



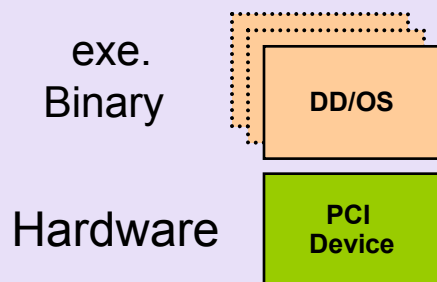
PCI Device Address Space Abstraction Model June 14, 2004

**Dwight D. Riley
Chairman, PCI-X 2.0 Workgroup
Industry Standard Servers-Office of the CTO, HP**

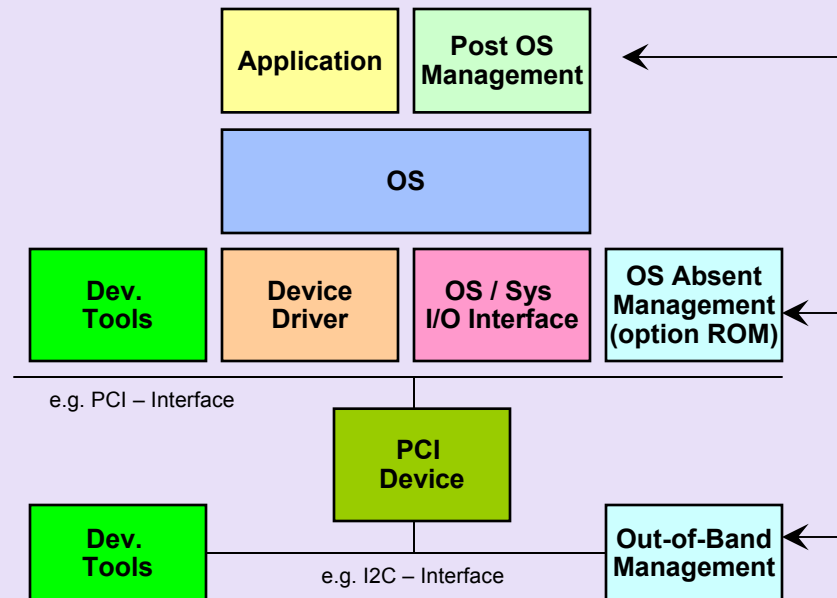
The Problem: “Hidden” PCI Config Space

- PCI configuration space is undefined
 - ✓ Configuration space is application specific
 - ✓ Non standard from vendor-to-vendor
 - ✓ Non standard within any of the PCI-SIG defined classes (class codes)
 - ✓ Tools must be custom written to a particular device so that registers can be interpreted
 - Formatted Output: Formatted data with no logical dependencies
 - Structured Output: Formatted data with logical dependencies
- PCI is ubiquitous

The Problem: IHV Deliverables



A single piece of hardware is supported on multiple OS platforms via custom OS device drivers, management controls and development tools



Limited industry standard control of PCI registers (Mem, I/O, or Config space)

- Through the OS (App or DD)
- Pre-OS Management (BIOS)
- Development tools

Out-of-Band Management (SMB)

- ACPI, BIOS, Option ROM/EFI – register control

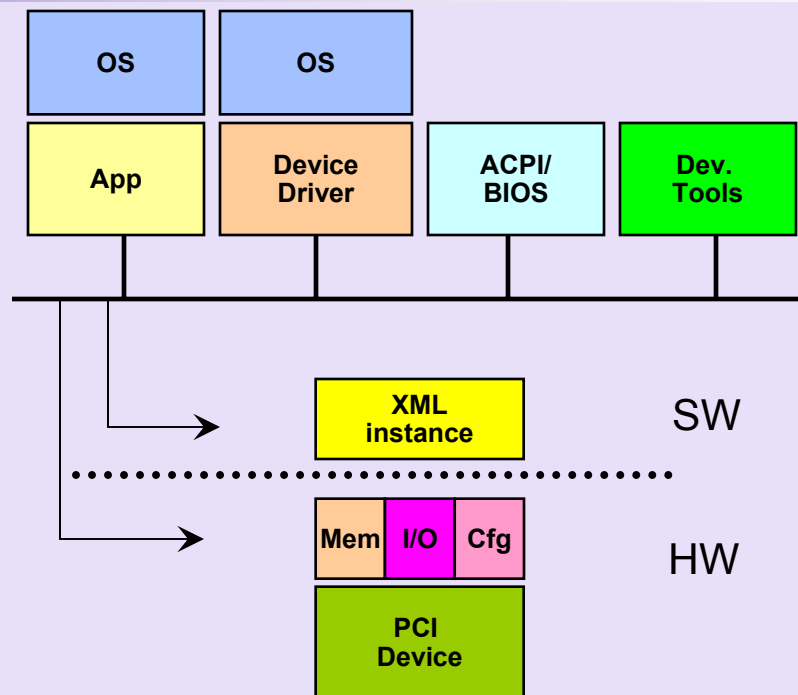
The Solution: Runtime XML Instance File

■ Pros:

- ✓ XML file sharing
- ✓ Flexibility across OS and app
- ✓ Visibility of PCI device configuration space
- ✓ Realtime upgrade of DD

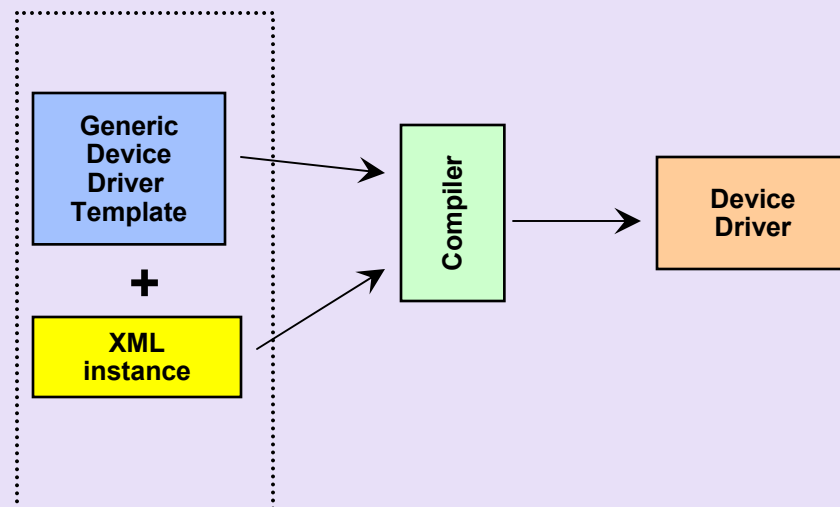
■ Cons:

- ✓ Performance
- ✓ Maintaining XML files including version controls



The Solution: Compiled XML Instance File

- Pros:
 - ✓ Source code reusability
 - ✓ Compatible with existing infrastructure
 - Validation
 - Certification
 - Support
 - ✓ Performance
- Cons:
 - ✓ No XML sharing
 - ✓ Less flexibility





Limited Visibility into Config Space: PCI Configuration Space Header

31		16		15		0		
Device ID				Vendor ID				00h
Status				Command				04h
Class Code						Revision ID		08h
BIST		Header Type		Latency Timer		Cache Line Size		0Ch
Base Address Registers								10h
								14h
								18h
								1Ch
								20h
								24h
Cardbus CIS Pointer								28h
Subsystem ID				Subsystem Vendor ID				2Ch
Expansion ROM Base Address								30h
Reserved						Capabilities Pointer		34h
Reserved								38h
Max_Lat		Min_Gnt		Interrupt Pin		Interrupt Line		3Ch

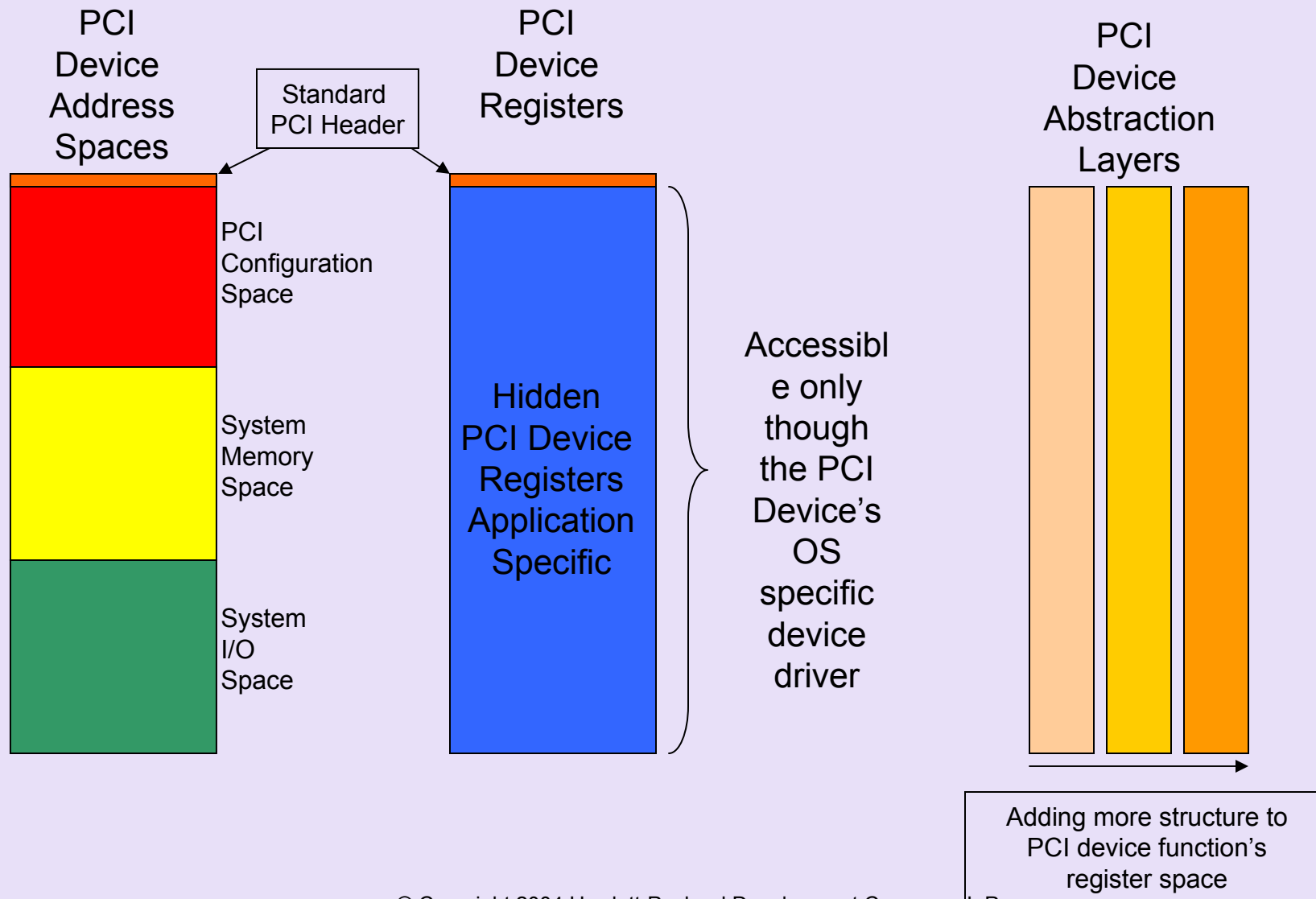
Configuration Space

- Only the first 64-bytes of the 4096 bytes of the device function configuration space is defined by the PCI Specification.
- Memory mapped register space not defined
- I/O space not defined

Making the Invisible Visible

Create a standard mechanism for abstracting a PCI device function's configuration registers regardless of the PCI address space, configuration, memory, or I/O

Using XML to add Structure



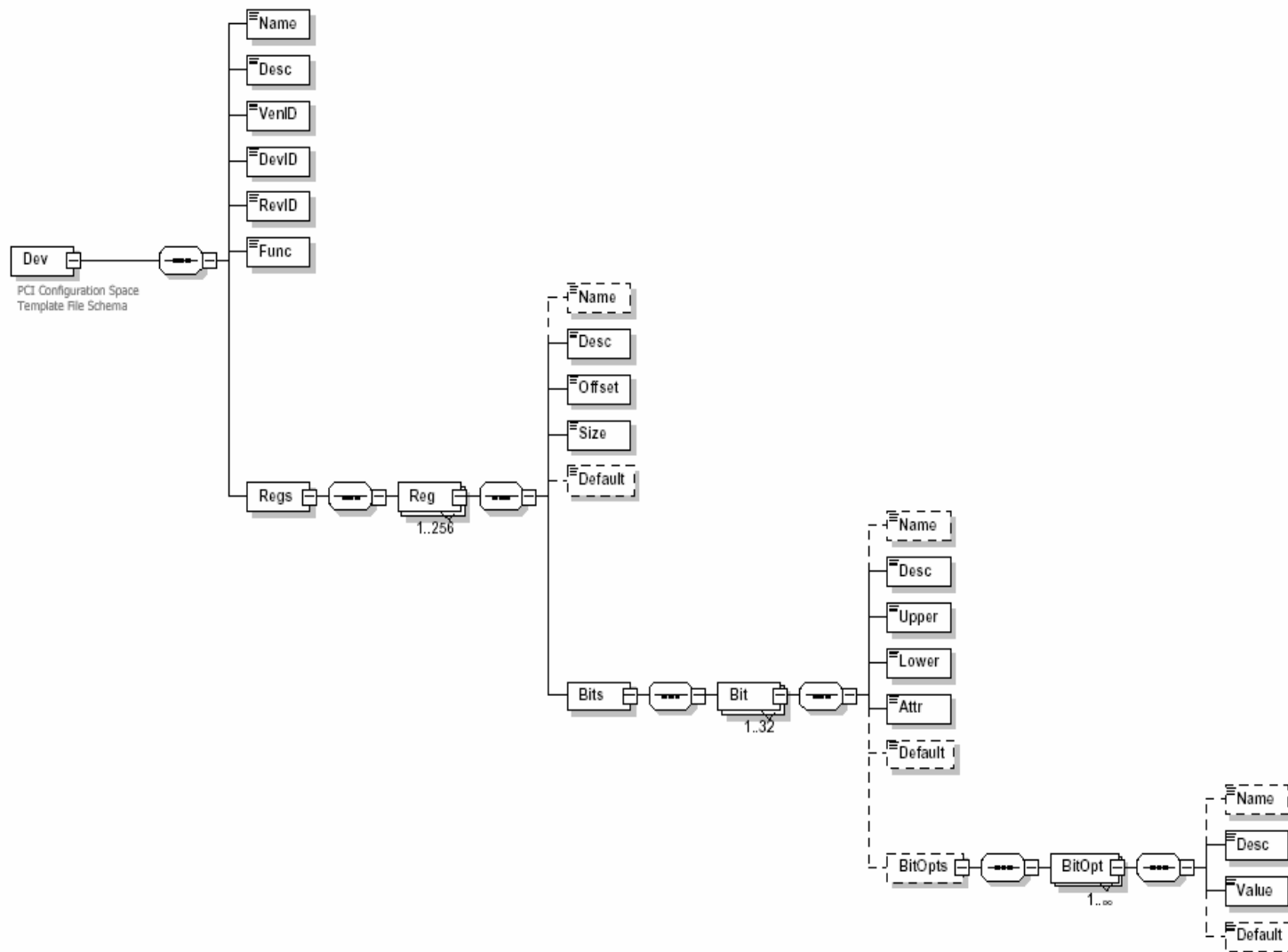
The Benefits:

- IHV
 - ✓ Simplifies the delivery mechanism
 - ✓ Reduced development cycle
- OSV
 - ✓ Simplifies OS development and testing
 - ✓ Standardizes interface across defined PCI Device Function classes
- System Vendor
 - ✓ Reduced system development time through re-use of mature PCI code base (BIOS, OS, etc)
 - ✓ Simplified system configuration and performance tuning
- End User Application
 - ✓ Improves development tools
 - ✓ Improved management through structured description of system devices

How Do We Get There

- **Phase 1:** Industry standardizes on a schema definition representing PCI Configuration Space for any device
- **Phase 2:** Device vendors provide XML description files of their devices based on the schema in Phase 1
- **Phase 3:** Users develop applications that use the XML description files such as management tools, debug and configuration utilities, device drivers, performance tools,

Phase 1: The Schema

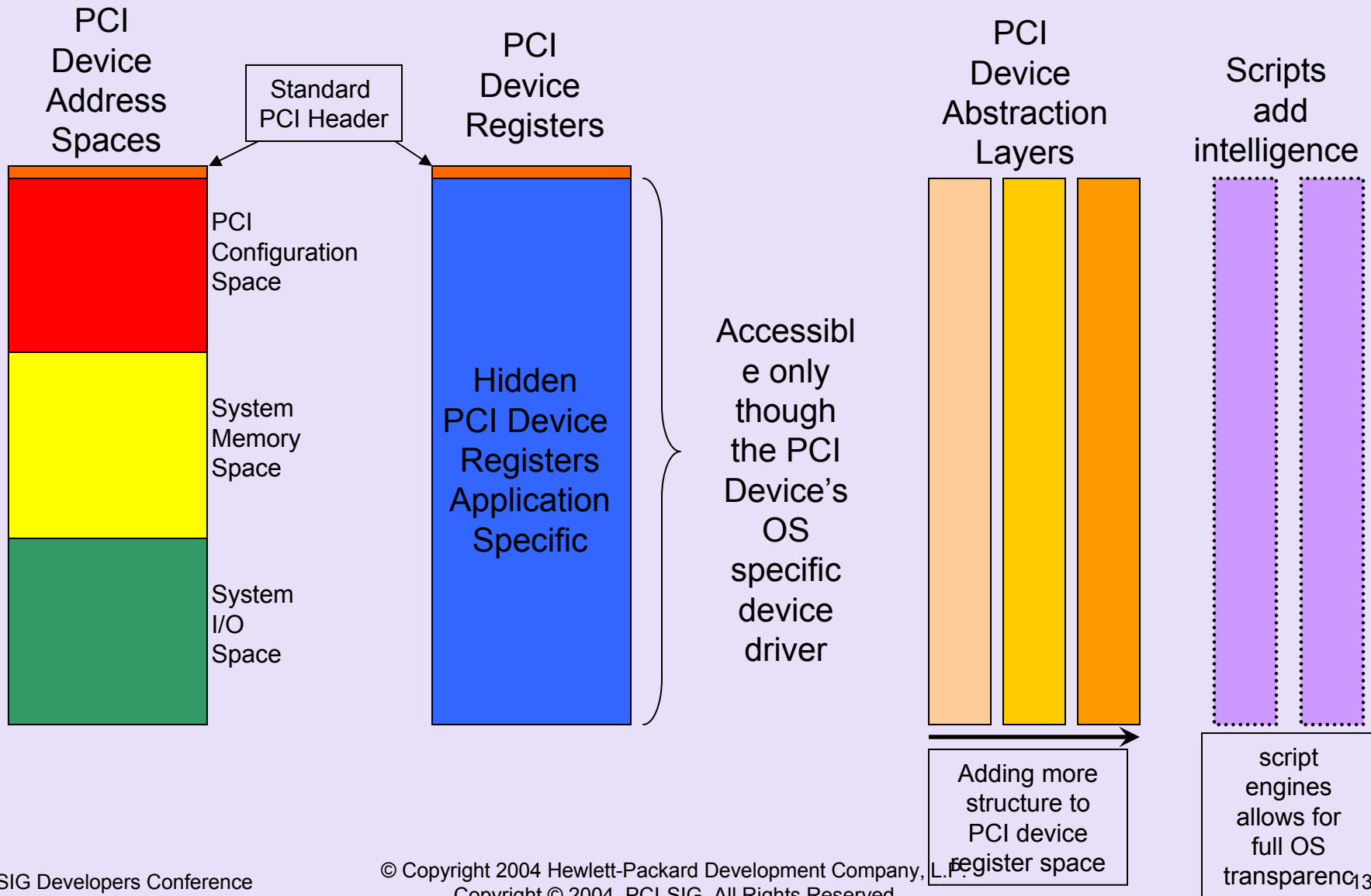




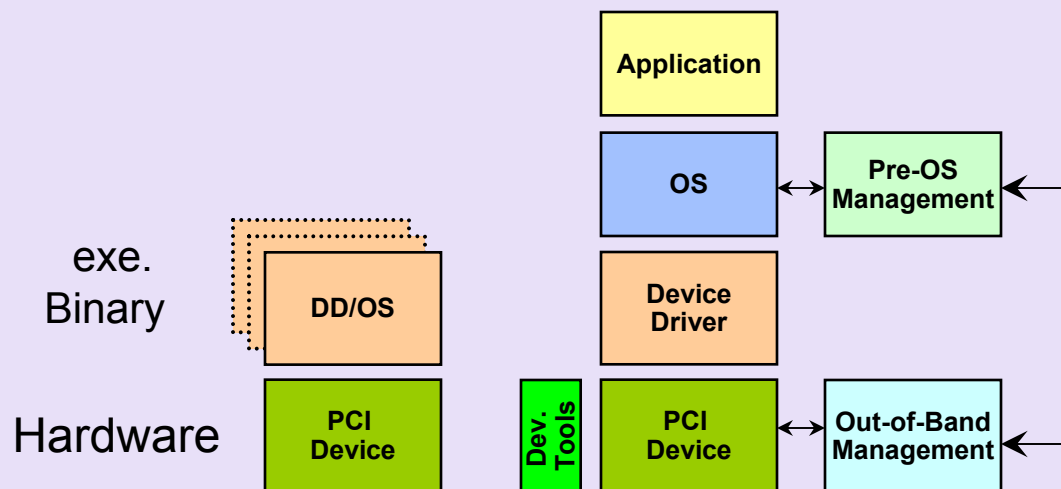
Phase 2: Describes PCI Device Function Configuration Space

```
<?xml version="1.0" encoding="UTF-8"?>
<!--PCI Configuration Space Definition File Template-->
<!--(c) 2003 Hewlett-Packard Corporation-->
<Dev xmlns="http://www.pcisig.com/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.pcisig.com/namespace
C:\Public\PCIDUMP\XMLFOR~1\PCITEM~1.XSD">
  <Name>UNKNOWN</Name>
  <Desc>Unknown PCI Device</Desc>
  <VenID unit="hex">FFFF</VenID>
  <DevID unit="hex">FFFF</DevID>
  <RevID unit="hex">FF</RevID>
  <Func>0</Func>
  <Regs>
    <Reg>
      <Name>VID</Name>
      <Desc>Vendor ID Reg</Desc>
      <Offset unit="hex">0</Offset>
      <Size unit="bytes">2</Size>
      <Default unit="hex"/>
      <Bits>
        <Bit>
          <Name>VID</Name>
          <Desc>Vendor ID</Desc>
          <Upper unit="bits">15</Upper>
          <Lower unit="bits">0</Lower>
          ...
```

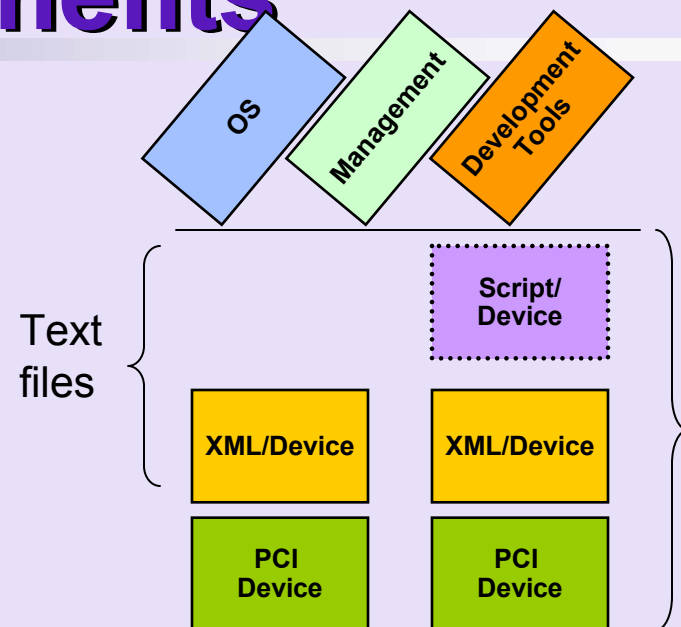
Full Abstraction Allows for Full Transparency



Phase 3: The Benefits



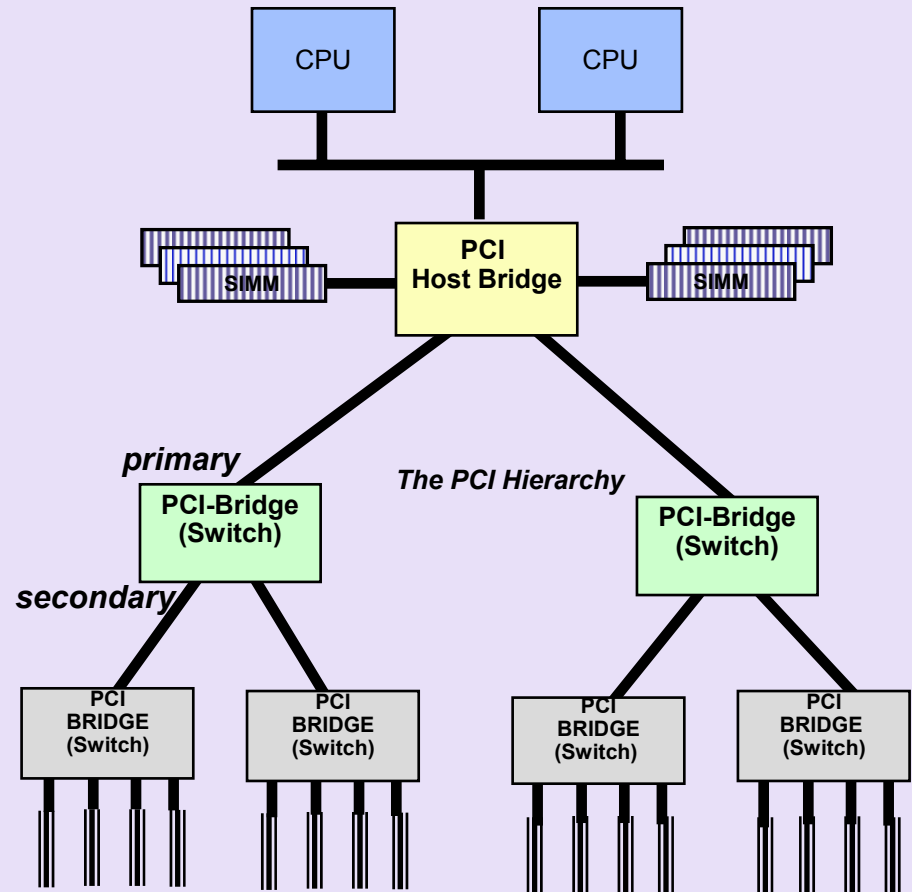
The old way



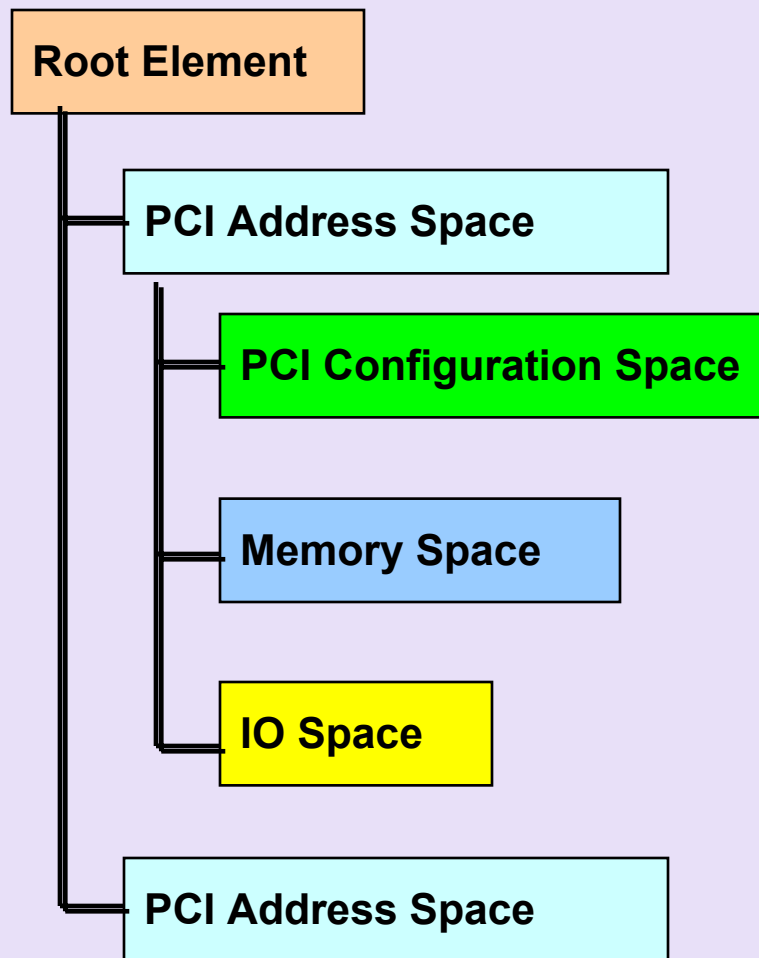
The new way

Local I/O linking it all together: The PCI hierarchy

- PCI Address Models
 - ✓ Configuration Space
 - ✓ Memory Space
 - ✓ I/O Space
- Host-Bridge/Root Complex
 - ✓ 1 per PCI hierarchy
- Bus Numbers
 - ✓ Bus 1 is the primary bus
 - ✓ 1-256 bus segments
- Devices
 - ✓ 1-32 devices per bus segments
- Function
 - ✓ 1-8 functions per PCI Device



PCI Device Address Space Schema



Root Element

```

<PCIAddressSpaceModel>
  <PCIConfigurationSpaceModel>
</PCIConfigurationSpaceModel>
  <MemorySpaceModel>
</MemorySpaceModel>
  <IOSpaceModel>
</IOSpaceModel>
</PCIAddressSpaceModel>
  
```


Defining XML Tags: e.g. Video Controller

```
1. <BitField>
2.   <Name>Video Mode</Name>
3.   <Range>15:12</Range>
4.   <BitValue>
5.     <Name>640 by 480</Name>
6.     <Value>0000</Value>
7.   </BitValue>
8.   <BitValue>
9.     <Name>1024 by 768</Name>
10.    <Value>0001</Value>
11.  </BitValue>
12. </BitValue>
13.   <Name>1280 by 1024</Name>
14.   <value>0010</Value>
15. </BitValue>
```

Data :

Formatted – Yes
Structured -- No
Logic -- No

Defining application specific XML formatting Tags

Adding Industry Standard Tags: e.g. DMTF Management Anchor Tags

```

1. <BitField>
2.   <Name>Video Mode</Name>
3.   <Range>15:12</Range>
4.   <ManagementData Owner="DMTF">
5.     <InstanceClass Name="CIM_VideoController"/>
6.   </ManagementData>
7. <BitValue>
8.   <Name>640 by 480</Name>
9.   <Value>0000</Value>
10.  <ManagementData Owner="DMTF">
11.    <Class Name="CIM_VideoController">
12.      <Property Name="CurrentHorizontalResolution" Value="640"/>
13.      <Property Name="CurrentVerticalResolution" Value="480"/>
14.    </Class>
15.  </ManagementData>
16. </BitValue>

```

Data :

Formatted – Yes
Structured -- Yes
Logic -- No

Using Standard Tags: e.g. Add Dependency Tags (1 of 3)

```

1. <BitField>
2.   <Name>Video Operating Mode</Name>
3.   <Range>1:0</Range>
4.   <ManagementData Owner="DMTF">
5.     <InstanceClass Name="CIM_VideoController"/>
6.     <SettingsClass Name="CIM_VideoControllerResolution">
7.       <Dependencies>
8.         <Dependency Name="RefreshSetting" Path="../BitField[Name='Refresh Frequency']">
9.           <Dependency Name="ColorSetting" Path="../BitField[Name='Colors']">
10.            </Dependency>
11.          </Dependencies>
12.        </SettingsClass>
13.      </ManagementData>
14.    <BitValue>
15.      <Name>640x480</Name>
16.      <Value>01</Value>
17.      <ManagementData Owner="DMTF">
18.        <Class Name="CIM_VideoController">
19.          <Property Name="CurrentHorizontalResolution" Value="640"/>
20.          <Property Name="CurrentVerticalResolution" Value="480"/>
21.        </Class>
22.      </ManagementData>
23.    <ManagementData Owner="DMTF">
24.      <Class Name="CIM_VideoControllerResolution">
25.        <Property Name="HorizontalResolution" Value="640"/>
26.        <Property Name="VerticalResolution" Value="480"/>
27.        <SettingDependencies>
28.          <Dependency SettingName="RefreshSetting" Select="BitValue"/>
29.          <Dependency SettingName="ColorSetting" Select="BitValue"/>
30.        </SettingDependencies>
31.      </Class>
32.    </ManagementData>
33.  </BitValue>

```

Data :

Formatted – Yes
Structured -- Yes
Logic -- Yes

Logic Dependency:

Video Operating Mode
- Refresh Frequency
- Colors

Using Standard Tags: e.g. Add Dependency Tags, con't (2)

```

1. <BitField>
2.   <Name>Refresh Frequency</Name>
3.   <Range>3:2</Range>
4.   <ManagementData Owner="DMTF">
5.     <InstanceClass Name="CIM_VideoController"/>
6.     <SettingsClass Name="CIM_VideoControllerResolution">
7.       <Dependencies>
8.         <Dependency Name="ResolutionSetting" Path="../BitField[Name='Video Operating Mode']">
9.         <Dependency Name="ColorSetting" Path="../BitField[Name='Colors']">
10.      </Dependencies>
11.    </SettingsClass>
12.  </ManagementData>
13.  <BitValue>
14.    <Name>60Hz</Name>
15.    <Value>01</Value>
16.    <ManagementData Owner="DMTF">
17.      <Class Name="CIM_VideoController">
18.        <Property Name="CurrentRefreshRate" Value="60"/>
19.      </Class>
20.    </ManagementData>
21.    <ManagementData Owner="DMTF">
22.      <Class Name="CIM_VideoControllerResolution">
23.        <Property Name="RefreshRate" Value="60"/>
24.        <SettingDependencies>
25.          <Dependency SettingName="ResolutionSetting" Select="BitValue"/>
26.          <Dependency SettingName="ColorSetting" Select="BitValue"/>
27.        </SettingDependencies>
28.      </Class>
29.    </ManagementData>
30.  </Bitvalue>

```

Data :

Formatted – Yes
Structured -- Yes
Logic -- Yes

Logic Dependency:

Refresh Frequency
- Video Operating Mode
- Colors

Using Standard Tags: e.g. Add Dependency Tags, con't (3)

```

1. <BitField>
2.   <Name>Colors</Name>
3.   <Range>3:2</Range>
4.   <ManagementData Owner="DMTF">
5.     <InstanceClass Name="CIM_VideoController"/>
6.     <SettingsClass Name="CIM_VideoControllerResolution">
7.       <Dependencies>
8.         <Dependency Name="ResolutionSetting" Path="../BitField[Name='Video Operating Mode']">
9.           <Dependency Name="RefreshSetting" Path="../BitField[Name='Refresh Frequency']">
10.            </Dependency>
11.          </Dependencies>
12.        </SettingsClass>
13.      </ManagementData>
14.    <BitValue>
15.      <Name>16 colors</Name>
16.      <Value>01</Value>
17.      <ManagementData Owner="DMTF">
18.        <Class Name="CIM_VideoController">
19.          <Property Name="CurrentColors" Value="16"/>
20.        </Class>
21.      </ManagementData>
22.    <ManagementData Owner="DMTF">
23.      <Class Name="CIM_VideoControllerResolution">
24.        <Property Name="Colors" Value="16"/>
25.        <SettingDependencies>
26.          <Dependency SettingName="ResolutionSetting" Select="BitValue"/>
27.          <Dependency SettingName="RefreshSetting" Select="BitValue"/>
28.        </SettingDependencies>
29.      </Class>
30.    </ManagementData>
31.  </BitValue>

```

Data :

Formatted – Yes
Structured -- Yes
Logic -- Yes

Logic Dependency:

Colors

- Video Operating Mode
- Refresh Frequency

Hierarchy of the PCI Configuration Space

Element Name	Parent	Children	Description
<PCIConfigurationSpace>	<PCIAddressSpaceModel>	<Name> <Desc> <Buses>	Parent element of the PCI Configuration Space branch of the schema
<Name>	<PCIConfigurationSpace>	None	The name of the PCI Configuration Space.
<Desc>	<PCIConfigurationSpace>	None	The description of the PCI Configuration Space.
<Buses>	<PCIConfigurationSpace>	<Bus>	The collection of PCI buses in the parent configuration space.
<Bus>	<Buses>	<Name> <Desc> <Devs>	A particular instantiation of a bus. There can be from 1 to 256 bus instantiations in the <Buses> collection.
<Name>	<Bus>	None	The name of the PCI bus.
<Desc>	<Bus>	None	The description of the PCI bus.
<Devs>	<Bus>	<Dev>	The collection of devices on the parent bus.
<Dev>	<Devs>	<Name> <Desc> <Funcs>	A particular instantiation of a device. There can be from 1 to 32 device instantiations in the <Devs> collection.
<Name>	<Dev>	None	The name of the device.
<Desc>	<Dev>	None	The description of the device.
<Funcs>	<Dev>	<Func>	The collection of functions in the parent device.

Hierarchy of the PCI Configuration Space, con't

Element Name	Parent	Children	Description
<Func>	<Funcs>	<Name> <Desc> <Regs>	A particular instantiation of a function. There can be from 1 to 8 function instantiations in the <Funcs> collection.
<Name>	<Func>	None	The name of the function.
<Desc>	<Func>	None	The description of the function.
<Regs>	<Func>	<Reg>	The collection of registers in the parent function.
<Reg>	<Regs>	<Name> <Desc> <Offset> <Size> <BitFields>	A particular instantiation of a register. There can be from 1 to 256 register instantiations in the <Regs> collection.
<Name>	<Reg>	None	The name of the register.
<Desc>	<Reg>	None	The description of the register.
<Offset>	<Reg>	None	The offset of the register. The unit=<bin,dec,hex> attribute indicates the unit of measurement.
<Size>	<Reg>	None	The size of the register. The unit=<bits,bytes> attribute indicates the unit of measurement.
<BitFields>	<Reg>	None	The collection of bit fields in the parent register.

Hierarchy of the PCI Configuration Space, con't

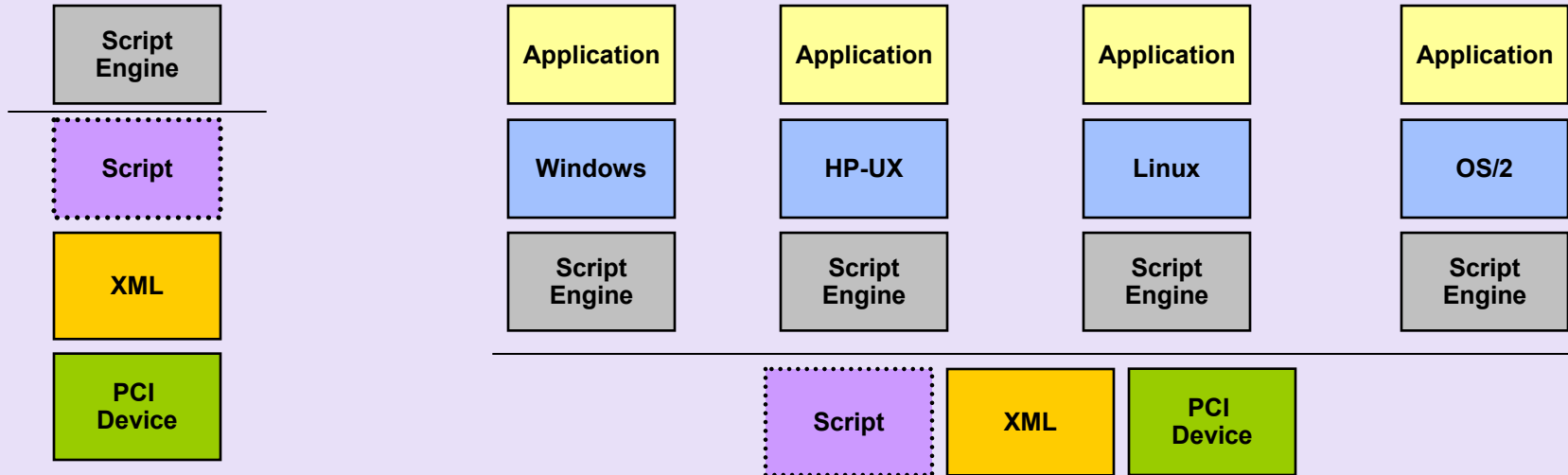
Element Name	Parent	Children	Description
<BitField>	<BitFields>	<Name> <Desc> <Range> <Attr> <Value> <Default> <BitValues>	A particular instantiation of a bit field. There can be from 1 to infinite bit field instantiations in the <BitFields> collection.
<Name>	<BitField>	None	The name of the bit field.
<Desc>	<BitField>	None	The description of the bit field.
<Range>	<BitField>	None	The range of the bit field. Expressed as <Range>h:l</Range>, where h represents the uppermost bit position and l represents the lowermost bit position within the bit field. A single bit is represented by either <Range>a</Range> or <Range>a:a</Range>
<Attr>	<BitField>	None	The attributes of the bit field. The attribute list includes: RO – Read Only RW – Read/Write WC – Write to clear
<Value>	<BitField>	None	The actual value of the bit field. A placeholder to store the value during run-time operation. The unit=<bin,dec,hex> attribute indicates the unit of measurement.
<Default>	<BitField>	None	The default value of the bit field. The unit=<bin,dec,hex> attribute indicates the unit of measurement.

Hierarchy of the PCI Configuration Space, con't

Element Name	Parent	Children	Description
<BitValues>	<BitField>	<BitValue>	The collection of bit values in the parent bit field.
<BitValue>	<BitValues>	<Name> <Desc> <Value> <Default>	A particular instantiation of a bit value. There can be from 1 to infinite bit value instantiations in the <BitValues> collection.
<Name>	<BitValue>	None	The name of the bit value.
<Desc>	<BitValue>	None	The description of the bit value.
<Value>	<BitValue>	None	The value of the bit value. The unit=<bin,dec,hex> attribute indicates the unit of measurement.
<Default>	<BitValue>	None	A Boolean indication of whether or not the current bit value is the default value.



XML Provides Full Transparency



Conclusion: XML Abstraction Benefits

- IHV
 - ✓ Simplifies the delivery mechanism
 - ✓ Reduced development cycle
- OSV
 - ✓ Simplifies OS development and testing
 - ✓ Standardizes interface across defined PCI Device Function classes
- System Vendor
 - ✓ Reduced system development time through re-use of mature PCI code base (BIOS, OS, etc)
 - ✓ Simplified system configuration and performance tuning
- End User Application
 - ✓ Improves development tools
 - ✓ Improved management through structured description of system devices



SIGTM