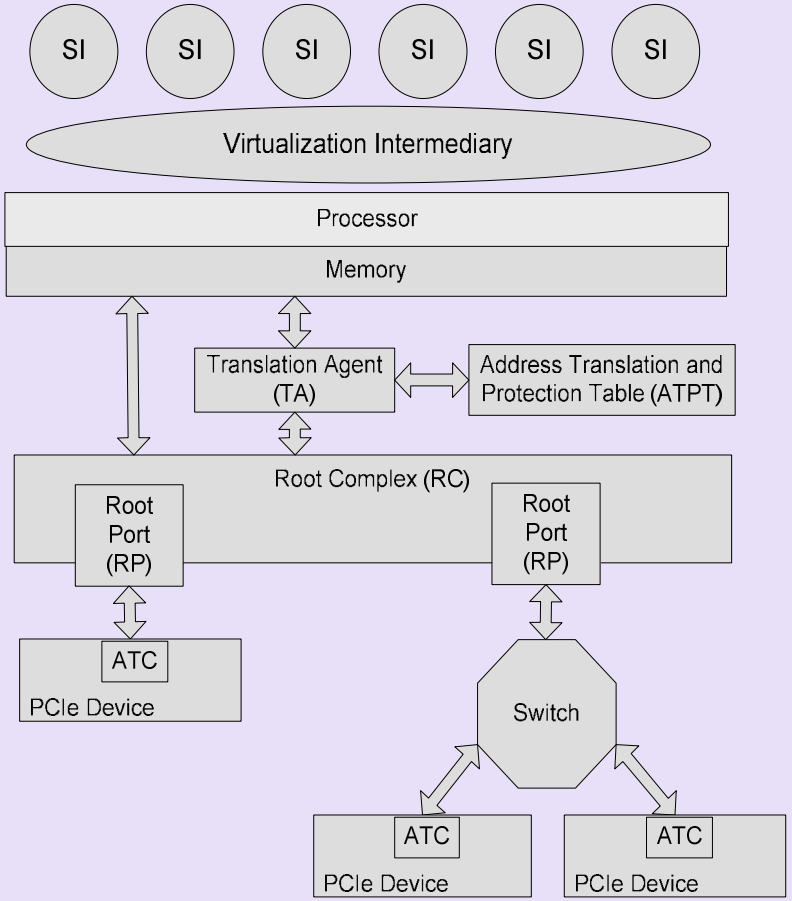
- Attributes:
  - ✓ Scalability:
    - Up to 8 PCI Functions with unique configuration space / BAR / etc.
    - ARI Capability enables up to 256 Functions to be supported
  - ✓ Function 0 required to manage shared resource / process shared events
  - ✓ Interrupt support:
    - INTx, MSI, MSI-X or combination of MSI and MSI-X
  - ✓ Function dependencies through vendor-specific mechanisms
  - ✓ Function-specific resource arbitration through vendor-specific mechanisms
  - ✓ Cannot be shared by more than one SI without VI involvement

SR-IOV Device



- Attributes:
  - ✓ Scalability
    - – Up to ~2^16 Functions
      - • ARI enables up to 256
      - • IOV enables additional Bus Numbers to be associated
  - ✓ Function 0 required
  - ✓ Interrupt support - MSI, MSI-X
  - ✓ One configuration space / BAR per Function
    - – Multiple VF share Function's resource space
  - ✓ Function dependencies through vendor-specific mechanisms
  - ✓ Function-specific resource arbitration through vendor-specific mechanisms
  - ✓ Can be shared by more than one SI
    - – VI responsible for all configuration access

Single-Root PCIM (SR-PCIM)



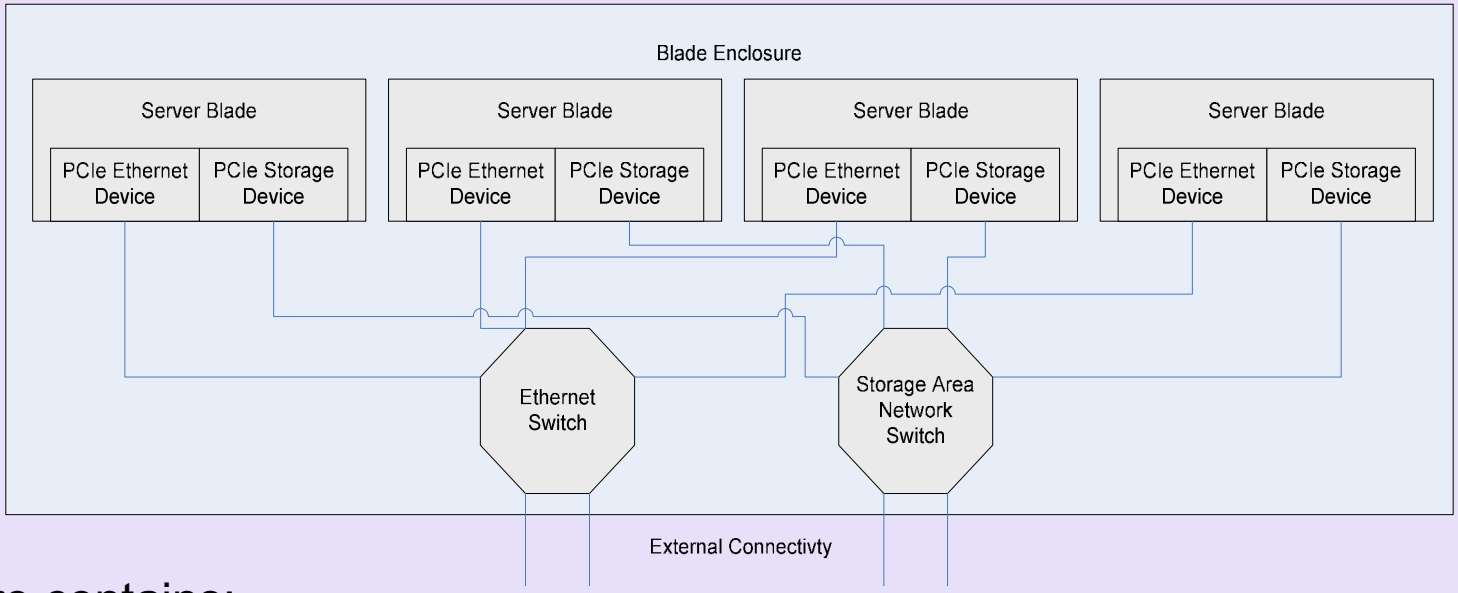
- PCIM – PCI Manager
  - ✓ System software that controls configuration, management and error handling of PFs and VFs.
  - ✓ SR-PCIM may be integrated into a VI or other software
    - Implementation-specific

Alternative Routing Identifier (ARI)

- New PCI Express Capability
  - ✓ Draft ECN applicable to PCI Express 1.1 and 2.0
  - ✓ Applicable to
    - Multi-Function Devices at Upstream Ports (ARI Devices)
    - Downstream Ports (Root Ports and Switches)
  - ✓ An ARI Device interprets its directly associated ID (Routing, Requester, Completer) as having an 8-bit Function Number instead of a traditional 3-bit Function Number.
    - An ARI Device has no Device Number.
    - An ARI Device supports up to 256 Function Numbers.
      - Function 0 is required
      - Function 0 acts as head of link list of Function Numbers
        - Software walks link list to find next Function to configure – improves enumeration performance.
  - ✓ When ARI Forwarding is enabled, Downstream ports do not enforce the Device Number = 0 restriction.
  - ✓ ARI Functions can be organized into Function Groups
    - ARI Functions assigned to a Function Group Number (FGN)
    - Multi-VC arbitration can use using FGN instead of FN
    - Access Control Services can use FGN instead of Function Numbers

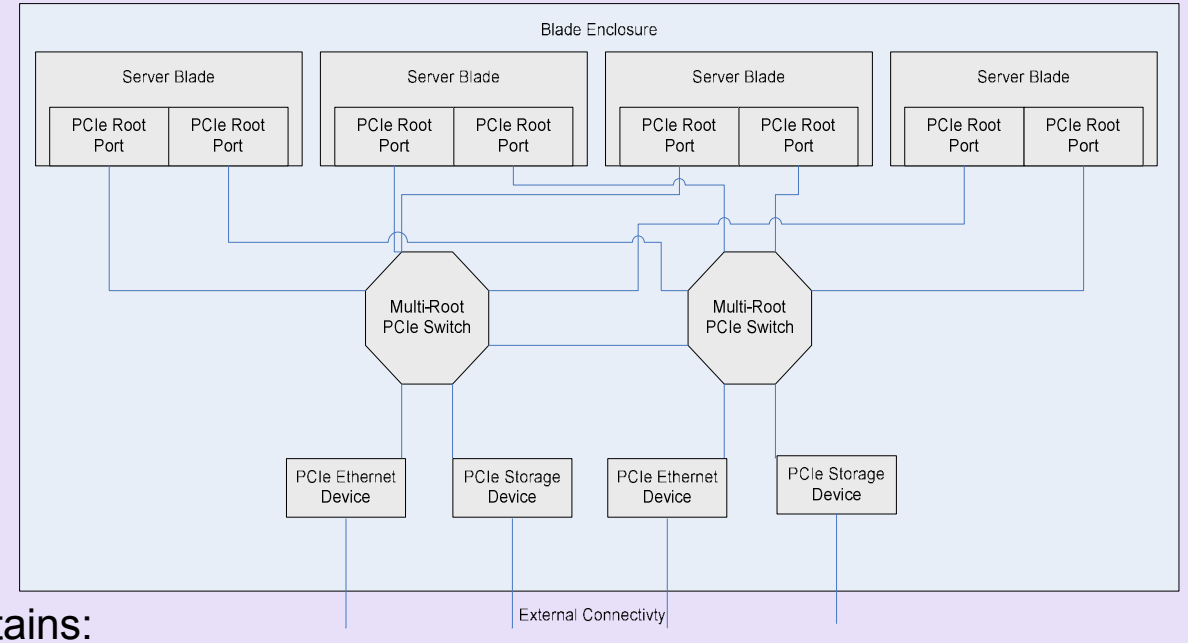
Multi-Root (MR) IOV

Example Blade Enclosure



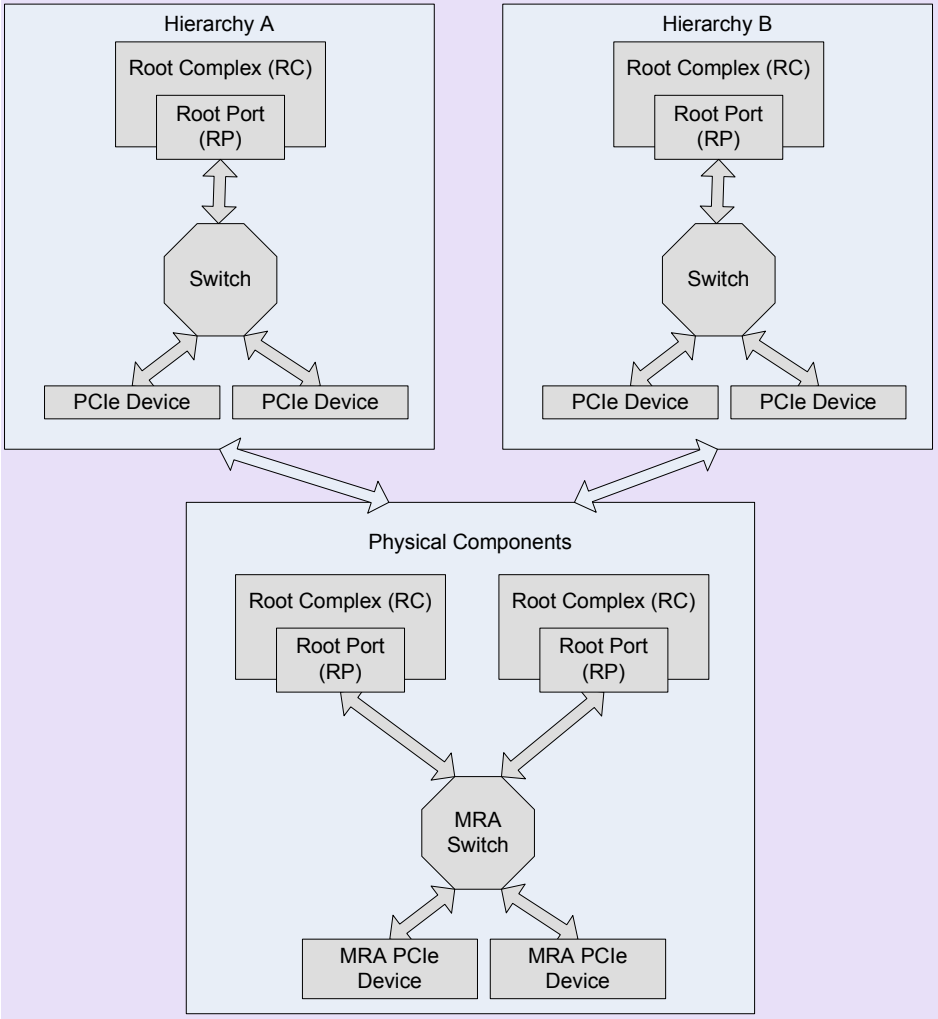
- Enclosure contains:
  - ✓ 4 server blades with 2 PCIe Devices each for a total of 8 Devices
    - – Each server blade contains an Ethernet and a storage device (e.g. Fibre Channel)
    - – Assumes point-to-point connectivity to PCIe RP
  - ✓ Enclosure contains 2 (4 in the case of a High Availability solution) “external” switches
- Each I/O device and switch port is typically provisioned to enable any I/O device to operate at full bandwidth.

MR-IOV Blade Enclosure



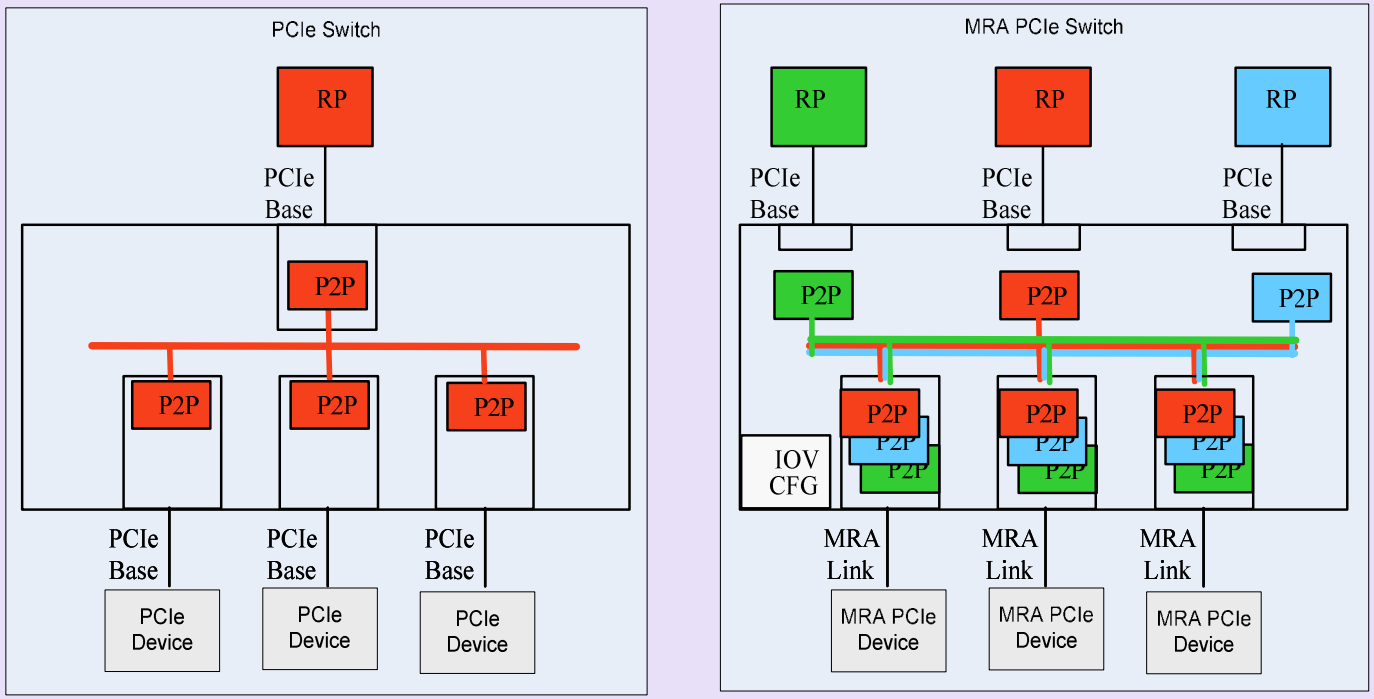
- Enclosure contains:
  - ✓ 4 server blades with no PCIe Devices
  - ✓ 4 MRA PCIe Devices – 2 Ethernet and 2 Storage
    - – Can horizontally scale to increase aggregate bandwidth based on workload needs
  - ✓ Enclosure contains 2 MRA PCIe Switches
    - – No external switches required
- I/O Device is shared
  - ✓ Can be dynamically provisioned to meet workload requirements

Goal of MR-IOV



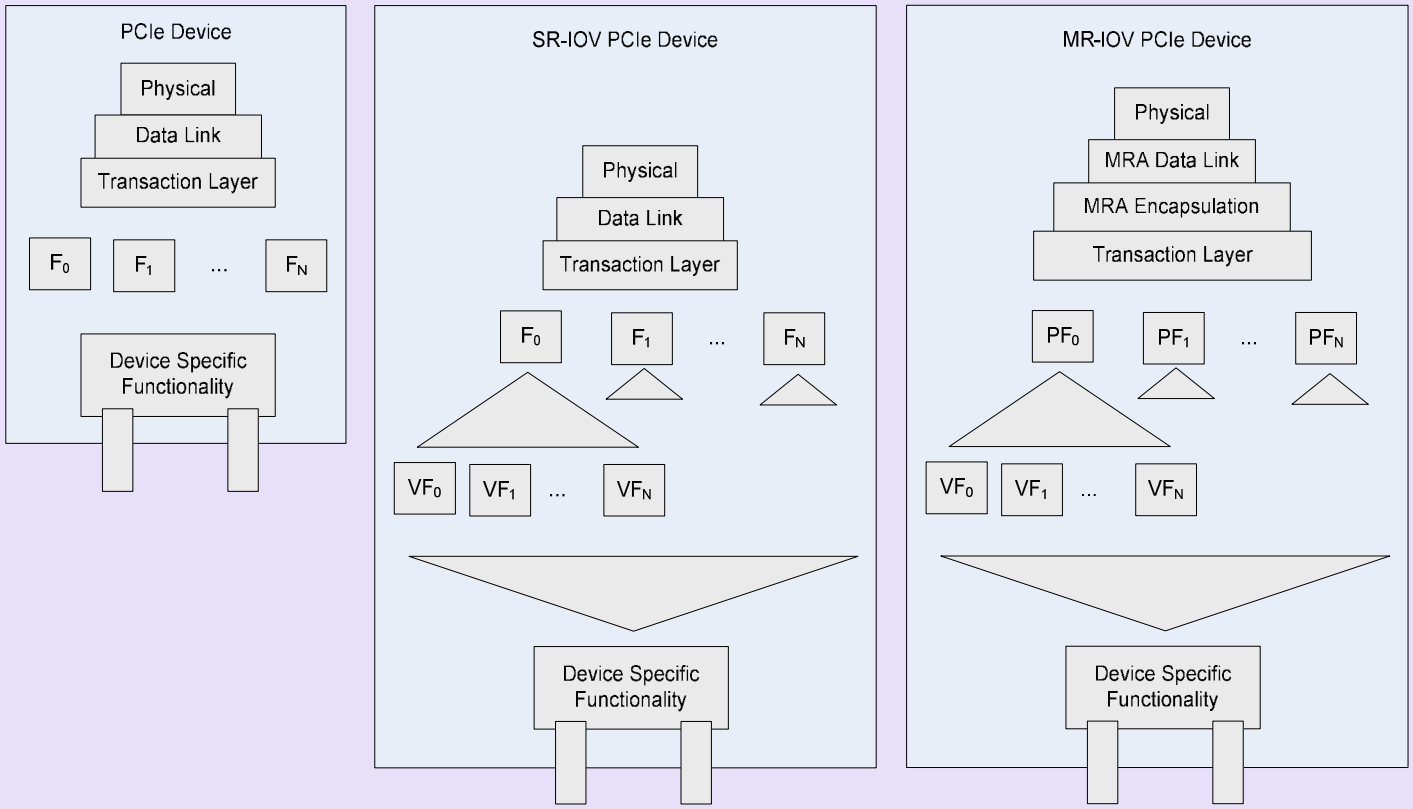
- PCI components underneath each RP must be virtualized and logically overlaid on the MRA PCIe Switches and Devices
- The virtualized PCI components are referred to as a Virtual Hierarchy (VH). A VH has the following attributes:
  - ✓ Each VH must contain at least one PCIe Switch.
    - The PCIe Switch will be a virtualized component implemented over of a MRA Switch.
    - The PCIe Switch functionality and semantics are per the *PCI Express Base Specification*.
  - ✓ Each VH may contain any mix of PCIe Devices, MRA PCIe Devices, or PCIe to PCI / PCI-X Bridges

Multi-Root Aware (MRA) Switch

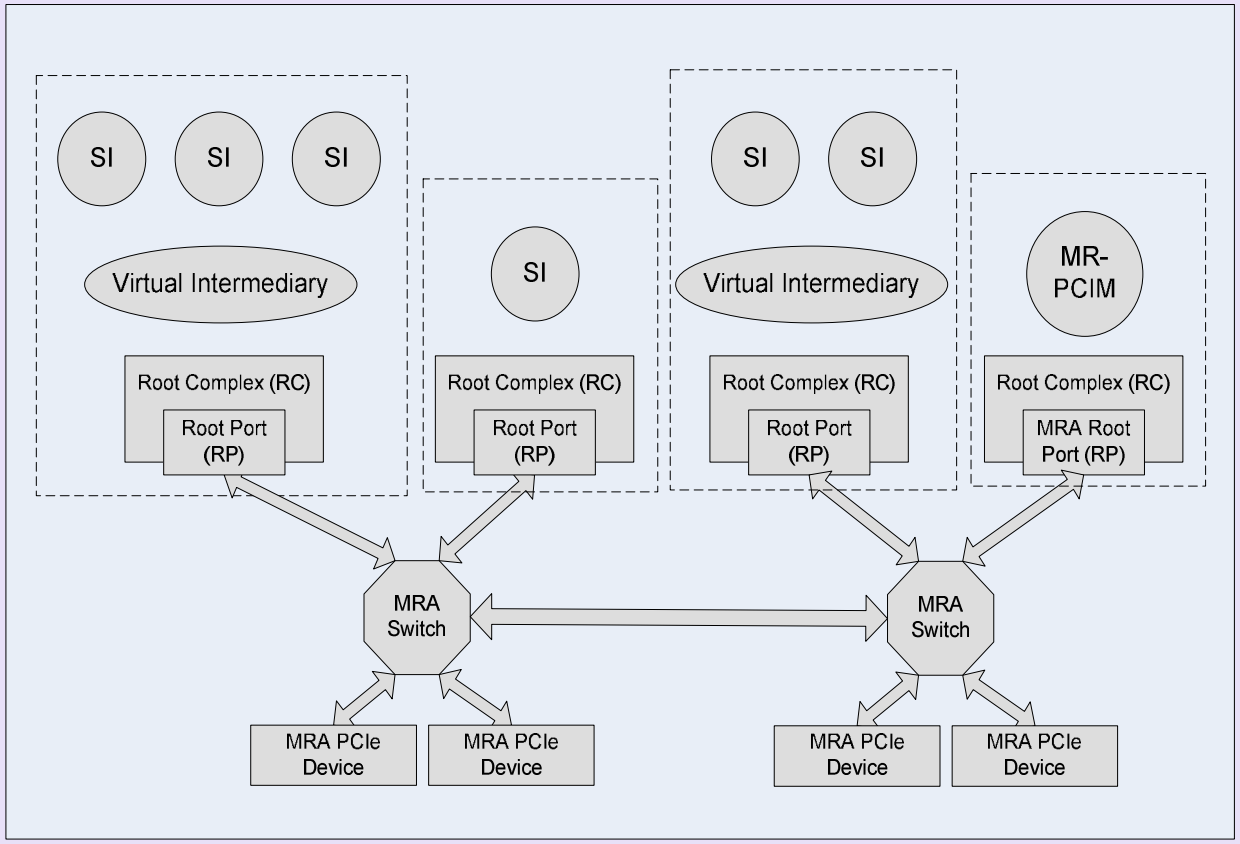


- The MR-IOV topology must contain at least one MRA PCIe Switch.
  - ✓ Multiple MRA PCIe Switches can be provisioned and interconnected in a variety of topologies – tree, fat-tree, star, mesh, etc.
  - ✓ A MRA PCIe Switch typically contains two or more upstream Ports. In a single-stage or a switch at the top of the topology, each upstream Port connects to a RP which acts as the root of the VH.

Comparison of Device Types



MR-PCIM



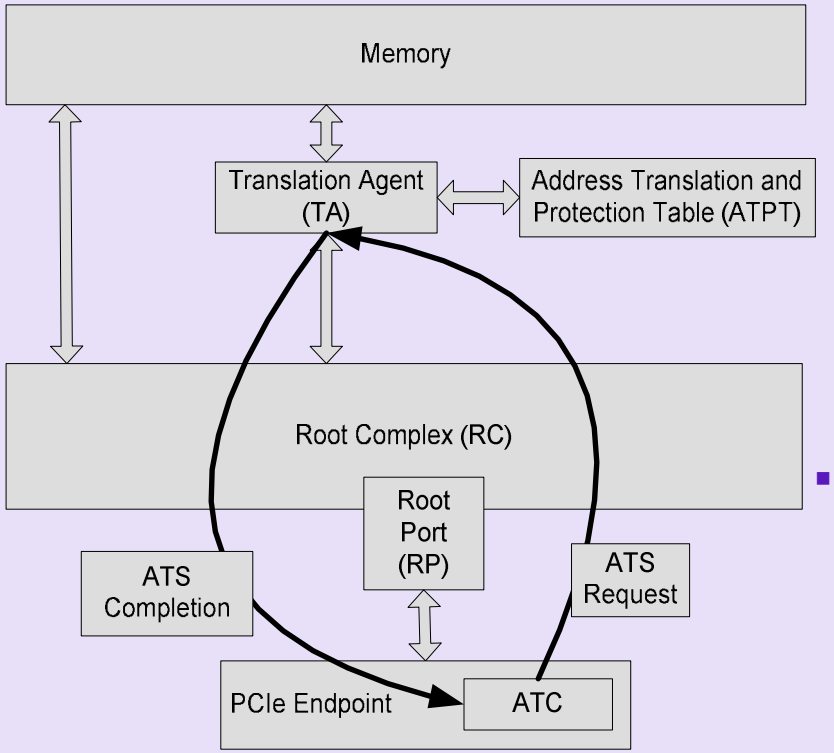
- MR-PCIM
  - ✓ Responsible for MR configuration and event management
    - – Creation VH, MR resource assignment, MR hot-plug, RESET, error handling, etc.
  - ✓ Can be implemented anywhere – above a RC, sideband off switch, etc.

Address Translation Services (ATS)

Introduction to DMA Address Translation

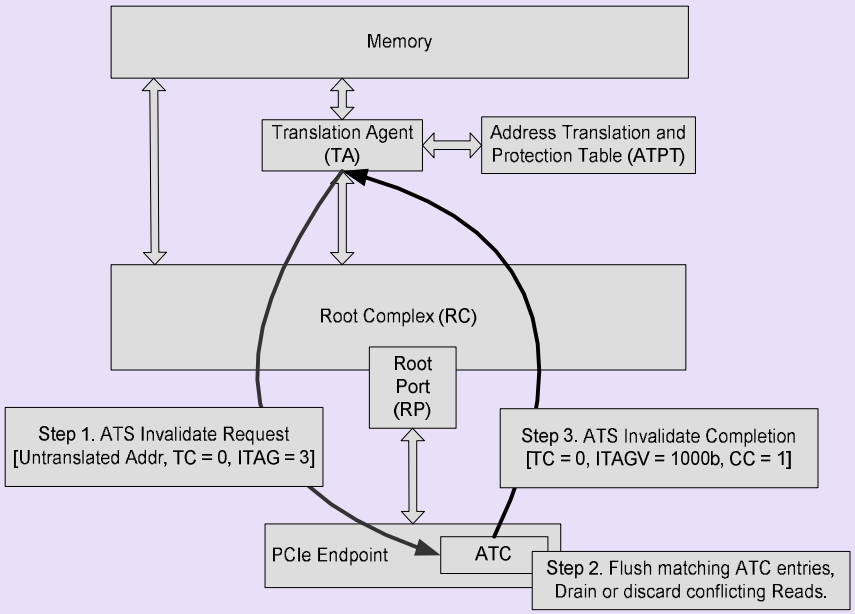
- Translation Agent (TA) performs:
  - ✓ Address translation and an access right validation per Device DMA request
  - ✓ One or more accesses into the ATPT to acquire translation
    - – May need to walk multiple table entries to acquire platform physical address
- Potential Issues with DMA Translation
  - ✓ Increased latency of accesses
    - – Might need one or two accesses to find address of tree associated with a BDF
    - – Might need 3 or 4 accesses to walk the tree
  - ✓ Translation caches (*ATC* or *IOTLB*) will be necessary to reduce overhead.
  - ✓ Caches may not provide good behavior if not sized correctly
    - – Only two possibilities for sizing caches: too large or too small
  - ✓ “Untimely” latency may cause issues with isochronous devices

ATS to the "rescue"



- ATS attempts to mitigate the impact of DMA translation by providing ways for Devices to participate in translation cache management
  - ✓ Device can maintain their own cache of translations – an "Address Translation Cache" (ATC)
  - ✓ TA provides table-walking services to device to avoid excess bus traffic – also means that translation table format is uniform in a system
- Device manages its ATC using its intimate knowledge of future access pattern
  - ✓ Look-ahead for isochronous devices to avoid "untimely" table walk latencies.
  - ✓ High-load devices (graphics) don't thrash ATC in TA.
  - ✓ Application specific caching in devices – ring buffer
  - ✓ Enable peer-to-peer in virtualized bus

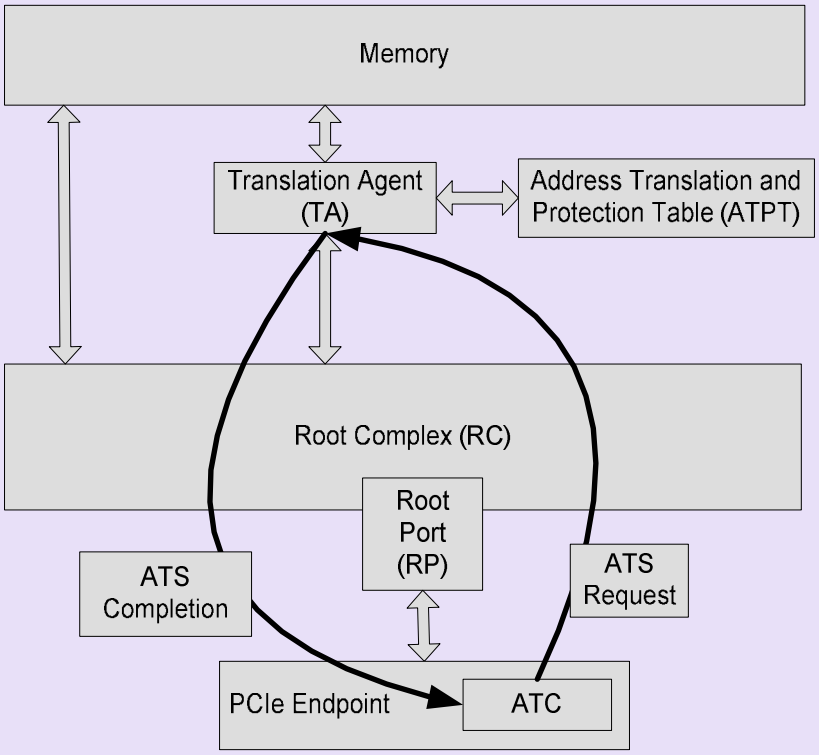
New Protocols for ATS



- Differentiated memory address type – device will be issuing Requests that use both translated and non-translated addresses
- Translation Request – Address Translation Cache (ATC) in device requests a translation from central TA
- Translation Completion – translation is returned in response to Translation Request
- Invalidation Request – when change occurs in central table, need to inform remote ATCs
- Invalidation Completion – when ATC completes the invalidation operation, it needs to tell TA.

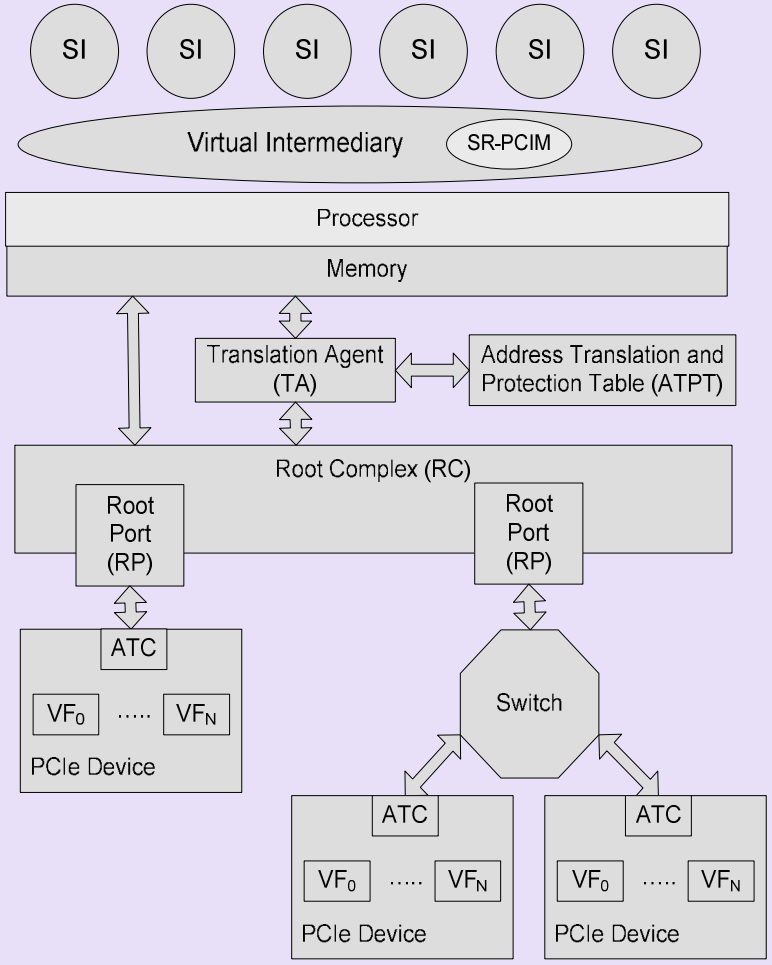
Specification Status and Schedule

ATS Specification Status



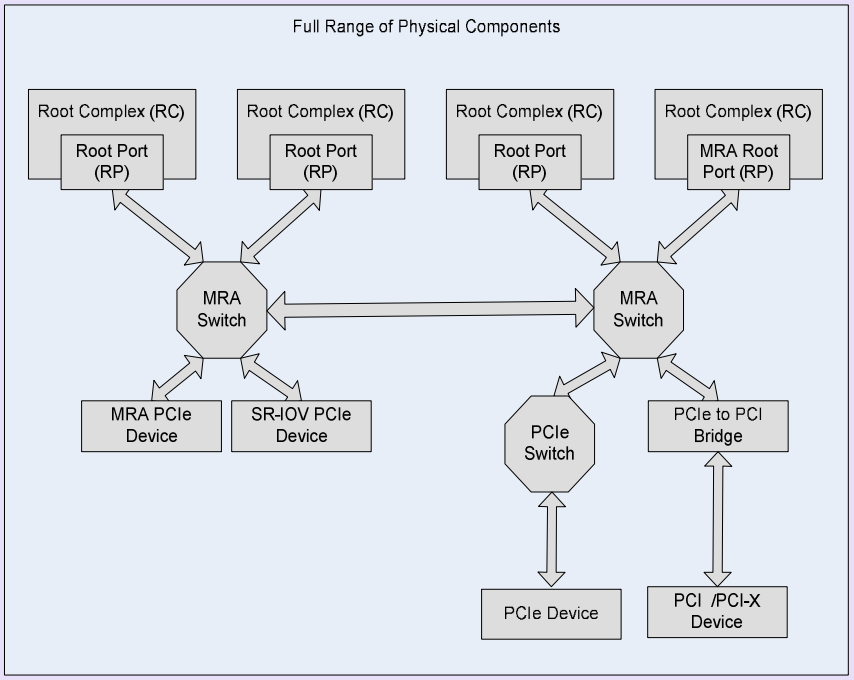
- Per PCI-SIG Specification Development process
  - ✓ Completed draft 0.9 60-day review on November 6, 2006
  - ✓ Workgroup received feedback from multiple member companies
  - ✓ Incorporating feedback now
- Version 1.0 December 2006

SR-IOV Specification Status



- Per PCI-SIG Specification Development process
- Draft 0.5 completed
- Draft 0.7 December 2006
- Draft 0.9 1Q2007
- Version 1.0 2Q2007

MR-IOV Specification Status



- Per PCI-SIG Specification Development process
- Draft 0.5 deliver November 2006
- Draft 0.7 1Q2007
- Draft 0.9 2Q2007
- Version 1.0 early 3Q2007

Questions


PCI
SIG®