

PCI



SIG[®]



PCI Express® Port Bus Driver Support for Linux®

**Tom Long Nguyen
Software Engineer
Intel® Corporation**

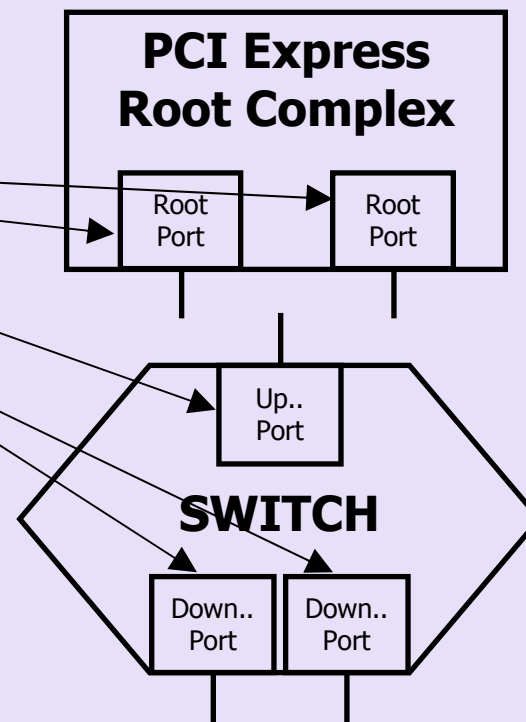


Agenda/Overview

- PCI Express Port Topology
- Linux® PCI Driver Model
- Port Bus Driver
- Service Driver and Examples
- Device Driver Impact and Next Steps

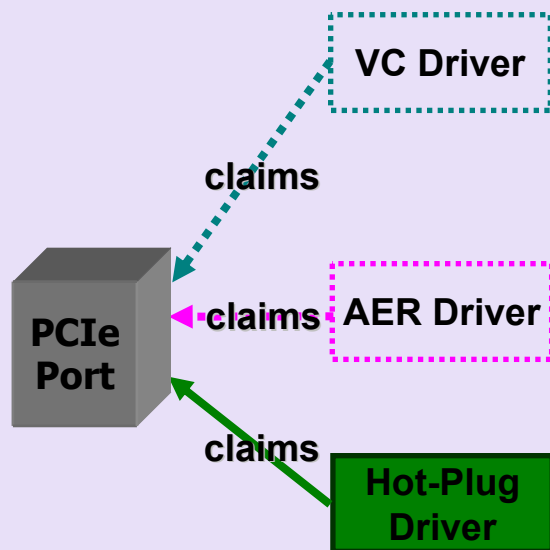
PCI Express Port Topology

- Categorizes PCI Express Ports into:
 - ✓ Root Port
 - ✓ Switch Upstream Port
 - ✓ Switch Downstream Port
- Each PCI Express Port may support none, some or all the following services:
 - ✓ Native hot-plug (HP),
 - ✓ Power management (PME),
 - ✓ Advanced error reporting (AER), and
 - ✓ Virtual channels (VC)

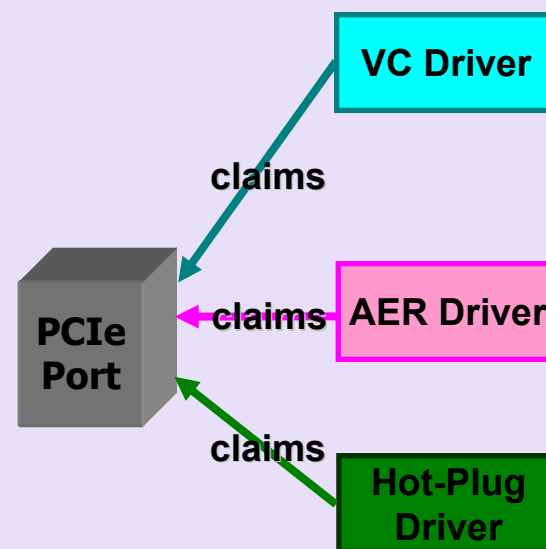


Linux® PCI Driver Model

- Restricts a device to a single driver based on PCI device ID
 - ✓ Once a driver is loaded, no other drivers for that device can be loaded



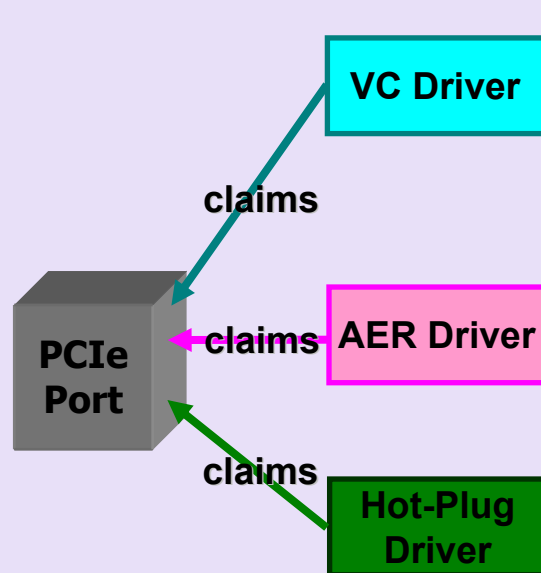
Current Model



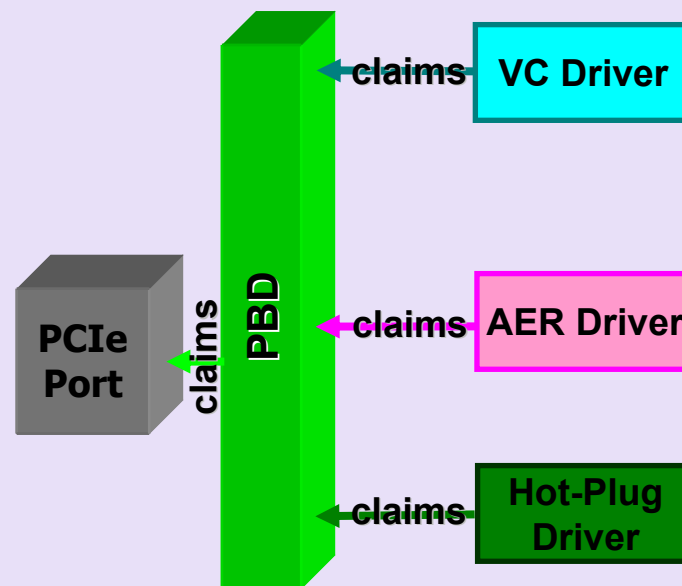
Desired Model

Motivation

- Numerous services available on a PCI Express port
- Enable multiple services to run concurrently
- Fit within Linux® PCI Driver Model

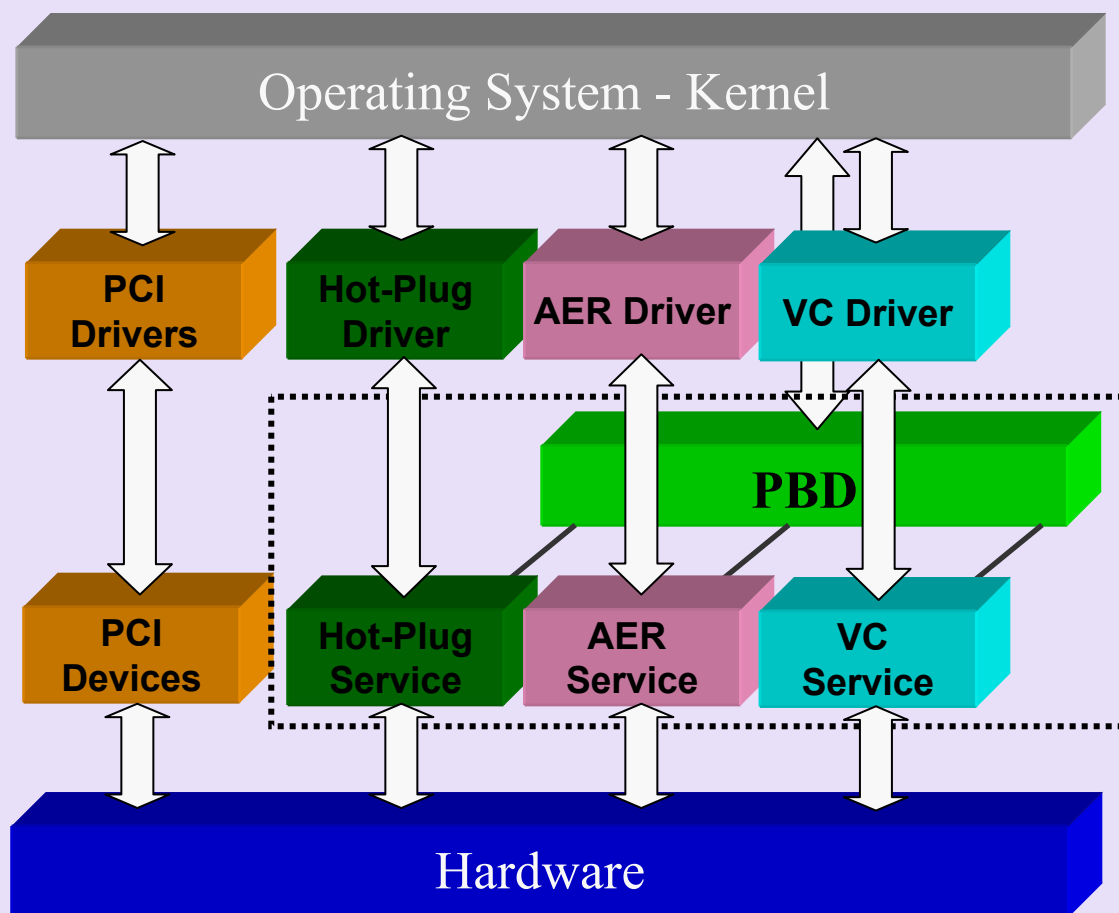


Desired Model



PBD Model

Port Bus Driver Architecture

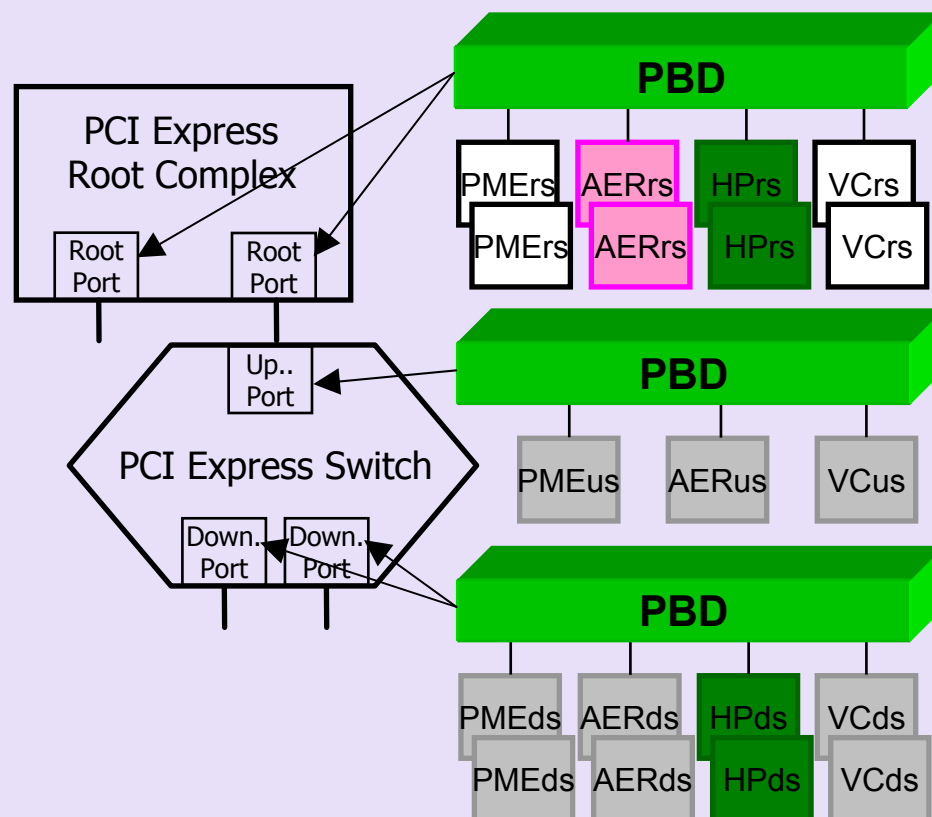


Port Bus Driver Architecture

- Dynamically loads service drivers on demand
- Handles interrupt resource management
- Maintains compatibility with existing Linux® PCI driver model
- Simplifies service driver design
 - ✓ Implement a service driver in a modular fashion
 - Changes in one driver do not impact other service drivers
 - Can easily be extended to support new services
 - ✓ Load or unload a service driver independently

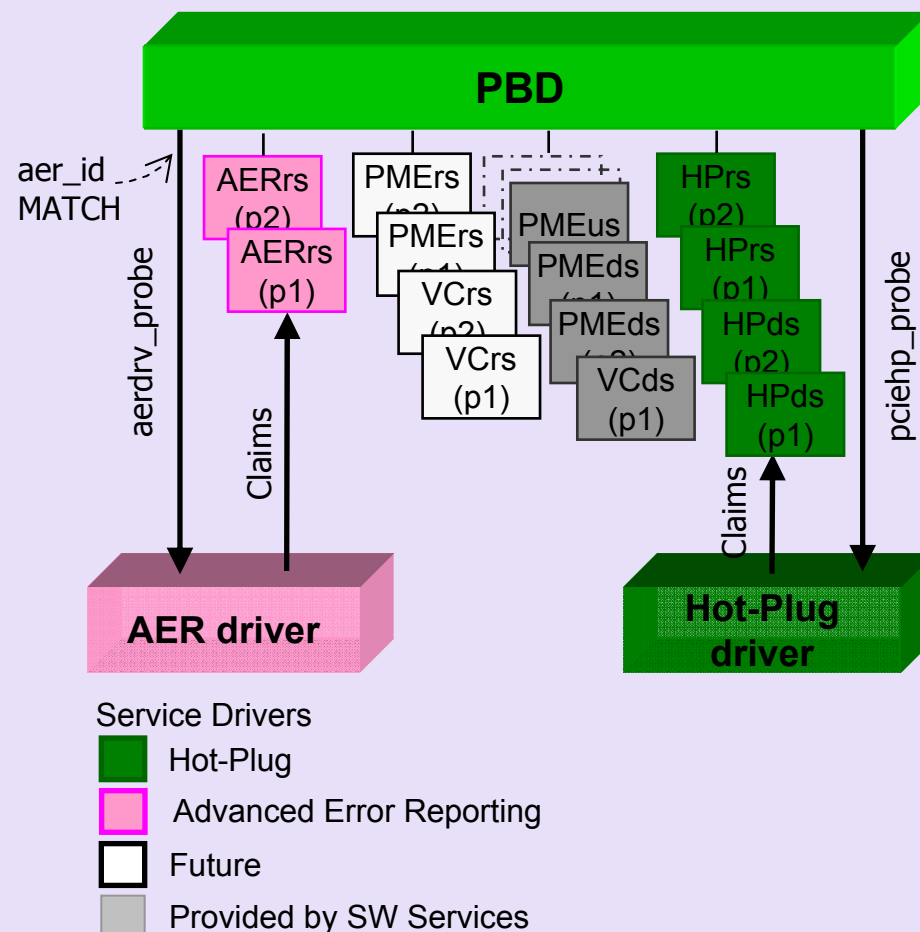
Port Bus Driver Architecture

- Claims PCI Express Ports during normal PCI enumeration
- Creates a **service device**
 - ✓ Assigned with resources and service ID (vendor ID, device ID, port type and service type)
 - ✓ Managed by port bus
 - ✓ Added into the system device hierarchy



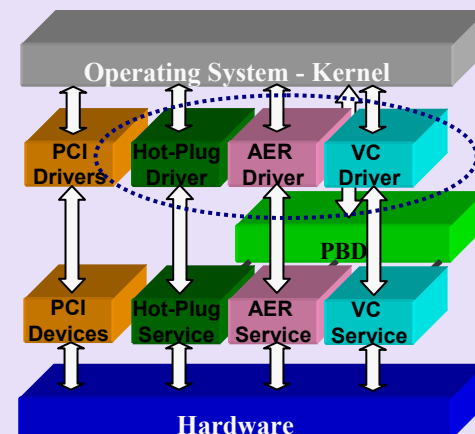
Port Bus Driver Architecture

- Loads registered service drivers dynamically based on the service ID
- Informs the PBD of an available service
 - ✓ `pcie_port_service_register` – adds the service
 - ✓ `pcie_port_service_unregister` – removes the service



Service Driver

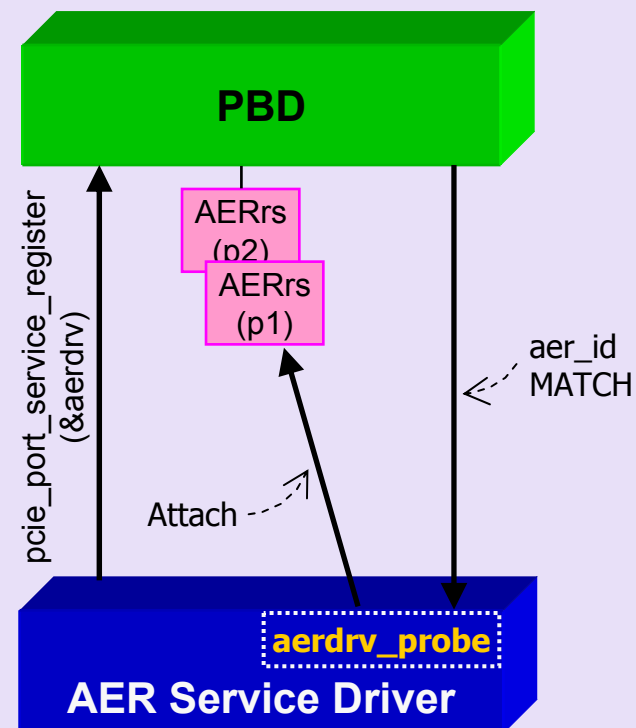
- Provides support for a specific PCI Express service
 - ✓ Some service drivers may support multiple port types (Hot-Plug)
 - ✓ Service drivers can be customized based on port type for the same service
- Can be designed and implemented independently
- Service Driver Attributes
 - ✓ Service id in terms of vendor/device/port type/service type
 - ✓ Service callbacks necessary in port service driver structure
 - .probe
 - .remove
 - .suspend
 - .resume



Service Driver Initialization

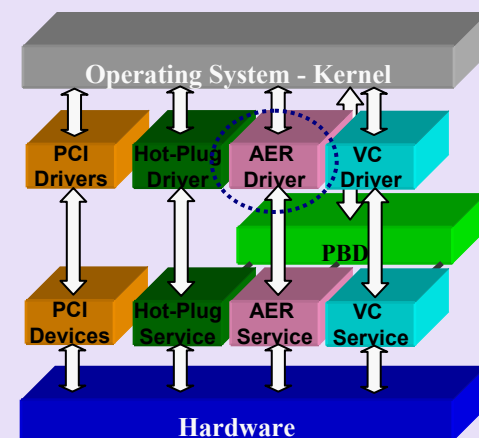
- Prior to call the PBD
 - ✓ First specify Service ID


```
struct pcie_port_service_id aer_id[] = { {
    .vendor = PCI_ANY_ID,
    .device = PCI_ANY_ID,
    .port_type = PCIE_RC_PORT,
    .service_type = PCIE_PORT_SERVICE_AER,
  }, {}};
```
 - ✓ Second, initialize port service driver *aerdrv* data structure
- Then register service driver with the PBD by calling `pcie_port_service_register (&aerdrv)`



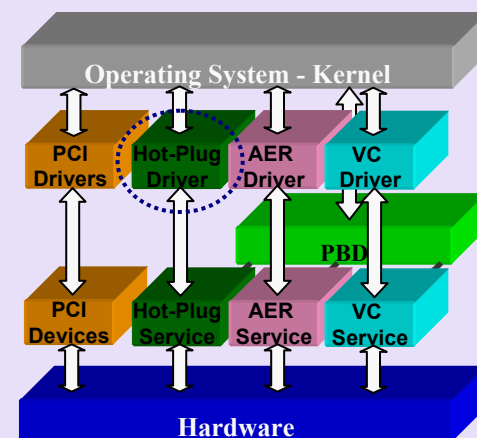
AER Service Driver

- Handles AER in-band error messages sent by PCI Express components to the Root Port
- Functions:
 - ✓ Gather comprehensive error information
 - ✓ Perform error recovery actions
 - Alert the downstream drivers within the hierarchy
 - If error is fatal
 - Reset link
 - Notify drivers of link reset completion
 - Restart downstream drivers
 - ✓ Report error
- AER driver is under review on LKML



Hot-Plug Service Driver

- Supports native endpoint hot-plug operations
- Native Hot-Plug features are:
 - ✓ Root ports and downstream ports of switches are hot-pluggable ports
 - ✓ Hot plug registers are part of the PCI Express Capability register set
 - ✓ Based on SHPC usage model
- Hot-Plug driver has been ported to PBD framework (2.6.11)



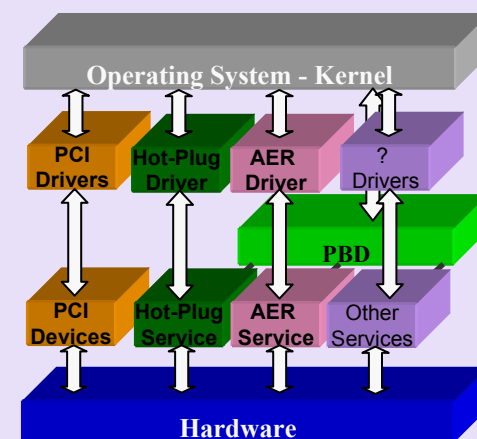
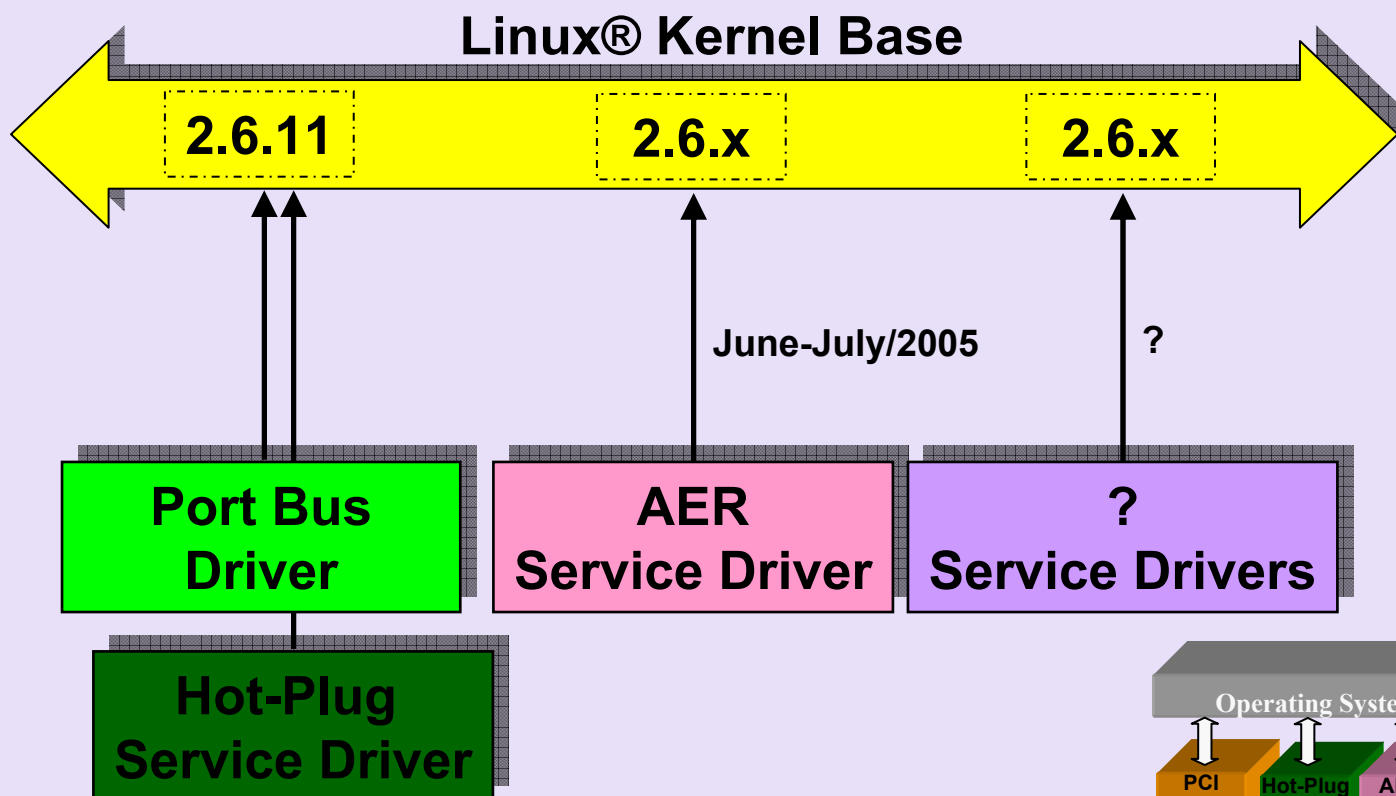
Device Driver Impact

- Endpoint Device Drivers
 - ✓ Be aware as AER Root service driver will get accepted into kernel base
 - ✓ Be aware as new service drivers (VC or PME) become available
- Port Service Drivers
 - ✓ New Root service drivers (VC,PME) must conform to the PBD framework
 - ✓ Switch service drivers must conform to the PBD framework

Next Steps

- PCI Express Root Ports
 - ✓ Develop service drivers for VC and PME
- PCI Express Switch Ports
 - ✓ Develop service drivers for AER, VC, and PME
- For More information
 - ✓ Paper to be presented at 2005 Ottawa Linux® Symposium (*July*) at <http://www.linuxsymposium.org/2005/>
 - ✓ Kernel source tree at <http://www.kernel.org/>
 - ✓ PCIEBUS-HOWTO.txt available from 2.6.11/Documentation

PBD/Service Driver Status



Thank you for attending the
PCI-SIG Developers Conference 2005.

For more information please go to
www.pcisig.com