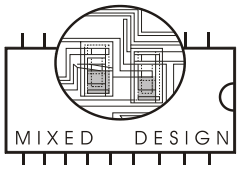
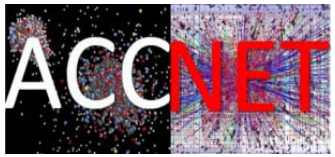




PCIExpress Hot-Plug Mechanism in Linux-based ATCA Control Systems

The research leading to these results has received funding from the European Commission under the FP7 Research Infrastructures project EuCARD, grant agreement no. 227579.

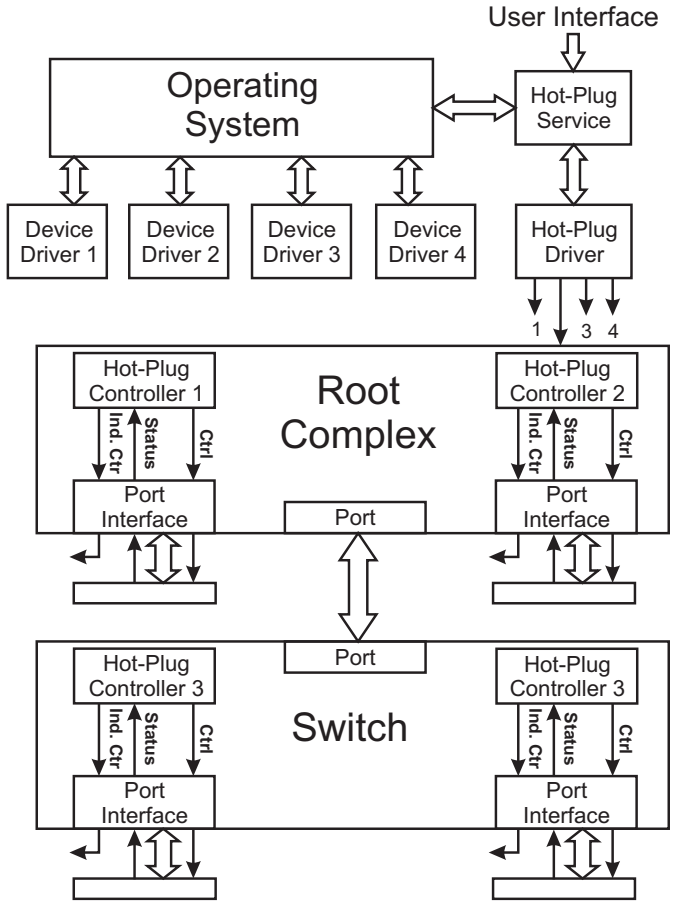
Outline

Hot-Plug Mechanism

Hot-Plug/Hot-Swap solution provide methods to replace modules without turning system off, keeping operating system services running correctly after component removal and restarting or shutting down software associated to removed device.

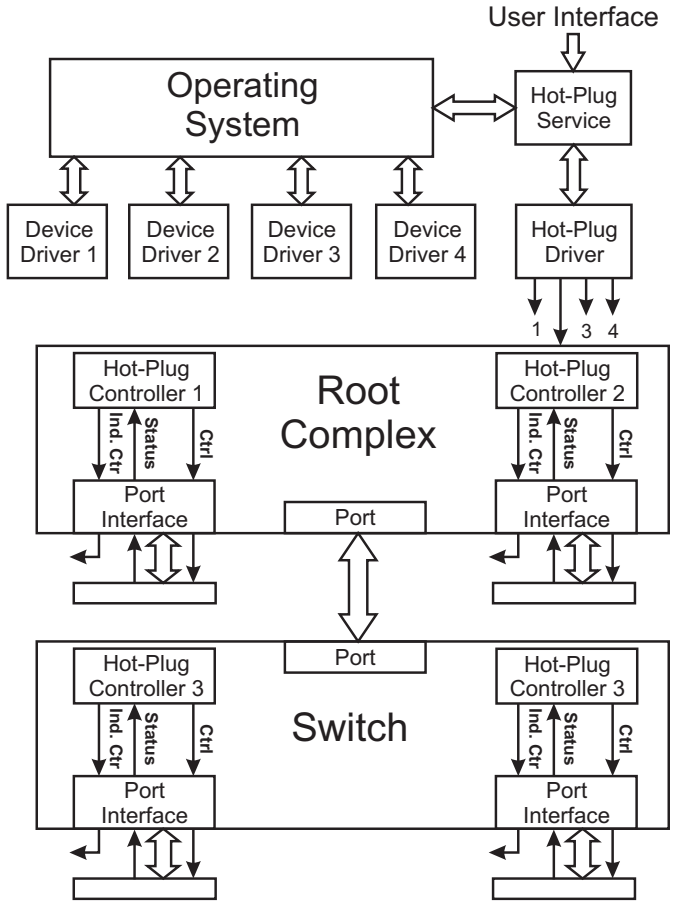
Hardware Support for Hot-Plug Mechanism

- Hot-Plug Controller



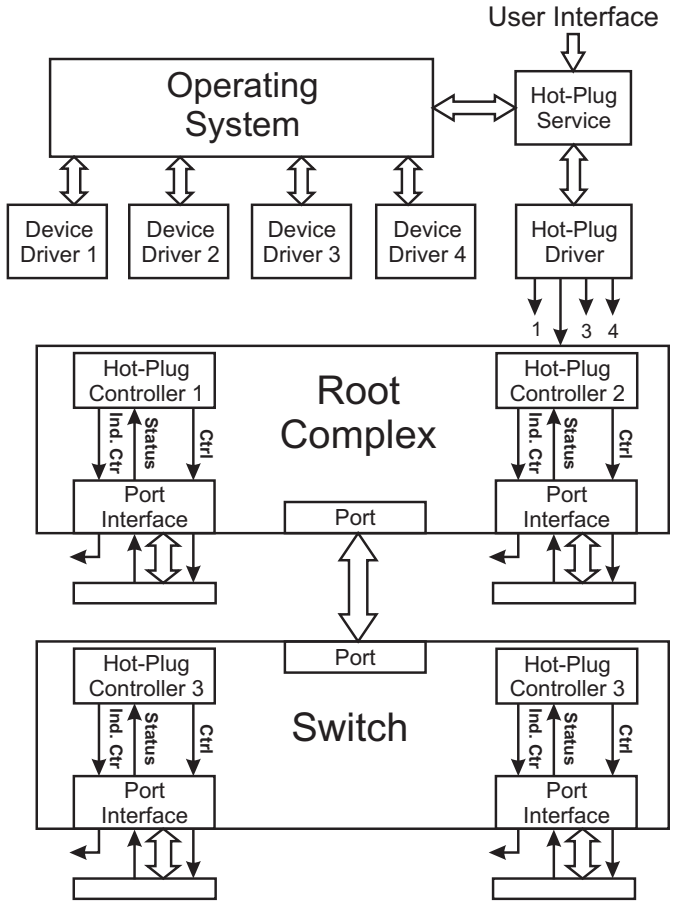
Hardware Support for Hot-Plug Mechanism

- Hot-Plug Controller
- Card Slot Power Switching and Card Reset Logic



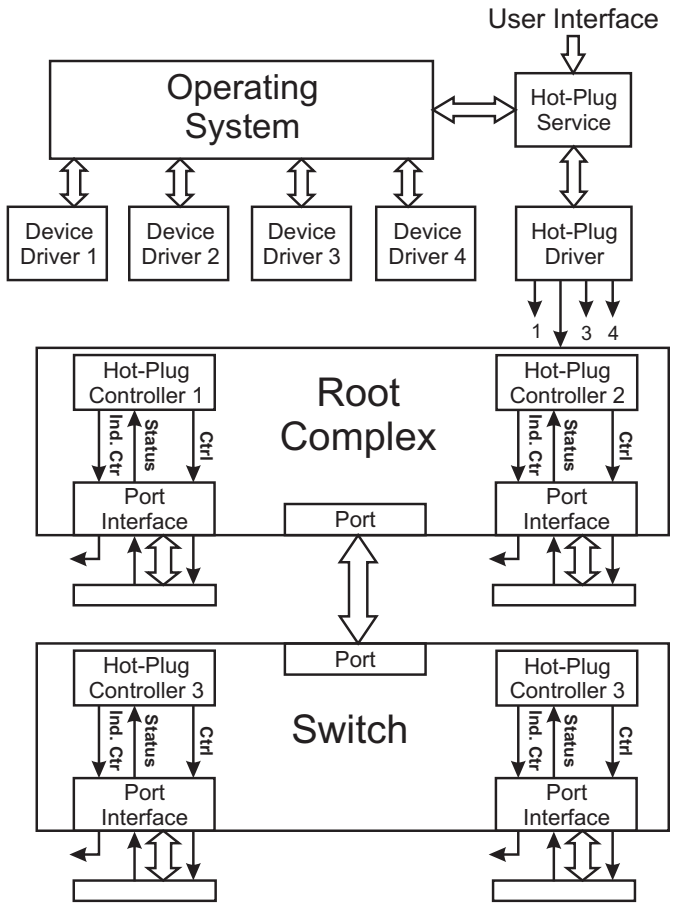
Hardware Support for Hot-Plug Mechanism

- Hot-Plug Controller
- Card Slot Power Switching and Card Reset Logic
- Power Indicator and Attention Indicator



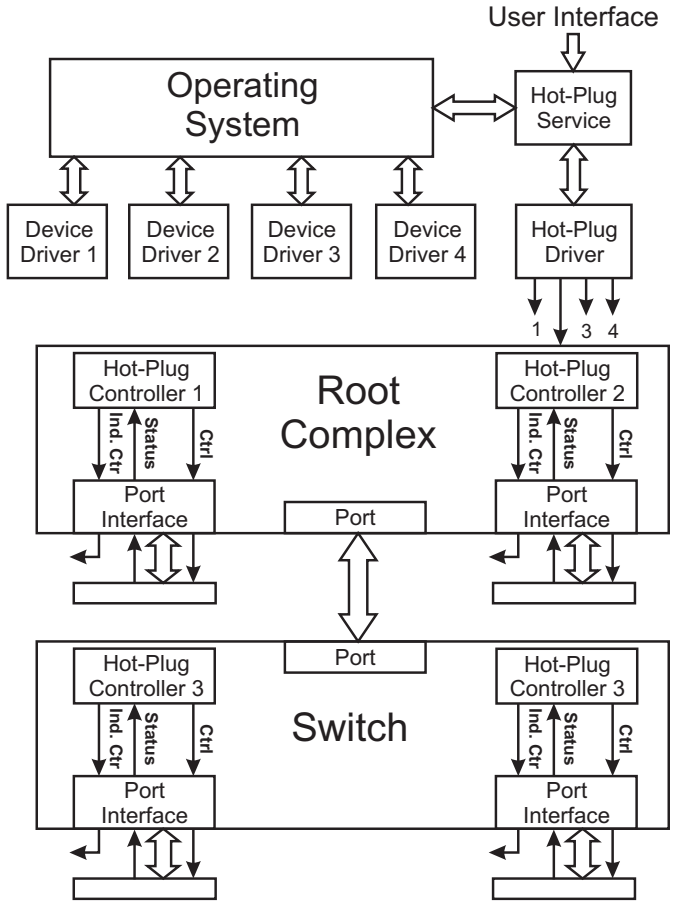
Hardware Support for Hot-Plug Mechanism

- Hot-Plug Controller
- Card Slot Power Switching and Card Reset Logic
- Power Indicator and Attention Indicator
- Attention Button



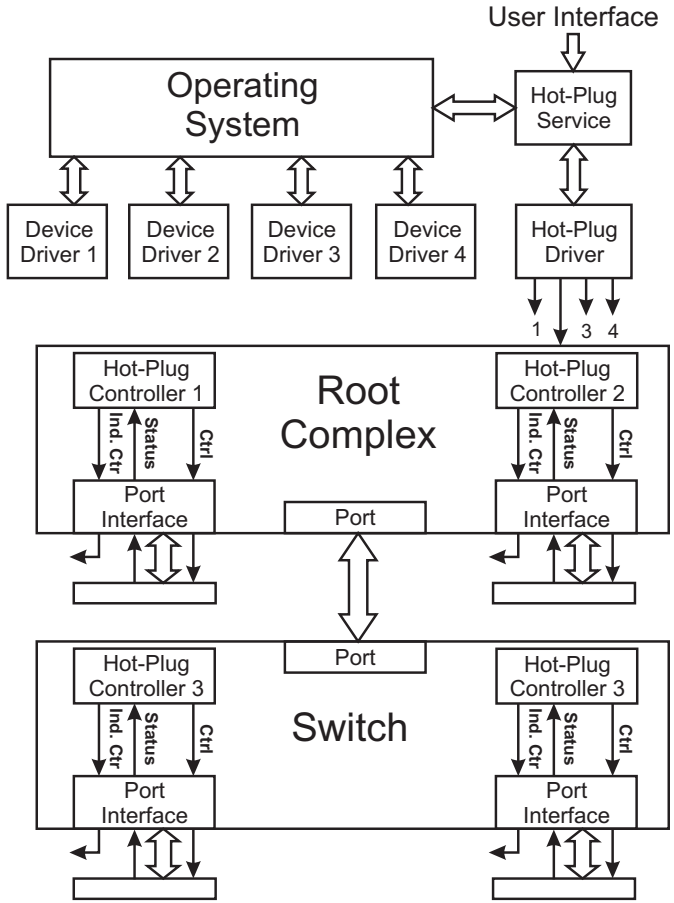
Hardware Support for Hot-Plug Mechanism

- Hot-Plug Controller
- Card Slot Power Switching and Card Reset Logic
- Power Indicator and Attention Indicator
- Attention Button
- Card Preset Detection Pins



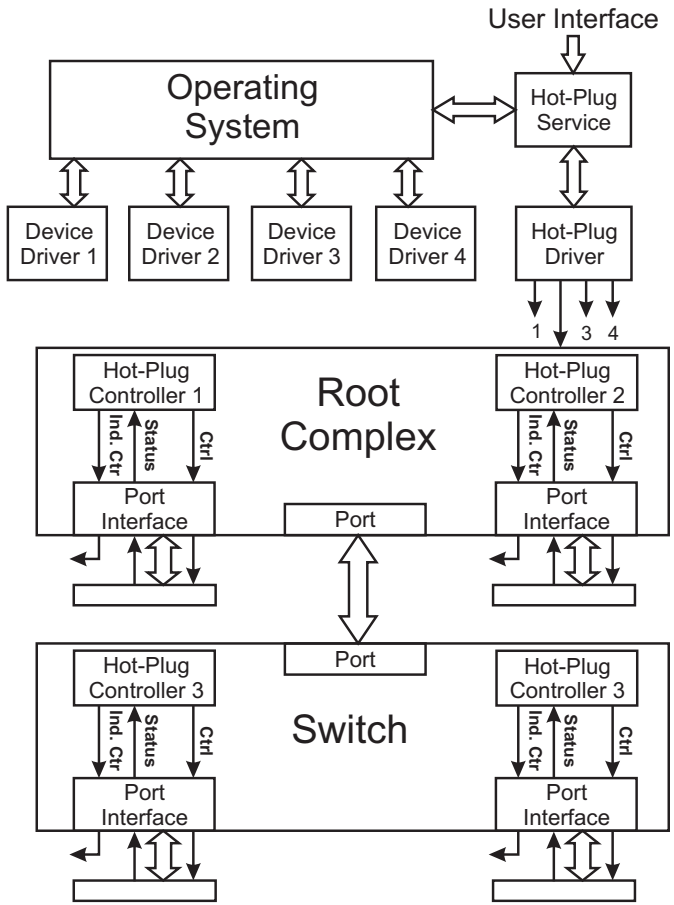
Software Support for Hot-Plug Mechanism

- Hot-Plug Service



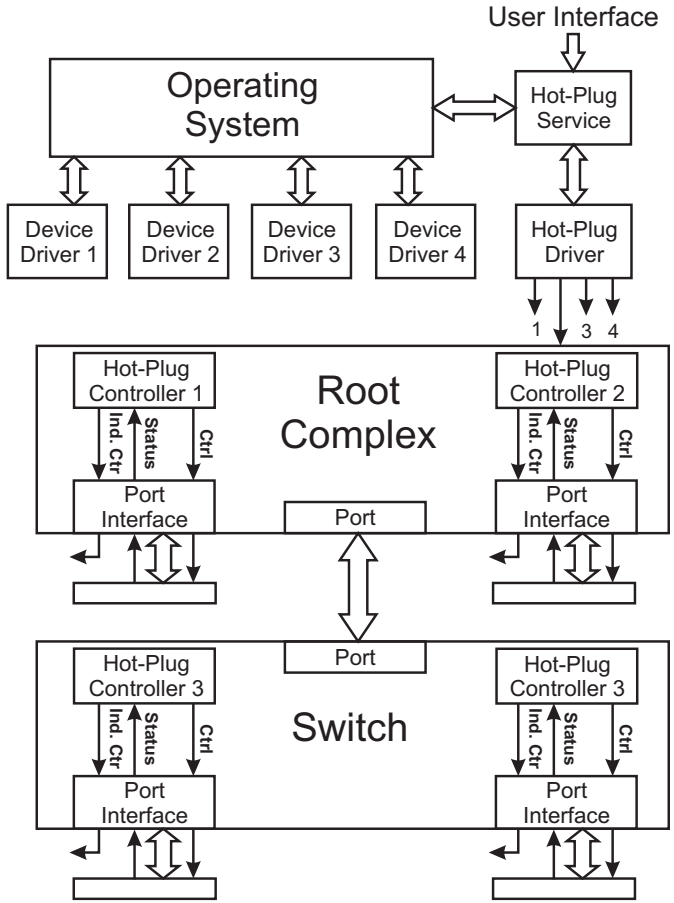
Software Support for Hot-Plug Mechanism

- Hot-Plug Service
- Hot-Plug Driver



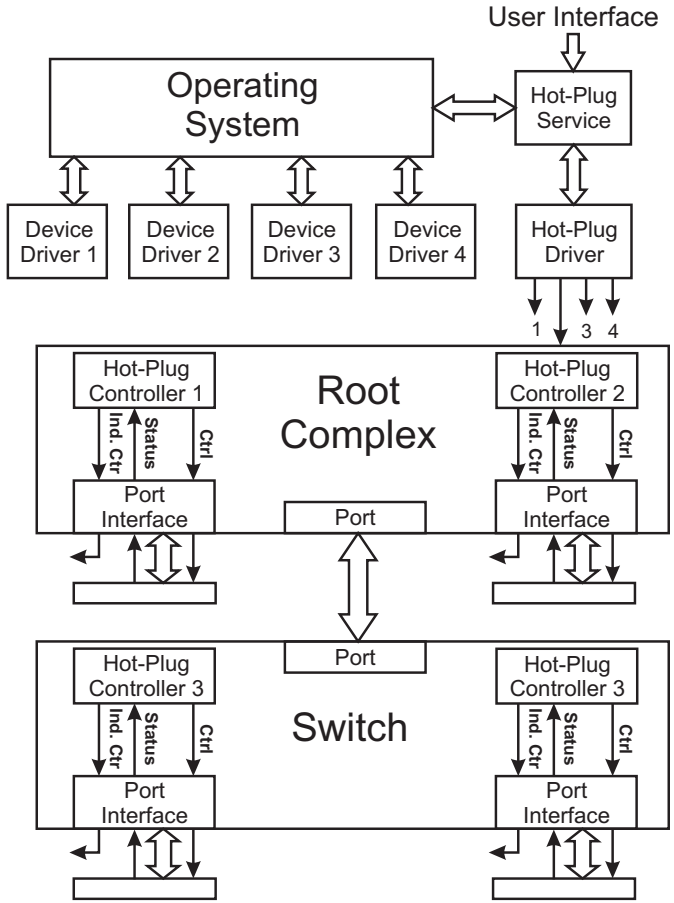
Software Support for Hot-Plug Mechanism

- Hot-Plug Service
- Hot-Plug Driver
- Device Drivers



Software Support for Hot-Plug Mechanism

- Hot-Plug Service
- Hot-Plug Driver
- Device Drivers
- User Interface

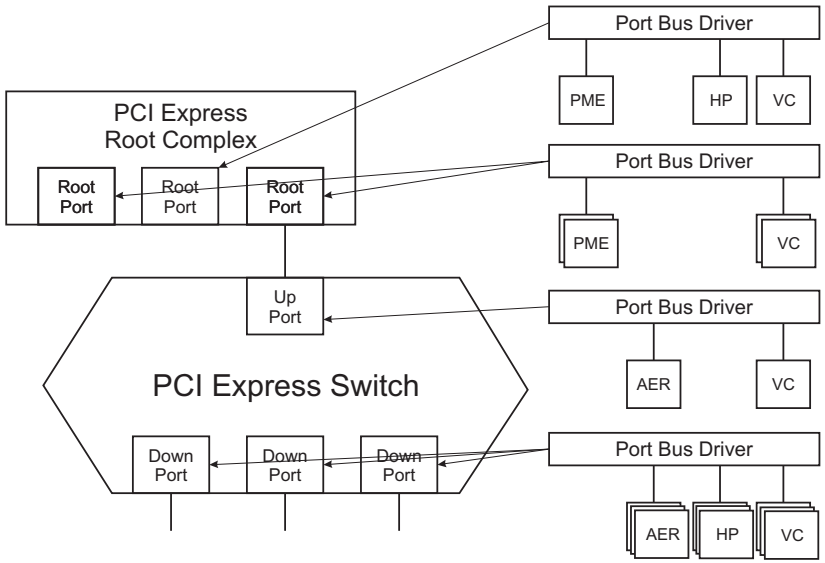


PCI Driver Model vs. PCI Express Driver Model

- Standard PCI Driver Model allows to load one driver for one device
- Standard PCI Express Ports support up to four different functions
- PCI Express Port Bus Driver was designed to support PCI Express functionalities in PCI Driver Model

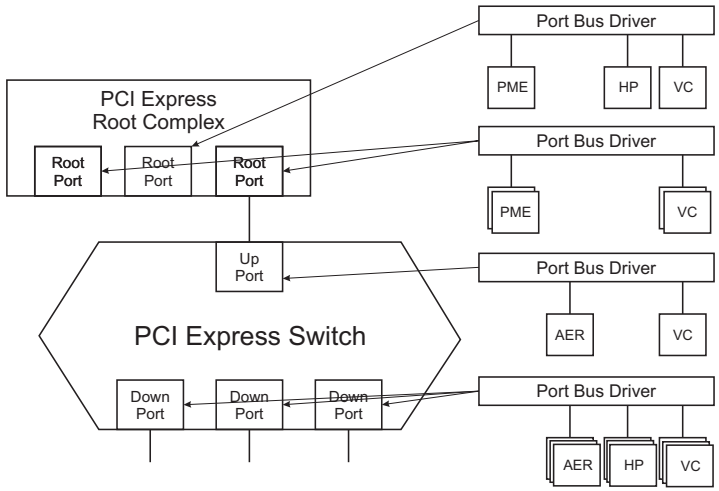
PCI Express Bus Driver

- Hot-Plug
- Power Management
- Virtual Channel
- Advanced Error Reporting



PCI Express Bus Driver

sysfs file name	*service description*
0000:00:01.0:pcie01	PME on first Root Port
0000:00:01.0:pcie08	VC on first Root Port
0000:00:1c.0:pcie01	PME on second Root Port
0000:00:1c.0:pcie04	HP on second Root Port
0000:00:1c.0:pcie08	VC on second Root Port
0000:04:00.0:pcie12	AER on switch upstream port
0000:04:00.0:pcie18	VC on switch upstream port
0000:05:08.0:pcie22	AER on switch first downstream port
0000:05:08.0:pcie24	HP on switch first downstream port
0000:05:08.0:pcie28	VC on switch first downstream port
0000:05:09.0:pcie22	AER on switch second downstream port
0000:05:09.0:pcie24	HP on switch second downstream port
0000:05:09.0:pcie28	VC on switch second downstream port



Hot-Plug for FPGA-based Devices

- Reconfiguration of FPGA-based PCI Express Endpoint does not emit standard Hot-Plug event,
- After reprogramming, PCI Express Endpoint is in uninitialized state,
- Reenumeration of PCI Express Bus must be force manually by user,
- Fake Hot-Plug Driver from newest version of Linux kernel is able to correctly handle reinitialization of FPGA-based device

Fake Hot-Plug Driver

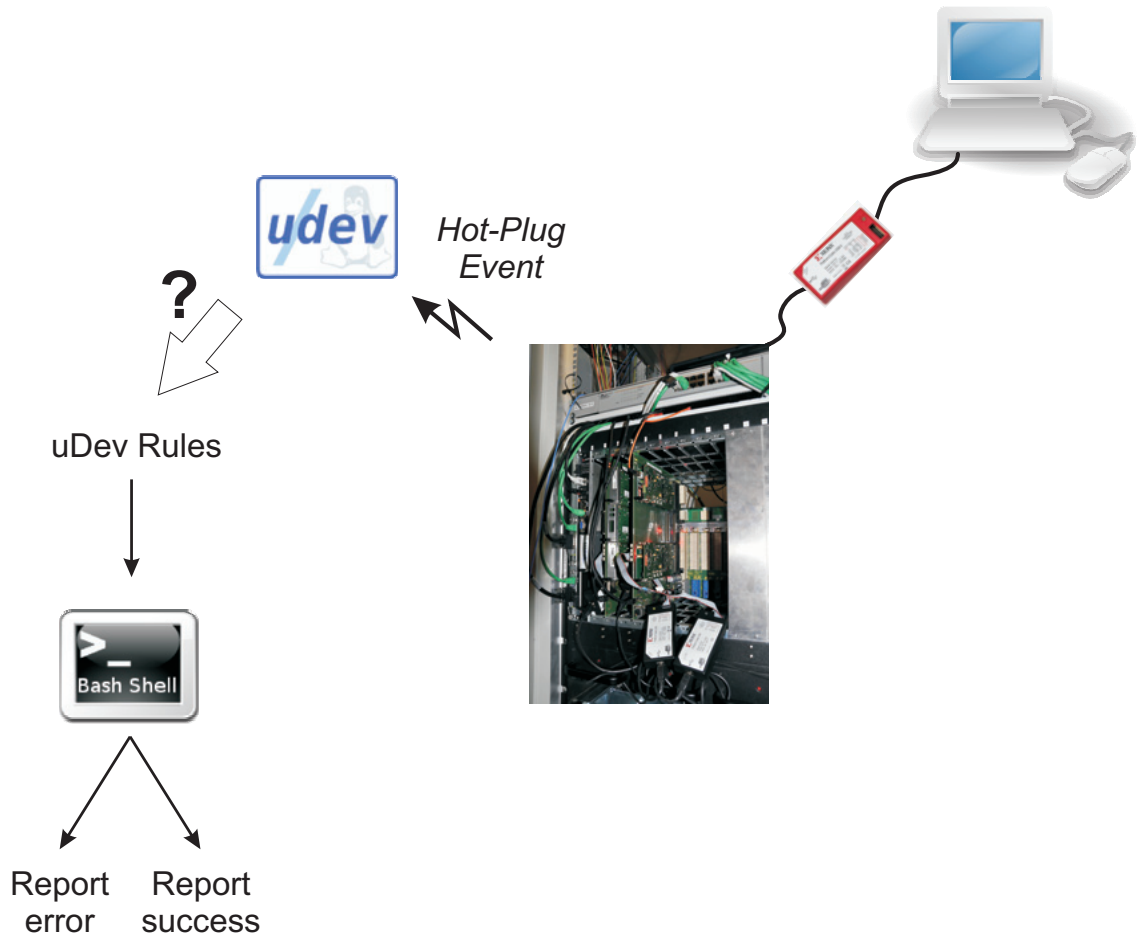
Fake Hot-Plug Driver is able to logically remove device from the system:

- `echo 0 > /sys/bus/pci/slots/0000:03:00.0/power`

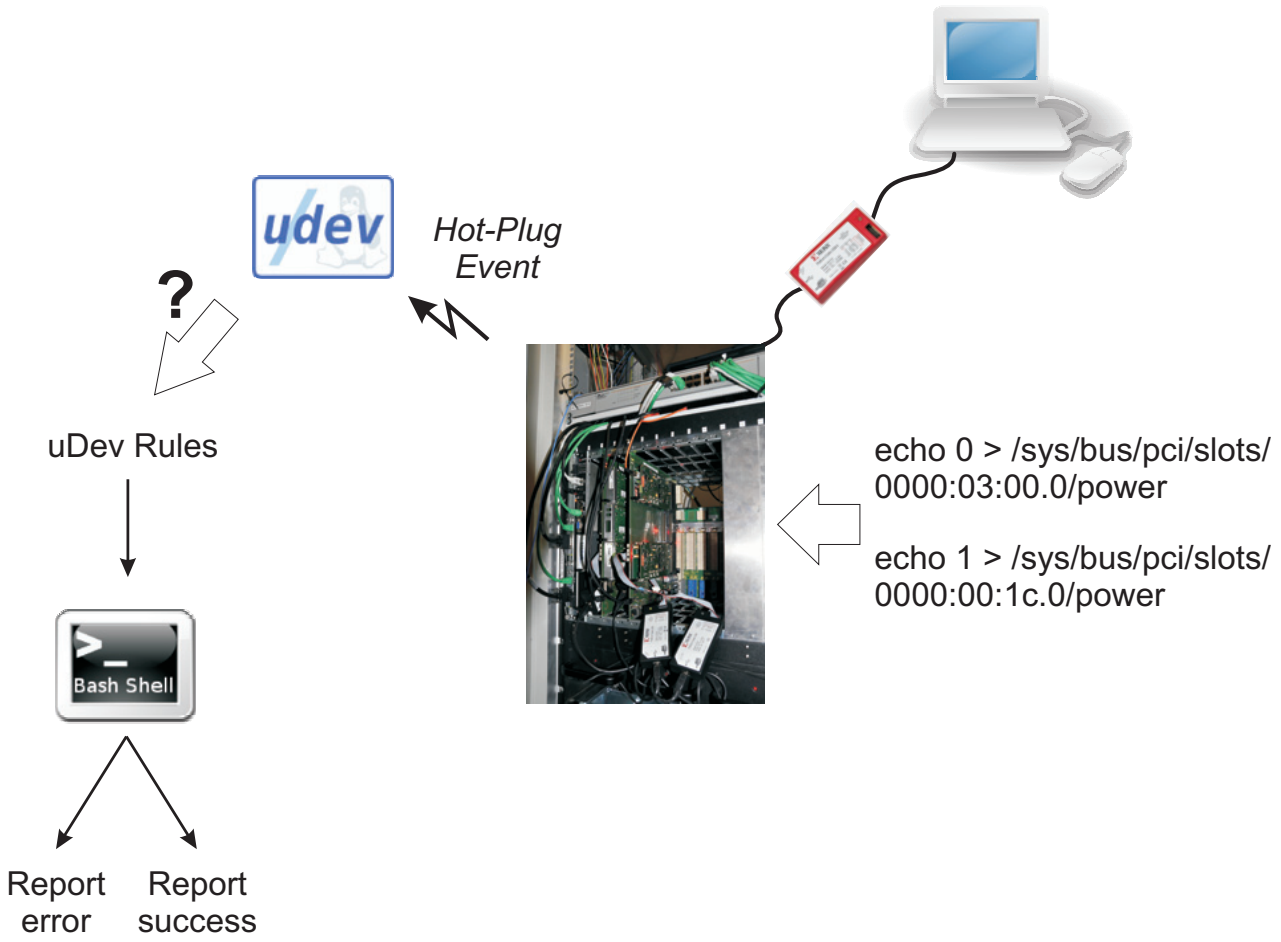
Fake Hot-Plug Driver is able to force enumeration of the device:

- `echo 1 > /sys/bus/pci/slots/0000:00:1c.0/power`

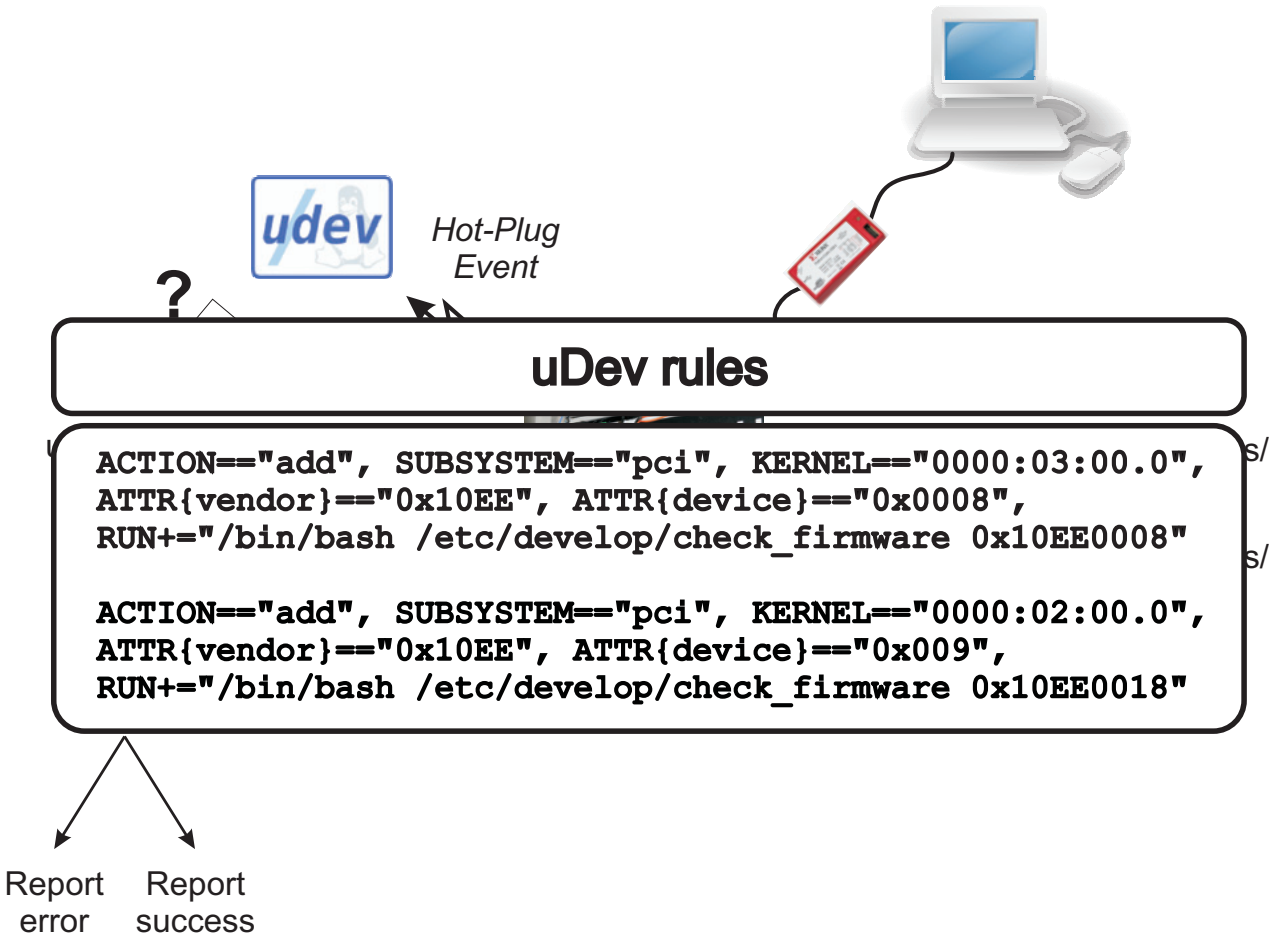
Example


udev
Hot-Plug
Event
?
uDev Rules
Bash Shell
Report
error
Report
success

Example


udev
Hot-Plug
Event
?
uDev Rules
Bash Shell
Report
error
Report
success
echo 0 > /sys/bus/pci/slots/
0000:03:00.0/power
echo 1 > /sys/bus/pci/slots/
0000:00:1c.0/power

Example - uDev Rules



Example - device parameters

Device parameters

```
udevadm info --path=/sys/bus/pci/devices/0000\:03\:00.0/
--attribute-walk


  looking at device '/devices/pci0000:00/0000:00:1c.0/
  0000:03:00.0':
    KERNEL=="0000:03:00.0"
    SUBSYSTEM=="pci"
    DRIVER==""
    ATTR{vendor}=="0x10ee"
    ATTR{device}=="0x0008"
    ATTR{subsystem_vendor}=="0x10ee"
    ATTR{subsystem_device}=="0x0007"
    ATTR{class}=="0x050000"
    ATTR{irq}=="255"
    ATTR{local_cpus}=="ffffffff"
    ATTR{local_cpulist}=="0-31"
    ATTR{modalias}=="pci:v000010EEd00000008sv000010
                     EEsd00000007bc05sc00i00"
    ATTR{enable}=="0"
    ATTR{broken_parity_status}=="0"
    ATTR{msi_bus}==""
```

Example - bash script

bash script

```
#!/bin/bash
cd /etc/develop/

./load_simple_driver

VER_R=$(./pcie-rw /dev/pcie_bar_0 r 0i 1i h)
if [ ${1} != ${VER_R} ]; then
      logger "Wrong firmware version (${VER_R})"
      echo 0 > /sys/bus/pci/slots/0000:00:1c.0/power
else
      logger "Loaded firmware version ${1}"
fi

./unload_simple_driver
cd -
```

Report error

success

Conclusions

- New version of Fake Hot-Plug Driver is able to correctly handle reprogramming of the FPGA-based PCI Express Endpoint device,
- uDev Device Manager ability to execute additional scripts during Hot-Plug Event processing allows to increase safety of the system

Thank You