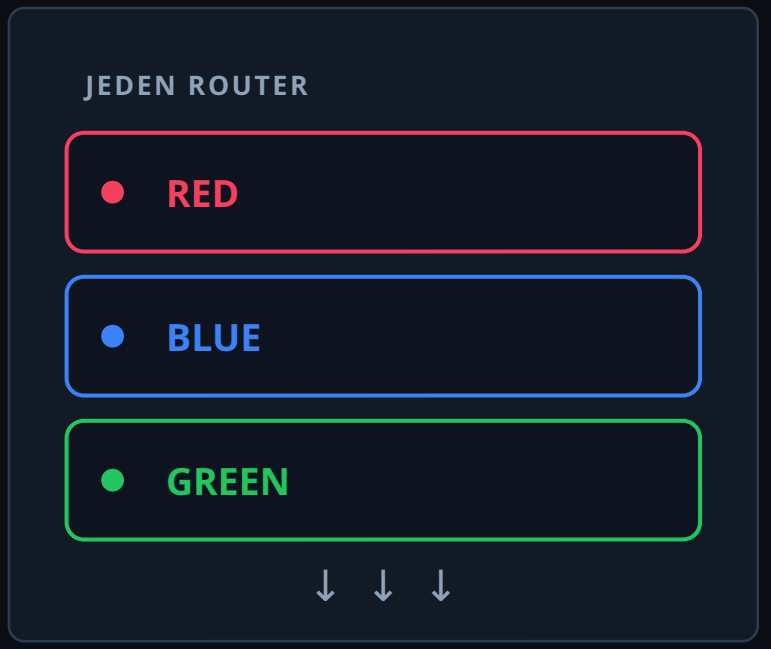
Jeden MikroTik, wiele routerów.

VRF od RIB-u do ASIC-a **w RouterOS 7.24**



jeden RouterBOARD, wiele niezależnych światów routingu

Jacek Rokicki

1–2 października 2026

MBUM 2026 — VRF w RouterOS

Jeden router. Trzy identyczne sieci.

- **RED** `10.0.0.0/24`
- **BLUE** `10.0.0.0/24`
- **GREEN** `10.0.0.0/24`

każdy tenant ma hosta 10.0.0.10

Który 10.0.0.10?

VLAN rozwiązuje separację L2. Nie rozwiązuje problemu wielu niezależnych przestrzeni routingu.

Jedna tablica. Jedna prawda.

KLASYCZNY ROUTING

Destination	Next hop
0.0.0.0/0	203.0.113.1
10.0.0.0/24	— ??? —
172.31.0.0/24	bridge-shared

10.0.0.0/24 może istnieć tylko raz w pojedynczym FIB-ie.

→

ROZWIĄZANIE: WIĘCEJ TABLIC

- RED FIB
- BLUE FIB
- GREEN FIB

VRF zmienia nie tylko trasę. Zmienia tablicę, w której trasy szukamy.

VLAN ≠ VRF

VLAN — warstwa 2

- izolacja L2
- osobna domena broadcast
- wspólny kontekst routingu

VRF — warstwa 3

- izolacja L3
- osobny RIB / FIB
- nakładające się adresacje
- niezależne tablice routingu

VLAN mówi „inny segment L2”. VRF mówi „inny świat routingu”. To dwie różne osie separacji — często używane razem, nigdy zamienne.

VRF w RouterOS — minimalna konfiguracja

```
/ip vrf
add name=RED   interfaces=vlan-red
add name=BLUE  interfaces=vlan-blue
add name=GREEN interfaces=vlan-green
```



Osobna tablica routingu dla każdego VRF.

Programowy limit: do 1024 (512) VRF.

Kolejność wpisów /ip vrf ma znaczenie.

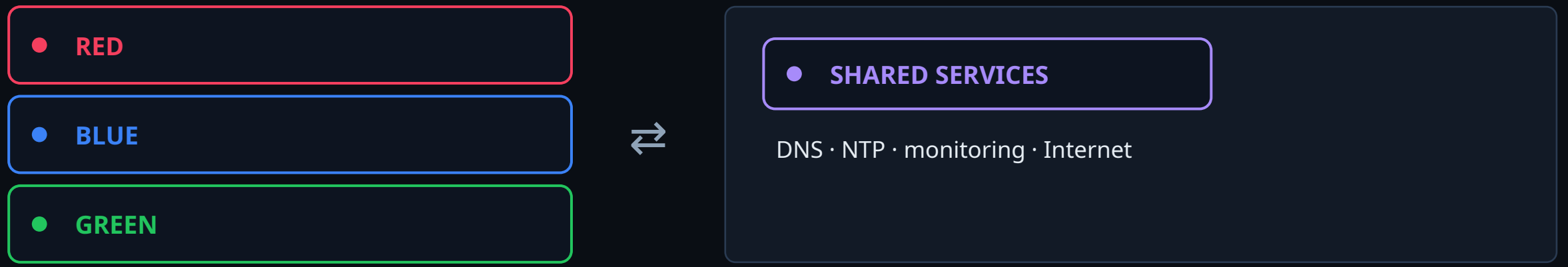
Który 10.0.0.10? Teraz już wiemy.

```
# ten sam adres, trzy różne konteksty routingu
ping 10.0.0.10 vrf=RED
ping 10.0.0.10 vrf=BLUE
ping 10.0.0.10 vrf=GREEN
```



Aktualna składnia RouterOS 7.x: parametr vrf= w ping/traceroute/telnet — nie notacja @VRF.

Odizolować jest łatwo.
Trudniej kontrolowanie połączyć.



Jak udostępnić wspólne usługi bez zniszczenia separacji?

Route leaking: model statyczny

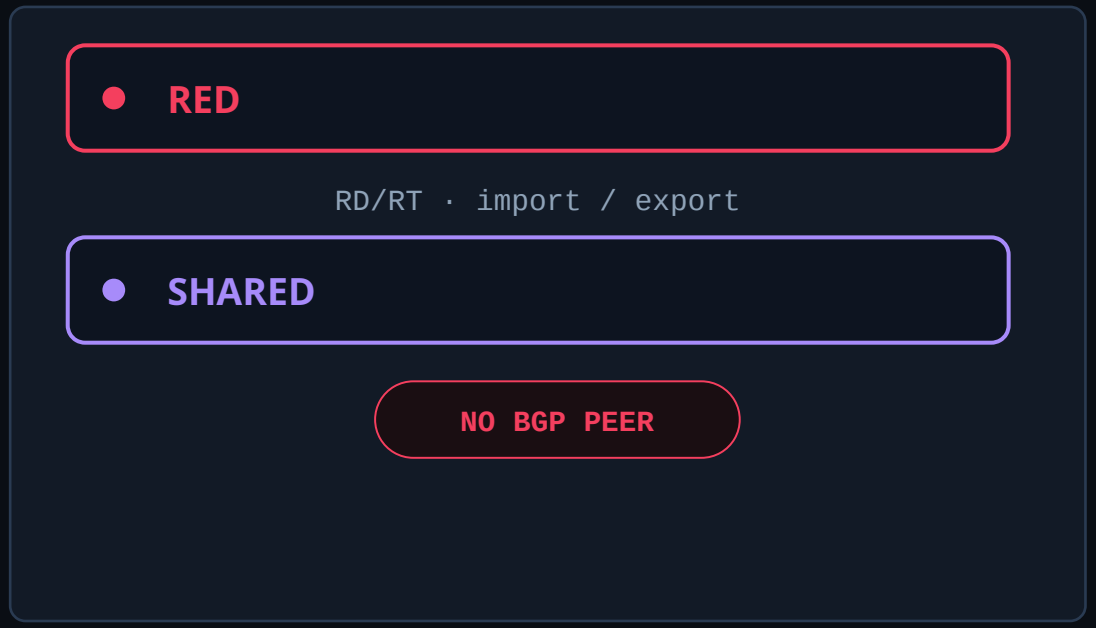
```
/ip route add
  dst-address=172.31.0.0/24
  gateway=10.0.0.1@main
  routing-table=RED
```



Next-hop również ma swój kontekst routingu.

Dwa modele: (1) statyczny leaking jak wyżej, (2) dynamiczny — BGP VPN import/export.

BGP route leaking bez BGP neighbora



- route target — kto ma prawo zobaczyć trasę
- import / export — reguły wymiany
- lokalna wymiana tras — bez sesji BGP między VRF-ami

RouterOS używa lokalnego BGP VPN jako policy engine do wymiany tras między VRF-ami — bez zestawiania sesji BGP.

Czy to naprawdę są osobne routery?

Routing table	TAK
Overlapping IP	TAK
Route leaking	TAK
BGP	TAK
DNS	?
SSH / WinBox	?
Firewall	?
Hardware	?

Routing mamy. Ale routing to tylko jeden z podsystemów RouterOS. Czas sprawdzić resztę.

VRF-aware services — nie raz, a stopniowo

7.14 VRF interface w firewallu	**7.15** DNS + DHCP Relay (vrf=)	**7.19** Remote Logging
7.22 Reverse Proxy	**7.24** Bandwidth Test / Speed Test	**7.25beta3** TFTP

DNS · SSH · WinBox · NTP · SNMP · RADIUS · WireGuard · Proxy — reprezentatywne przykłady, nie pełna lista.

VRF nie pojawia się w systemie raz. Każda usługa musi zostać osobno nauczona działania w konkretnym kontekście routingu.

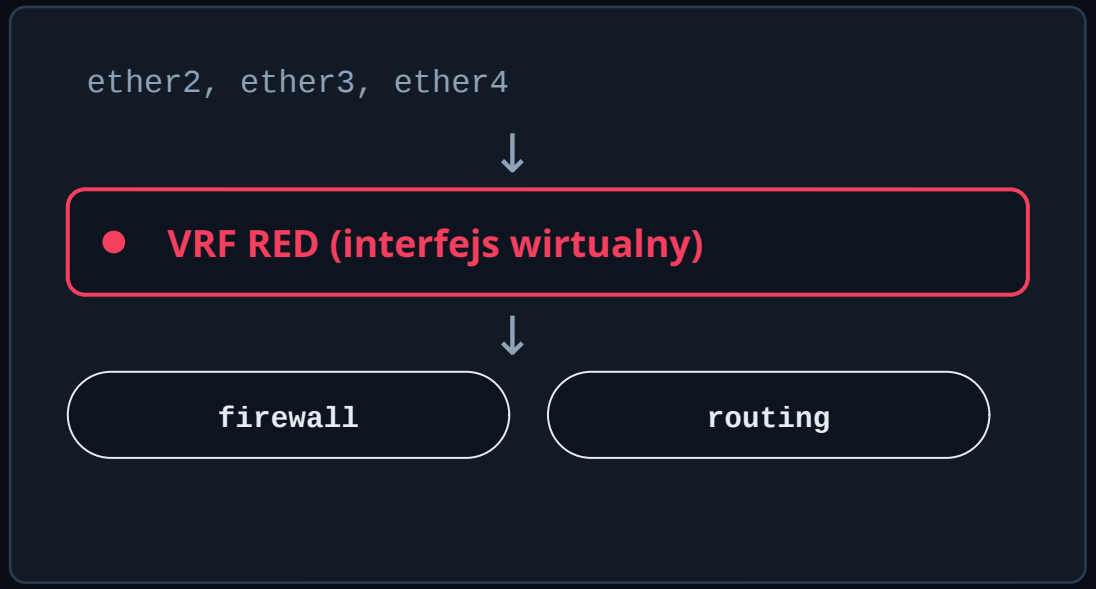
Management VRF




```
/ip service
set ssh    vrf=MGMT
set winbox vrf=MGMT
set api    vrf=MGMT
```

VRF nie jest tylko dla klientów. Management plane też może żyć w osobnym świecie.

Firewall a VRF interface




```
/ip firewall filter
add chain=forward
    in-interface=RED
    action=accept
```

Od RouterOS 7.14: firewall widzi logiczny interfejs VRF, nie pojedyncze porty fizyczne w środku.

Jeśli trzeba rozróżnić konkretne porty w obrębie jednego VRF — potrzebne inne mechanizmy, np. connection marks.

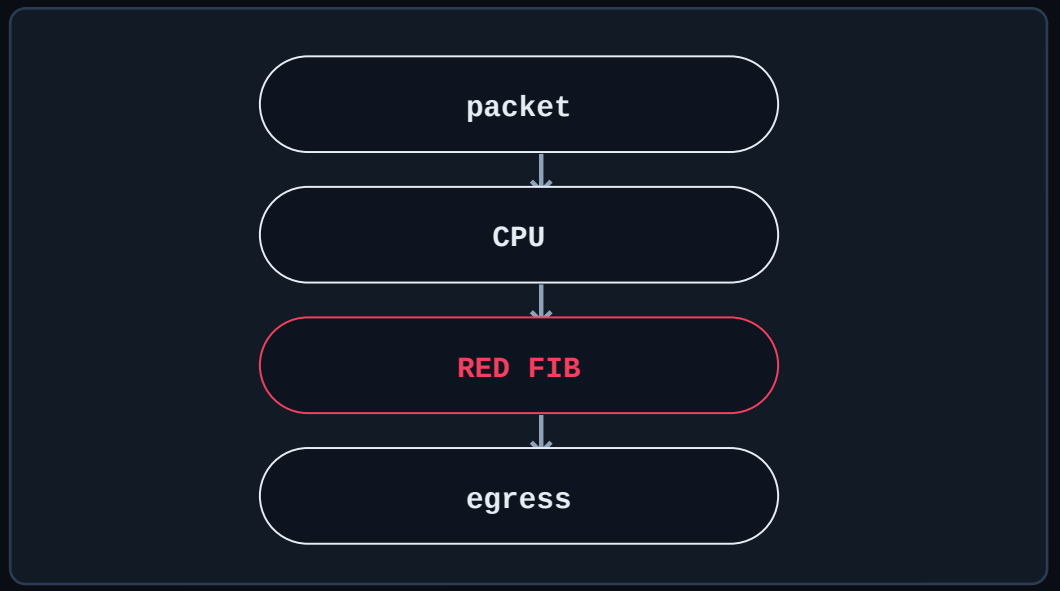
FastTrack

	main	non-main VRF
L3 routing	TAK	TAK
FastTrack	TAK	NIE
NAT HW offload	TAK	NIE
VXLAN HW offload	TAK	NIE

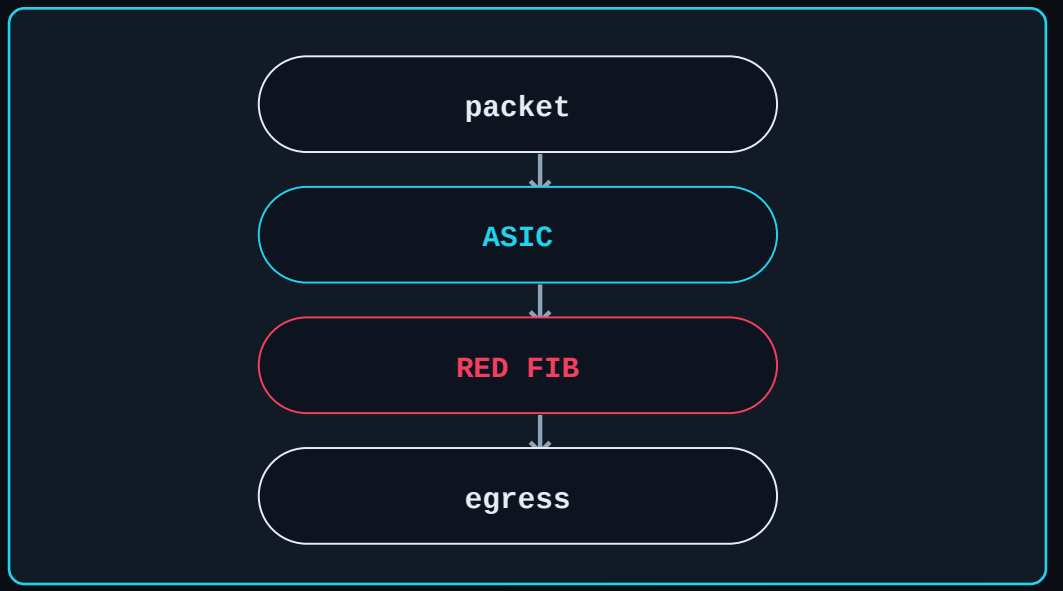
VRF działa. Ale cały RouterOS data plane nie jest jeszcze VRF-aware.

Wg dokumentacji L3 Hardware Offloading: FastTrack, NAT i VXLAN offload są ograniczone do głównej tablicy routingu (main).

Od CPU do ASIC-a



A co przy 10 / 25 / 100 Gb/s?



VRF trafia do hardware

7.23 `stable · 2026-05-25`

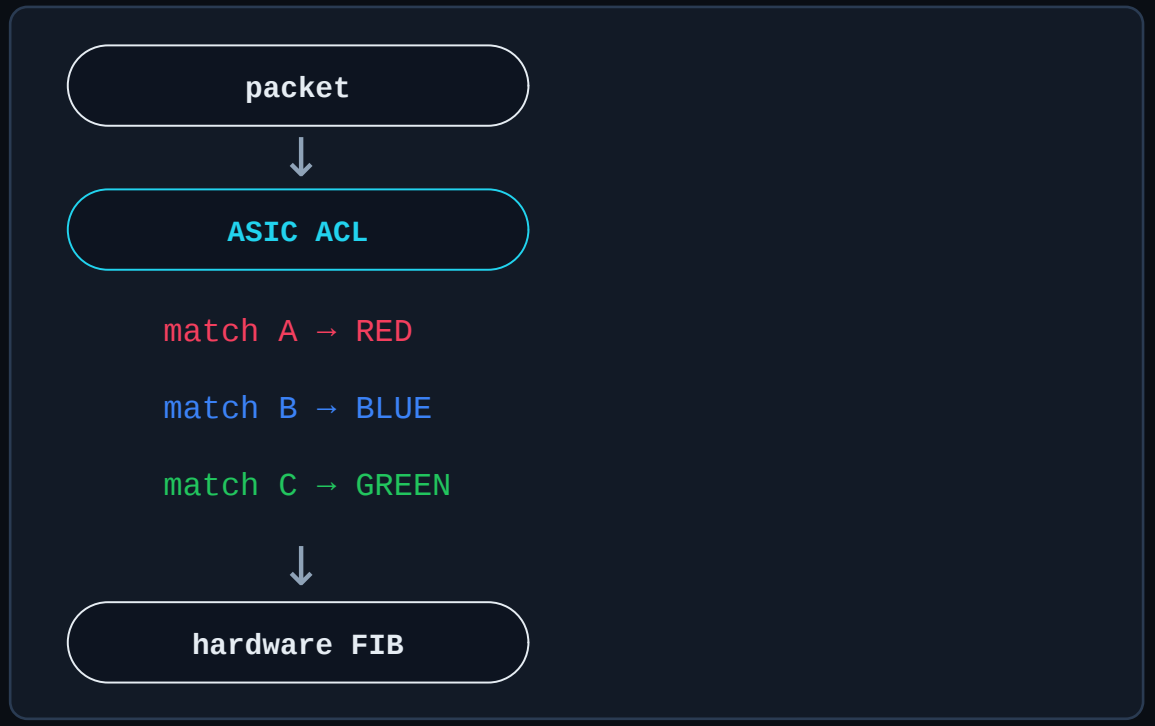
HW-offloaded VRF na **CRS8xx** · new-vrf w switch ACL (tylko CLI)

7.24 `stable · 2026-08-14`

HW VRF rozszerzone na rodzinę **Marvell Prestera**: 98DX8208 / 8212 / 8216 / 8332, **98DX3257 (CRS354)**, 98DX3255, 98DX4310, 98DX8525, 98CX8410

Nie każdy Prestera wspiera HW VRF — starsze 98DX2xxx / 98DX3xxx (np. CRS3xx niższej klasy) nie mają tego mechanizmu. Sprzętowy limit: 512 lub 1024 VRF, zależnie od konkretnego układu (98DX3257/CRS354 → 512).

`new-vrf`: decyzja o świecie routingu już w ASIC-u




```
/interface/ethernet/switch/rule
add switch=switch1 ports=ether5
    new-vrf=RED
```

Od 7.25beta3: liczniki packets / bytes na regułach switch ACL na urządzeniach Marvell Prestera — widać, że klasyfikacja realnie dzieje się w ASIC-u.

Nie tylko forwarding może być hardware-offloaded. Już ingress classifier może przypisać pakiet do konkretnego VRF.

To jeszcze nie koniec

7.23 HW VRF na CRS8xx · ACL new-vrf (CLI)

7.24 HW VRF na szerszej rodzinie Prestera · więcej narzędzi VRF-aware

7.25beta3 (obecnie 7.25beta5 · 30.09.2026 · dev channel)

- `tftp — add VRF support`
- `bgp — fix BGP unnumbered connections within a VRF`
- `ip-service — fix services unreachable after VRF disable/enable`
- `switch — packet/byte counters for ACL rules na Prestera`
- `ipsec — add XFRM interface support (nowość, jeszcze nie funkcja VRF)`

VRF nie jest skończonym feature'em. RouterOS nadal uczy kolejne subsystemy życia w wielu kontekstach routingu.

Czy to już naprawdę osobny router?

Routing table	✓
Overlapping IP	✓
Route leaking	✓
BGP	✓
Management	✓
DNS / services	✓
Kontekst firewalla	✓
Hardware FIB	✓
Everything?	x

Z każdą wersją RouterOS ten ostatni krzyżyk robi się trochę mniejszy.

Trzy rzeczy do zapamiętania

1. VRF to kontekst routingu, nie „lepszy VLAN”.
2. Dojrzałość VRF mierzy się tym, ile subsystemów naprawdę rozumie ten kontekst.
3. RouterOS 7.24 przenosi VRF z RIB/FIB w CPU do hardware data plane.

Jedna platforma. Wiele niezależnych światów routingu.

Dziękuje za uwagę.