The Patient Brush:

How to Clean up a 16 Year Old Linux Kernel API

Philipp Stanner
Kernel Engineer for GPUs @ Red Hat
OFTC: phasta
phasta@kernel.org

Obligatory Disclaimer

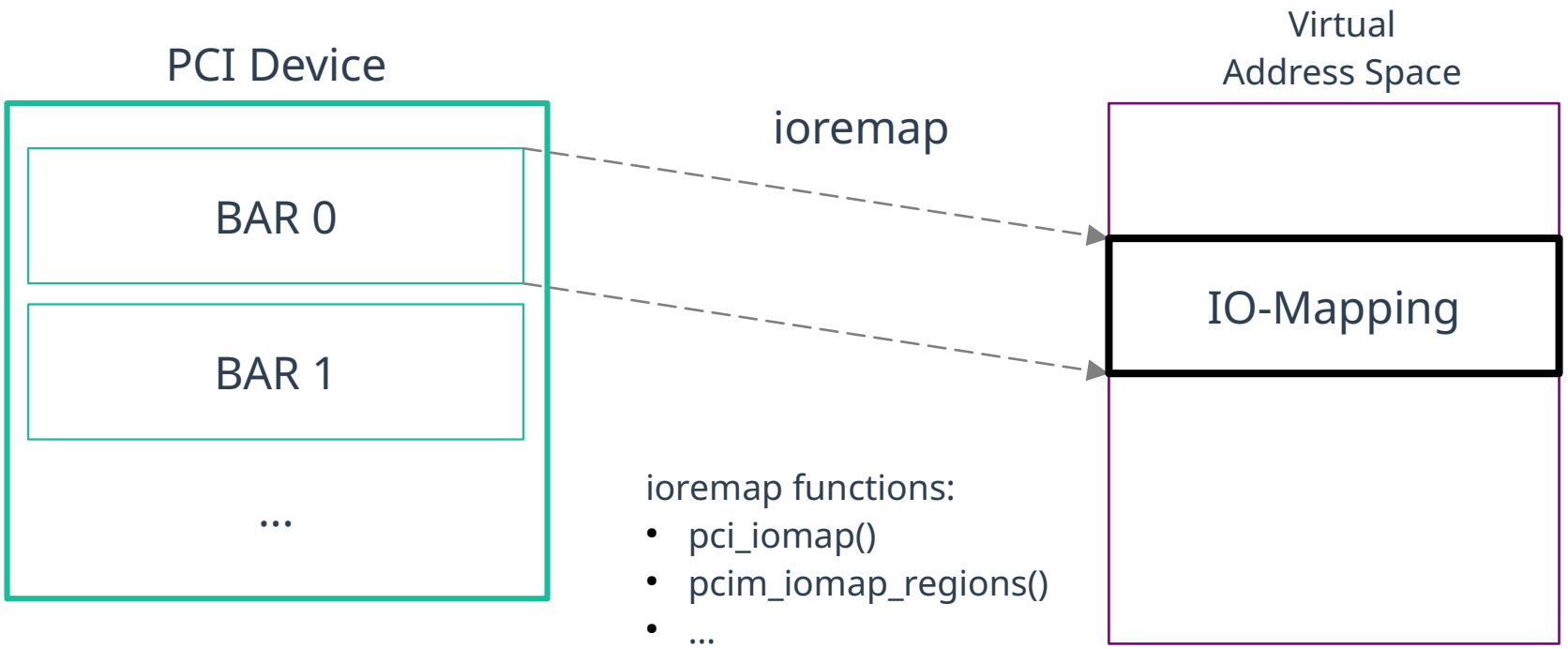
- This talk criticizes old *code*
- We don’t condemn its *authors*
- *“As a software developer, at first you think the others can’t code; ultimately you realize that no one can.”*

  – Anonymous former colleague of mine

The PCI Subsystem

- **PCIe:**
  - Most important bus on your computer
  - Quite old now (>20 years)
- **Subsystem:**
  - Currently has 1 full time maintainer
  - Has 2-3 problematic APIs (potential overflows / leaks) that I know of

PCI in a Nutshell



ioremap functions:

- pci_iomap()
- pcim_iomap_regions()
- ...

The Good Ol’ API (1)

```
void __iomem * const *pcim_iomap_table(struct pci_dev *pdev);
```

- Function in the PCI-Subsystem
- Written in ~2007-2009
- This function does… ahm… it… io-remaps a PCI device? And the iomem is const, isn’t it?

⇒ I don’t get its purpose ⇒ let’s look for users!

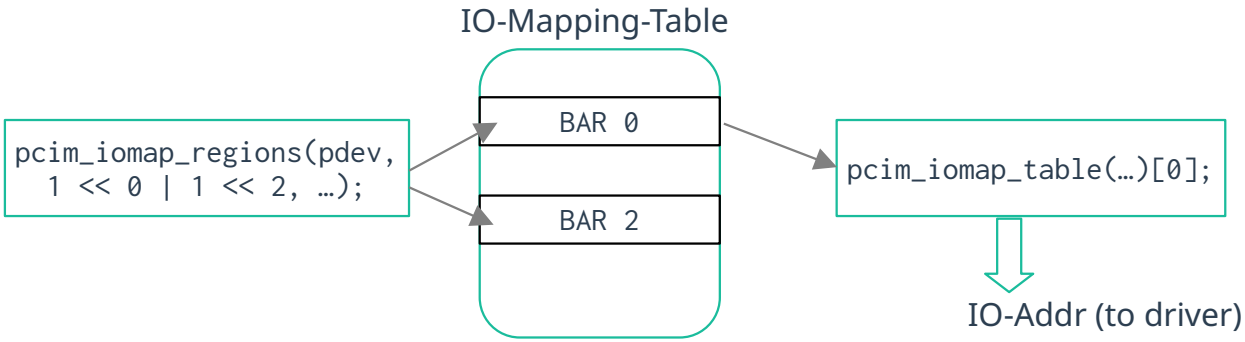
The Good Ol' API (2)

```
ret = pcim_iomap_regions(pdev, 1 << 0, pci_name(pdev));
if (ret) {
     dev_err(&pdev->dev, "I/O memory remapping failed\n");
     return ret;
}

ioaddr = pcim_iomap_table(pdev)[0];
```

- Ahm… OK…
- First request (i.e., reserve) and ioremap a BAR ("region")
- Specify which BAR through a bitmask
- Then get the mapping addr through the table function
- … by indexing "over the function" ò_ó

API-Designer's Intention

```
ret = pcim_iomap_regions(pdev, 1 << 0 | 1 << 2, pci_name(pdev));
if (ret)
     return ret;

ioaddr = pcim_iomap_table(pdev)[0];
```



- Original intention:
  Allow requesting multiple BARs with one call through bitmask
- C functions have (almost) no way to return multiple pointers
  ⇒ table-function to access those

API Problems – Overflows / UB

```
ret = pcim_iomap_regions(pdev, 1 << 0, pci_name(pdev));
if (ret) {
     dev_err(&pdev->dev, "I/O memory remapping failed\n");
     return ret;
}

ioaddr = pcim_iomap_table(pdev)[42 * 9001]; // overflow!
```

- table-function-index can’t be bounds-checked
- PCI devices currently have at most 6 BARs
- bitmask is an int (32 bits), so not that extensible anyways

API Problems – Hackyness

```
rc = pcim_iomap_regions(pdev, 0x3 << base,
                        dev_driver_string(gdev));
```

From: drivers/ata/libata-sff.c

- 0x3 = 0b11 = “first two BARs”
- Or maybe not, depends on base
- Some APIs encourage hackyness :)

How is it actually being used?

- **Let's search for users in the kernel**
- **Search result:**
  - 131 users (in early 2024)
  - Almost all request 1 BAR, setting 1 bit in the bitmask
- **Conclusion:**
  - API is overengineered
  - We want: `pcim_iomap_region(pdev, bar_index, …)`

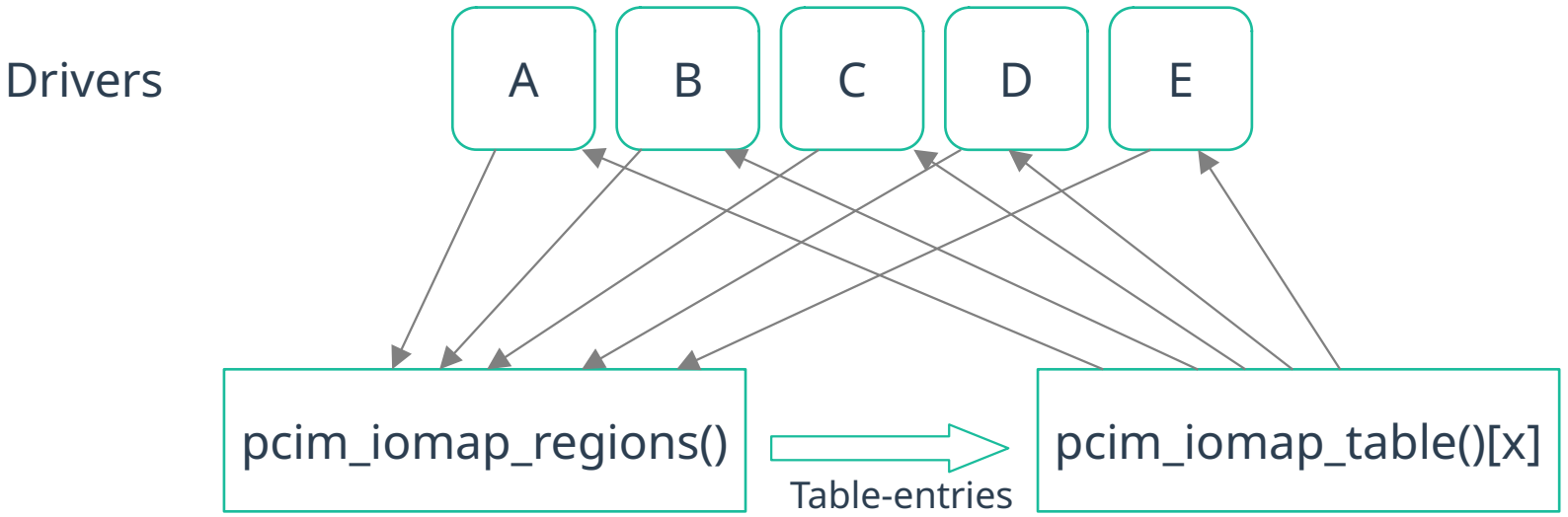
Replacing the old API

- **Obstacles:**
  - Hundreds of users
  - Dozens of drivers / subsystems → many maintainers involved
  - Patches are typically merged per subsystem
  - You need a review / ack for each driver

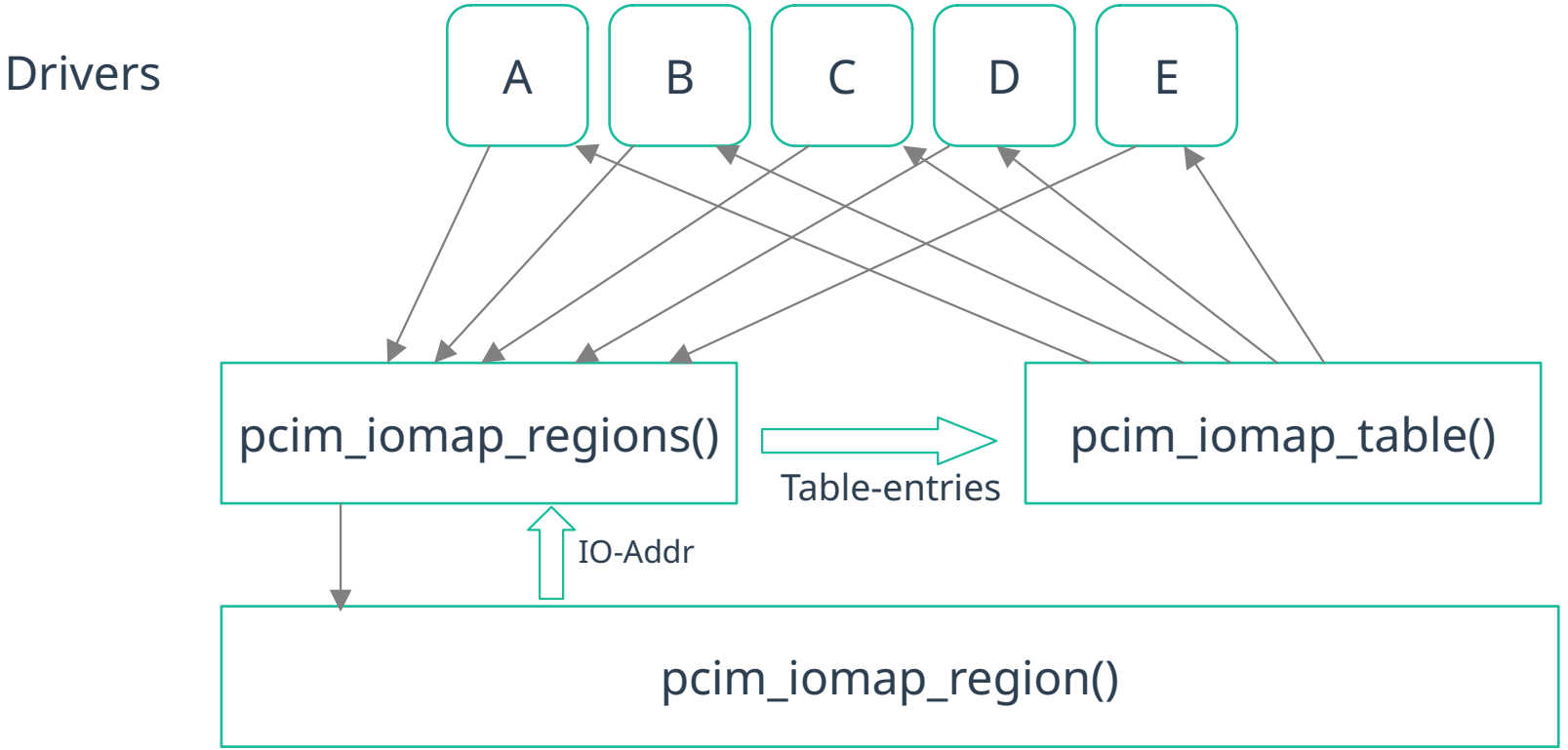
⇒ **Replacing the API at once is impossible**

(Side note: Kernel development is not only tech, but a lot of “politics”)

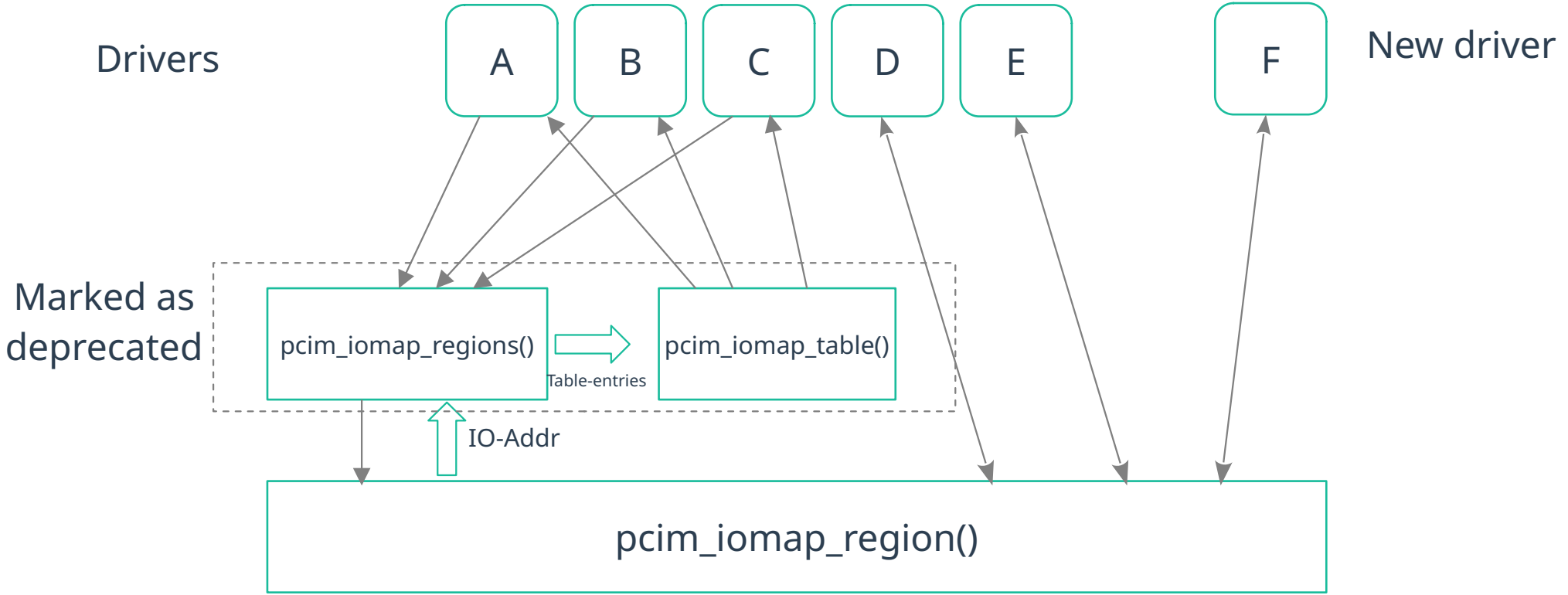
Pre-solution state



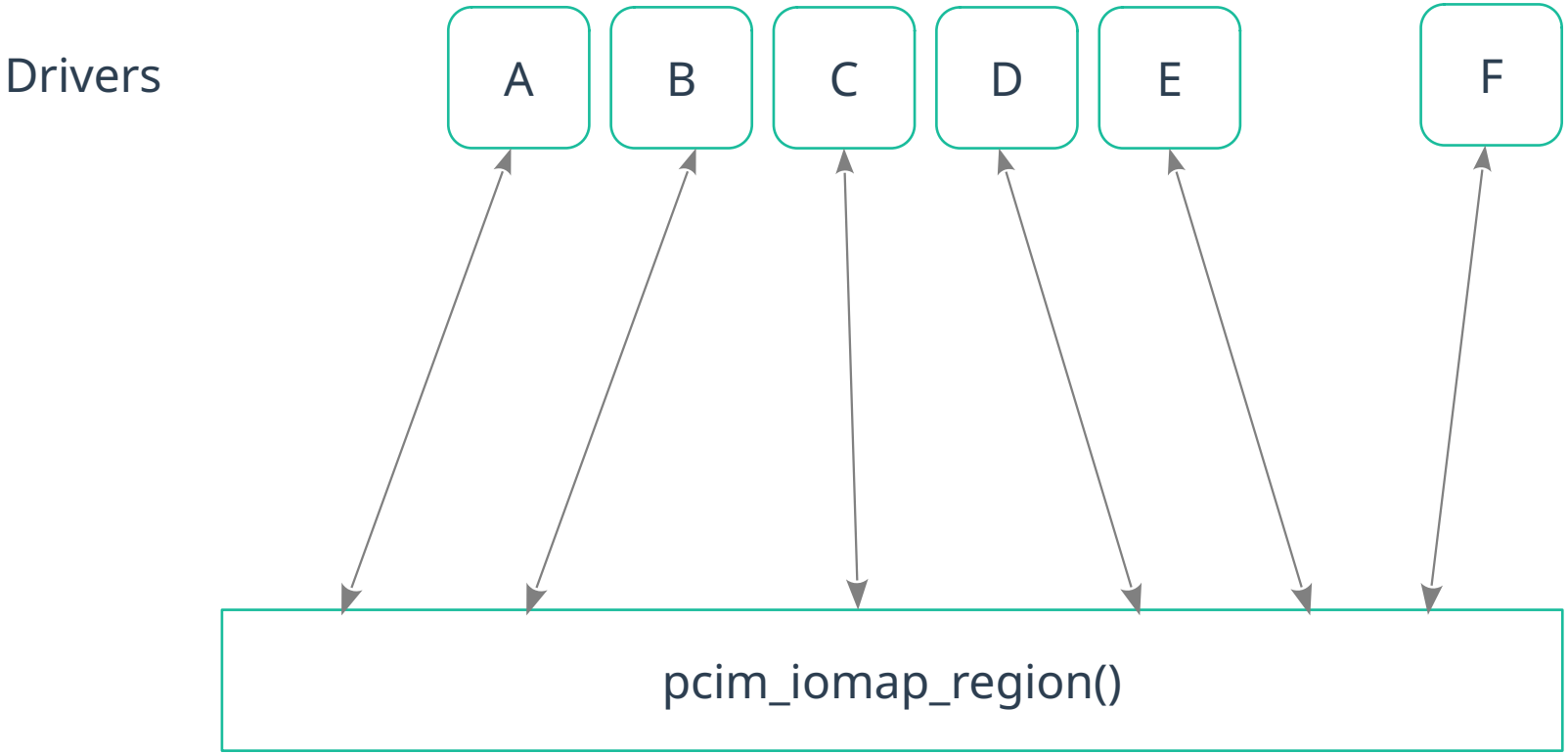
Solution – Step 1: Create a simpler alternative



Solution – Step 2: Port first users



Solution – Step 3: Success! (after years ^^)



Contributing

- **You’d like to get some commits into the kernel?**
- **Try this guide:**

1) Browse code you’re interested in
2) If something looks broken, it likely is!
   (Tip: use git blame to grasp the code’s background)
3) Try to repair it
4) Never hesitate to ask on-list. It’s the maintainer’s duty to guide you
5) Success \o/

Happy Hacking!