PCIe Device Drivers Common Interface

Martin Killenberg(DESY),
Ludwig Petrosyan (DESY),
Adam Piotrowski (FastLogic Sp. z o.o.)

- A device driver is loadable kernel module that operates or controls a particular type of device that is attached to a computer
- A device driver contains all the device-specific code necessary to communicate with a device
- Application programs use the device driver API to access the device.

- More devices → more drivers
  - Over the time with increasing in number of drivers, their support is at a limit
- The drivers from different producers have different API
 - That leads to certain difficulties at the level of the user programming

- We want to make live easier
 - On the user level programming
  - On the driver level programming

- Define very basic functionality in the Common API
  - Structures and defines for *read()* and *write()*

  - Some common *ioctl()* commands

- Implementation on the driver level based on the Common API definition
- Will lead as to the principle "write for one use for all"
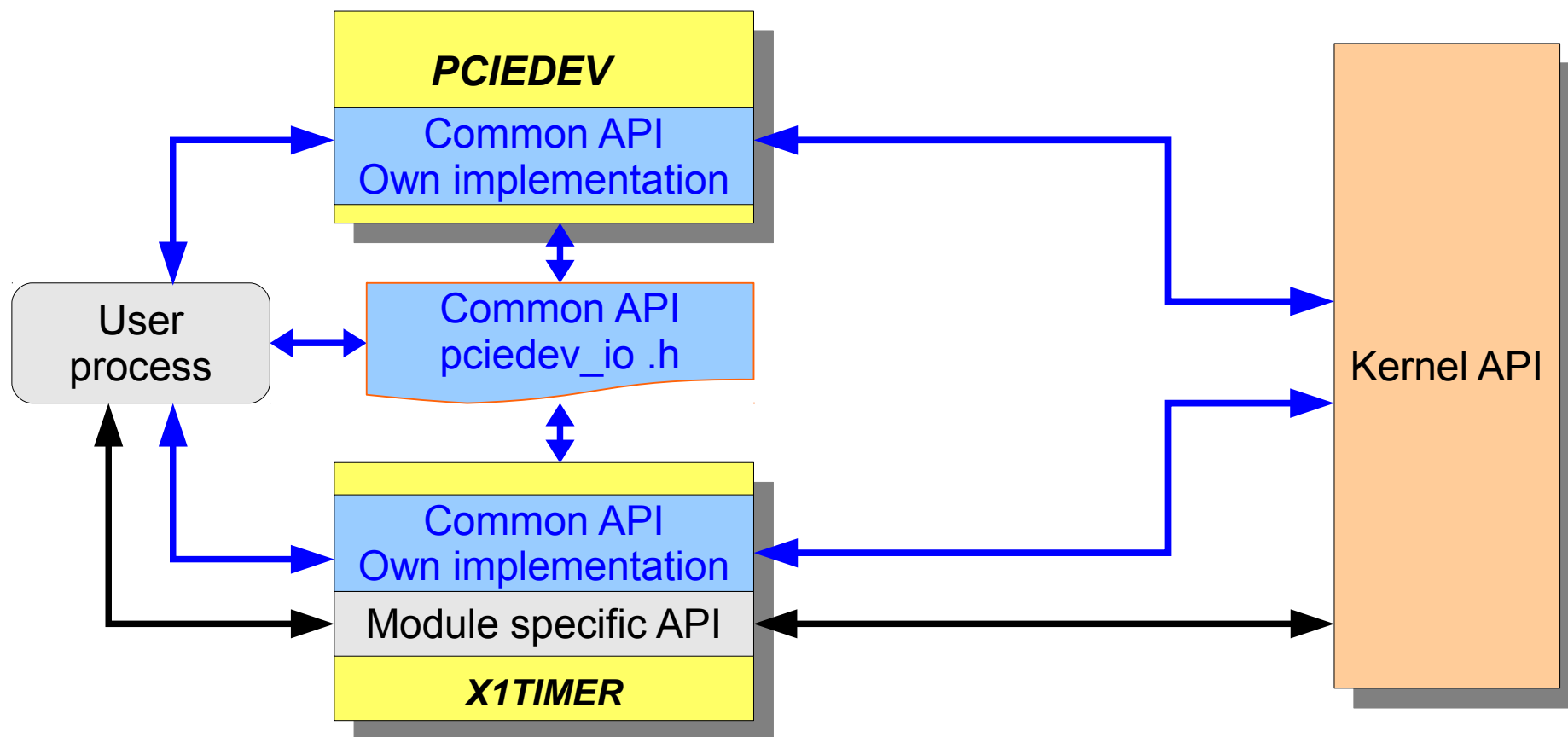
- Basic functionality of the PCIe device driver does not dependent on device type and could be common for all drivers
- Linux kernel allows modules stacking, which basically means one module can use the symbols defined in other modules

⬇
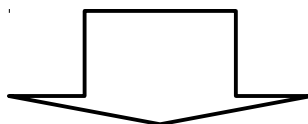
- Split PCIe driver into multiple parts
- Add basic functionality in the top level module

⬇
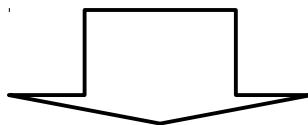
Will facilitate the tasks of creation and supporting of the device drivers

- The top level module provides all common structures and functions for PCIe communications
- The driver for current PCIe module will use the PCIe common part provided by the top level module

- Driver to device binding
- BARs checking-mapping
- Device class and file creating
- DMA mask and IRQ numbers
- */proc* file system
- *read(), write()*
- Some common **ioctl**
- Prepare the device to be removed, clean all resources

- *Ioctl()* supposed to use for:
- *Operations which do not fit into standard input/output model*
- Common for all devices
  - GET_SLOT_NUM ...
- Common API but Device specific implementation
  - DMA_R/W ...
- Full device-specific IOCTL
  - Defined and implemented In the driver side
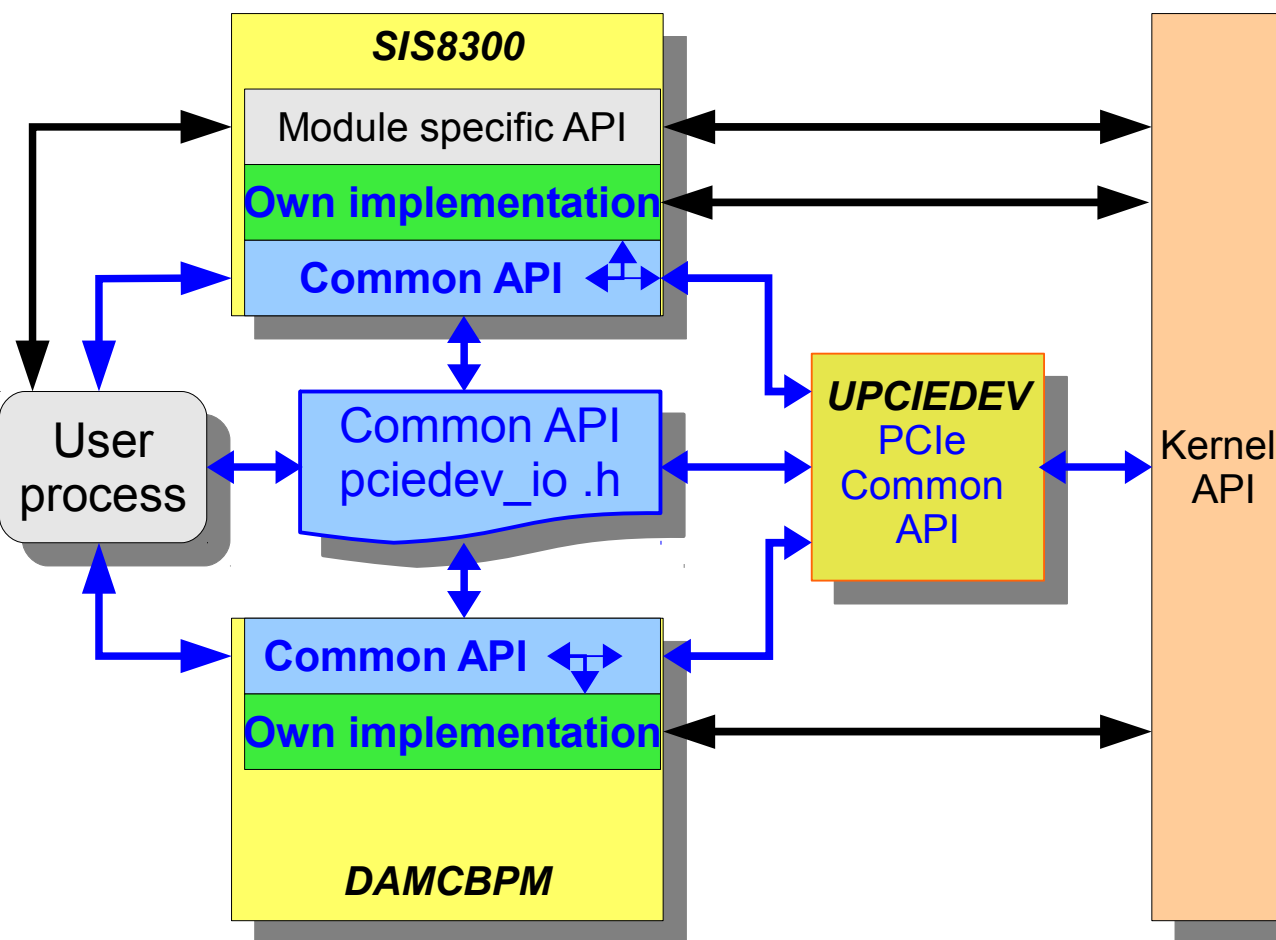
PCIEDEV: device driver for PCIe devices
- Based on common **UPCIEDEV** driver
- used for boards in a development stage
- **MPS, BLM** : based on DAMC2

X1TIMER: device driver for x1timer and x2timer modules
- Based on **Common API**
- Own implementations of *read*, *write* and *ioctl*

UPCIEDEV: top level module provides all common structures and functions for PCIe communications

Common API
pciedev_io.h

MTCAMONITOR:
- GUI to manage and monitor PCIe devices, provides direct hardware access
- Based on Common driver interface

DAMCBPM: based on DAMC2
LCBPM: based on SIS8300
TEWS: TAMC100, TAMC200
ESD ADIO
LLRF drivers(*A.Piotrowski*)

- based on top level **UPCIEDEV** driver and/or **Common API**

UTCApcielib (*O.Hensler*)
Doocs Library

SIS8300: device driver for SIS8300 module
- Based on top level **UPCIEDEV** driver
- DMA IOCTL uses common header but has own implementation

mTCA-MONITOR

Scan PCIe Bus	Run PCIe-Monitor	Rescan Bus	Quit

	10	11	12	2	3	4	5	6	7	8	9
Dev:	0000:04:10.0	0000:04:0a.0	0000:04:0b.0	0000:04:02.0	0000:04:01.0	0000:04:00.0	0000:04:08.0	0000:04:09.0	0000:04:13.0	0000:04:12.0	0000:04:11.0
	10b5:8748	10b5:8748	10b5:8748	10b5:8748	10b5:8748	10b5:8748	10b5:8748	10b5:8748	10b5:8748	10b5:8748	10b5:8748
	SWITCH ON	SWITCH OFF	SWITCH OFF	SWITCH ON	SWITCH ON	SWITCH OFF	SWITCH OFF	SWITCH ON	SWITCH OFF	SWITCH ON	SWITCH ON
DEV:		0000:0a:00.0	0000:0c:04.0			0000:05:00.0	0000:08:00.0		0000:10:00.0		
IDs:		10ee:0020	10b5:9056			1796:0018	10ee:0088		10ee:0088		
		3300:0020	12fe:0600			1796:0018	3300:0088		3300:0088		
Driver:		x1timer	NO DRIVER			sis8300	pciedev		pciedev		
		1.8.0				1.4.0	1.6.0		1.6.0		
BARs:		16777215	511			16383	67108863		67108863		
		0	255			0	67108863		67108863		
		0	65535			0	16777215		16777215		
		0	0			0	0		0		
		0	0			0	0		0		
	PCIe-monitor	PCIe-monitor	PCIe-monitor	PCIe-monitor	PCIe-monitor	PCIe-monitor	PCIe-monitor	PCIe-monitor	PCIe-monitor	PCIe-monitor	PCIe-monitor
	Bind pciedev	Bind pciedev	Bind pciedev	Bind pciedev	Bind pciedev	Bind pciedev	Bind pciedev	Bind pciedev	Bind pciedev	Bind pciedev	Bind pciedev
	INFO	INFO	INFO	INFO	INFO	INFO	INFO	INFO	INFO	INFO	INFO

```
root@mcscpu2:~$ lsmod
    sis8300              21403  5
    x1timer              66799  2
    pciedev               7170  0
    upciedev             28297  6 sis8300, pciedev
```

- Common interface gives possibility to use the same read/write tools for different devices

And the same procedures to get board and driver information

- Common interface and mtcamonitor allow as to bind *pciedev* driver to any PCIe device

- Drivers based both on the common interface and on the common kernel module are created

- Use of the above approach showed the advantages both from support and creation of device drivers and from the user programming

- Add more functionality to the common part
  - More PCIe specific tasks (ErrorHandling, HotPluging, Transaction Orderimg)
  - More Linux Kernel specific tasks (avoid kernel API changes from the version to the version)
  - More R/W and IOCTL (atomic bit operations …)
  - Parallel DMA ...

- ...

offers, remarks, comments …

THANK YOU