



# DPDK

SUMMIT

May 12-13 2026 | Stockholm, Sweden

## Integrating FRR with Grout

Maxime Leroy, Free Mobile  
Robin Jarry, Red Hat

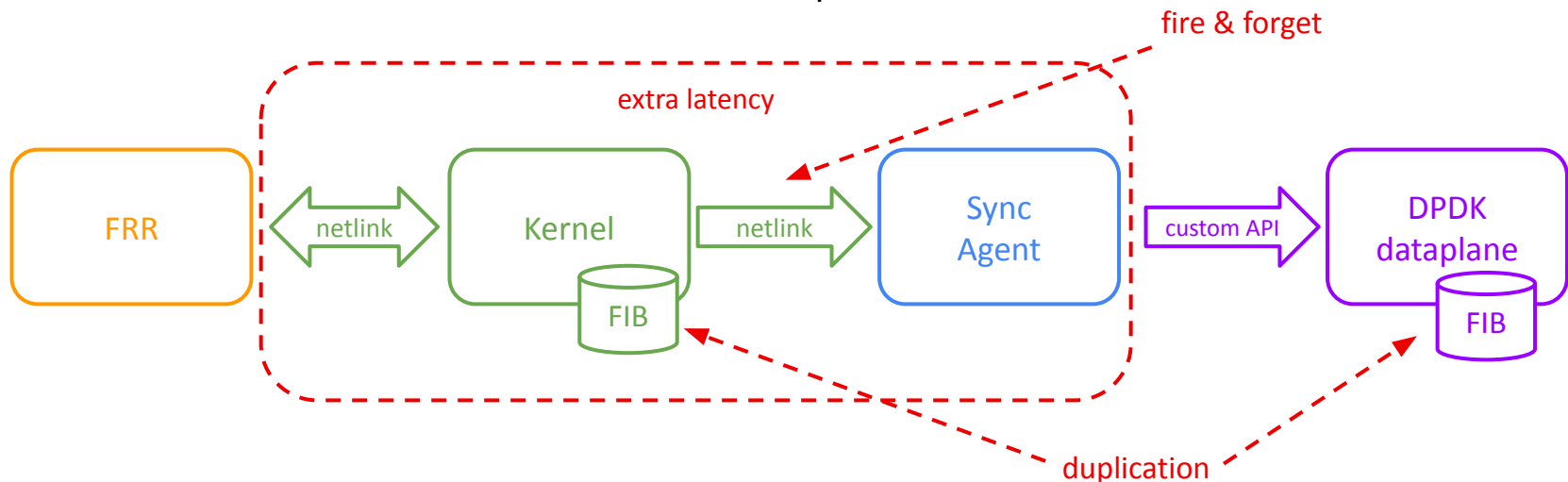
# FRR?

- **Free Range Routing: open-source**, multi-protocol routing stack. Production-grade successor to Zebra/Quagga (GPLv2): BGP, OSPF, IS-IS, RIP, LDP, BFD ...
- **Born multi-dataplane**: Zebra/Quagga ran on \*BSD, Solaris, Linux, even private Windows kernel ports, and pushed routes into ASICs. But Linux dominates today.
- Modular architecture? The **dpPlane plugin API** is the natural hook for programming a custom dataplane



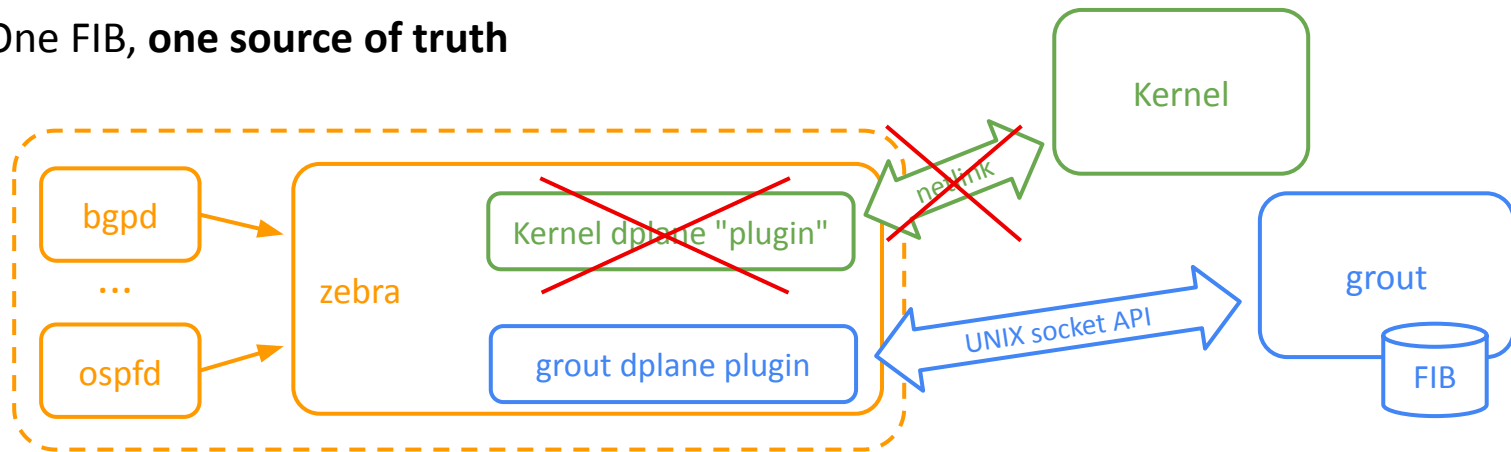
# The problem

- Most DPDK routers **mirror the kernel FIB** into userspace
- Two FIBs, two places where things go wrong
- FRR ⇒ kernel ⇒ netlink listener ⇒ DPDK datapath

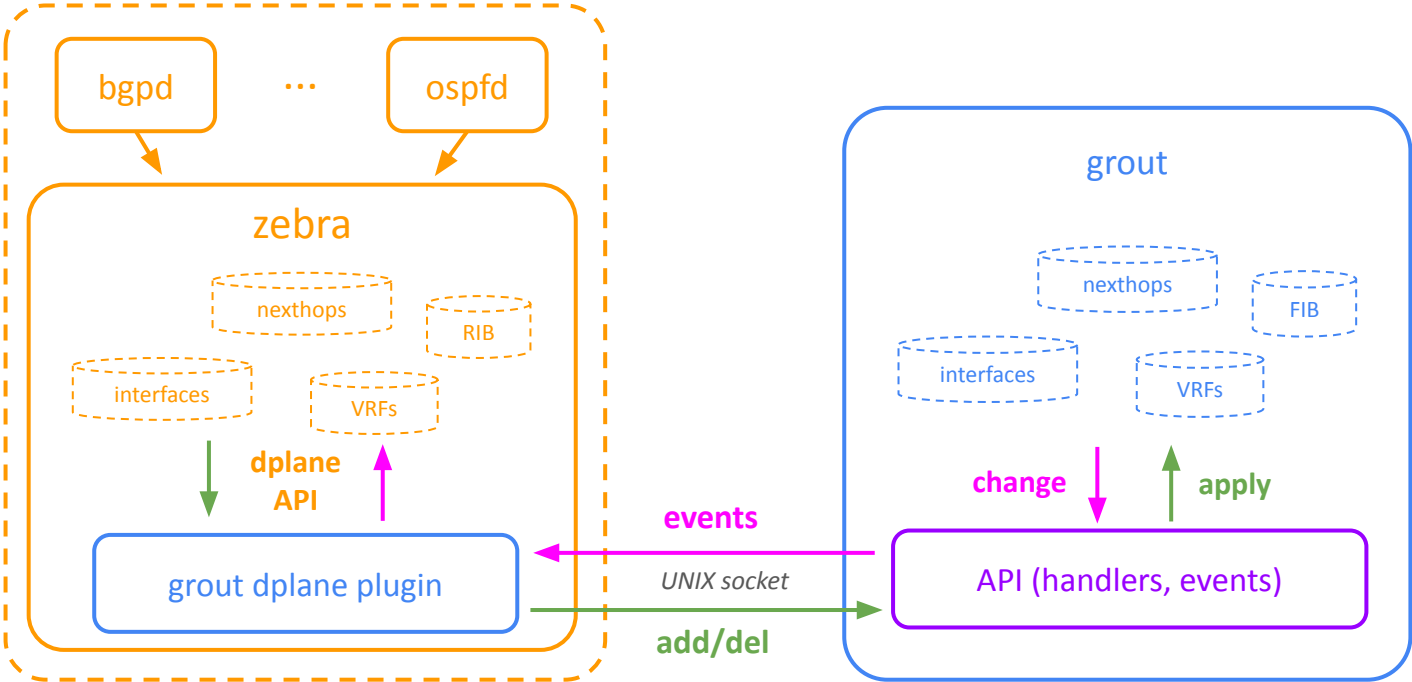


# Skip the kernel entirely

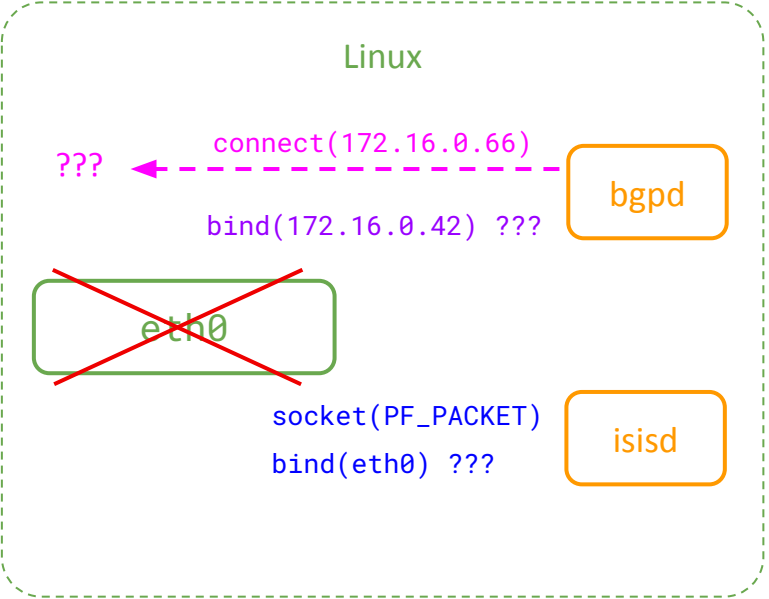
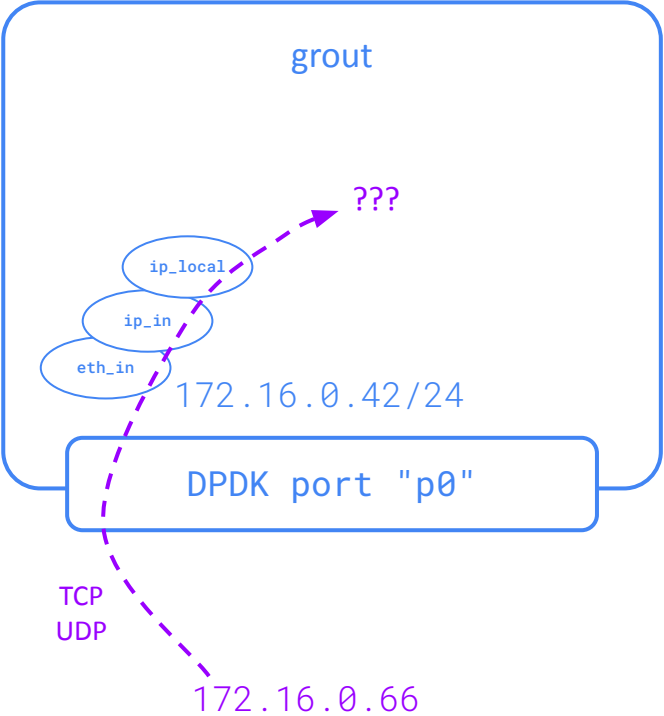
- FRR has a **dataplane plugin API (dplane)**
- Intercept route installs **before** they hit the kernel
- Program routes straight into the DPDK FIB
- One FIB, **one source of truth**



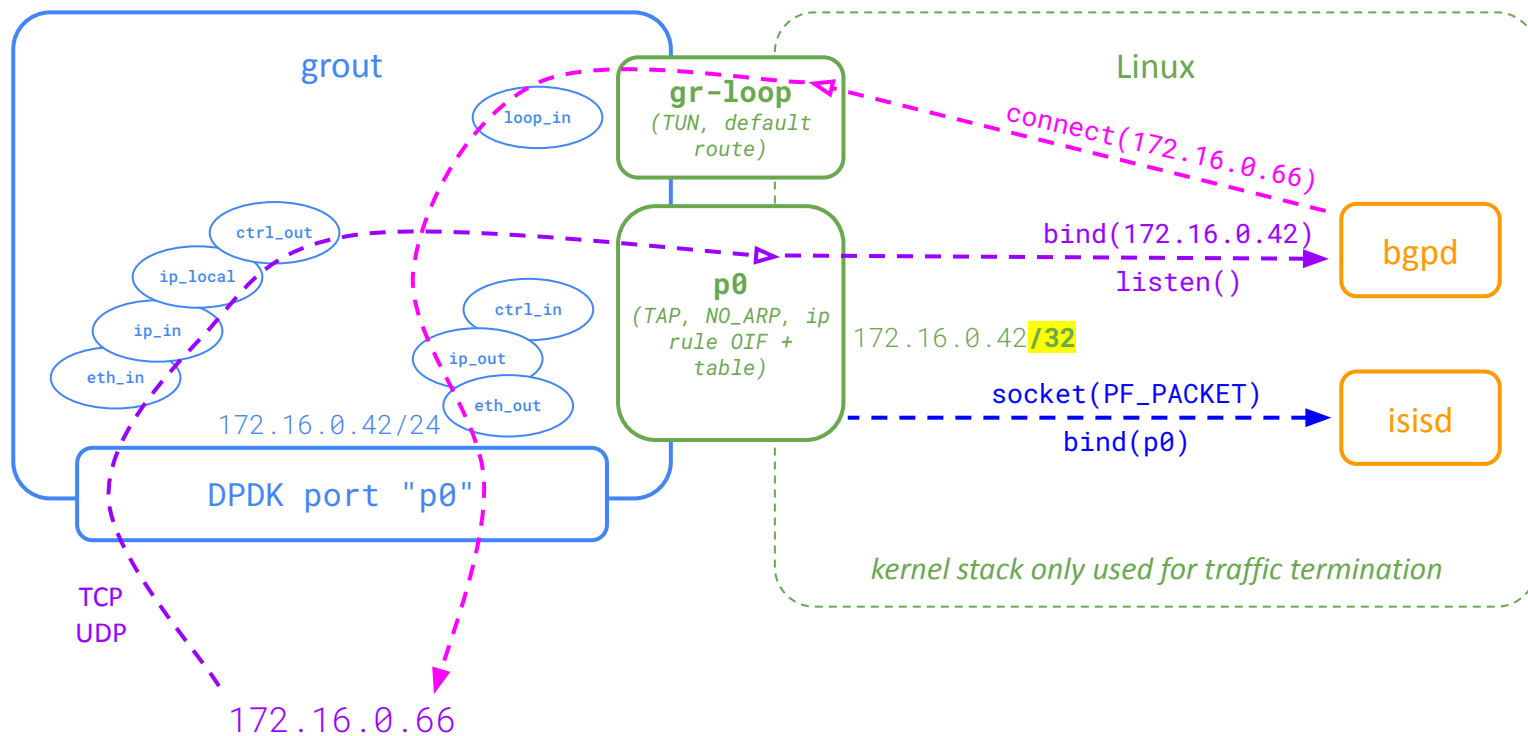
# Bidirectional sync



# Enabling local traffic termination

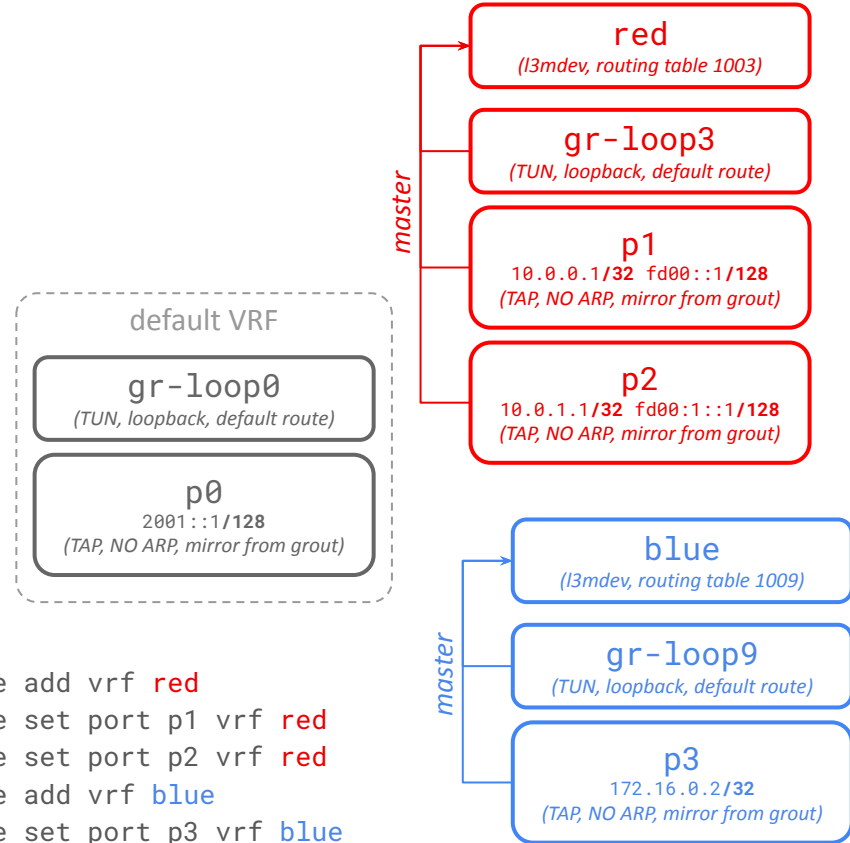


# Control-Plane "representors"



# VRFs: bridging Grout and Linux

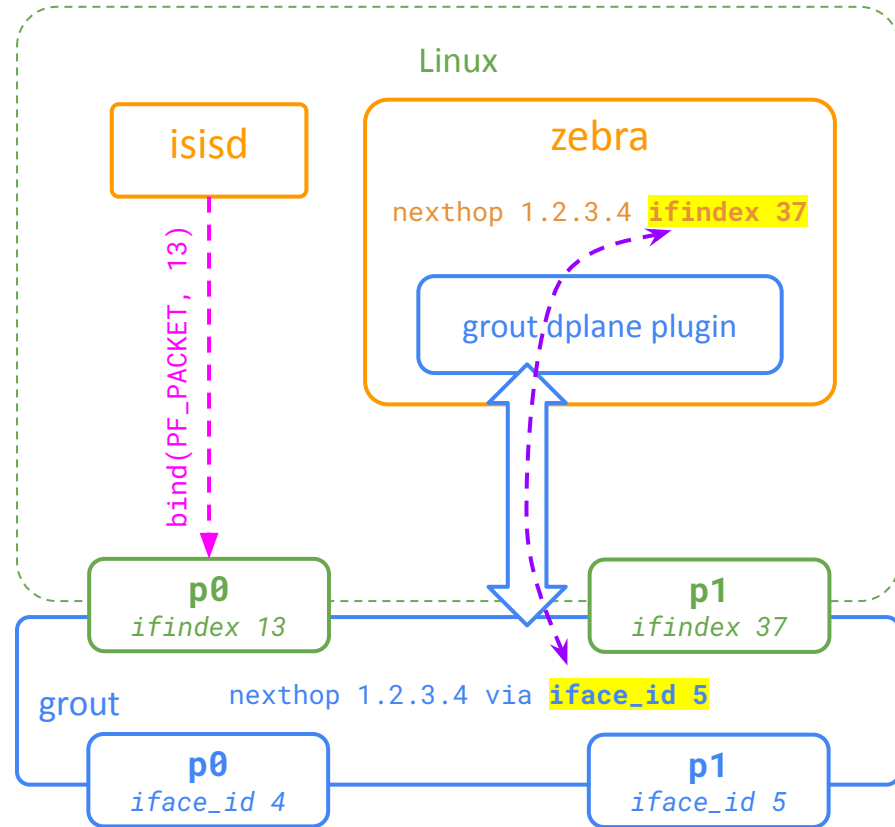
- FRR relies on Linux `l3mdev` (VRF interfaces)
- **Non-default** VRFs create a **real Linux VRF device** via netlink
- Grout TUN/TAP interfaces enslaved to the VRF device
- **IP addresses mirrored** onto each iface's own kernel representor (TAP) so daemons can `bind()` them per VRF
- FRR daemons see proper VRFs and bind sockets **per VRF**



```
grcli interface add vrf red
grcli interface set port p1 vrf red
grcli interface set port p2 vrf red
grcli interface add vrf blue
grcli interface set port p3 vrf blue
```

# The ifindex problem

- zebra stores the **kernel ifindex** of the TAP representor
- isisd needs to `bind(PF_PACKET, ifindex)` to a real Linux iface
- Grout has **its own native interface IDs**
- The `dplane` plugin maintains bi-directional translation tables



# FRR + grout performance

Route insertion rate.

Xeon(R) Silver 4316 CPU @ 2.30GHz -- FRR [v10.5.3](#) -- grout [v0.15](#)

Profile	Num. routes	Time	Rate
<b>IPv4 standalone</b> <small>(2048 nexthops)</small>	1,000,000	20.3s	49.3K routes/s
<b>IPv6 standalone</b> <small>(2048 nexthops)</small>	200,000	3.7s	54.6K routes/s
<b>IPv4 <u>FRR + GoBGP sync</u></b> <small>(1 nexthop)</small>	544,674	5m31s	1.65K routes/s

# Currently supported FRR features

- **BGP / OSPF / IS-IS:** full adjacency and route exchange
- **L3VPN / SRv6:** per-VRF routing with BGP, local SIDs, encapsulation, compressed SIDs
- **EVPN / VXLAN:** MAC/IP learning, FDB sync (BGP type 2, 3 and 5 routes)
- **Restart recovery:** keep traffic forwarding during zebra restart, and automatically restore routes after grout restart.

# Lessons learned

1. **FRR** presumes Linux everywhere
2. `dpplane` was originally designed for offload, not kernel bypass
3. TAP/TUN representors keep the kernel out of forwarding
4. Certain protocols (IS-IS, OSPF) need special care

# Demo



DPDK  
SUMMIT  
May 12-13 2026 | Stockholm, Sweden

## Integrating FRR with Grout

Robin Jarry, Red Hat  
Maxime Leroy, Free Mobile



**Maxime Leroy,**  
**Free Mobile**



# DPDK

SUMMIT

May 12-13 2026 | Stockholm, Sweden

## Thank You / Questions

<https://github.com/DPDK/grout>

[grout@dpdk.org](mailto:grout@dpdk.org)

[#grout @ dpdkproject.slack.com](#)