An Introduction to Netkit

The BPF Programmable Network Device

AGENDA

01 **BACKGROUND**

About Me
Overview of Network Namespaces
Veth Performance Implications

02 **NETKIT DEVICE**

Performance Benefits
Development Benefits
Real World Results

03 **CLOSING**

Shout Outs
Questions/Feedback

Mike Willard

Software Engineer, Meta

About Me

- Software Engineer at Meta, Containers team
- Live in Brooklyn, NYC
- Previously worked at iZotope, XWing, and Honeywell
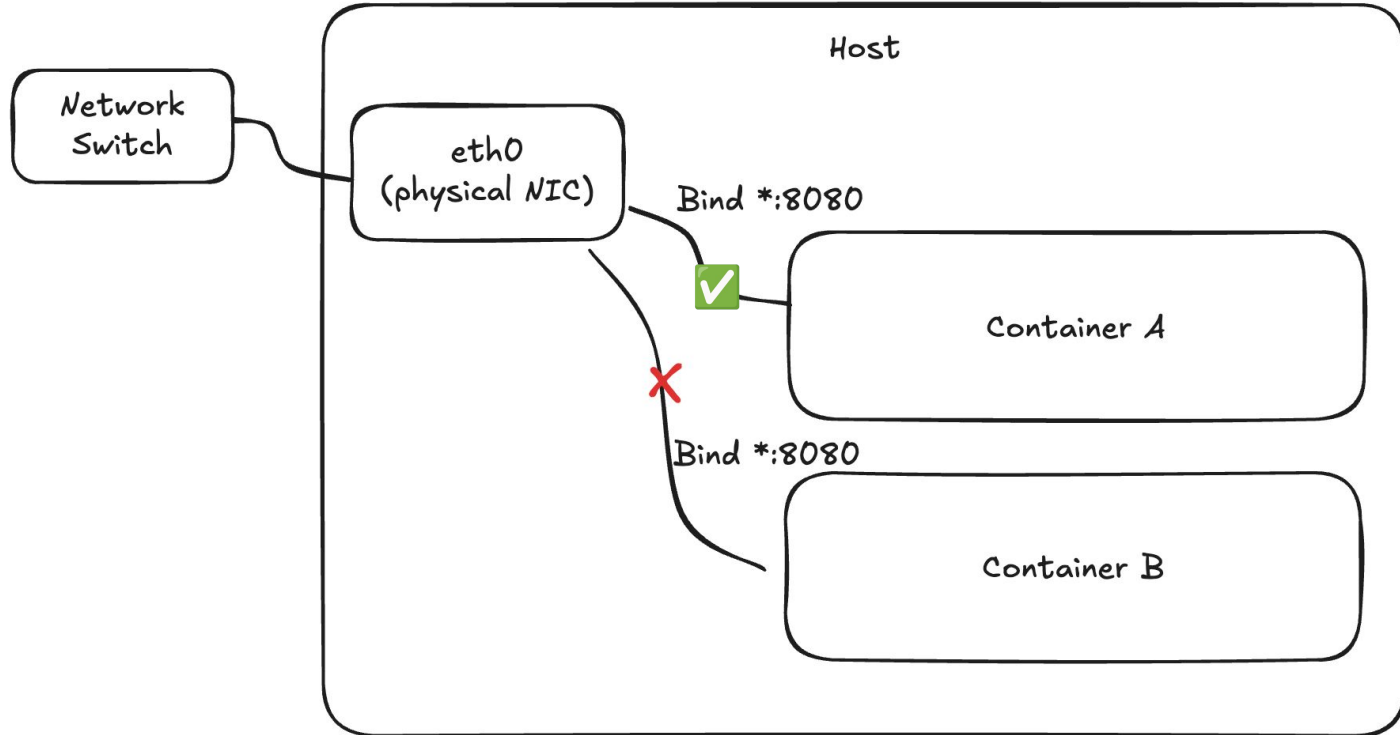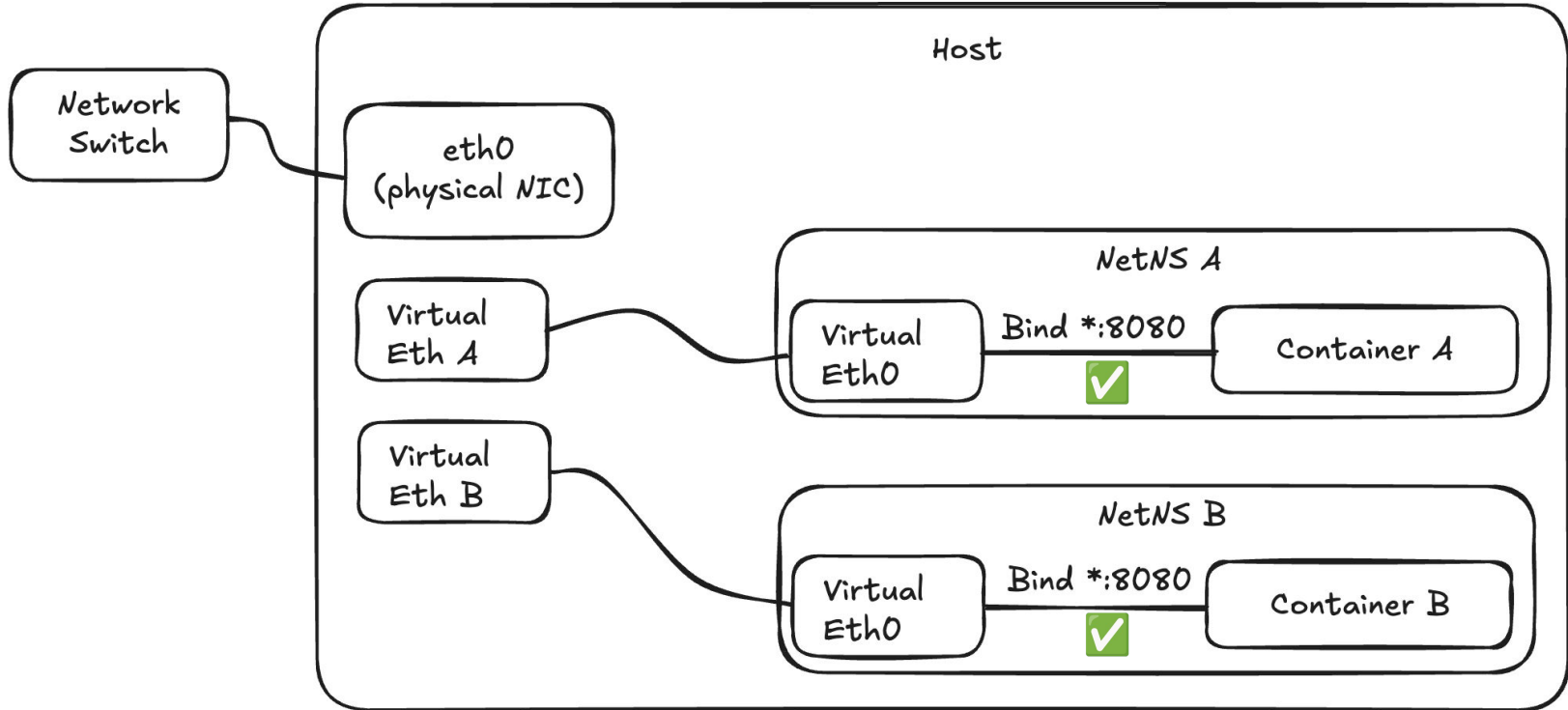- Hobbies: Rally codriving, snowboarding

01

Background

WITHOUT NETWORK NAMESPACES
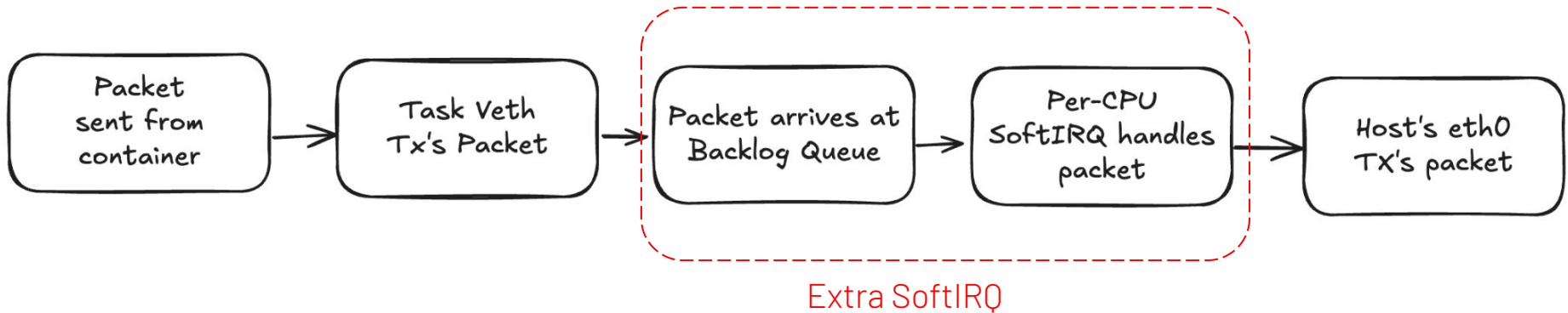
Containers share a single network environment

WITH NETWORK NAMESPACES

Each container has an isolated network environment

VETH PERFORMANCE IMPLICATIONS

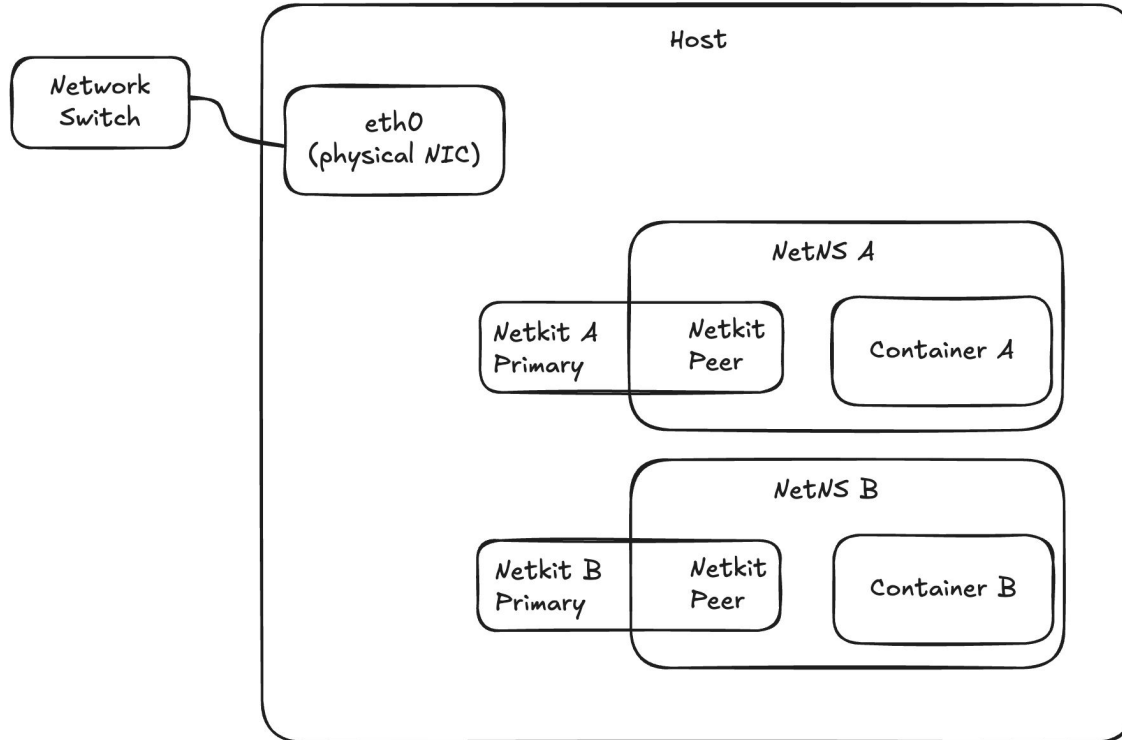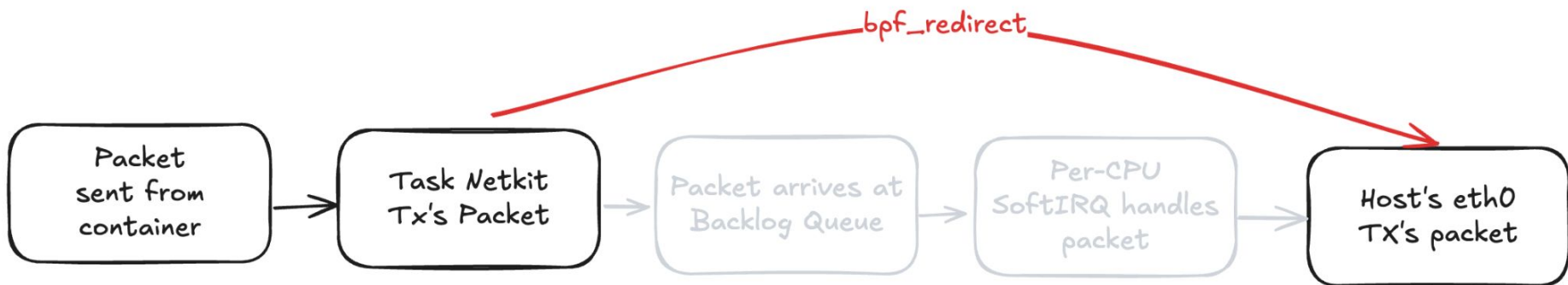Network isolation with Veth results in extra SoftIRQ usage

Extra SoftIRQ

02

The Netkit Device

THE NETKIT DEVICE
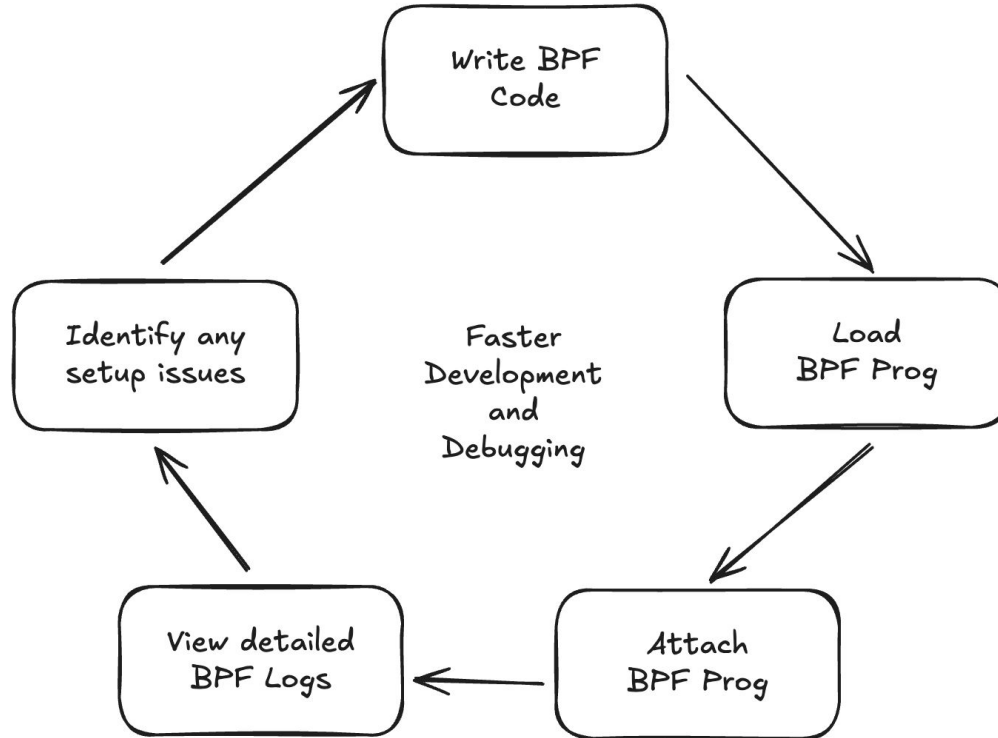
Netkit uses BPF to manage container←→host traffic

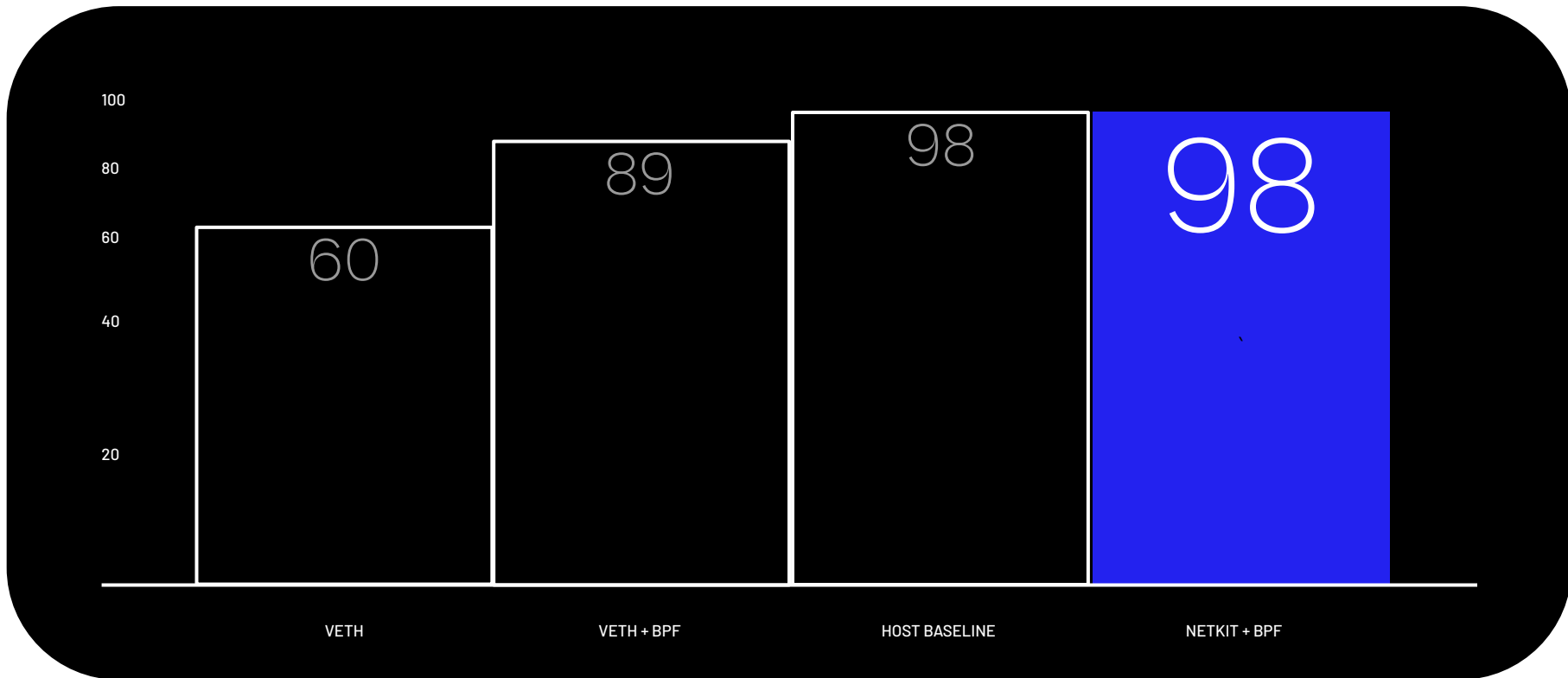# NETKIT BPF BENEFITS

Additional SoftIRQ Usage is Bypassed

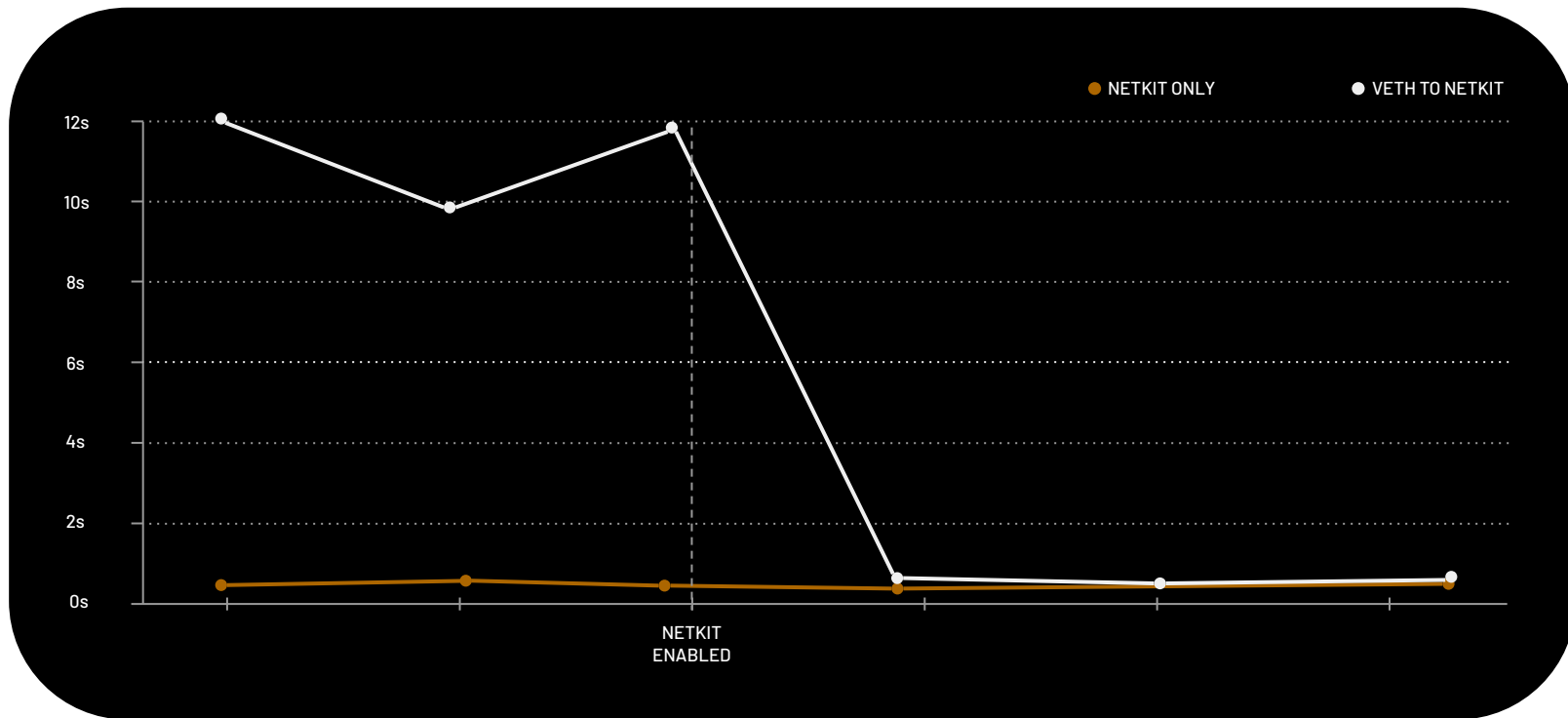# NETKIT BPF BENEFITS

Faster development and debugging

VETH VS NETKIT THROUGHPUT (MBPS)

iPerf testing shows Netkit performance can match baseline throughput

Source: https://isovalent.com/blog/post/cilium-netkit-a-new-container-networking-paradigm-for-the-ai-era/

P99 Latency, 4x Stacking

On a task with 4x copies running on one host, P99 latency was >12s due to high SoftIRQ usage. After enabling Netkit, latency dropped to ~100ms.

03

Closing

Shout Outs

- This work would not have been possible without the open source community and help from others working on this project
- Meta: Takshak Chahande, Martin Lau
- Isovalent: Daniel Borkmann, Nikolay Aleksandrov, Nico Vibert, Bill Mulligan

Conclusions

- Network namespaces are key for providing isolation between collocated containers
- With Netkit, network namespaces can be used without incurring a performance penalty
- BPF programs allow for greater flexibility and ease of development

"To play a wrong note is insignificant, to play without passion is inexcusable"

Ludwig Van Beethoven