Cisco Systems

Kernel Development Process

Roland Dreier

Let's talk about how code goes upstream

Why You Want Your Code Upstream

People won't break your code

- Other developers can see what APIs are being used
- If an API changes, your code will be fixed up:

```
Kay Sievers:

    driver core: replace "hotplug" by "uevent"
```

- ...and you don't have to do anything!

Why You Want Your Code Upstream

People will clean up your code

- For example:

```
Ingo Molnar:

    IB: convert from semaphores to mutexes
```

- ...and you don't have to do anything!

Why You Want Your Code Upstream

People will fix your code

- For example:

```
Adrian Bunk:
    drivers/infiniband/core/mad.c: fix
    use-after-release case
```

- ...and you don't have to do anything!

Why You Want Your Code Upstream

Distribution

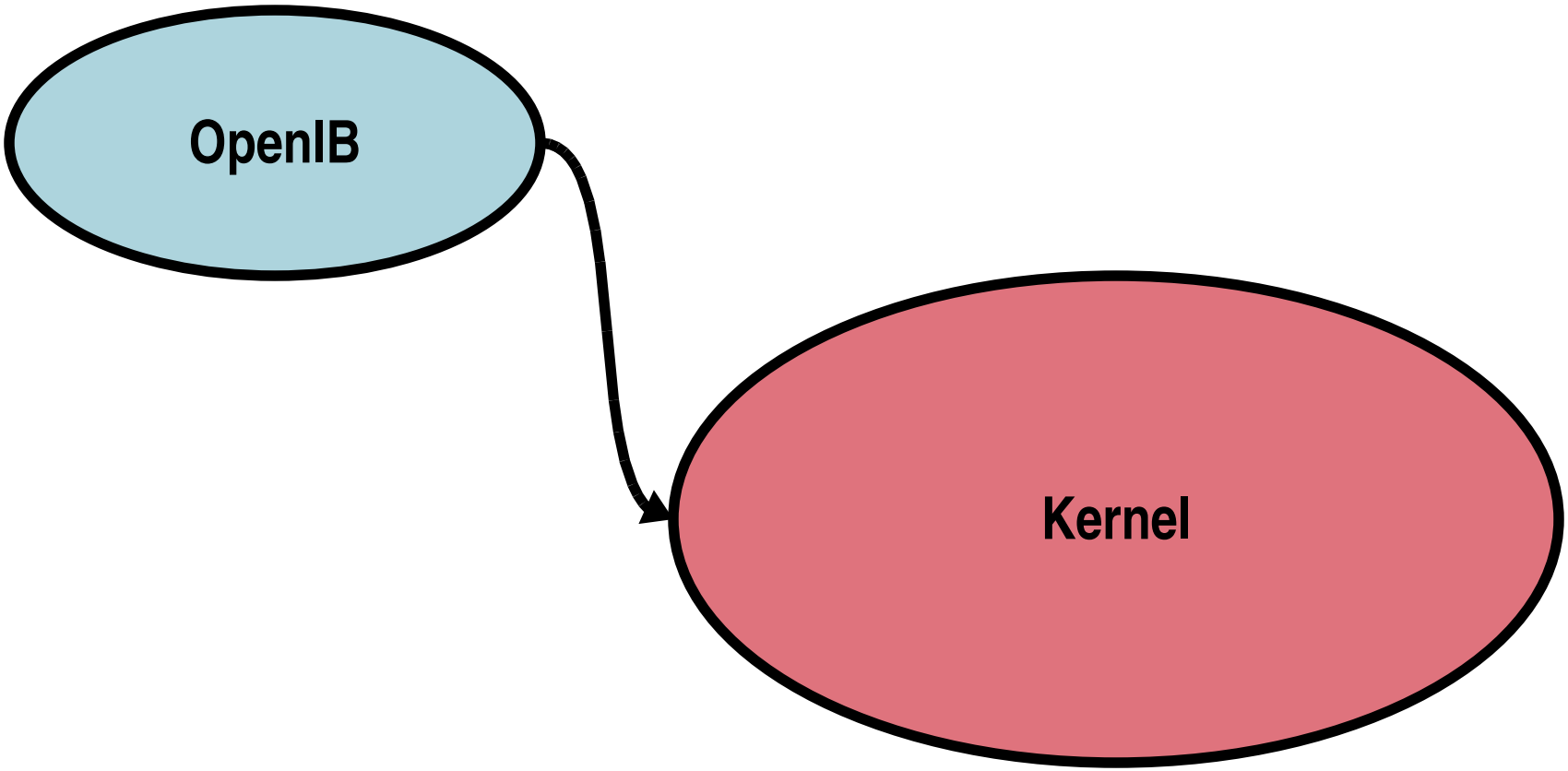
- No kernel patches for users to hunt down
- Linux distributions ship it
- ...and you don't have to do anything!

Why You Want Your Code Upstream

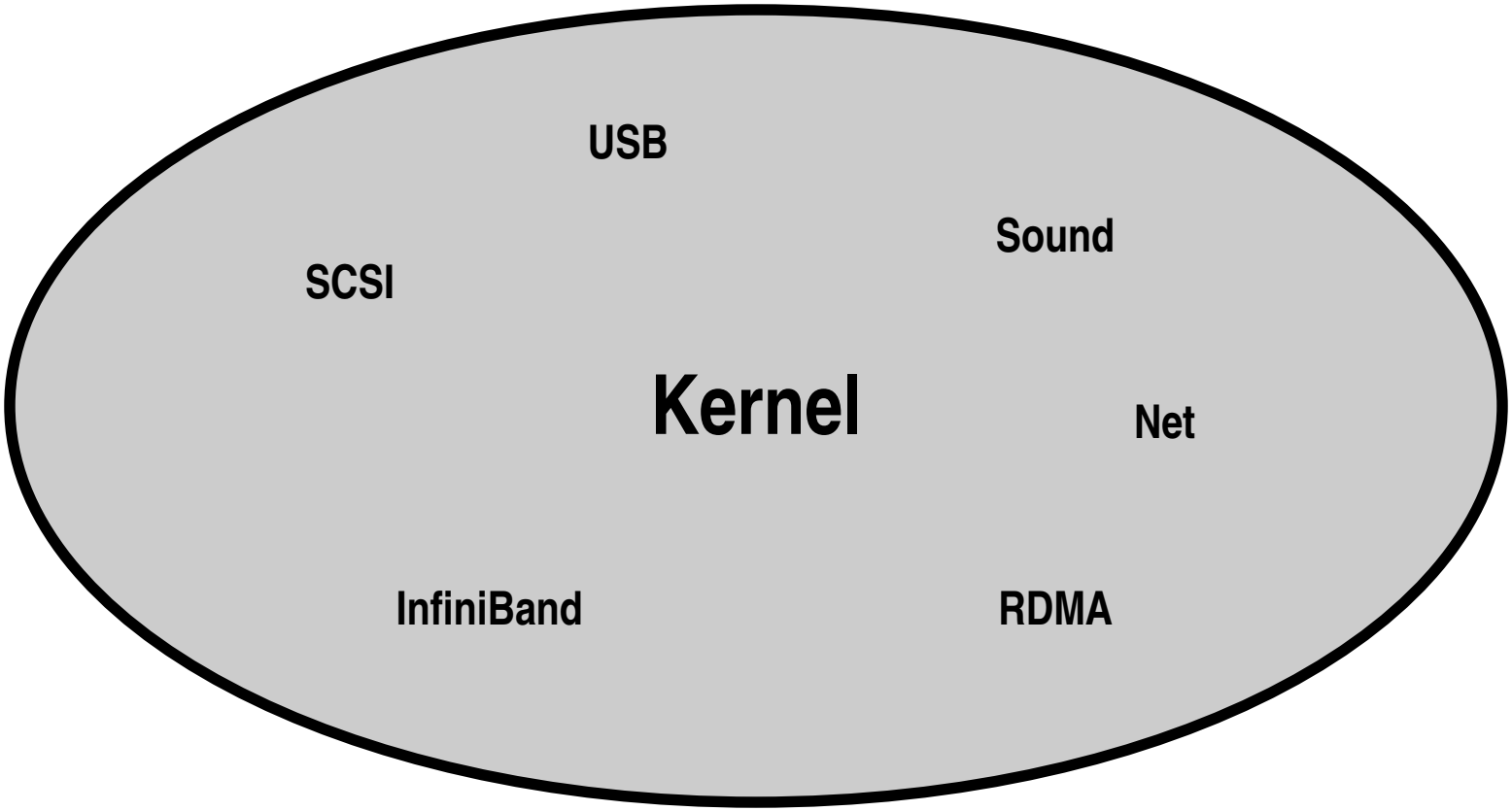
Save what remains of Roland's sanity

- Don't make me track more trees!

The Wrong Way to Look at the World



The Right Way to Look at the World



> “ Justifying the inclusion of a feature by the appearance and usefulness of the end result doesn't really work in this world. There are numerous unmerged kernel features out there which work well and look great. But we will look under the hood, and that's when problems start.”

Andrew Morton (http://lkml.org/lkml/2006/2/2/334)

What This Means

Write your code as if it's part of the kernel

- Read Documentation/CodingStyle
  - believe it, follow it
- Get review early
  - simpler is better
  - don't wait until you're “done”
- Stay engaged with merge process
  - take feedback and come back with better code quickly

It's a Conversation

“This patch is horrible.”

- **Doesn't mean, “go away and stop bothering us.”**
- **It means, “I'm interested enough to read your code.”**
- **Don't take it personally**
- **Disagreement is OK**

Things To Say About a Patch

- Bad: “This is the wrong thing to do, but we can fix it later.”
  - If your later never comes, someone else has to fix it.
- Good: “This is a simple step in the right direction. Next, we'll be able to make these improvements.”
  - Reviewers want patches they can understand (and read in a finite amount of time)
  - We want confidence we aren't making things worse

Things To Say About a Patch

- Bad: “This is ugly because we have to work around this core kernel API.”
  - It's your code too: don't “work around” it - fix it!
- Good: “This improves the kernel's API to make the implementation of the following feature better.”

Things To Say About a Patch

- Bad: “This 100,000 line patch integrates iWARP, fixes a bunch of bugs and reorganizes the source tree.”
  - Imagine reviewing that patch!
- Good: “This series of patches introduces one new feature in a step-by-step way.”

Q and A