PCI Express* Hot Plug Implementation for Servers

Steve Krig
Staff Software Engineer
Enterprise Platform Group
Intel Corporation

September 2003

Agenda

- Hot Plug Overview
- Conventional PCI Hot Plug
- PCI Express* Advancements
- PCI Express* Example
- Summary

Hot Plug
Overview

What is Hot Plug?

- Hot Plug Enables Devices to be Added/Removed to/from the System Without Interrupting Normal Server Operations (No Reset or Power Down).

Hot Plug Increases Overall Server RAS (Reliability, Availability and Serviceability)

Hot Plug
Overview

PCI Express* “Native” Hot Plug History:

- PCI Hot Plug Introduced 1997
  - SHPC Improvements 2001
- PCI Express Native Hot Plug 2002
  - Fully Supports Legacy PCI Hot Plug
  - Based on SHPC
  - SHPC Usage Model Becomes a Base Platform Capability and H/W Support Requirement
- *Native* Implies Bulk of Code Exists in the O/S
  - Shift away from Platform Vendor Specific Drivers/Firmware
  - Standard H/W interface (Replacing FW Interface)

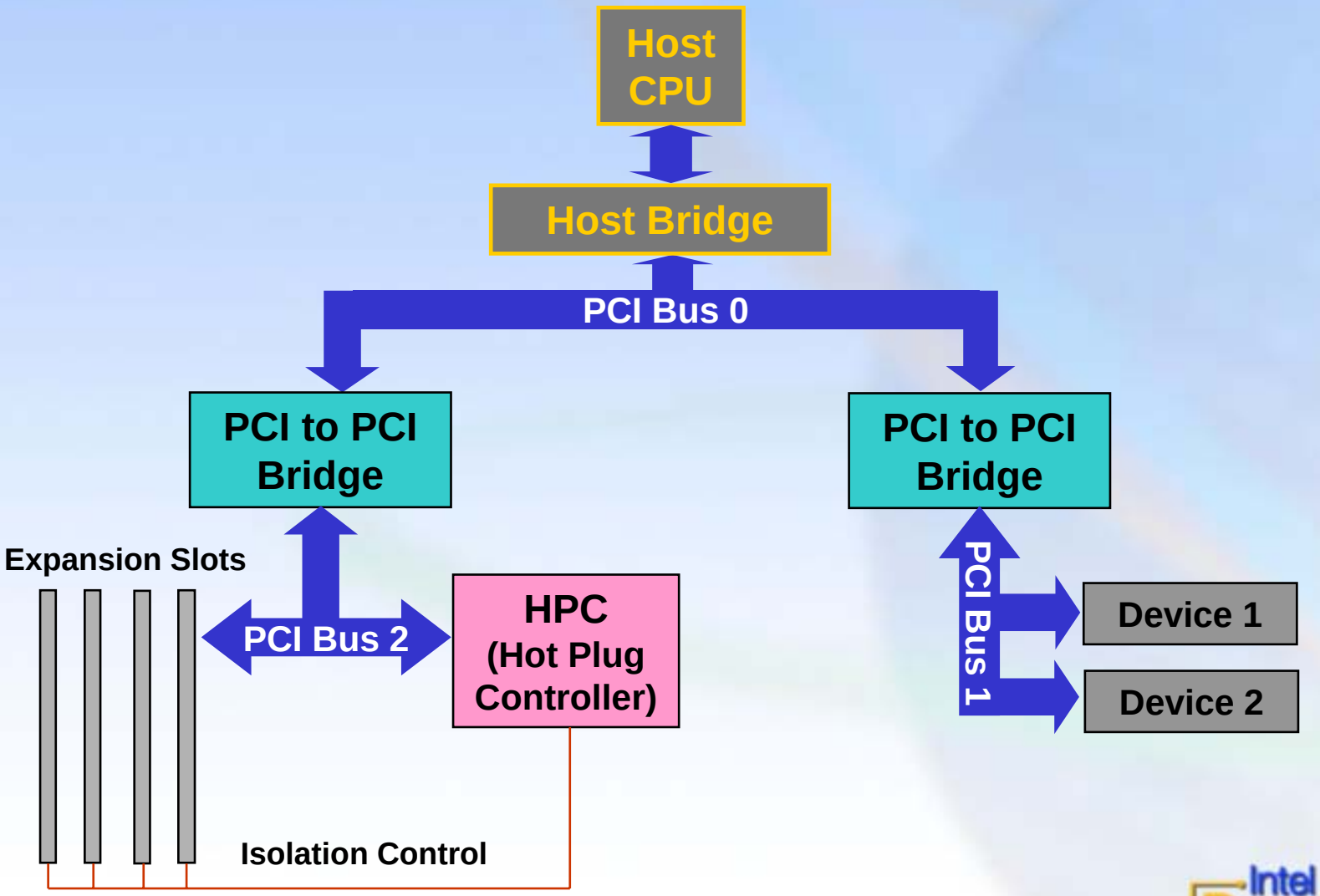
Based Upon Over Five Years of Hot Plug Industry Standards Experience

Agenda

- Hot Plug Overview
- Conventional PCI Hot Plug
  - PCI, ACPI
  - Implementation Issues/Considerations
- PCI Express* Advancements
- PCI Express* Example
- Summary

PCI Hot-Plug

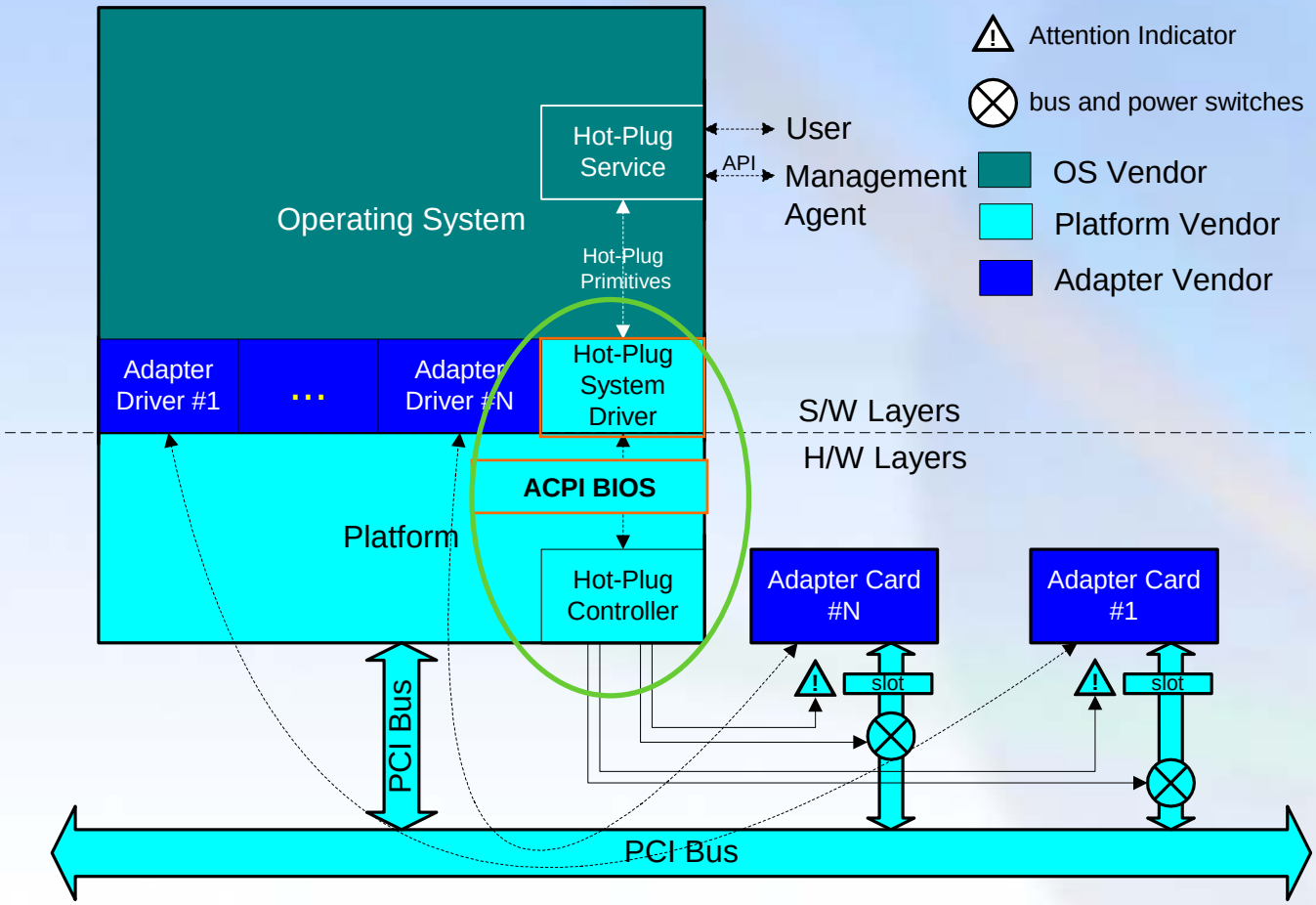
Conventional (Legacy) Bus Hierarchy



HPC and Related SW Infrastructure is Vendor Specific…

PCI Hot Plug

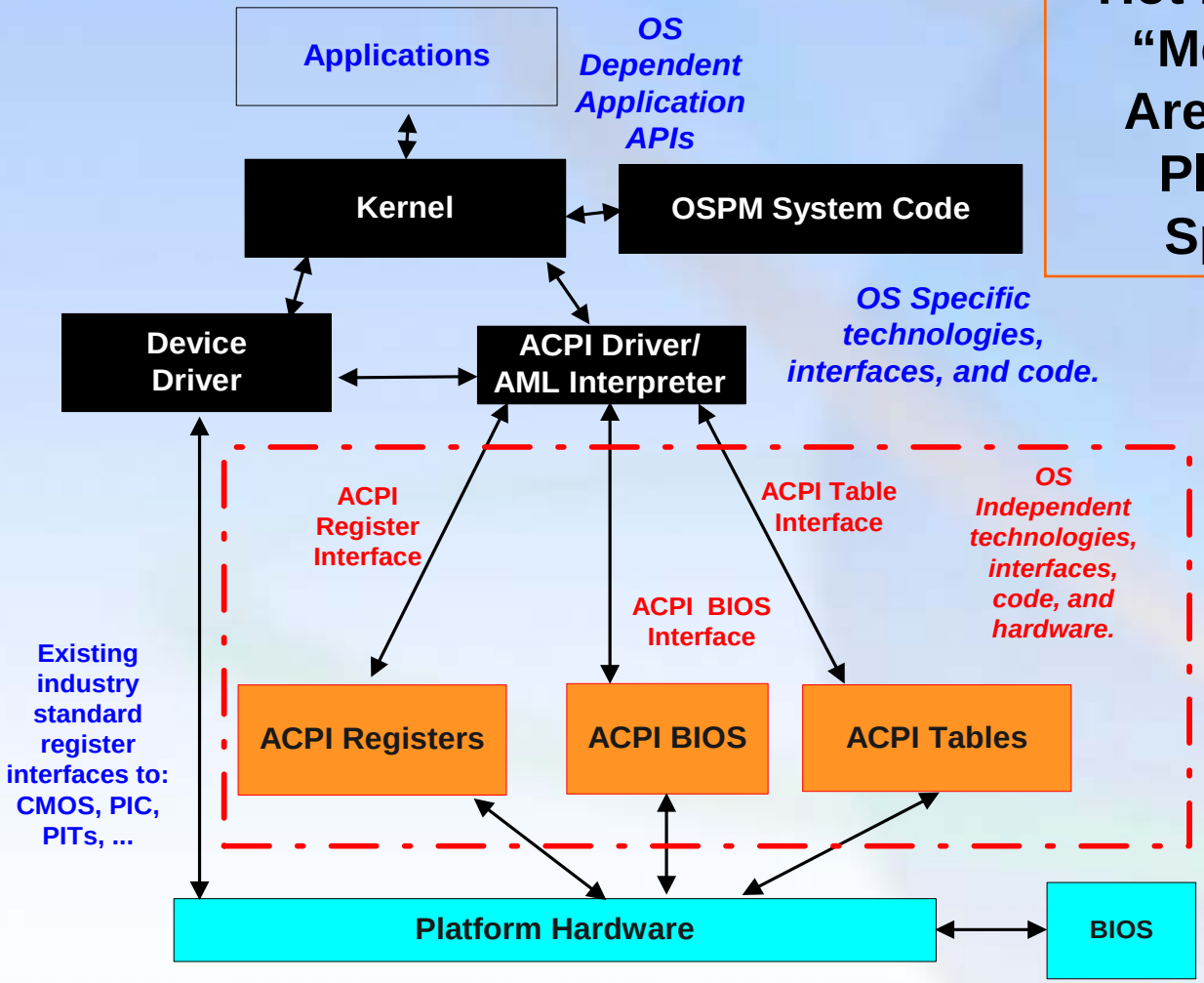
Conventional H/W and S/W Components



Relies on Vendor Specific ACPI BIOS and HPC Driver

Conventional Hot Plug
ACPI BIOS

Hot Plug ASL "Methods" Are Vendor Platform Specific



- ACPI Spec Covers this area.
- OS specific technology, not part of ACPI.
- Hardware/Platform specific technology, not part of ACPI.

* Other names and brands may be claimed as the property of others.

Conventional Hot Plug

Example: Hot Insertion Event Flow

1. User: opens chassis and inserts the hardware in the slot.
2. User: initiates slot power-up.
3. Hot-Plug PCI controller: Asserts a GPE.
4. Core chip set: Raises an SCI.
5. ACPI driver: Clears the GPE event - runs “Methods" to determine slot and then Notify() ACPI Driver
6. ACPI driver: Executes Methods for the devices specified in Notify() in the previous step.
7. ACPI driver: Tells the PCI driver to enumerate the bus.
8. PCI driver: Reads configuration space to identify the device.
9. PCI driver: Loads and starts the drivers for all functions of the device.
10. Device drivers: Request the functions be turned on.
11. PCI driver: Writes to configuration space to turn on the device according to PCI Power Management specifications.
12. Device driver: Begins using the device normally.

Conventional PCI Hot Plug
Issues/Considerations

- Disjoint Platform Architecture – Too Many Implementation Options
- Differing Usage Models
- Disjoint Compliance Model
  - Dissimilar Vendors - OEM Systems; IHV Adapters
- Differing Platform Support Models
- Differing BIOS/Drivers/Controllers

Vendor Specific Implementations Increase IT Support Costs/Burden and Degrade Overall Platform RAS

Agenda

- Hot Plug Overview
- Conventional PCI Hot Plug
- **PCI Express* Advancements**
  - **The Need for a Standard Usage Model**
  - **PCI Express Base Architecture: SHPC**
  - **Form Factor Implementation Considerations**
  - **Software Implementation Considerations**
- PCI Express* Example
- Summary

* Other names and brands may be claimed as the property of others.

PCI Express* Advancements

The Need for a Standard Hot Plug Usage Model

Why a "Standard" Usage Model?

- **Simplifies Overall Platform Model**
  - **Defines IHV and OEM Support Options and Requirements**
    - **PCI Express* Standardizes Register Requirements: Chassis/Slot/Endpoint**
  - **Compliance Tests Can Verify Register Support, Functionality and System Interoperability of Chassis, Slots and Endpoints**
    - **Standard Hardware Requirements Enable Interoperable Solutions**

Simplifies Overall Platform Implementation: Improves Reliability

PCI Express* Advancements

The Need for a Standard Hot Plug Usage Model

Why a "Standard" Usage Model ? (cont'd.)

- **IT/Data Center Manager Assurance**
  - **Compliance to Standard model**
    - **Dissimilar Vendors - OEM Systems; IHV Adapters**
  - **Serviceability:**
    - **Delivers True Slot based FRU Scheme to Server Racks**
    - **Data Center Technicians Can Focus on One Simple Formula for Floor Service Across Differing Vendor Systems/Adapters**

Simplifies Tech. Aspects of Service: Improves Serviceability, Increases Availability

Agenda

- Hot Plug Overview
- Conventional PCI Hot Plug
- **PCI Express* Advancements**
  - The Need for a Standard Usage Model
  - **PCI Express Base Architecture: SHPC**
  - Form Factor Implementation Considerations
  - Software Implementation Considerations
- PCI Express* Example
- Summary

PCI Express* Base Architecture

SHPC: Standard Hot Plug Controller

Component	Description	Platform Areas		
		System Software	Chassis/ Slot	Module /Card
Power Indicator	Visually Indicates power state (on/transitioning/off)	Yes	Yes	Yes
Attention Indicator	Identifies slot/card for Service	Yes	Yes	Yes
MRL	Manually Operated Retention Latch - Retains Card in Slot	No	Yes	Yes
MRL Sensor	Allows the port and system SW to detect MRL has been opened	Yes	Yes	No
Electromechanical Interlock	Prevents Opening of MRL (when slot is powered)	No	Yes	Yes
Attention Button	Allows Tech. to initiate Hotplug operation	Yes	Yes	Yes
Software Interface	SW Usage Model and Interface; Event/Indicator monitoring	Yes	No	No
Slot Numbering	Visually maps slot to id recognized by SW (slot register)	Yes	Yes	No

SHPC Defines the Usage Model and Platform Components

PCI Express* Base Architecture

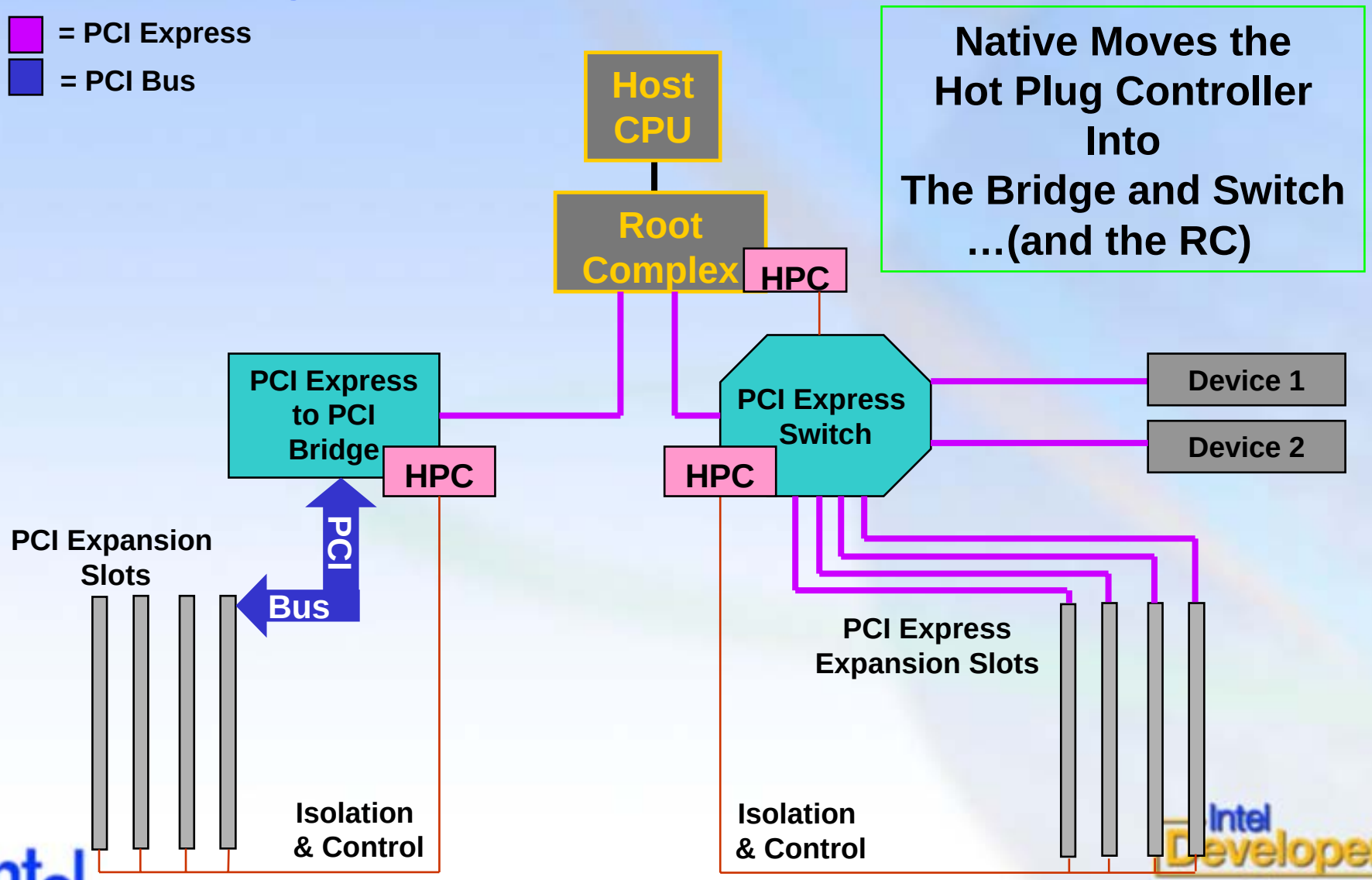
"Native" Hot Plug

- PCI Express Native Hot Plug Follows the SHPC Standard Usage Model
  - Same Basic Platform Components
- PCI Express Native Hot Plug Registers Differ from SHPC 1.0
  - Related Functionality is Same as SHPC 1.0
  - Native: Root Ports and Downstream Ports of Switches are Capable of Supporting Hot Plug
- Single O/S System Driver Can Service Both SHPC and Native Hot Plug

PCI Express Native and SHPC Export the Same Usage Model to the O/S Through Different Register Sets

PCI Express* Base Architecture

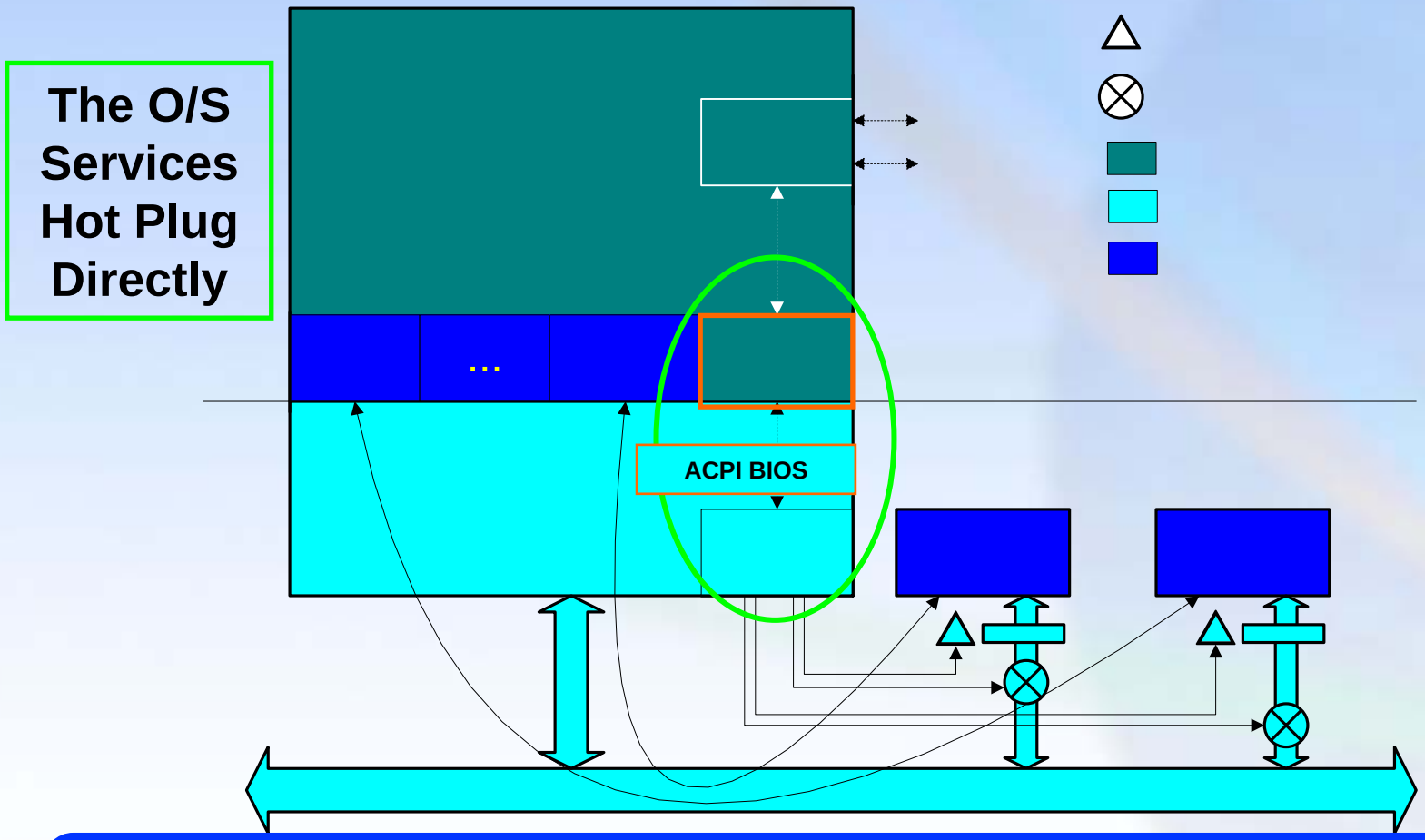
Hot-Plug Bus Hierarchy



* Other names and brands may be claimed as the property of others.

PCI Express* Base Architecture

Delta to Conventional PCI Hot Plug



Eliminates Hot Plug Dependencies on ACPI BIOS and Platform Specific HPC Driver

PCI Express* Base Architecture

Native Hot Plug vs. Firmware Control

- OS Calls ACPI BIOS Extension (Control Method)
- Required on PCI Express* Platforms Shipping Prior to "Native" Hot Plug Support in O/S
  - Platforms Ship With Native Support in H/W and BIOS
  - O/S Decides When to Turn Native On
  - Provided for Each Port That is Hot-Plug Capable and Controlled by ACPI BIOS
  - ACPI BIOS Restores any Hot Plug Signals (INTx/PME) for O/S Control

Legacy O/S Compatibility: An Operating System With Native Hot Plug Support Calls this Control Method to Request ACPI BIOS to Stay Out of the Way for Related Event Servicing

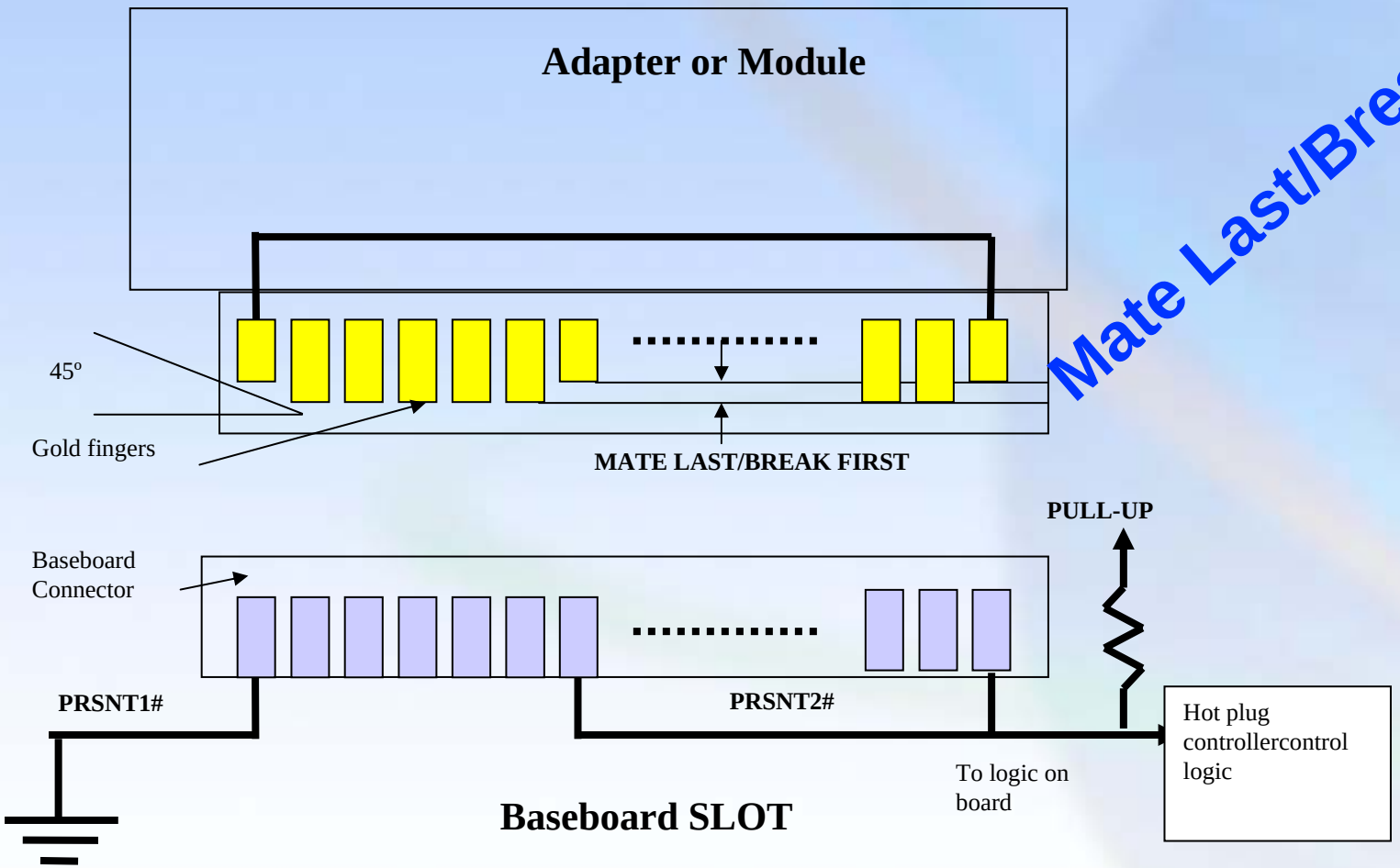
Agenda

- Hot Plug Overview
- Conventional PCI Hot Plug
- **PCI Express* Advancements**
  - The Need for a Standard Usage Model
  - PCI Express Base Architecture: SHPC
  - **Form Factor Implementation Considerations**
  - Software Implementation Considerations
- PCI Express* Example
- Summary

* Other names and brands may be claimed as the property of others.

Form Factor Considerations

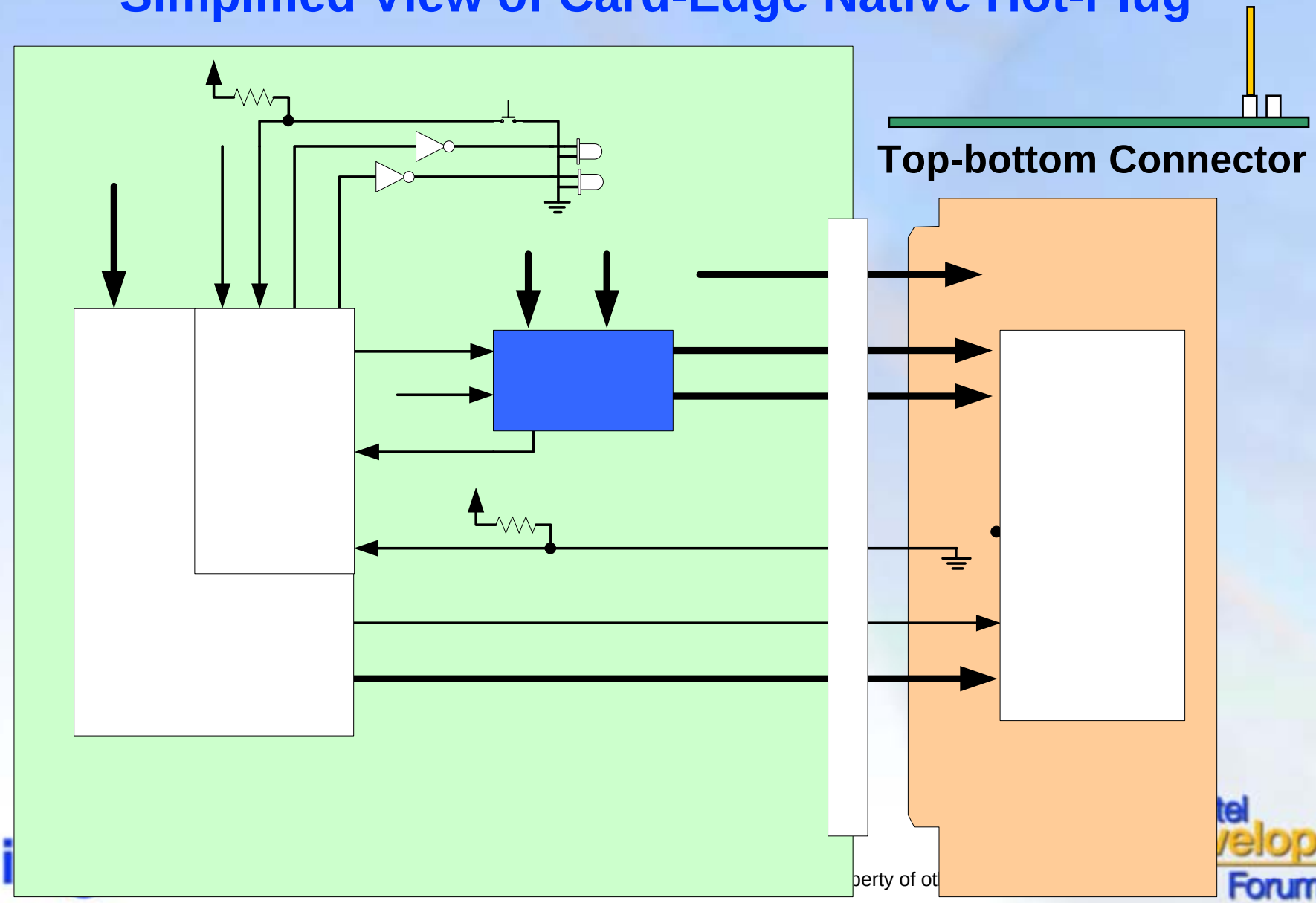
EVO Presence Detect (Hot Plug)



Hardware Protection for Surprise Removal - OS Requires Attention Button or MRL Sensor to Assure SW Integrity.

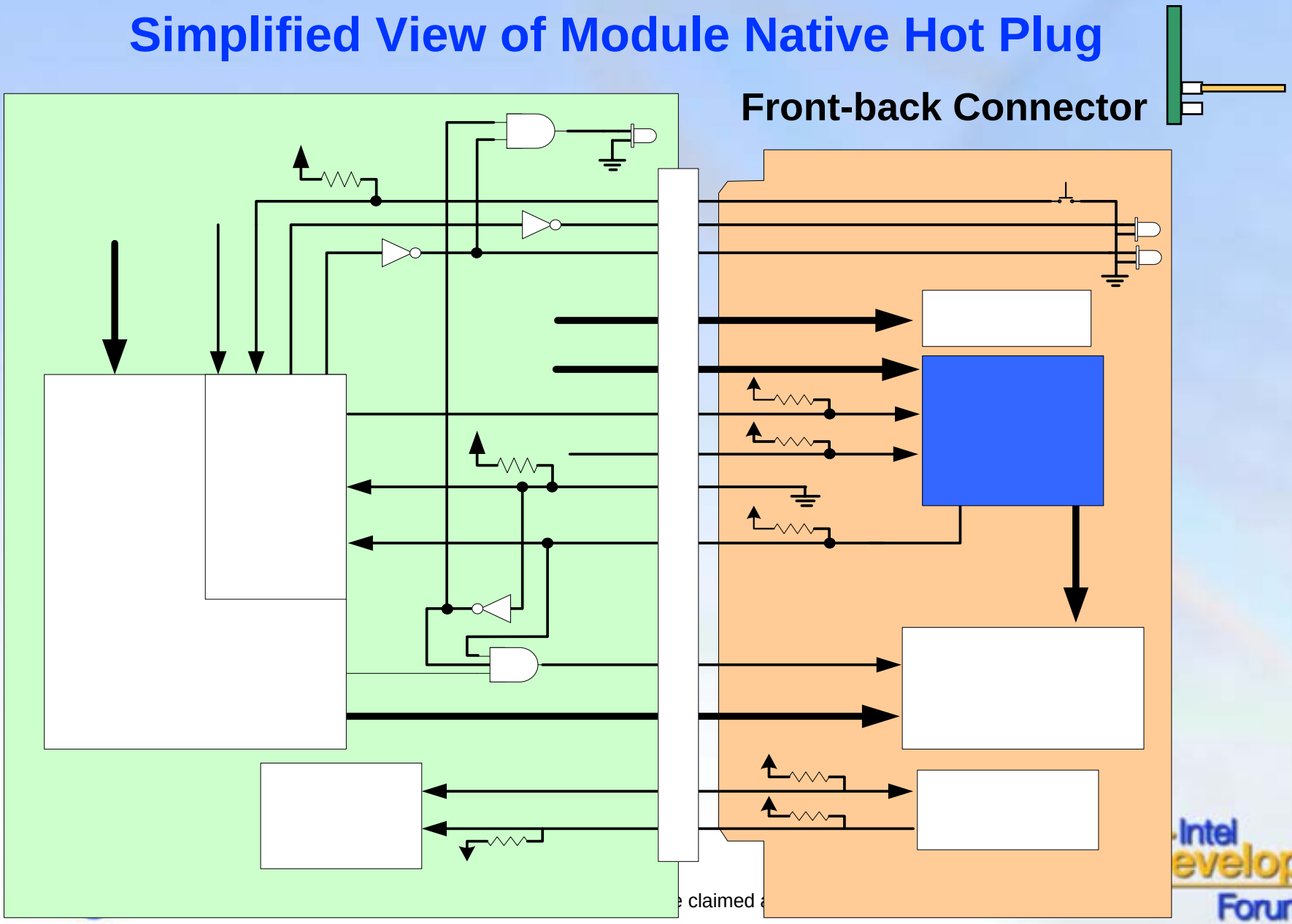
Form Factor Considerations

Simplified View of Card-Edge Native Hot-Plug



Form Factor Considerations

Simplified View of Module Native Hot Plug



Form Factor Considerations

A Note on Modules & Slots…

- Both Use Mate/Last Break/First Pins at Connector
  - "Hot Surprise" Protection for Hardware
  - Module: Front-Back Connector Eliminates Need to Open Chassis
- Support Same Software Model (Legacy & Native)
- Module: Additional Signals at Connector
  - Route Native Control Signals from Root Port/Switch to Actual Power Control on the Module
- PCI Express* Native is Less Expensive to Implement Than Legacy Hot Plug
- Module Hot Plug Shares Cost Between Host Side and Module – Reducing Host Costs Even More
  - Only Pay for Power Required to Drive Functionality of a Specific Module

Modules: "Pay as you Go/Grow"

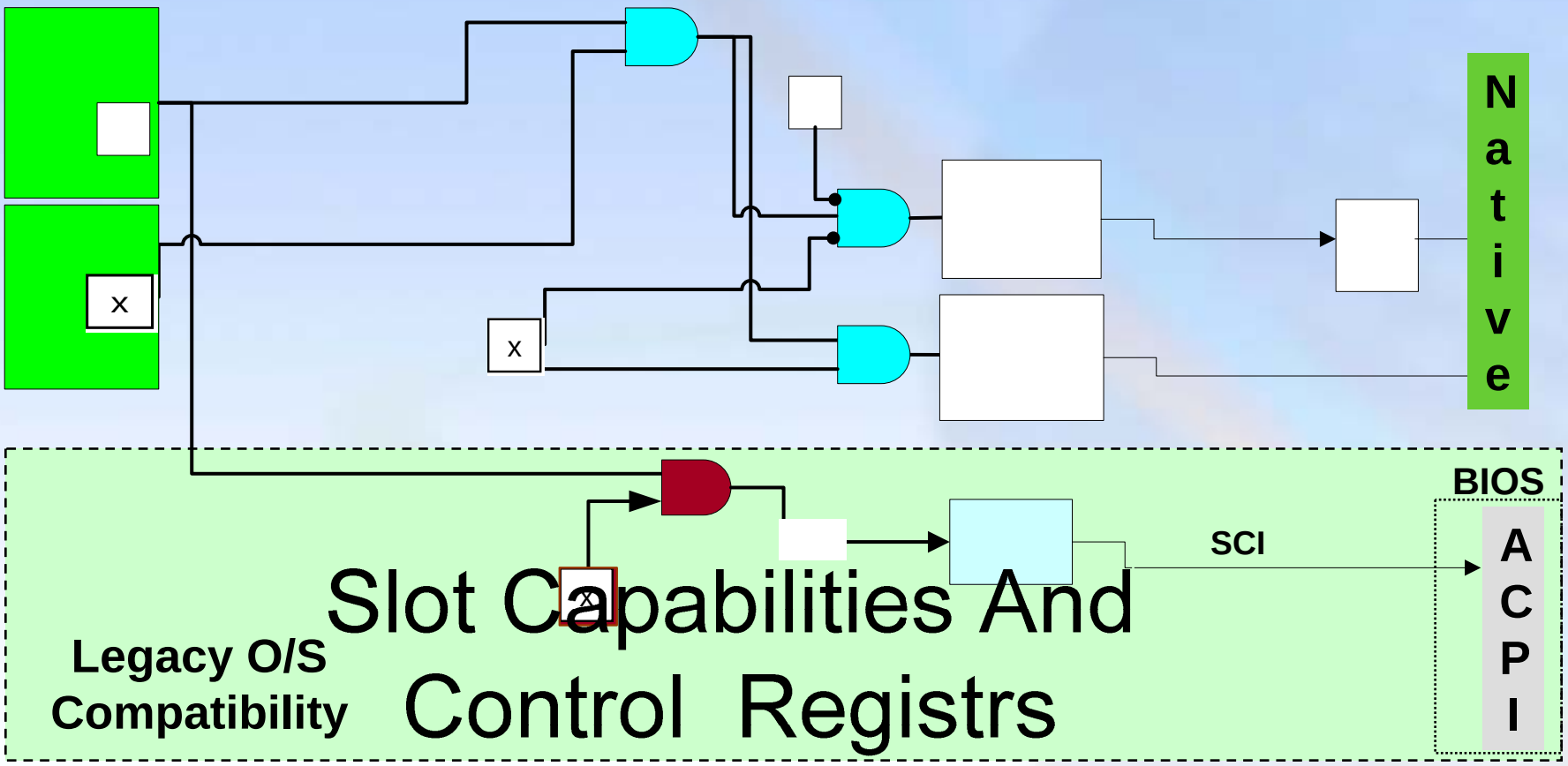
Agenda

- Hot Plug Overview
- Conventional PCI Hot Plug
- **PCI Express* Advancements**
  - The Need for a Standard Usage Model
  - PCI Express Base Architecture: SHPC
  - Form Factor Implementation Considerations
  - **Software Implementation Considerations**
- PCI Express* Example
- Summary

* Other names and brands may be claimed as the property of others.

Software Considerations

PCI Express* Hot Plug Event Logic



PCI Express "Native" Hot Plug Raises the Event Directly to Single/Common O/S System Driver

Software Considerations

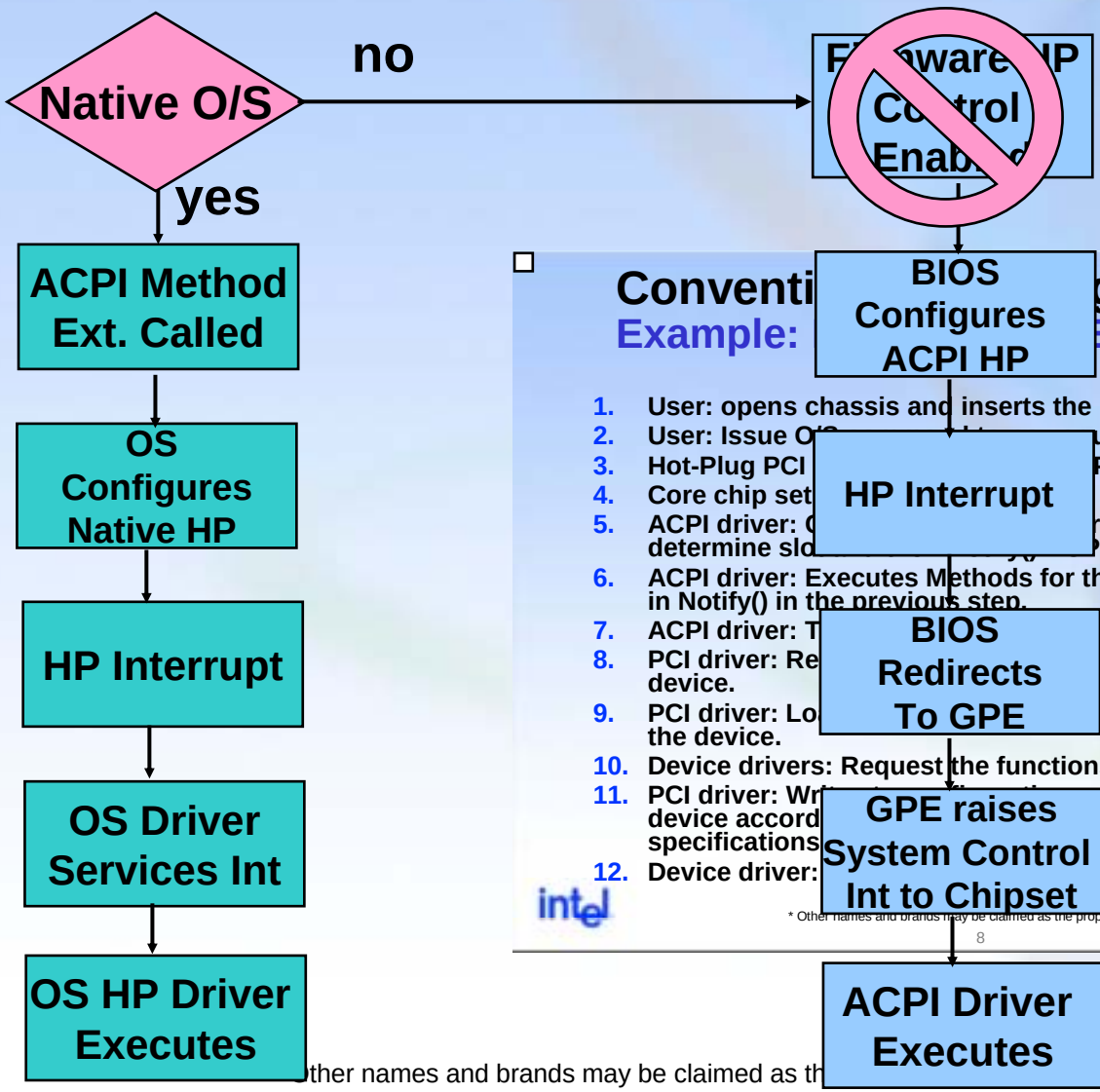
A Quick Note on MSI and PCI Express*…

- **Both INTx and MSI are Message Based on PCI Express (No Side-Band Signals)**
  - Int. Uses Sideband on PCI/PCI-X
- **MSI Means Greater Device Performance**
  - Multiple Interrupts per. Device
  - Direct Service to O/S Driver (bypass APIC)
  - MSI-X Increases Performance Opportunities on MP Systems
- **MSI Means Greater Platform Reliability**
  - MSI Resolves Legacy O/S Resource Balancing Issues Associated w/APIC and Shared INTx
  - Driver Additions are Needed for MSI (Dependent on O/S Implementation)

Message Signaled Interrupts Increase Performance and Reliability of Server Platforms

Software Considerations

Operating System Hot Plug Event Service Model



Example: Event Flow

1. User: opens chassis and inserts the hardware in the slot.
2. User: Issue O
3. Hot-Plug PCI
4. Core chip set
5. ACPI driver: runs "Methods" to determine slo I Driver
6. ACPI driver: Executes Methods for the devices specified in Notify() in the previous step.
7. ACPI driver: T umerate the bus.
8. PCI driver: Re e to identify the device.
9. PCI driver: Lo ers for all functions of the device.
10. Device drivers: Request the functions be turned on.
11. PCI driver: Wr ace to turn on the device accord gement specifications
12. Device driver: e normally.

Software Considerations

A Look at Hot-plug on Linux…

= OS/Platform Independent

= OS Vendor Specific

= Platform Vendor Specific

Hot Plug User Interface

User Space

Kernel

PCI Hot Plug Module

Platform Specific Hot Plug Driver Module

Kernel

Platform

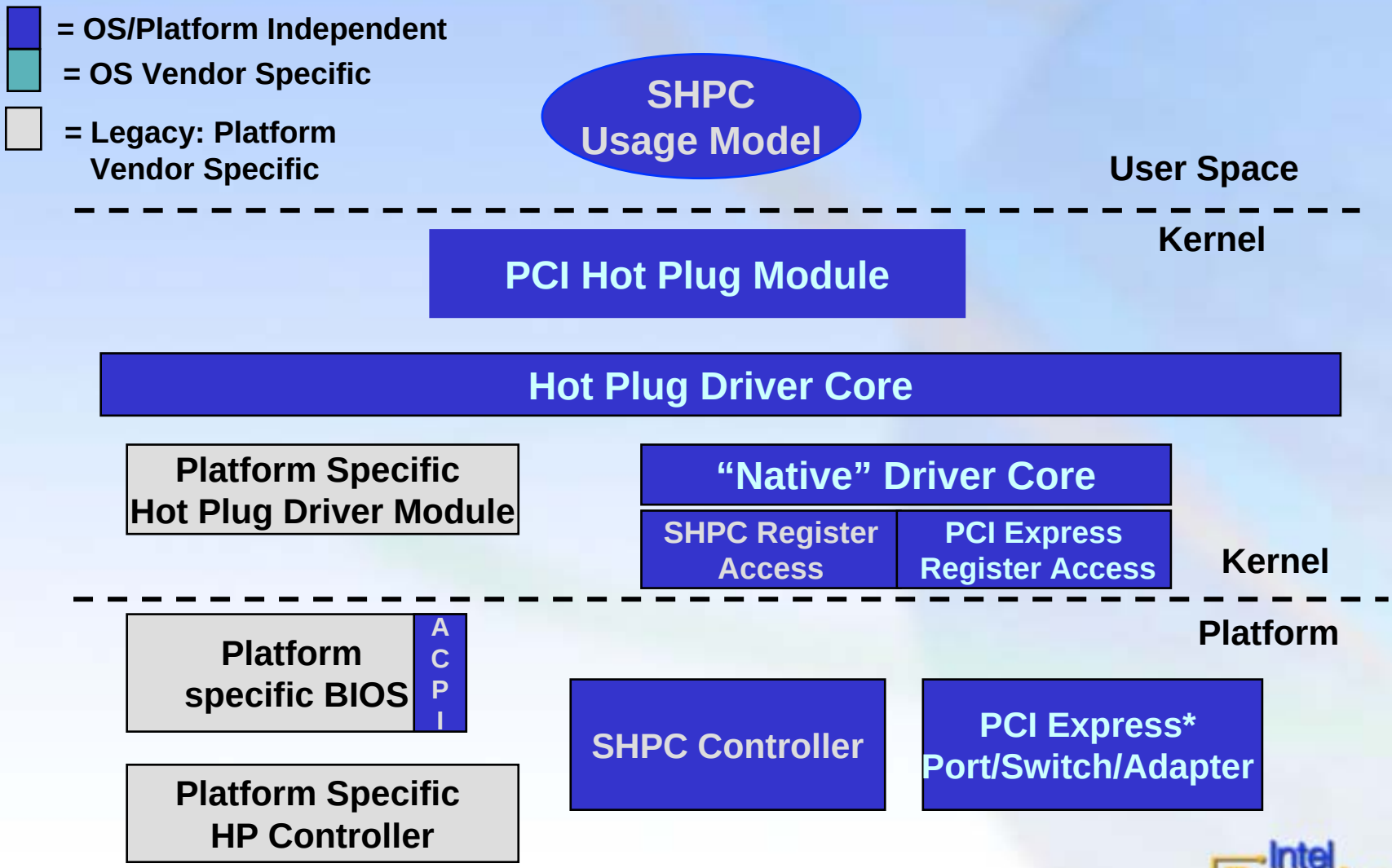
Platform specific BIOS

Platform Specific HP Controller

Functionality is Vendor/Platform Specific

Software Considerations

A Look at Linux: Legacy + Native



Common Cross-vendor Model Improves Reliability

Software Considerations

A Note on Power Budgeting…

- **PCI Express* Introduces Slot Power Limiting Messages**
  - **Downstream from RC or Switch to Upstream Port of a Device/Endpoint**
- **Allows the O/S to Budget Power by Fixing Limits on the Amount of Power Available (to Devices) per. Slot.**
  - **Eliminate Power "Brown-outs"**
- **A Device Plugged into a Limited Slot Must Not Consume More Power than Budgeted**

Power Budgeting Improves Reliability of Server Platforms

* Other names and brands may be claimed as the property of others.

Software Considerations

Summary: Native Hot Plug Driver Considerations

- **Operating System Drivers**
  - **PCI Configuration/Enumeration (Cap ID 10h)**
  - **Native/SHPC Driver (Slot/Device Registers)**
    - **Builds on SHPC Driver Base**
  - **Native Interrupt Servicing (INTx/MSI)**
  - **Power Budgeting**
- **Vendor Device Drivers**
  - ***No Changes Required to IHV Adapter Drivers. Vendor Hot Plug Driver no Longer Needed.***
  - **Eliminates Hot Plug Driver Dependency on ACPI BIOS**
    - **Vendor ACPI BIOS Must Include ACPI Control Method Extension to Disable Firmware Hot Plug Control**
  - **Reduced Cost: No Platform Vendor Specific HPC Drivers**

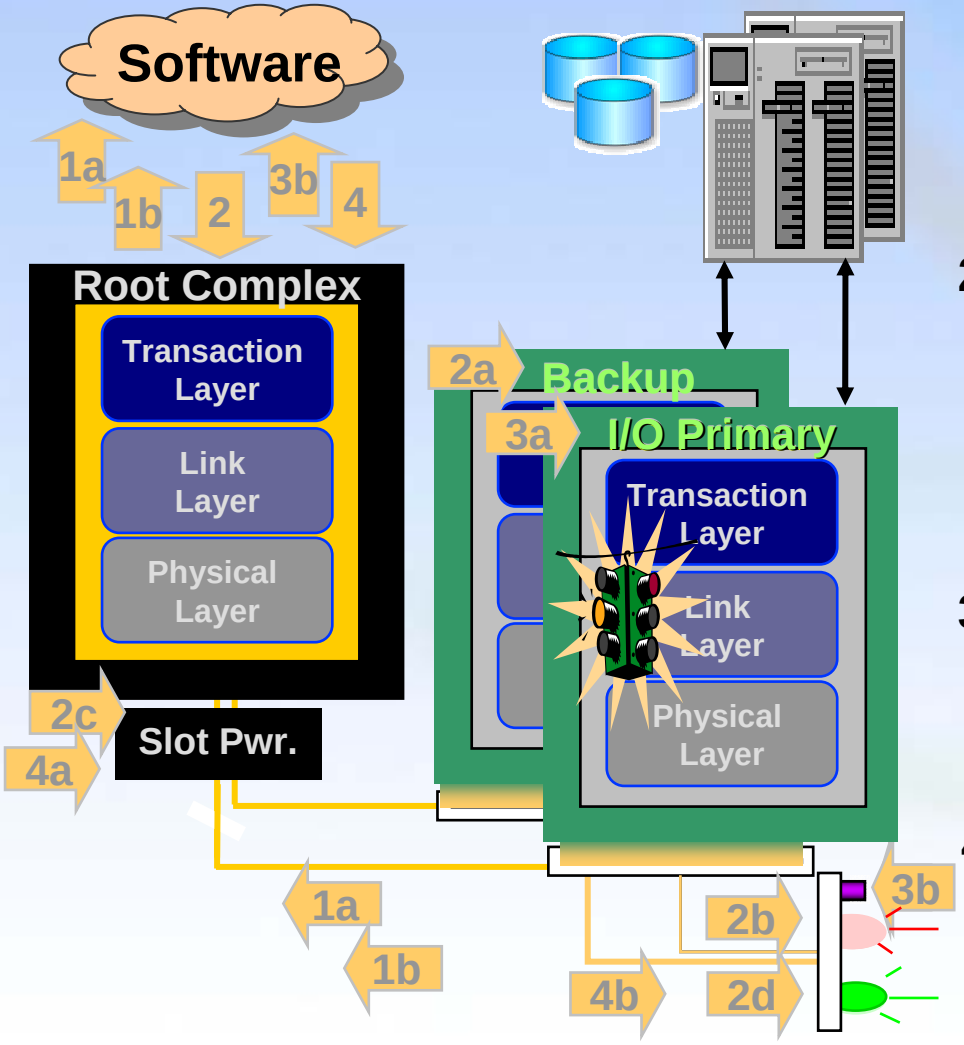
Native Hot Plug Greatly Reduces the OEM an IHV Development and Support Burden

Agenda

- Hot Plug Overview
- Conventional PCI Hot Plug
- PCI Express* Advancements
- **PCI Express* Example**
- Summary

PCI Express* Server RAS & Hot Plug

Example



1. I/O Primary Detects ERR_COR
   - a. *Reports Replay Timeout*
     SW Reads/Clears Err. Log
   - b. *Reports Receiver Error*
     SW Reads/Clears Err. Log
     - Predicts Link Failure
2. SW Initiates Corrective Action
   - a. Switches I/O to Backup
     - Pages Tech. To Replace
   - b. Flashes Attention Indicator
   - c. Disables Primary Slot Power
   - d. Turns off Power Indicator
3. Tech. Locates Server/Slot
   - a. "Hot Swaps" out/in I/O Adapter
   - b. Depresses Attention Button
   - – *Attn. Button* Event to SW
4. SW Brings New Backup on-line
   - a. SW Enables Slot Power
   - b. SW Lights Power Indicator
     - Turns off Attn. LED

Native Hot Plug Improves Platform RAS

Agenda

- Hot Plug Overview
- Conventional PCI Hot Plug
- PCI Express* Advancements
- PCI Express Example
- **Summary**

* Other names and brands may be claimed as the property of others.

PCI Express* Native Hot Plug

Summary

- **Based On SHPC 1.0**
  - **Builds on Over 5 Years Experience in SIG**
- **Improves Platform Reliability**
  - **Bulk of Support in O/S: Common Driver/Controller Model**
    - **Eliminates Platform Specific BIOS/Drivers/Controllers**
- **Improves Platform Availability/Serviceability**
  - **Common Usage Model Decreases Downtime**
- **Reduces Platform Costs**
  - **No Platform Vendor Specific HPC or SW Stack to Support**
  - **Slot Control Saves $$s Over Legacy (PCI/PCI-X)**
  - **Greatest Component Savings Coming With Modules**

PCI Express Native Hot Plug Improves RAS for the Enterprise

PCI Express* Native Hot Plug

Summary (cont'd.)

- Include PCI Express* Native Support in Your Hardware Designs
- Server OEMs: Build 2004 Servers on this Strong RAS Foundation
  - Software Developers: Turn on Slot Configuration of PCI Express (Cap List 010h) and Native Hot Plug.
  - OEMs: Include ACPI Control Method Extension in your Platform BIOS to Disable ACPI Hot Plug for Native Control
  - IHVs: Include Native Hot Plug and MSI support in your designs.
  - OEMs: Look for Adapters/Drivers Supporting MSI/MSI-X for Improved Platform Performance and Reliability
  - OEMs: Deploy Integrated Sever Products from Components Supporting PCI Express Native Hot Plug

Collateral

- **Intel® Developer Network for PCI Express* Architecture**
  - **http://developer.intel.com/technology/pciexpress/devnet/**
    - **See the Hot Plug Article in the Enterprise Area**
- **PCI Special Interest Group**
  - **http://www.pcisig.org**
- **Technical Book for Developers**
  - ***Introduction to PCI Express***
    A Hardware and Software Developer's Guide
    - **Essential for developers implementing PCI Express**
    - **More info at www.intel.com/intelpress**
    - **Purchase Intel Press books from Amazon.com, etc.**

* Other names and brands may be claimed as the property of others.

Definition of Acronyms used in this presentation

- ACPI: Advanced Configuration and Power Management Interface
- ASL: ACPI Source Language
- HPC: Hot Plug Controller
- INTx: Legacy Interrupt
- MSI: Message Signaled Interrupt
- O/S: Operating System
- RAS: Reliability, Availability, Serviceability
- MRL: Manual Retention Latch
- RC: Root Complex
- LED: Light Emitting Diode
- SHPC: Standard Hot Plug Controller
- FRU: Field Replaceable Unit