



Using Rust for out-of-tree kernel drivers

Alice Ryhl (aliceruhl@google.com)

Rust Microconference — Linux Plumbers Conference 2026



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026

Abstractions are uniquely hard out of tree

Where this comes up

- Backporting upstream drivers
- LTS-first development
- Pure OOT drivers

Sources of friction

- Missing abstractions
- Abstractions are hard to extend

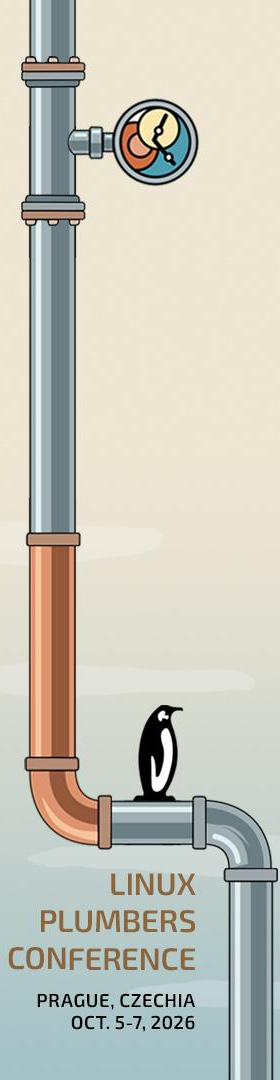


Missing abstractions

Scary to feature teams getting started with Rust

Harder to implement when not upstream:

- Modifying rust/kernel/ complicates LTS merges
- Difficult to extend abstractions from outside rust/kernel/
- Separate bindgen and C helper invocations are fragile



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

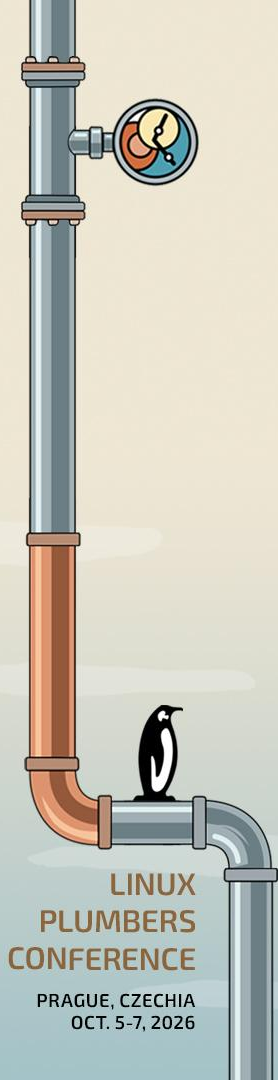
Abstractions are hard to extend

When it happens

- Abstraction is missing a method
- Older LTS lacks a newer feature

Why it's hard

- `as_raw()` / `from_raw()` are private
- Unsafe APIs are private
- Orphan rules



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

Extending abstractions: discussion

1. Make `as_raw()` / `from_raw()` pub by default?

2. Are there cases where that's too dangerous even for `unsafe`?

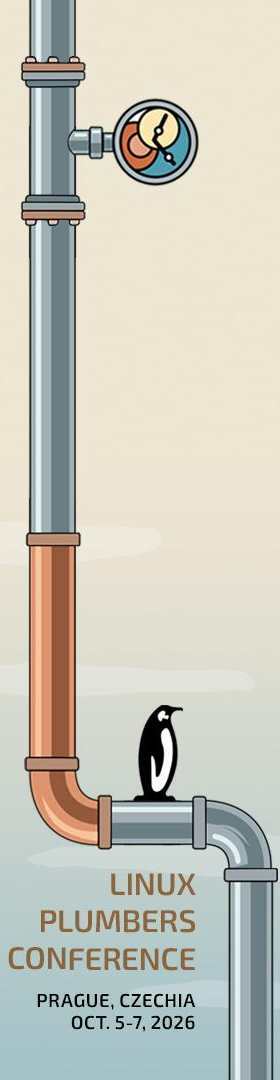
3. The kernel crate split needs the same boundaries.

New abstractions

Upstream rule: no abstraction without an in-tree user

Example: `dma_buf`

Will have in-tree users eventually - but none are planned yet



LINUX
PLUMBERS
CONFERENCE

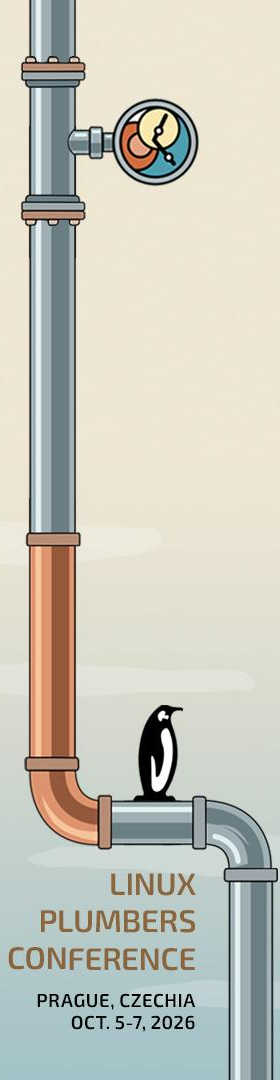
PRAGUE, CZECHIA
OCT. 5-7, 2026

New abstractions: discussion

1. Review abstractions that only have OOT users *right now*?

2. How do OOT drivers share abstractions with each other?

3. Land some upstream anyway, with KUnit / doctests / samples as the user?



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026