EEH Overview

Gavin Shan

Linux Technology Center, IBM, China
shangw@cn.ibm.com

May, 8, 2012

Agenda

- What's EEH?
- High-level Overview
- How EEH Core Works?
- Further Development Work

What's EEH?

- EEH is the abbreviation of Extended Error Handling.
- Isolate PCI errors within IO domains without affecting the rest of the system
  - Enhanced system reliability and availability
- A feature available on Power platforms only.

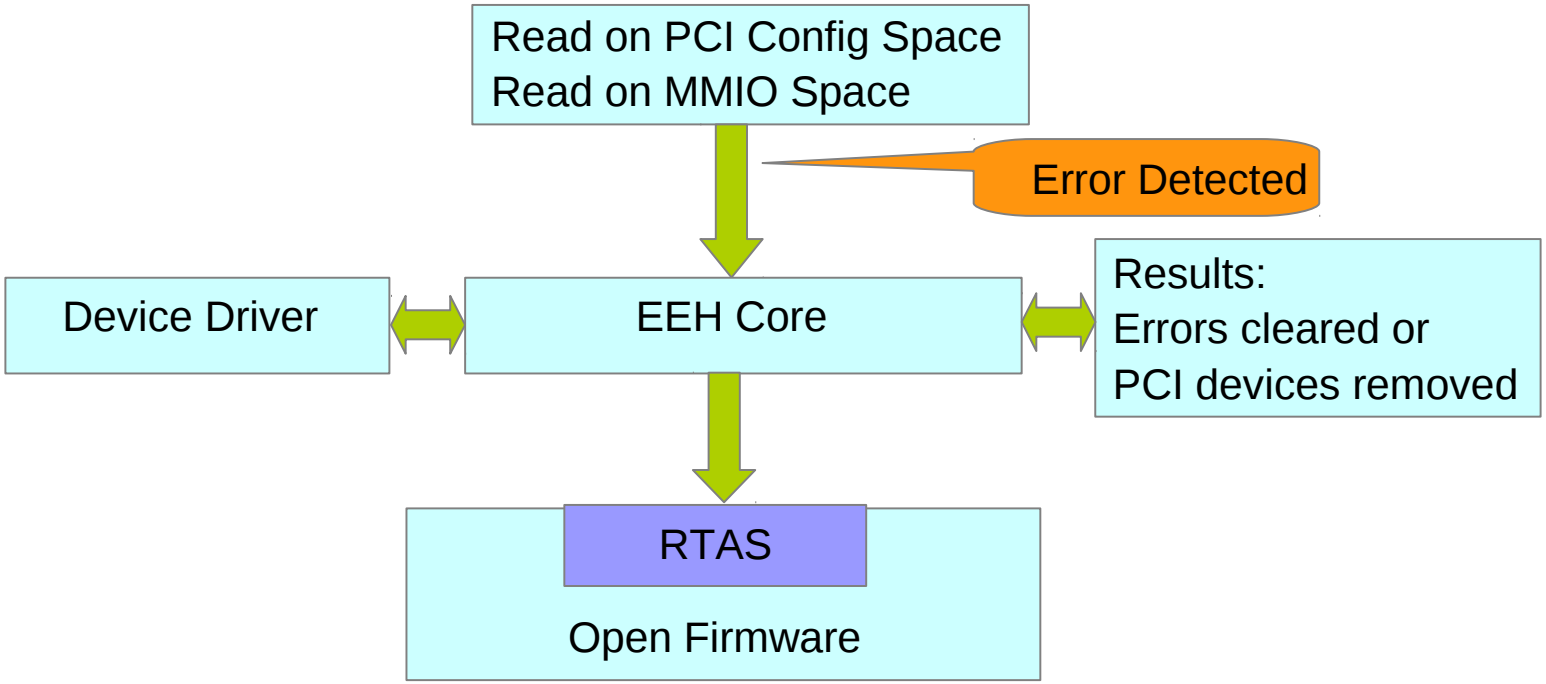
What is EEH?

- RTAS(Run-Time Abstraction Services) in the firmware provides PCI error related services to the OS.

- The EEH core in the OS handles the error by either
  - Taking appropriate corrective actions
  - Or resetting the IO domain responsible for the error.

- RTAS services are documented in PAPR (Power Architecture Requirements, www.power.org).

High-level Overview



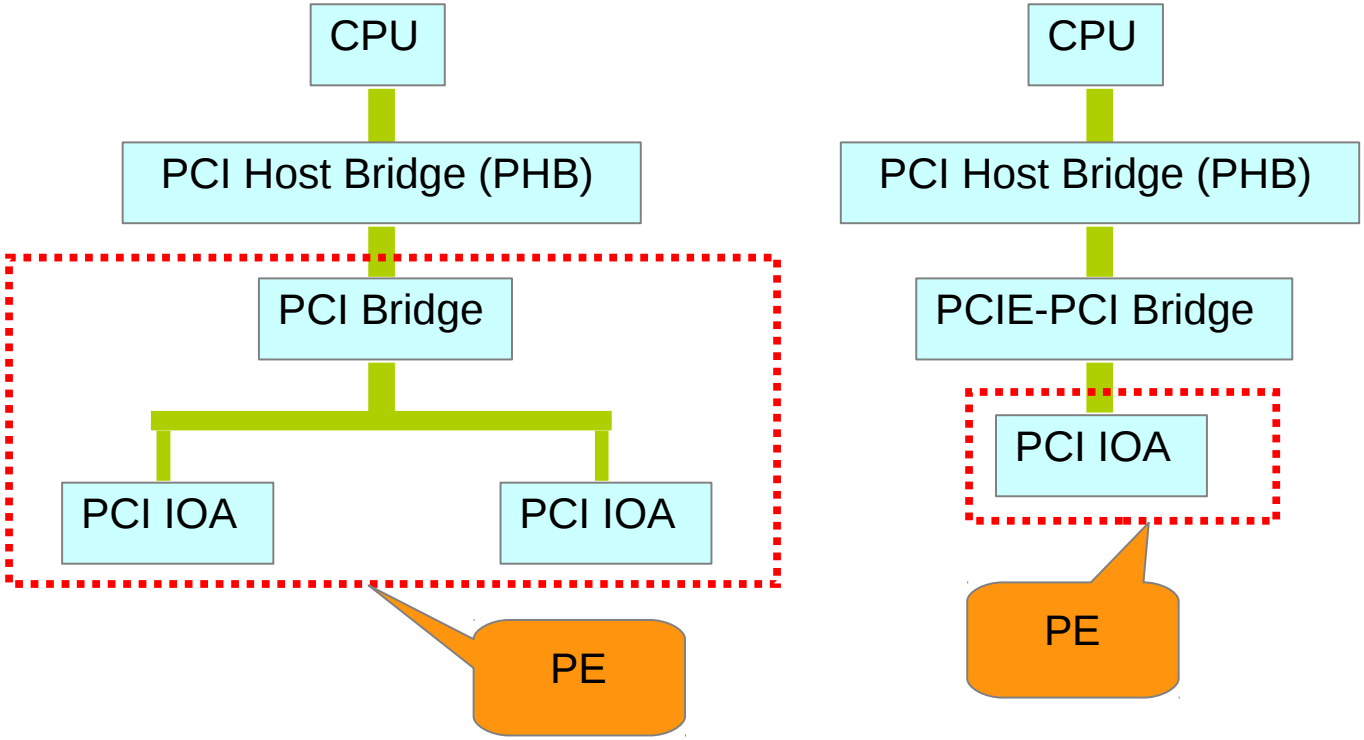
Partitionable Endpoint (PE)

- PE is an I/O error and recovery domain made up of
  - A single or multi-function IO Adapter or
  - A function of a multi-function IO Adapter or
  - Multiple IOAs, possibly includes upstream switches and bridges

- Partitionable Endpoint (PE) is defined in PAPR (Power Architecture Platform Requirements).

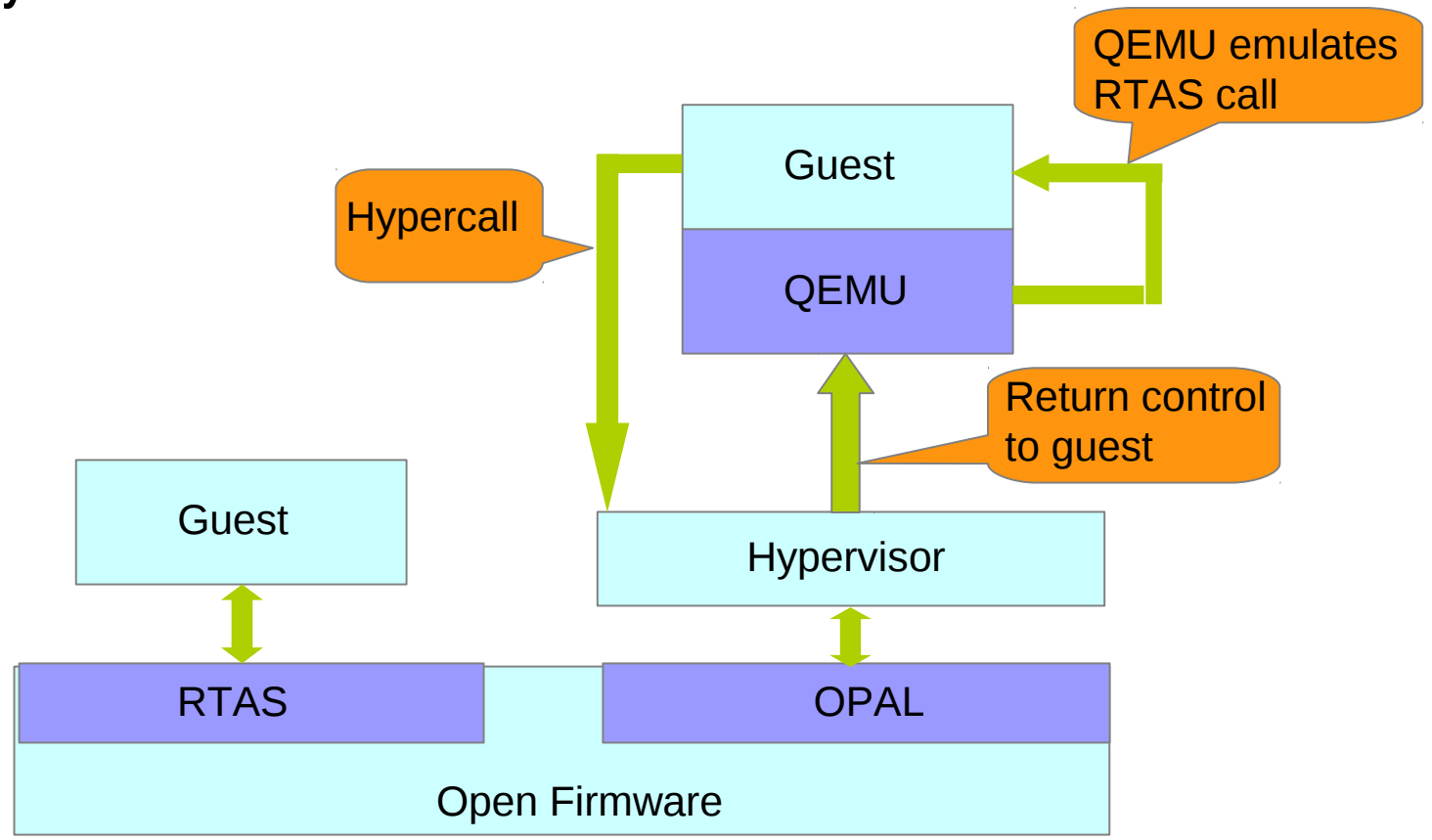
- RTAS compliant firmware supports EEH related operations at PE granularity.

Partitionable Endpoint (PE)



EEH RTAS Calls

- **The RTAS (Run-Time Abstraction Services) calls intend to insulate OS from having to know about and manipulate the platform hardware by registers directly.**



EEH RTAS Calls

- OF (Open Firmware) is using device tree to pass information to Linux kernel. Usually, we call those device tree nodes as OF nodes. OF node “/rtas” includes lots of properties to designate EEH calls.

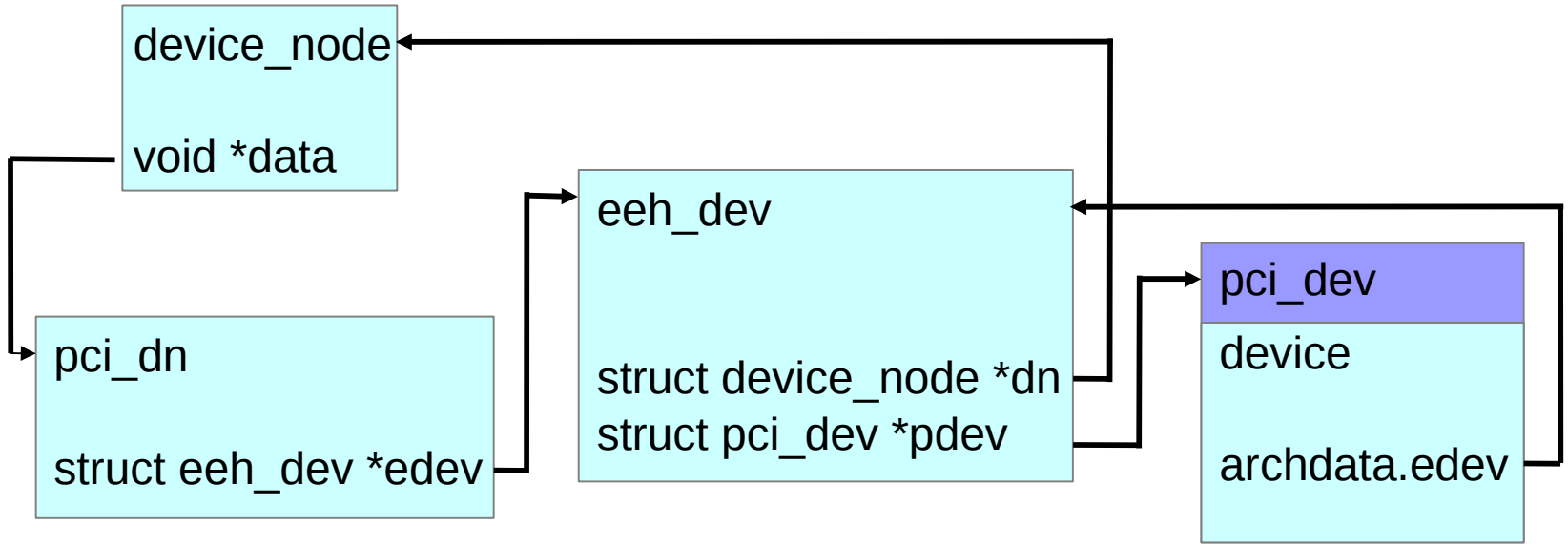
- “ibm,set-eeh-option”
  Enable/Disable EEH for PE, or enable/disable MMIO/DMA for PE.
- “ibm,set-slot-reset”
  Reset PE
- “ibm,read-slot-reset-state2”, “ibm,read-slot-reset-state”
  Query the state of PE
- “ibm,slot-error-detail”
  Retrieve error log
- “ibm,get-config-addr-info2”, “ibm,get-config-addr”
  Retrieve PE address
- “ibm,configure-pe”, “ibm,configure-bridge”
  Configure the PCI bridges in the PE

PCI Access with EEH Errors

- EEH traps PCI config and MMIO read errors and continue with normal operation without frozen state of the corresponding PE.
- Return of 0xFF's from PCI config and MMIO read is the criteria of trapping into EEH.
- A PE in frozen state drops all PCI config and MMIO writes quietly.

EEH Device

- EEH device , “struct eeh_dev”, traces the EEH related information for each PCI device.
- OF node, “struct device_node”, represents device tree nodes.
- For those PCI based OF nodes, “struct pci_dn” introduced to store the PCI related information (e.g. bus number, slot, etc.)



PCI Address Cache

- PCI address cache stores PCI devices using a RB tree.
- Helps finding the EEH device associated with a MMIO address.
- Each node in the RB tree stores MMIO physical window and its associated PCI device.

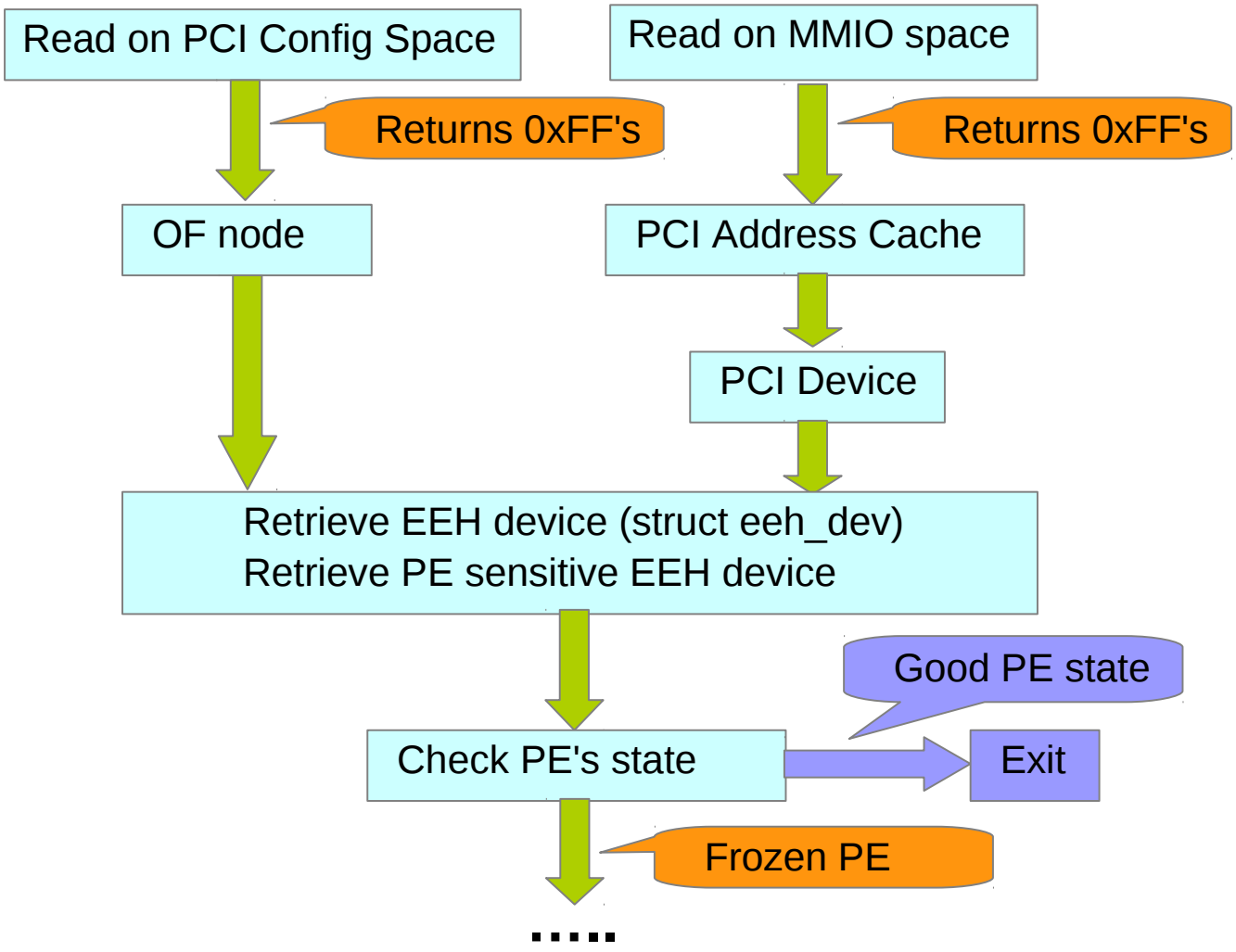
How EEH Talks to DD?

- EEH core uses EEH handlers registered by device drivers.

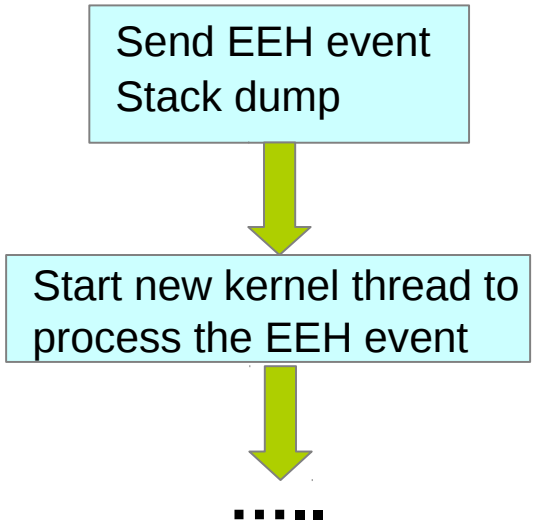
- EEH core and device drivers communicate and handle errors through the EEH handlers.

```
struct pci_error_handlers {
        pci_ers_result_t (*error_detected)(struct pci_dev *dev,
                                           enum pci_channel_state error);
        pci_ers_result_t (*mmio_enabled)(struct pci_dev *dev);
        pci_ers_result_t (*link_reset)(struct pci_dev *dev);
        pci_ers_result_t (*slot_reset)(struct pci_dev *dev);
        void (*resume)(struct pci_dev *dev);
};
```

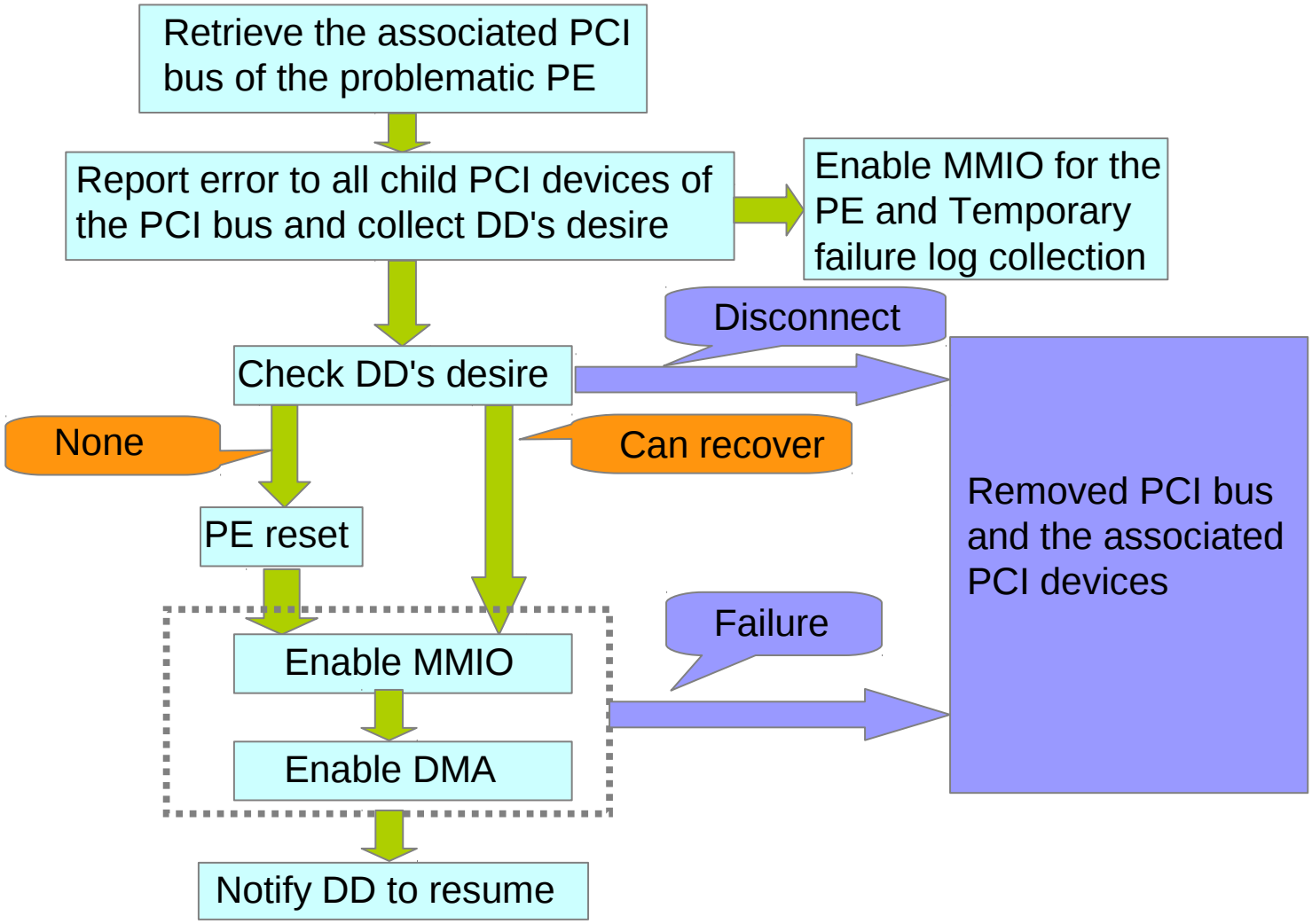
How EEH Core Works?



How EEH Core Works?



How EEH Core Works?



Further Development work

- PE explicit support. It will introduce data struct “eeh_pe” to represent PE so that EEH core becomes more data centralized. The EEH core needs somewhat rework accordingly.

- EEH support for P7IOC based powernv platform.

- EEH emulation for KVM based guests.

Legal Statement

This work represents the view of the author and does not necessarily represent the view of IBM.

IBM, IBM (logo), AIX, POWER, POWER6, POWER7 and PowerVM are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Thanks & Questions