LinuxFP: Transparently Accelerating Linux Networking

Marcelo Abranches*, **Erika Hunhoff***, Rohan Eswara*, Oliver Michel[+], Eric Keller*

*University of Colorado Boulder

[+]Princeton

University of Colorado Boulder

ICDCS 2024

Jersey City, New Jersey USA

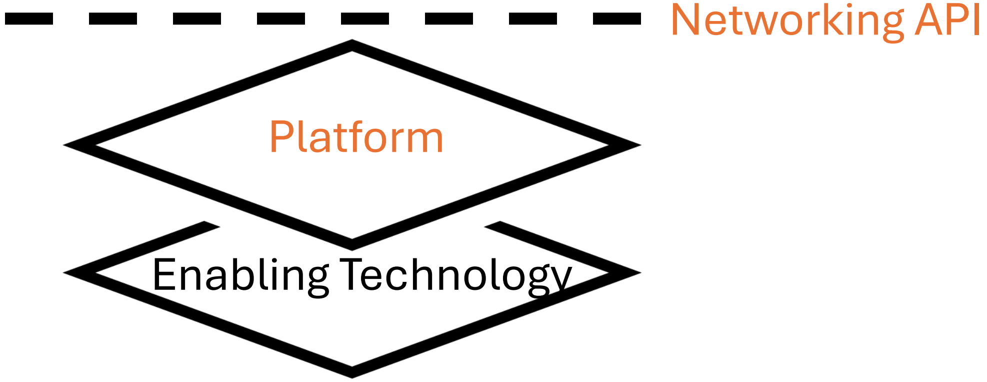
PRINCETON UNIVERSITY

Software Based Packet Processing

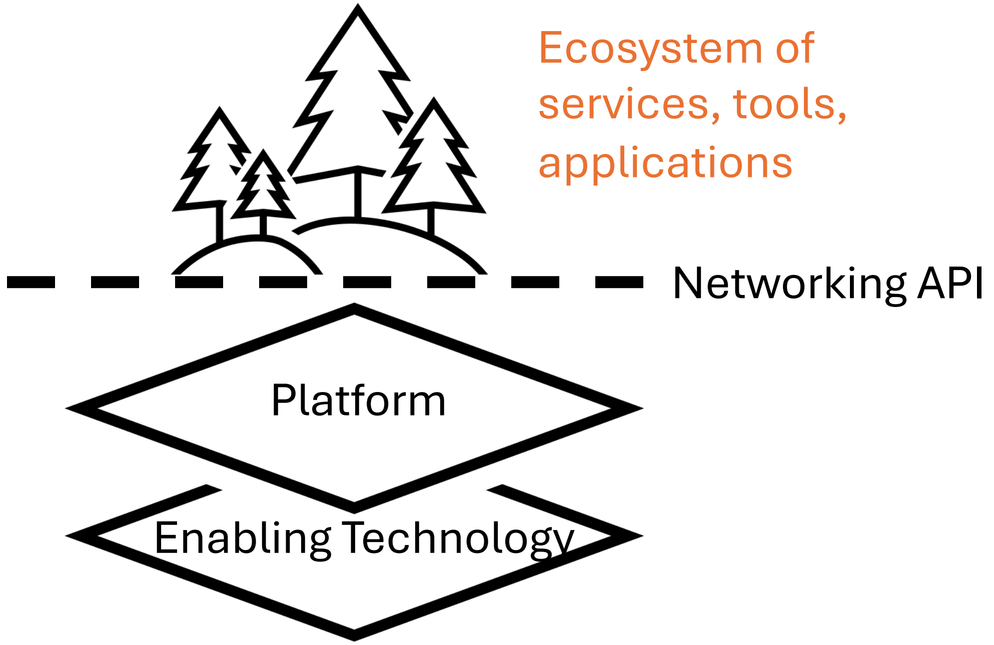
Software Based Packet Processing

Enabling Technology

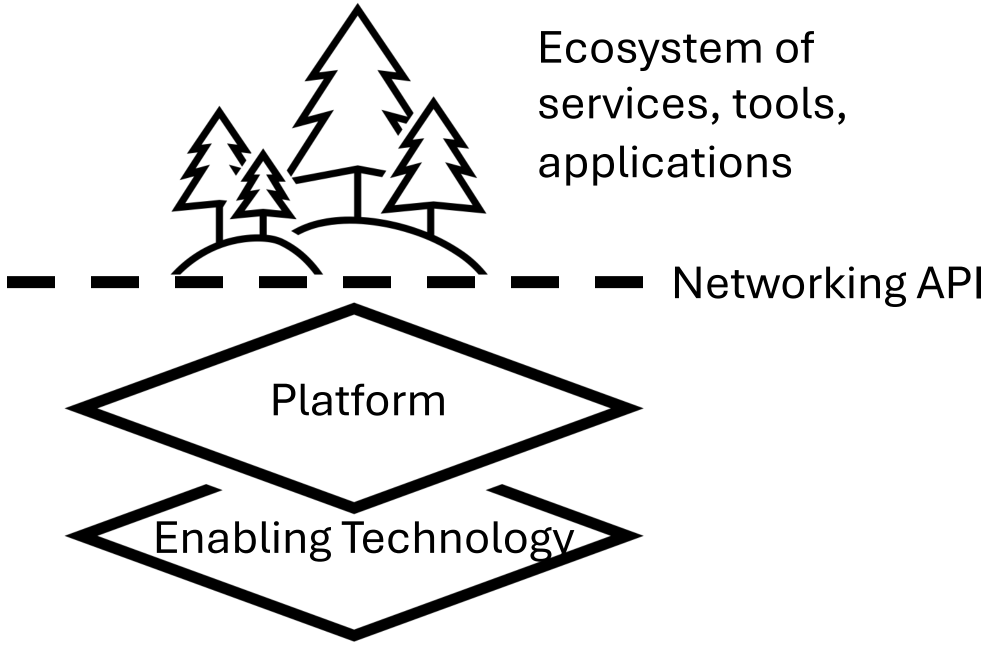
Software Based Packet Processing



Software Based Packet Processing



Software Based Packet Processing **Examples**



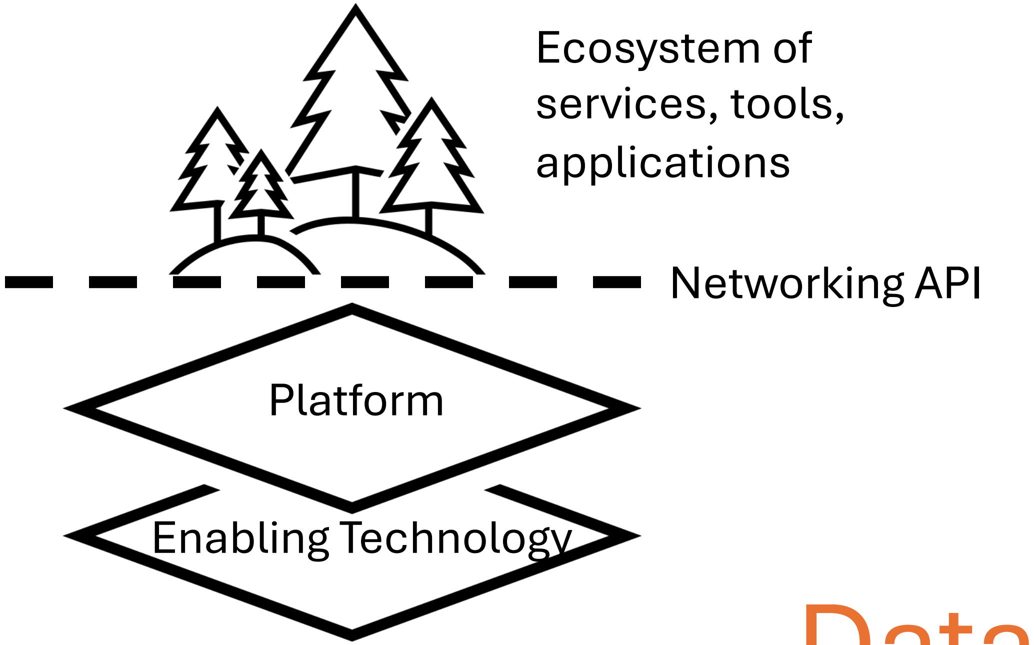
Ecosystem of services, tools, applications

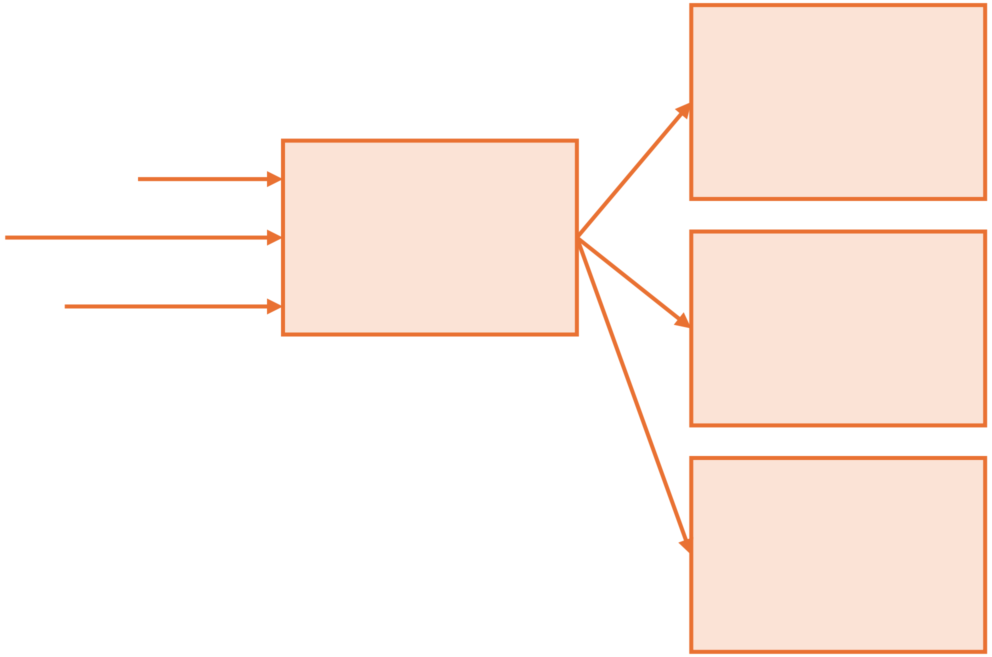
Networking API

Platform

Enabling Technology

Software Based Packet Processing **Examples**

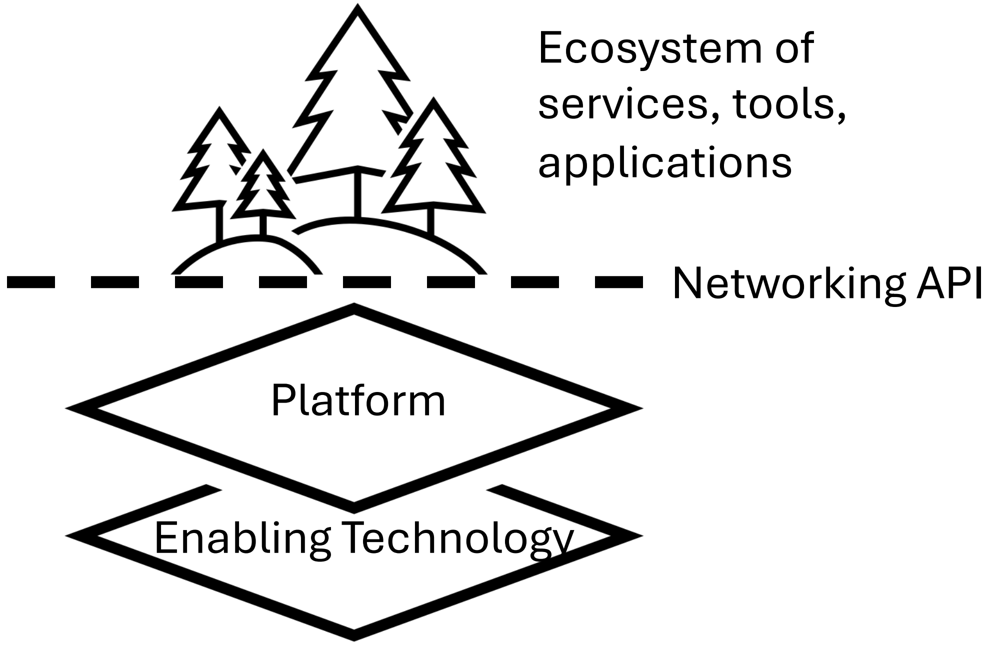




Data Center Load Balancing

University of Colorado **Boulder** PRINCETON UNIVERSITY

Software Based Packet Processing Examples

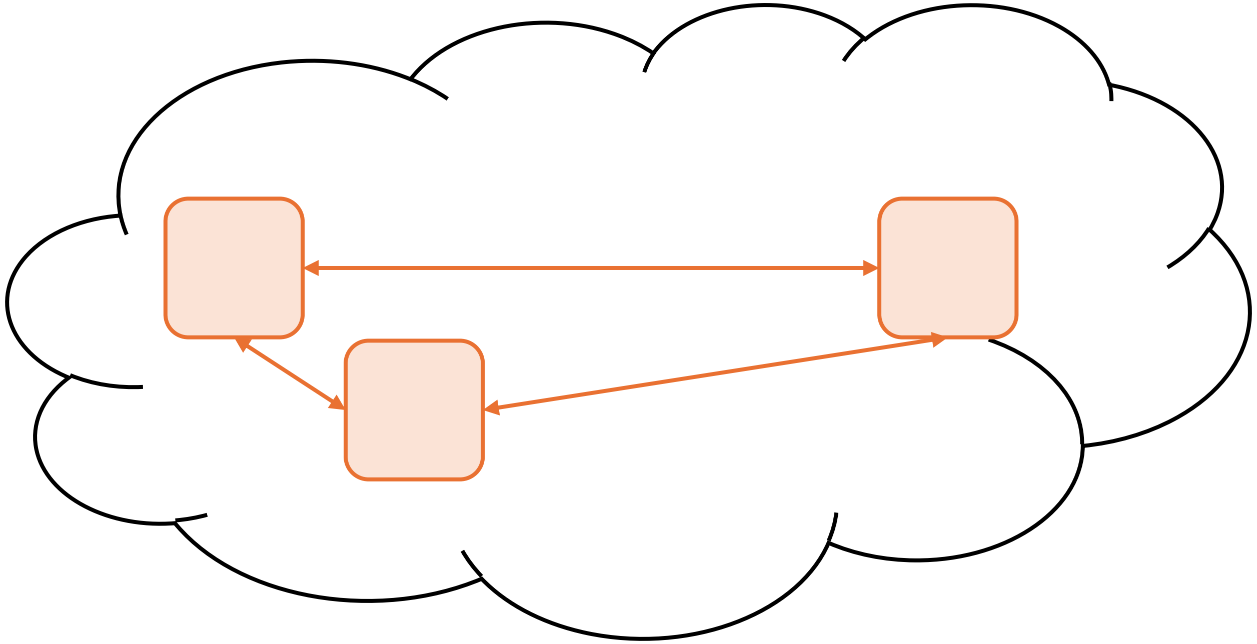


Ecosystem of services, tools, applications

Networking API

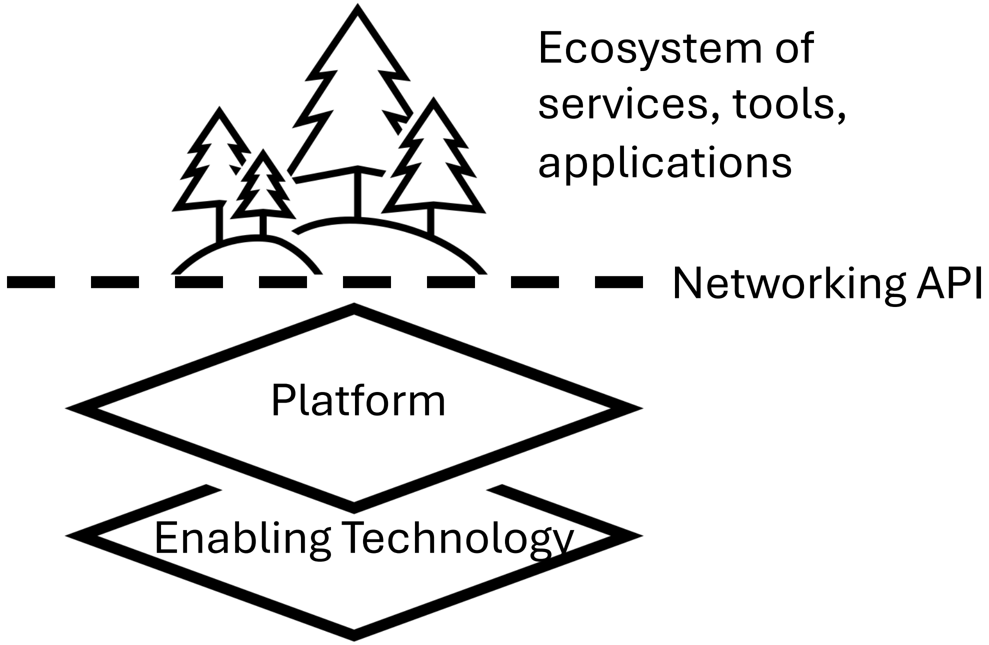
Platform

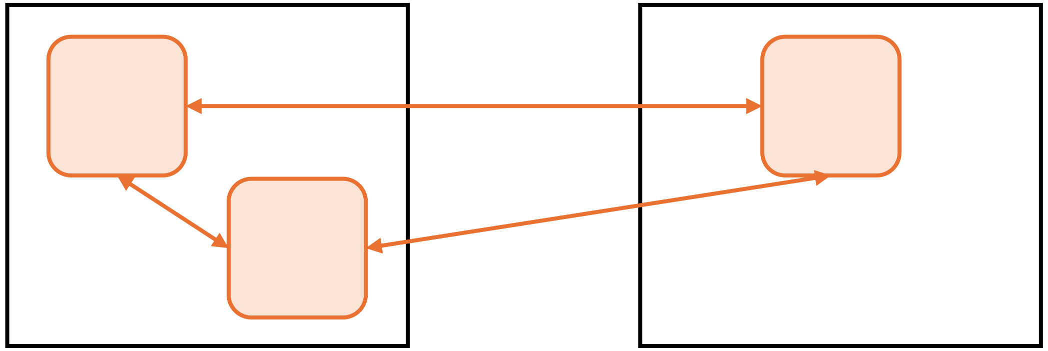
Enabling Technology



Cloud Overlay Networks

Software Based Packet Processing **Examples**



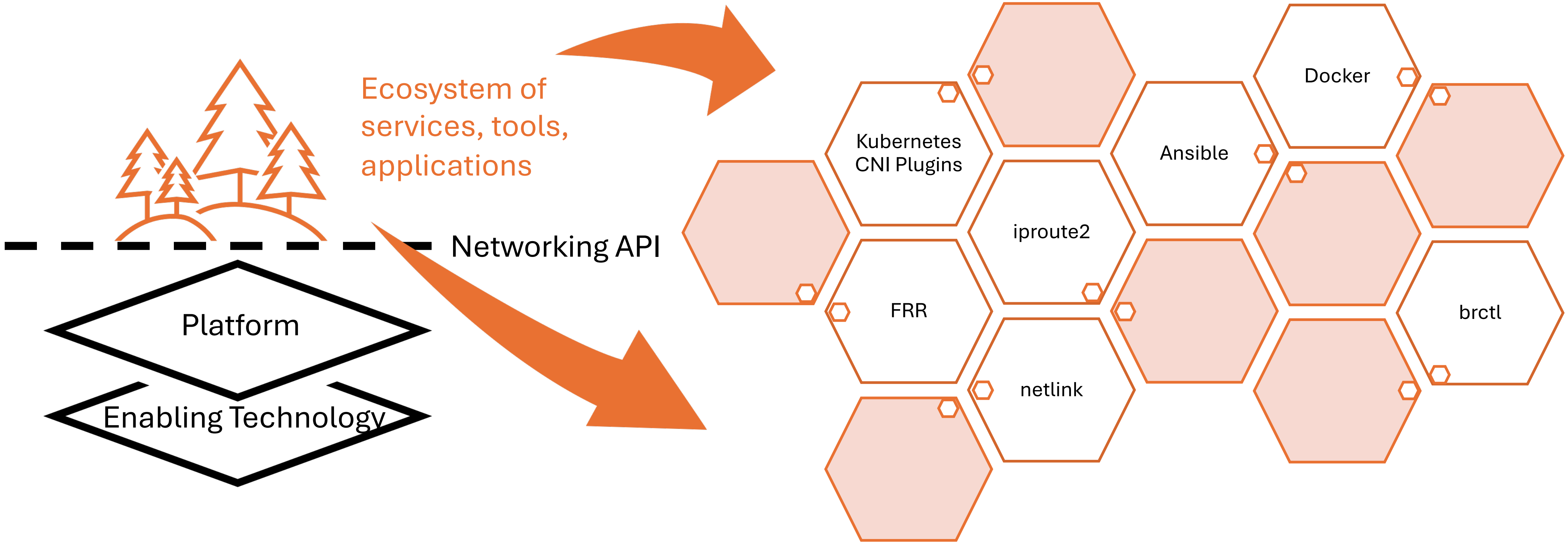


Virtual Networking Between Containers

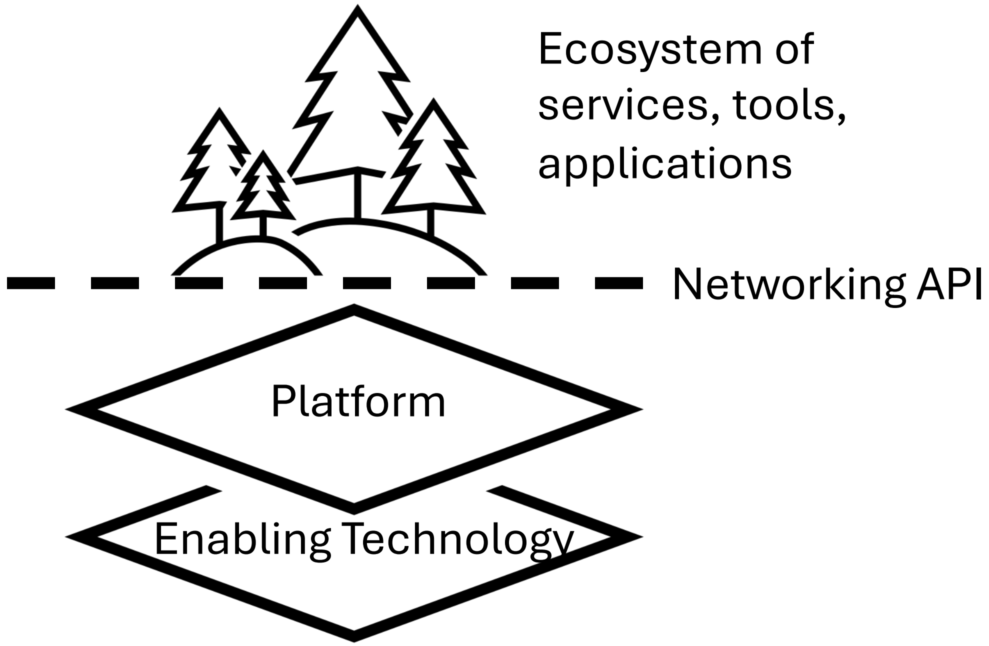
University of Colorado **Boulder**

PRINCETON UNIVERSITY

Software Based Packet Processing **in Linux**



Fast Software Based Packet Processing



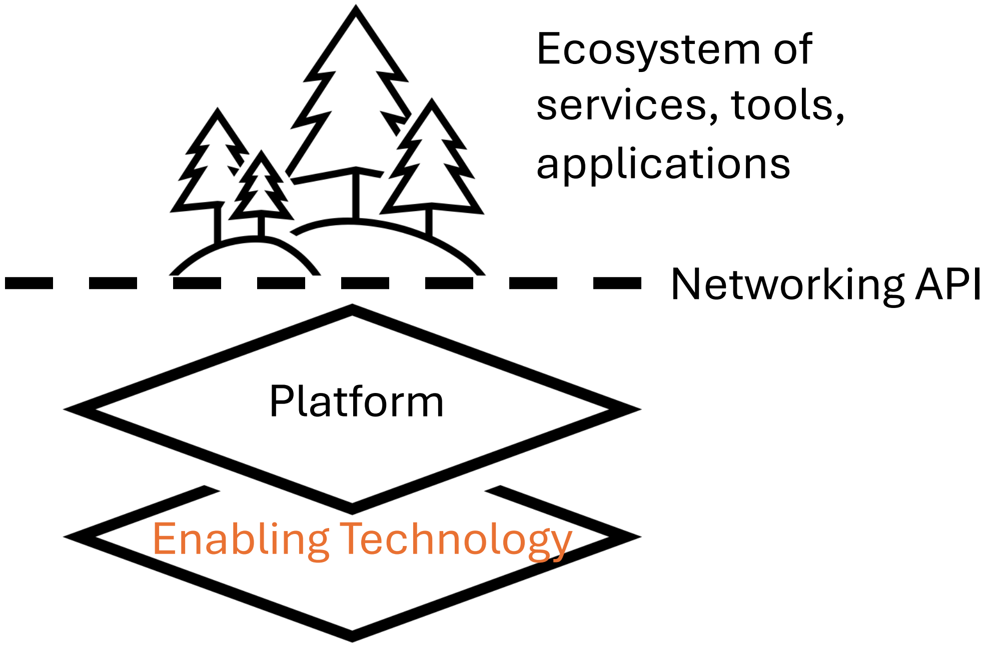
Ecosystem of services, tools, applications

Networking API

Platform

Enabling Technology

Fast Software Based Packet Processing

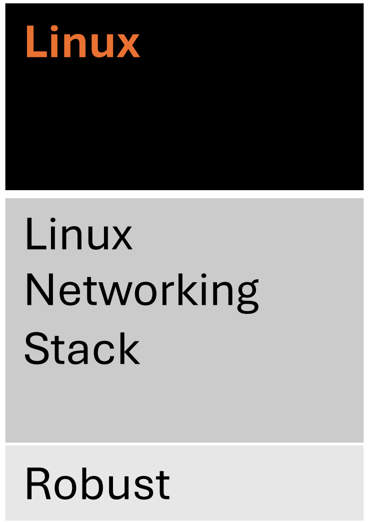


Ecosystem of services, tools, applications

Networking API

Platform

Enabling Technology

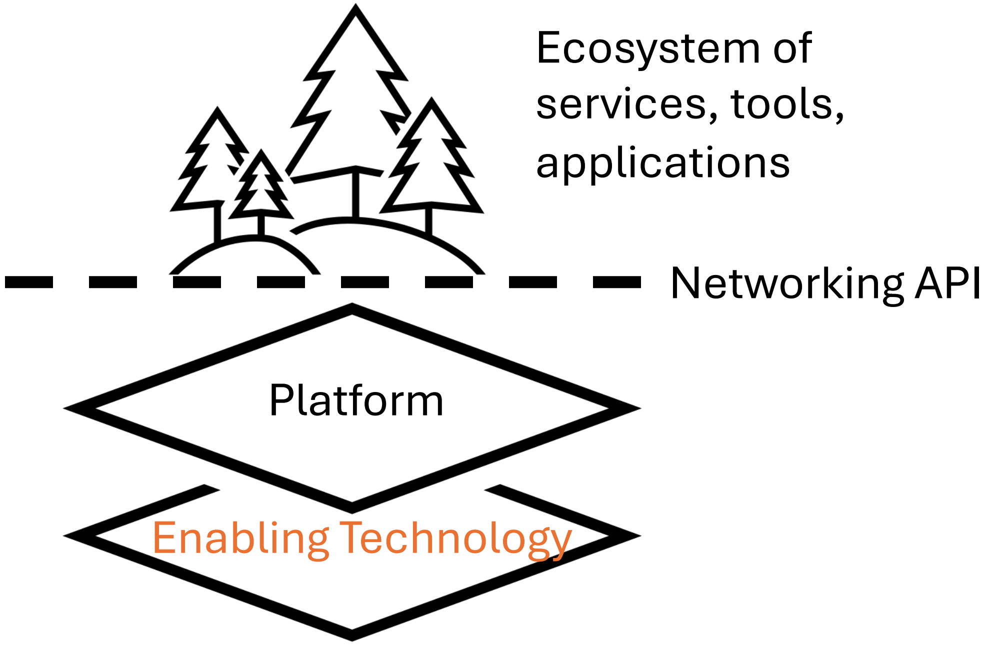


Linux

Linux Networking Stack

Robust

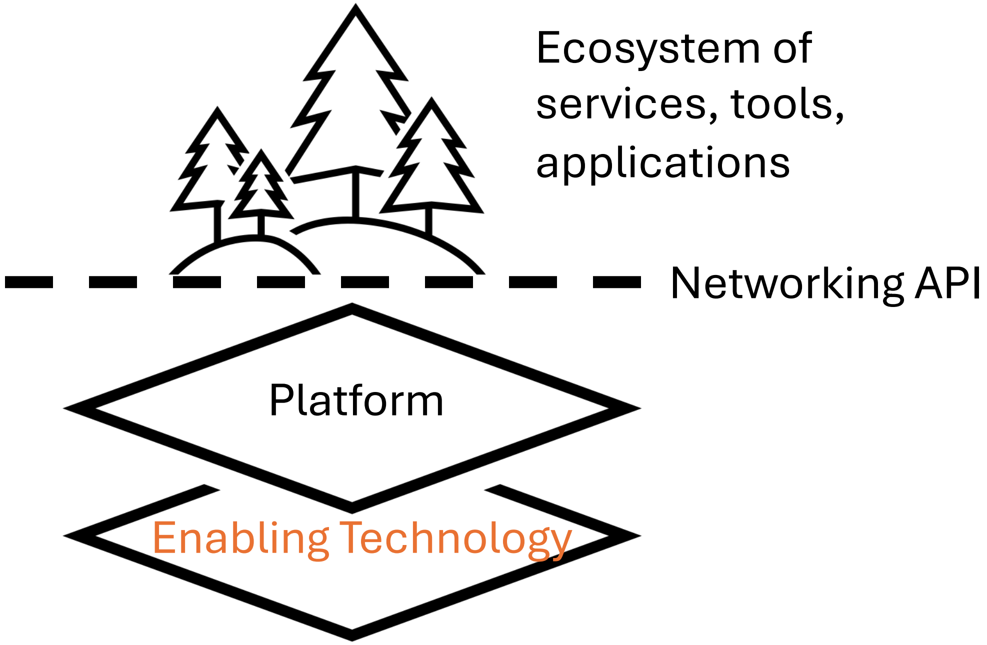
Fast Software Based Packet Processing



Linux	In-Kernel, network stack bypass
Linux Networking Stack	eBPF/XDP (eXpress Data Path) [0]
Robust	Specific

[0] THøiland-Jørgensen, *et. al*. The eXpress Data Path: Fast programmable packet processing in the operating system kernel. In *ACM CoNEXT*, 2018.

Fast Software Based Packet Processing

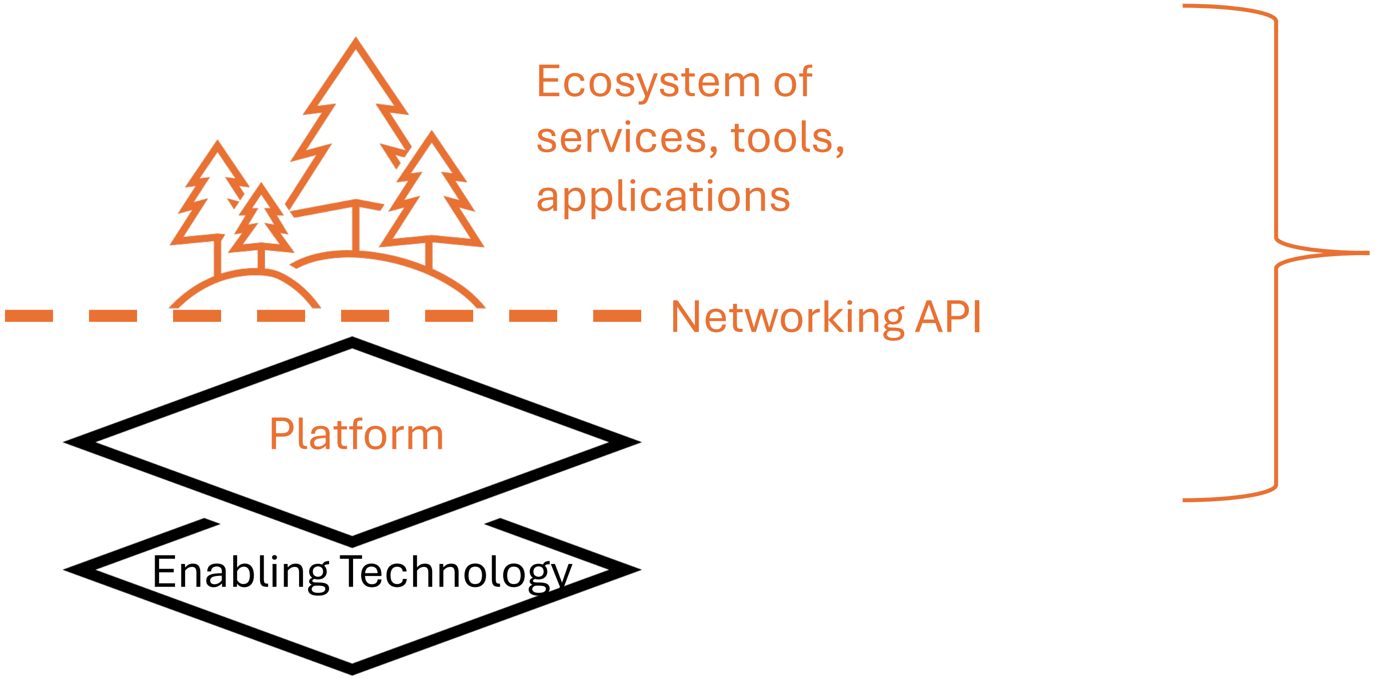


Linux	In-Kernel, network stack bypass	Kernel Bypass
Linux Networking Stack	eBPF/XDP (eXpress Data Path) [0]	DPDK (Dataplane Development Kit [1]
Robust	Specific	Specific

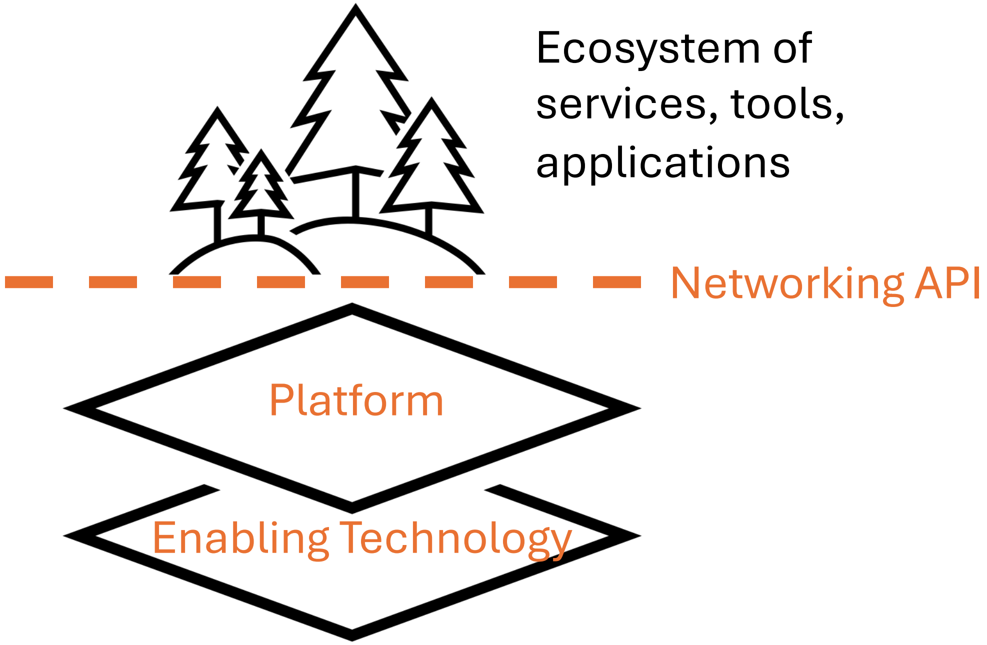
[0] THøiland-Jørgensen, *et. al*. The eXpress Data Path: Fast programmable packet processing in the operating system kernel. In *ACM CoNEXT*, 2018.

[1] DPDK Project. Data plane development kit, 2022. Retrieved June 13, 2022, from https://www.dpdk.org.

Fast Software Based Packet Processing



Fast Software Based Packet Processing

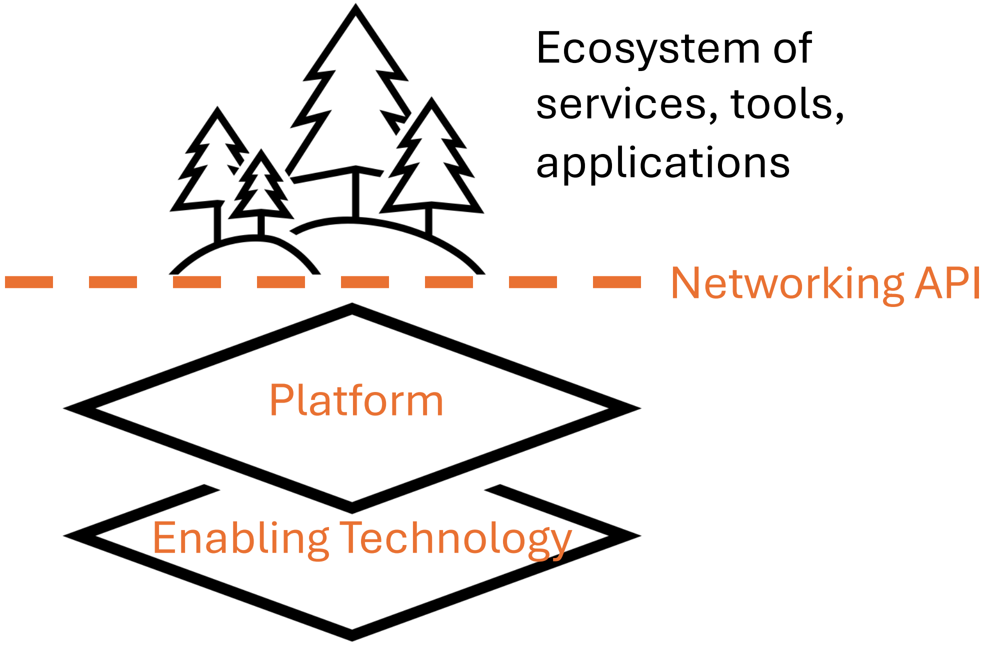


Linux

Linux Networking Stack

Robust, familiar programming model

Fast Software Based Packet Processing

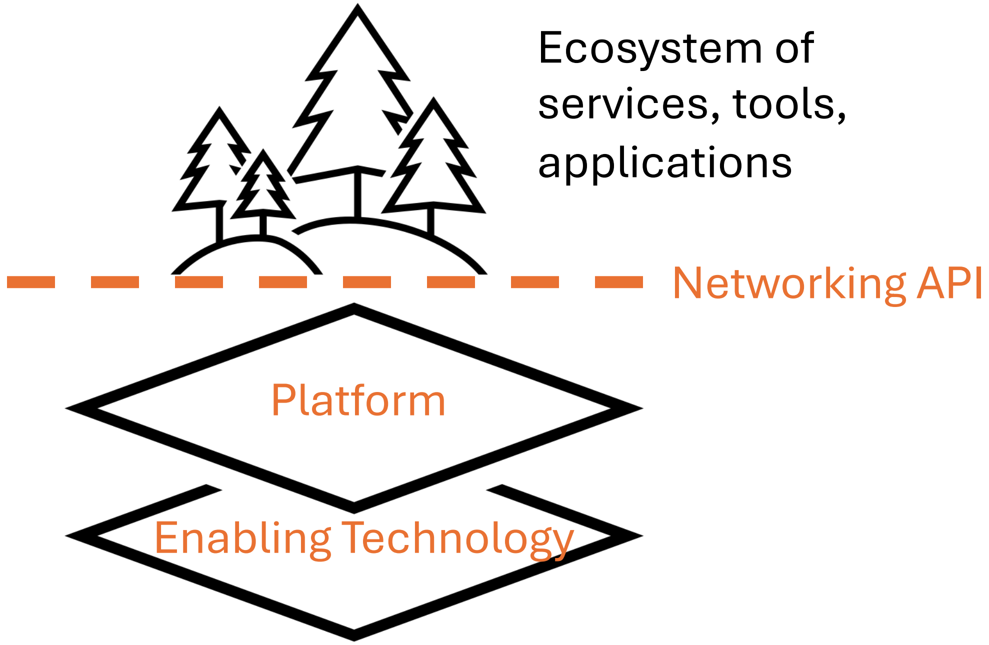


Linux	Polycube [0]
Linux Networking Stack	eBPF
Robust, familiar programming model	Limited by available network functions, custom API

[0] Miano, *et. al*. A Framework for eBPF-Based Network Functions in an Era of Microservices. *IEEE TNSM*, 18(1), 2021.

University of Colorado **Boulder** PRINCETON UNIVERSITY

Fast Software Based Packet Processing

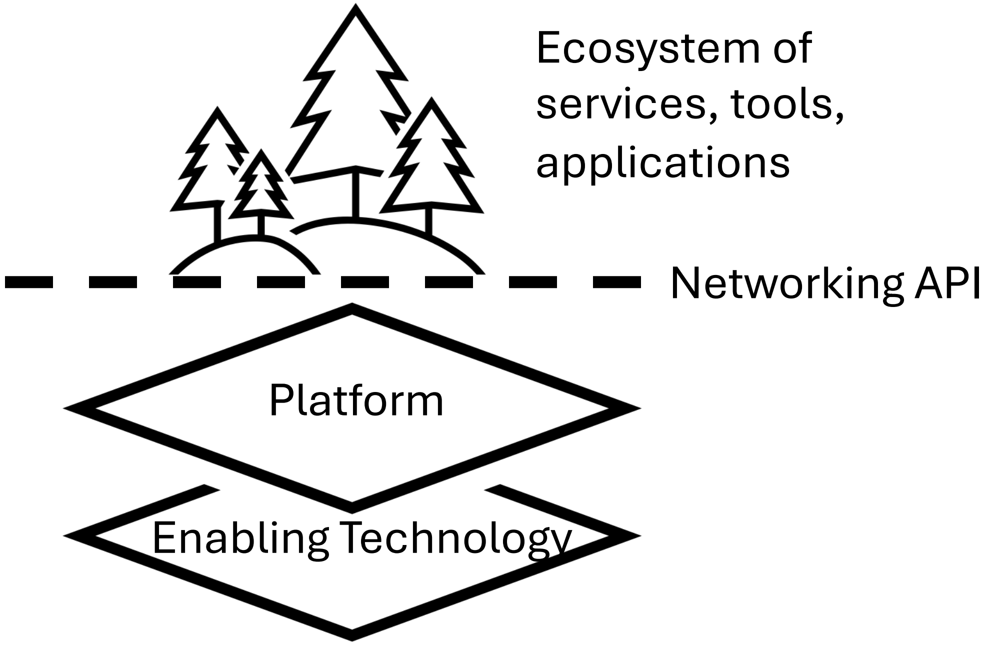


Linux	Polycube [0]	Vector Packet Processing (VPP) [1]
Linux Networking Stack	eBPF	DPDK
Robust, familiar programming model	Limited by available network functions, custom API	Manual programming model, custom API

[0] Miano, *et. al*. A Framework for eBPF-Based Network Functions in an Era of Microservices. *IEEE TNSM*, 18(1), 2021.
[1] FD.io: The worlds' secure networking dataplane, 2023. Retrieved October 20, 2023, https://fd.io.

University of Colorado **Boulder** PRINCETON UNIVERSITY

Fast Software Based Packet Processing



Ecosystem of services, tools, applications

Networking API

Platform

Enabling Technology

Existing Linux networking ecosystem

AND

Accelerated packet processing

LinuxFP (Linux Fast Path)

- **Transparently** enables **accelerated** packet processing while:
  - Maintaining compatibility with the Linux networking API
  - Maintaining access to the breadth of the Linux networking stack

LinuxFP Design

- **Dual processing pipelines**
  - Slow Path: Linux networking stack
  - Fast Path: eBPF/XDP

LinuxFP Design

- Dual processing pipelines
  - Slow Path: Linux networking stack
  - Fast Path: eBPF/XDP
- **Use Linux kernel networking state** for both paths to maintain consistency

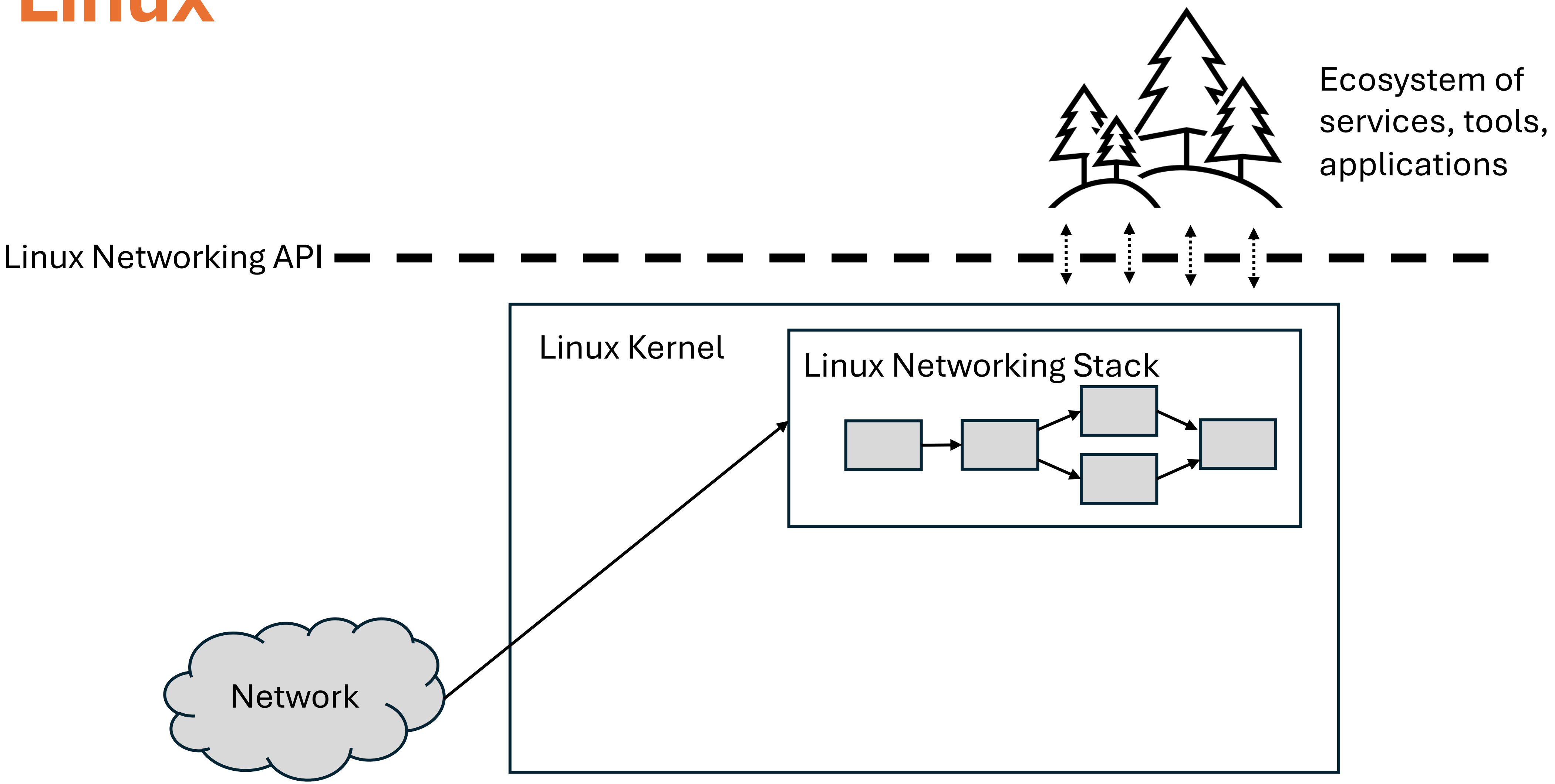
LinuxFP Design

- Dual processing pipelines
  - Slow Path: Linux networking stack
  - Fast Path: eBPF/XDP
- Use Linux kernel networking state for both paths to maintain consistency
- Modular design with **new Fast Path Module (FPM) abstraction**

LinuxFP Design

- Dual processing pipelines
  - Slow Path: Linux networking stack
  - Fast Path: eBPF/XDP
- Use Linux kernel networking state for both paths to maintain consistency
- Modular design with newFast Path Module abstraction
- **Dynamic and automatic installation of FPMs, as-needed**

Linux



Ecosystem of services, tools, applications

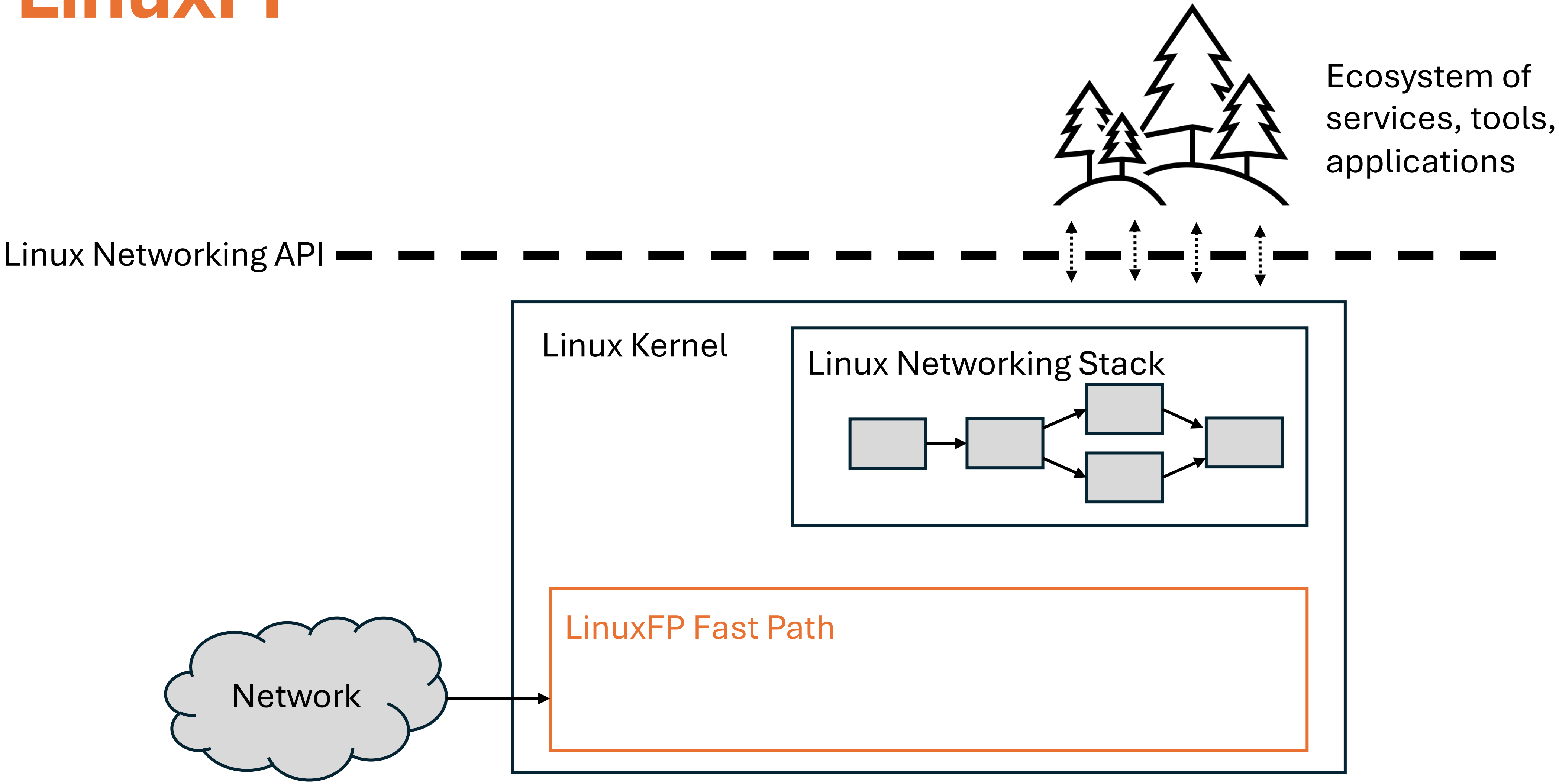
Linux Networking API

Linux Kernel

Linux Networking Stack

Network

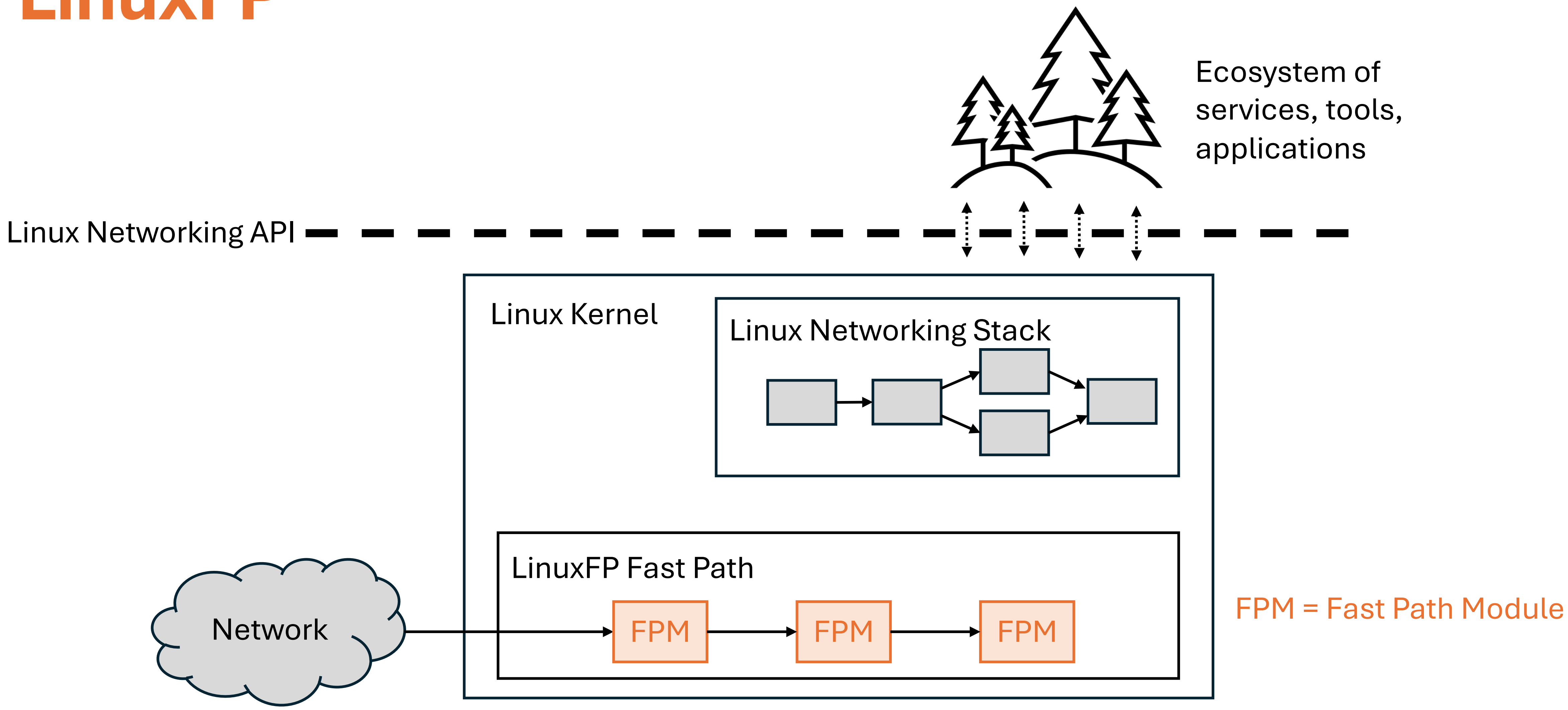
LinuxFP



University of Colorado **Boulder**

PRINCETON UNIVERSITY

LinuxFP

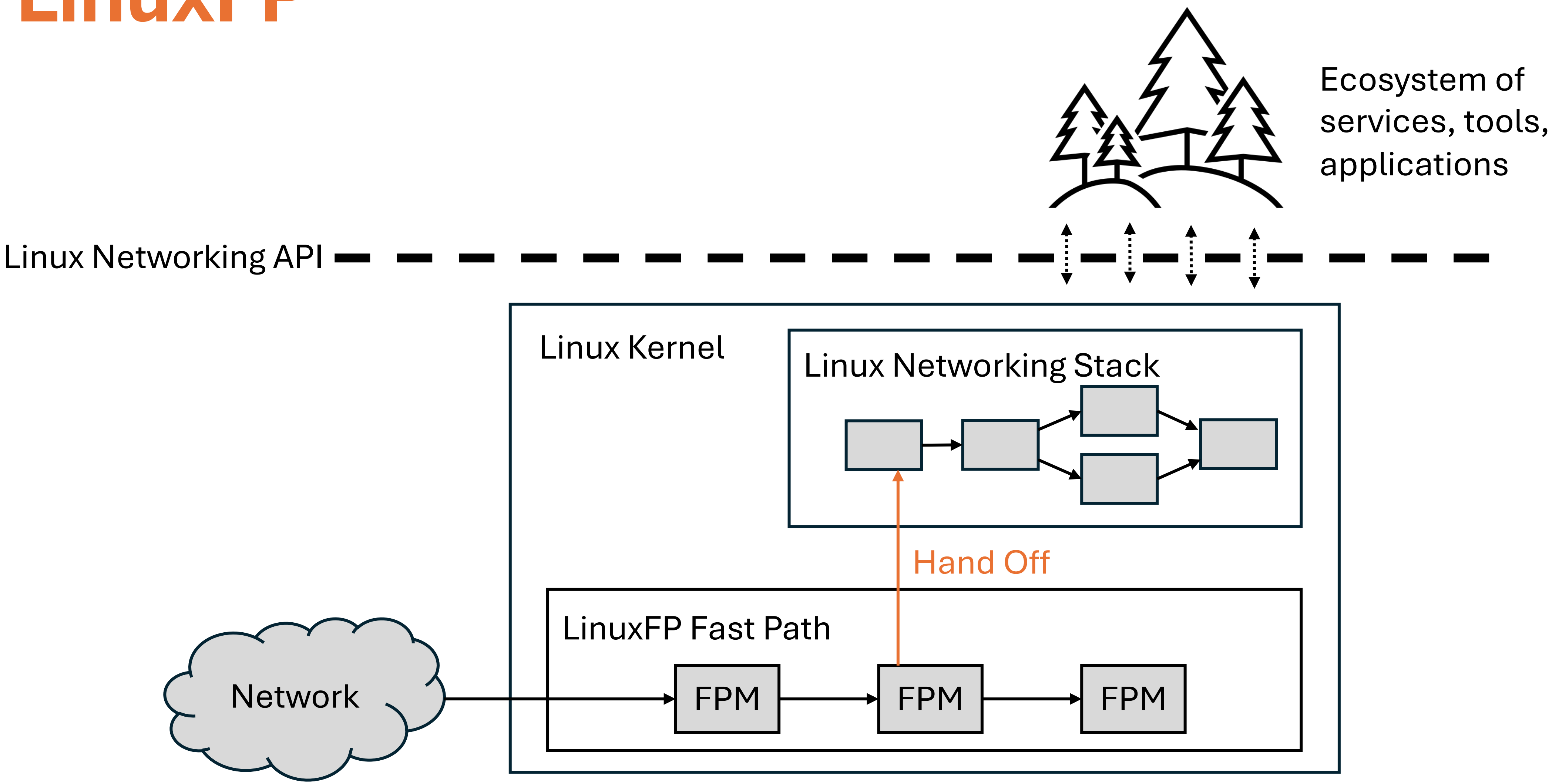


FPM = Fast Path Module

University of Colorado Boulder

PRINCETON UNIVERSITY

LinuxFP



Ecosystem of services, tools, applications

Linux Networking API

Linux Kernel

Linux Networking Stack

Hand Off

LinuxFP Fast Path

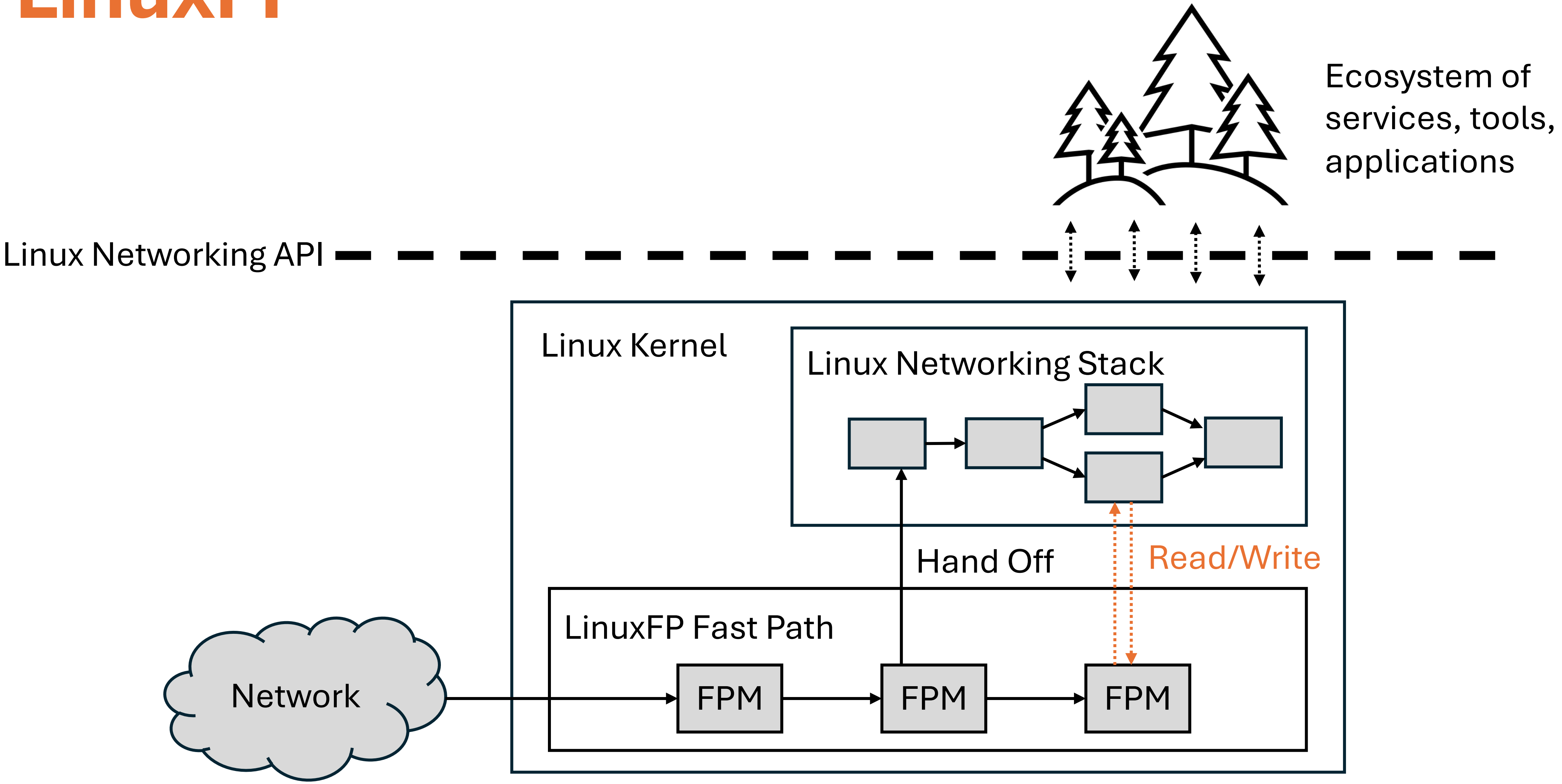
Network

FPM

FPM

FPM

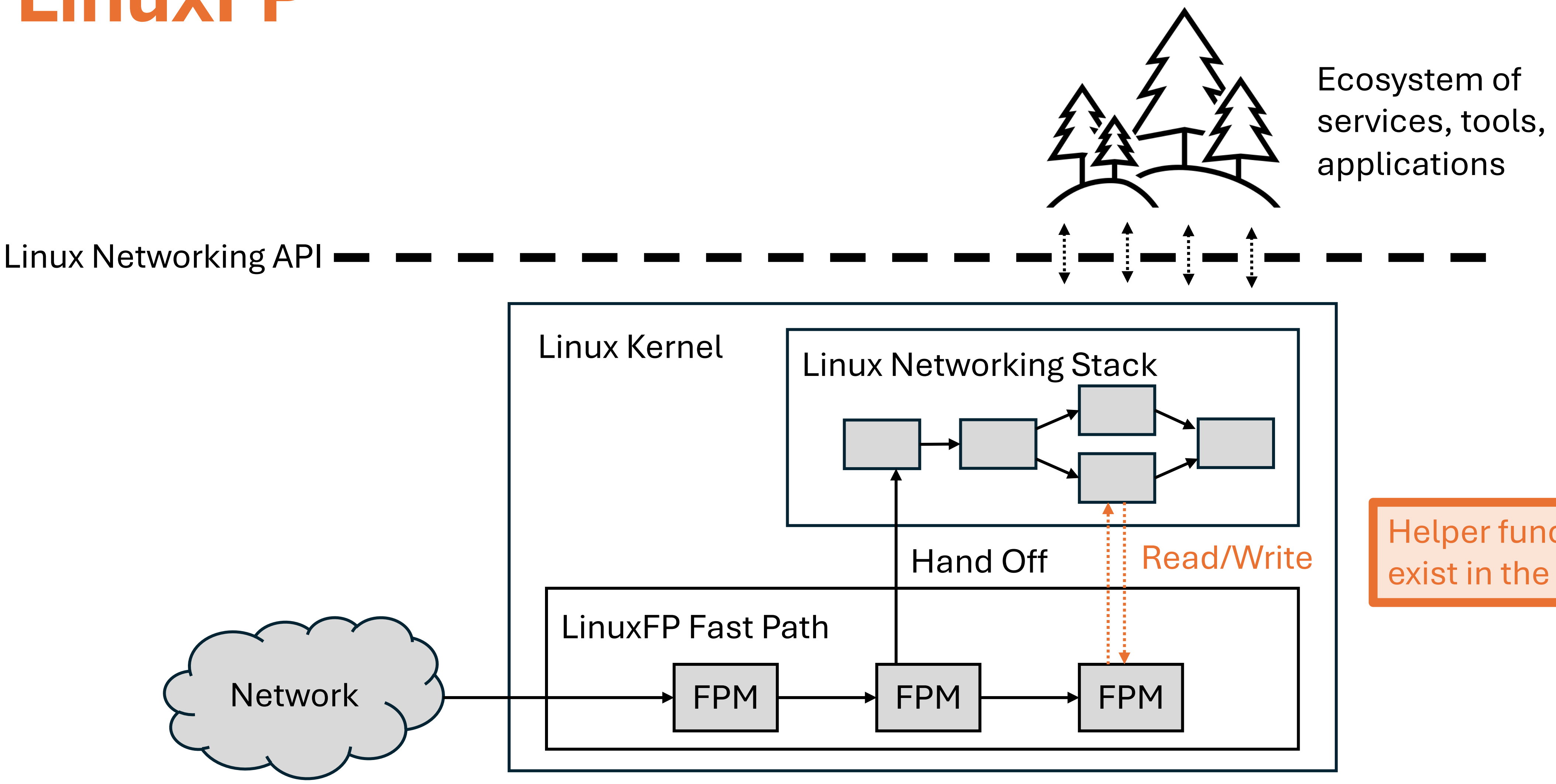
LinuxFP



University of Colorado **Boulder**

PRINCETON UNIVERSITY

LinuxFP

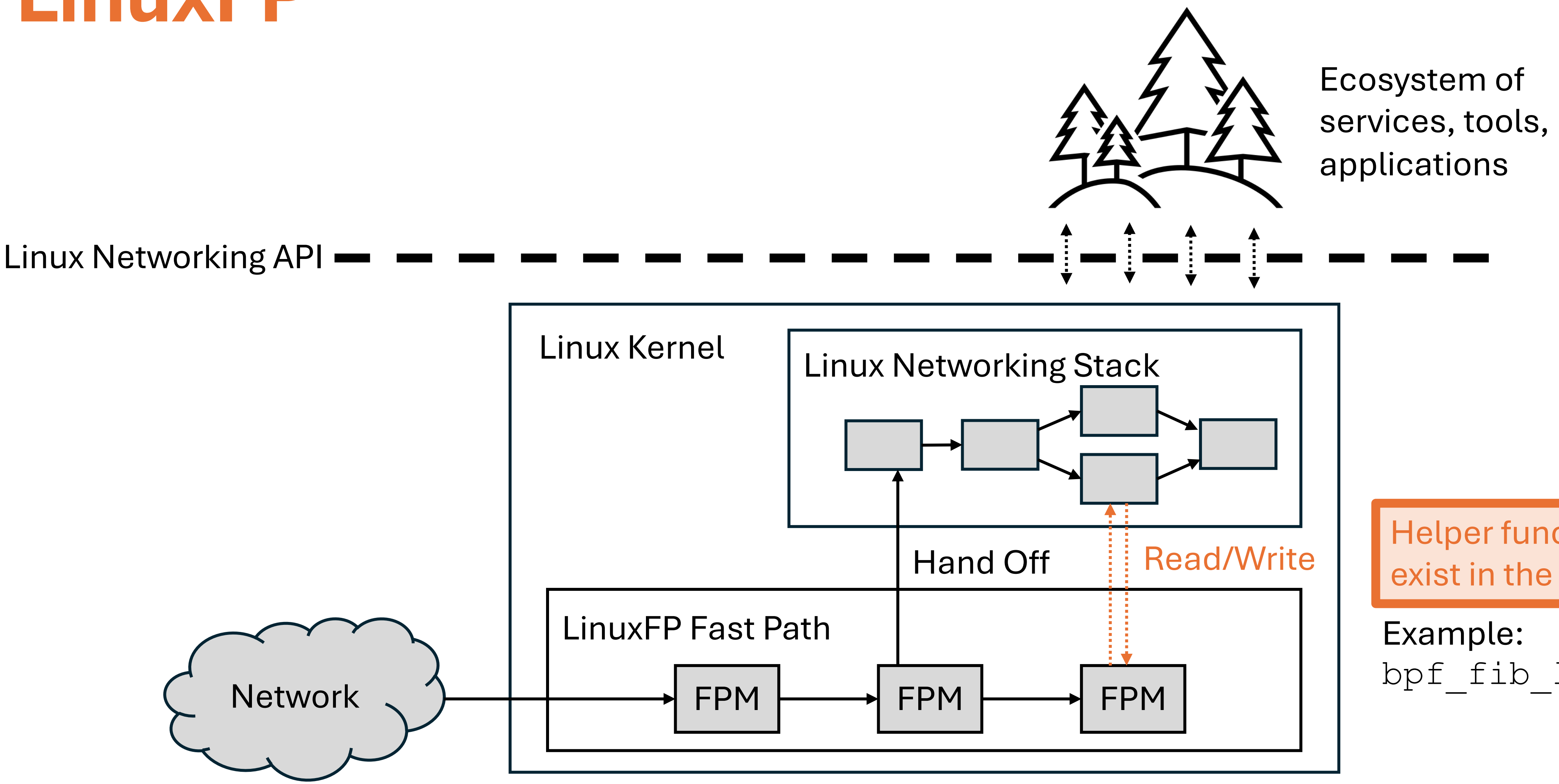


Helper functions exist in the kernel

University of Colorado **Boulder**

PRINCETON UNIVERSITY

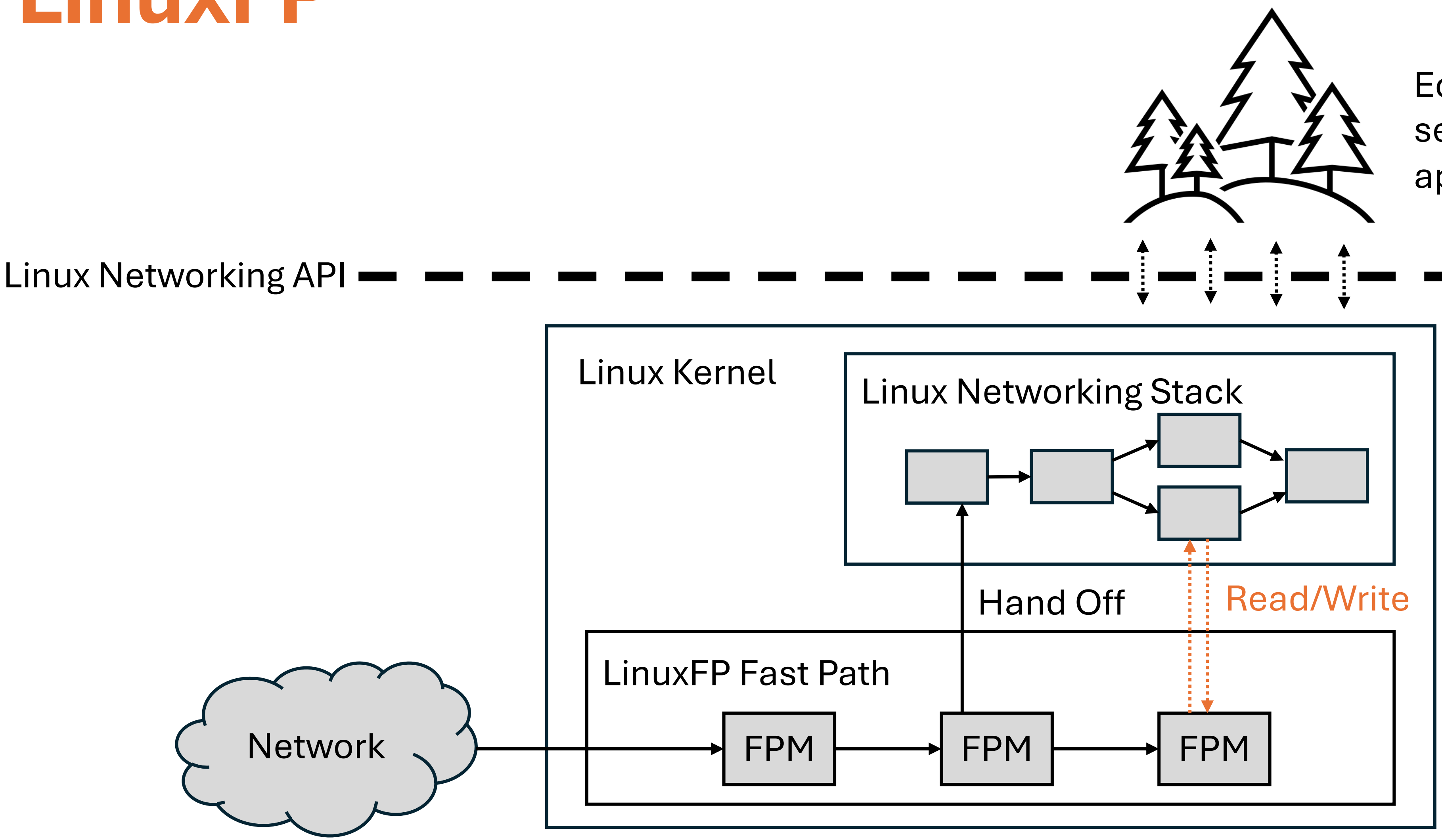
LinuxFP



Helper functions exist in the kernel

Example:
`bpf_fib_lookup()`

LinuxFP



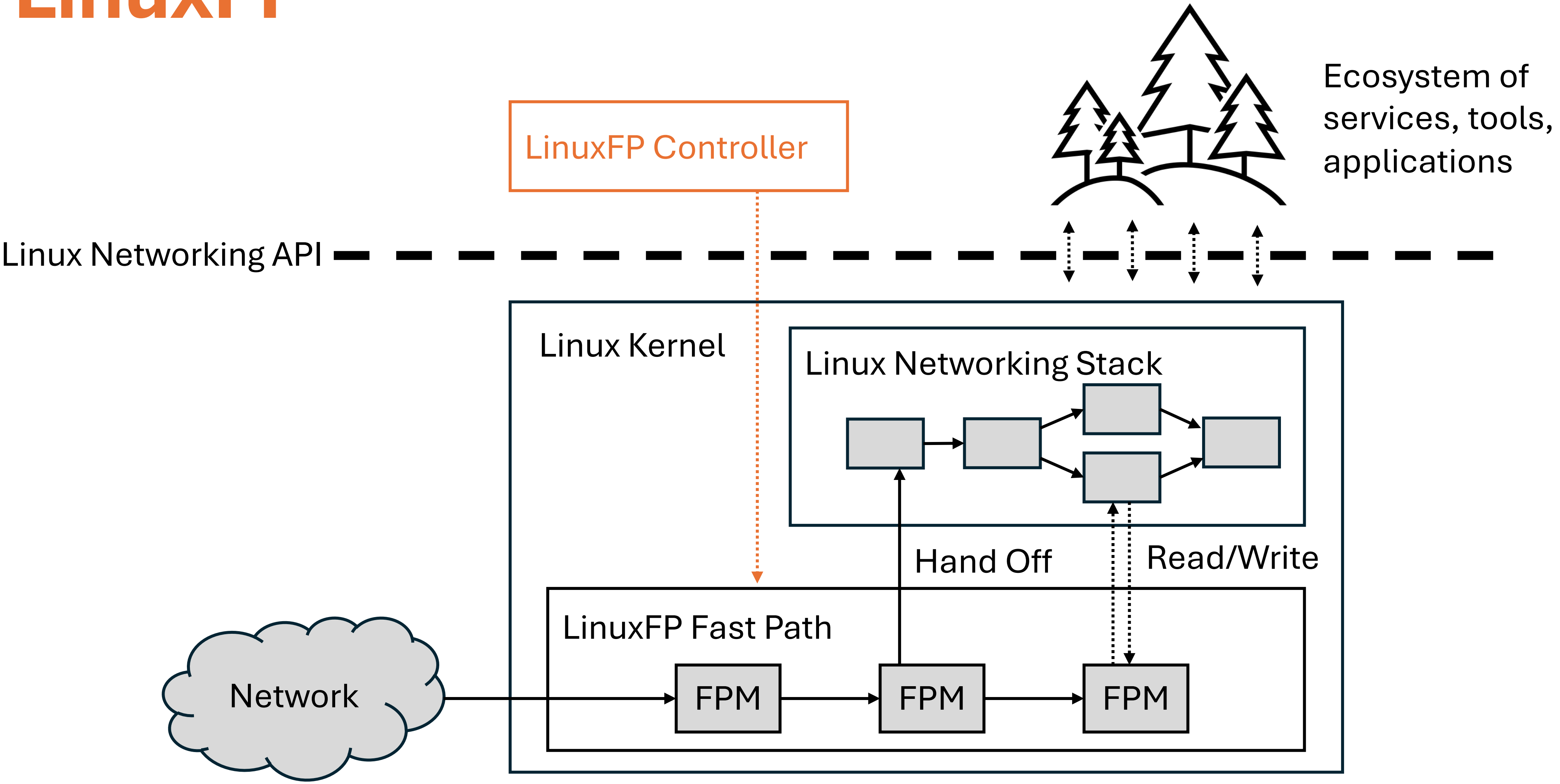
Ecosystem of services, tools, applications

Helper functions exist in the kernel*

Example:
`bpf_fib_lookup()`

*New Example:
`bpf_ipt_lookup()`

LinuxFP

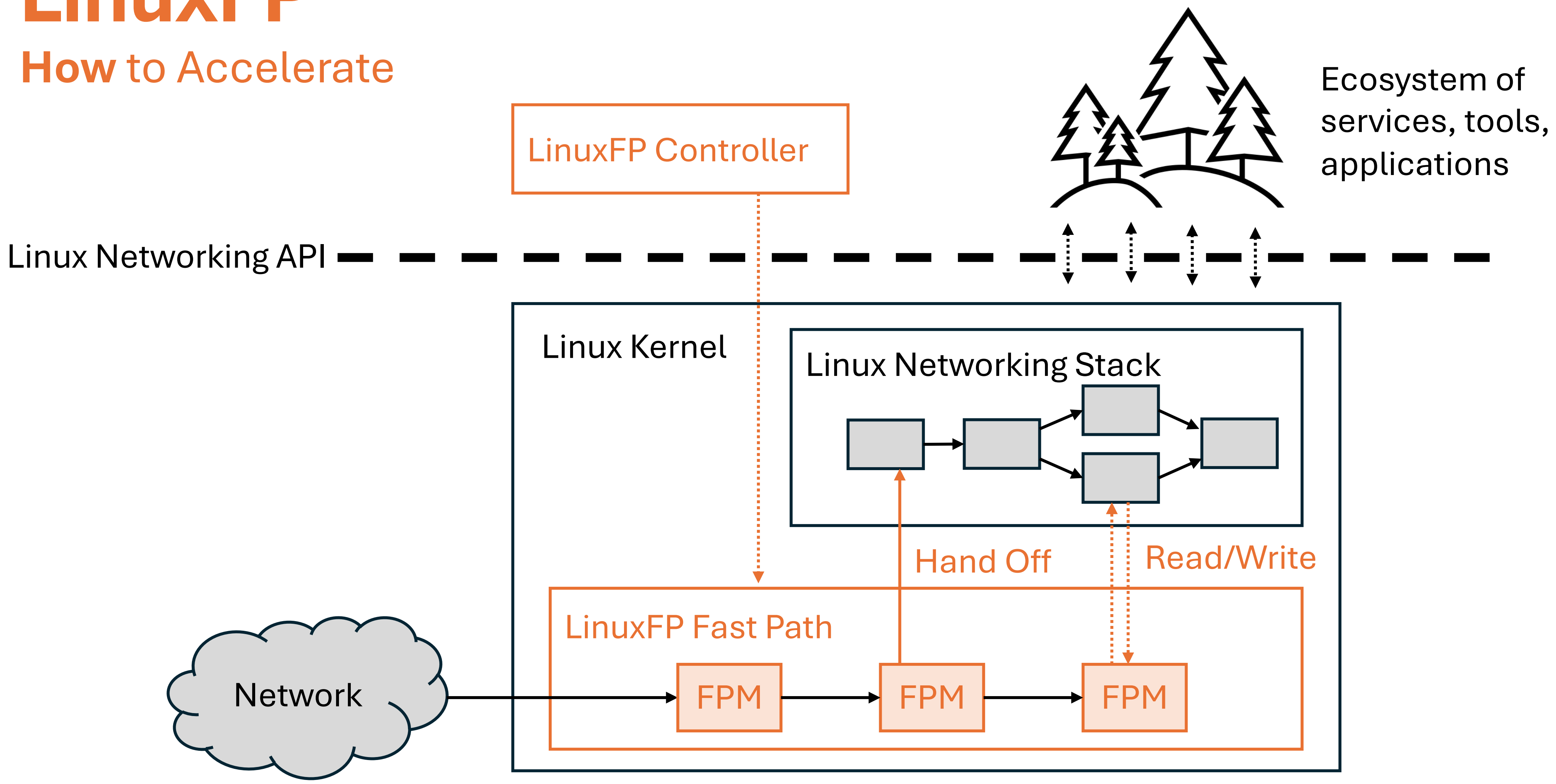


University of Colorado **Boulder**

PRINCETON UNIVERSITY

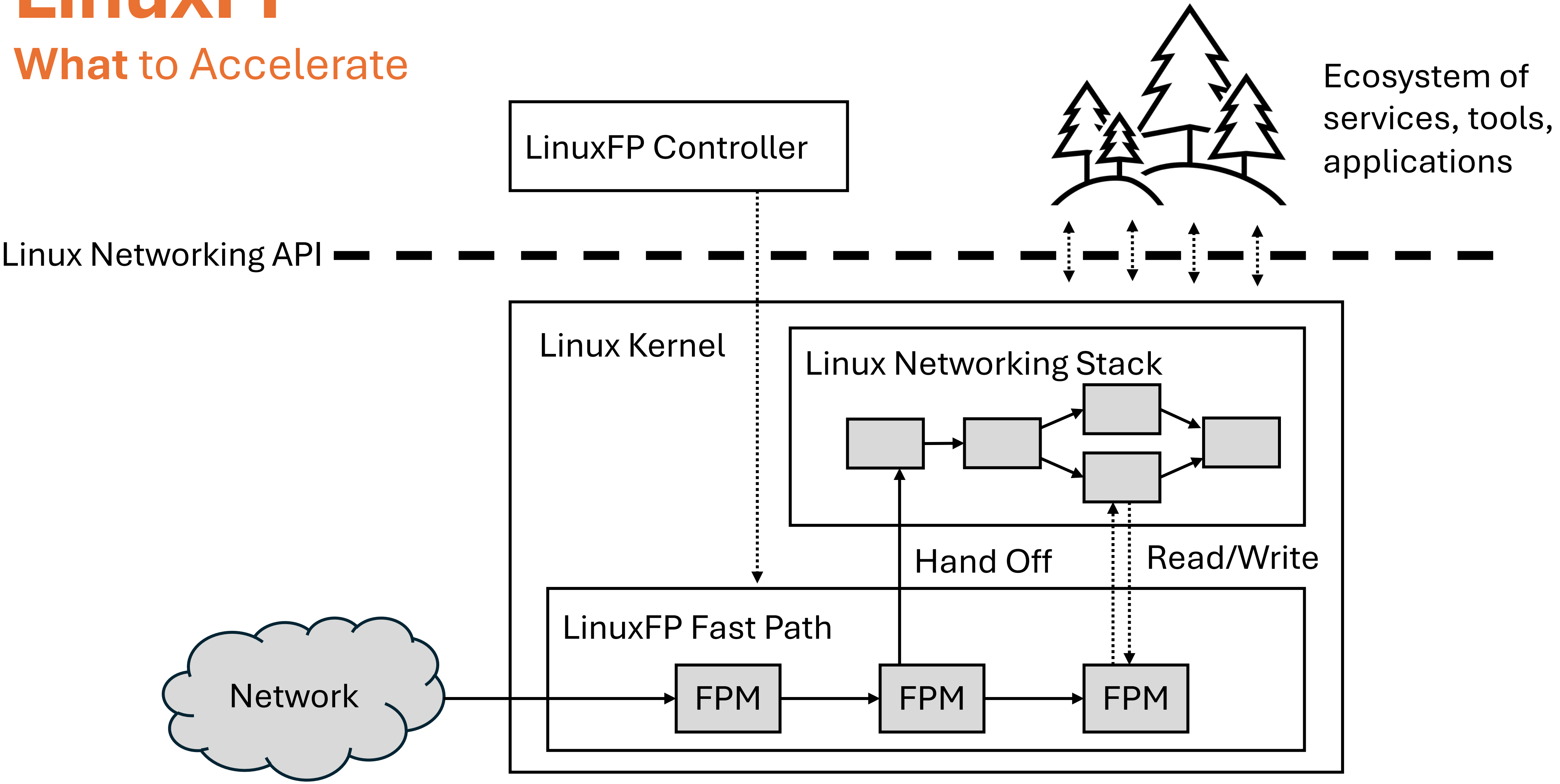
LinuxFP

How to Accelerate



LinuxFP

What to Accelerate

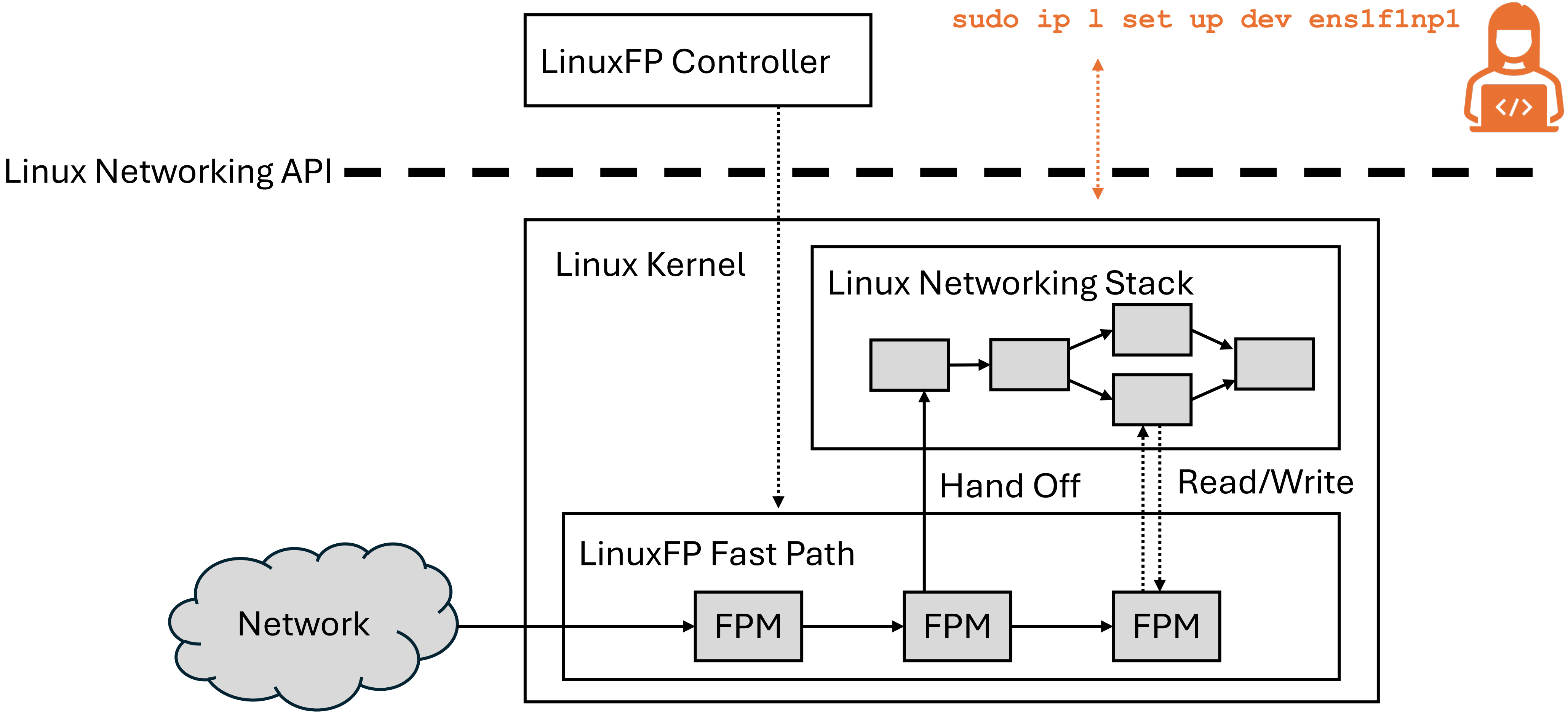


University of Colorado **Boulder** PRINCETON UNIVERSITY

LinuxFP

What to Accelerate

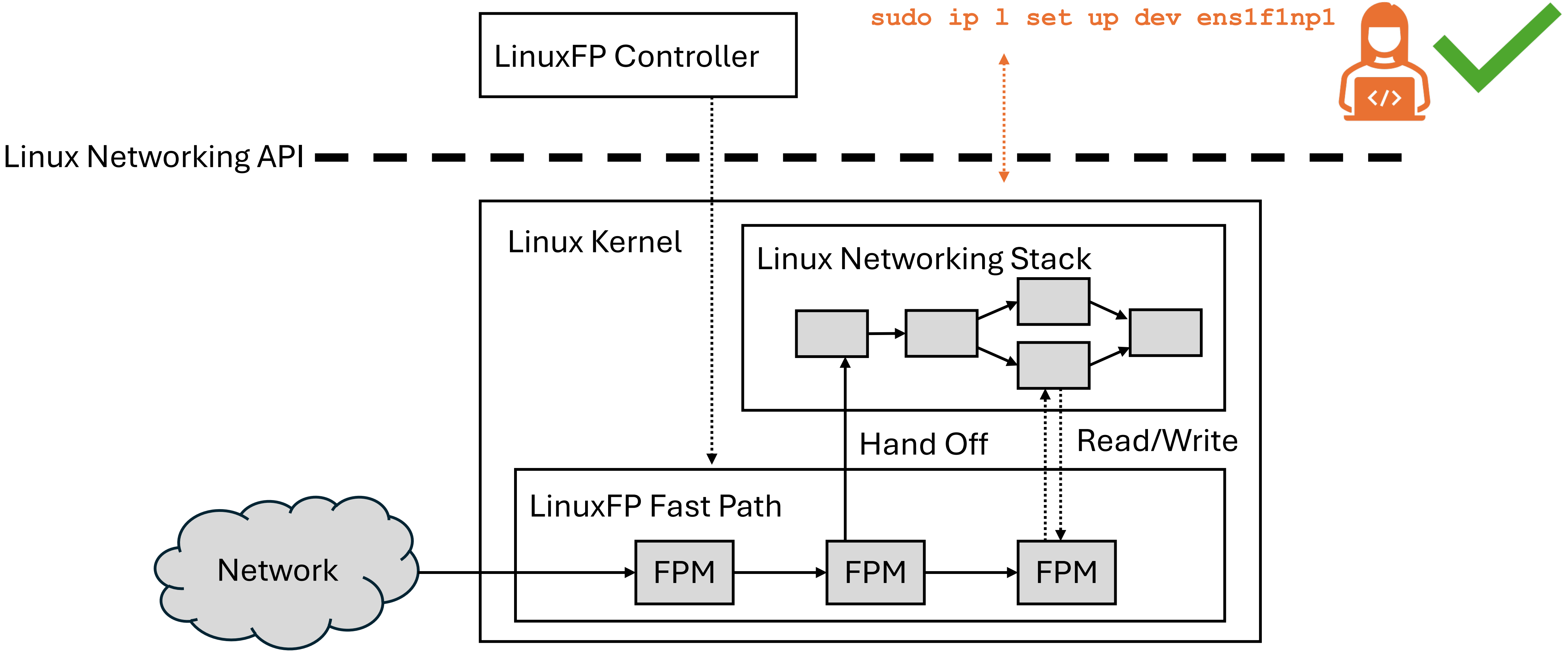
```
# This is my virtual router setup script
sudo ip a add 10.0.1.1/24 dev ens1f0np0
sudo ip a add 10.0.2.1/24 dev ens1f1np1
sudo ip l set up dev ens1f0np0
sudo ip l set up dev ens1f1np1
```



LinuxFP

What to Accelerate

```
# This is my virtual router setup script
sudo ip a add 10.0.1.1/24 dev ens1f0np0
sudo ip a add 10.0.2.1/24 dev ens1f1np1
sudo ip l set up dev ens1f0np0
sudo ip l set up dev ens1f1np1
```



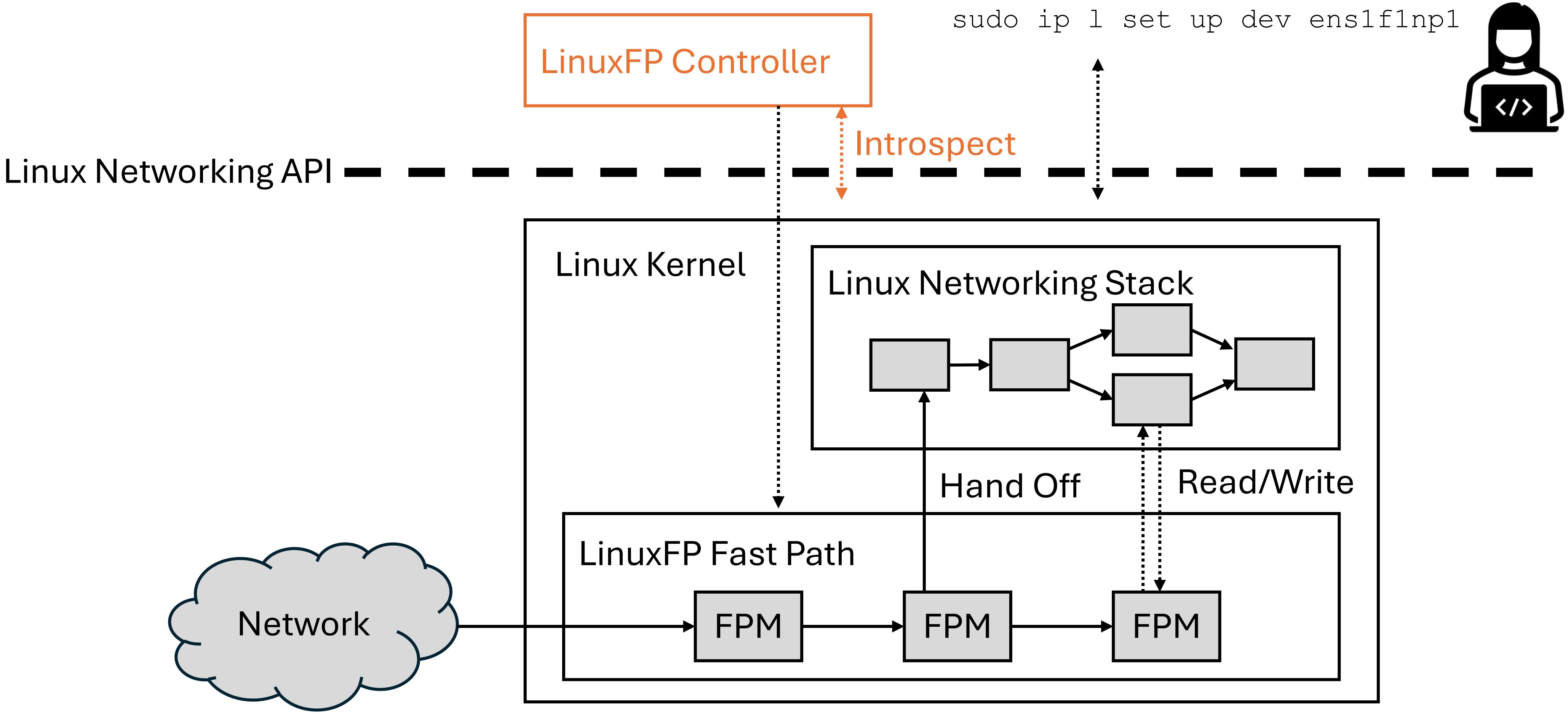
University of Colorado Boulder

PRINCETON UNIVERSITY

LinuxFP

What to Accelerate

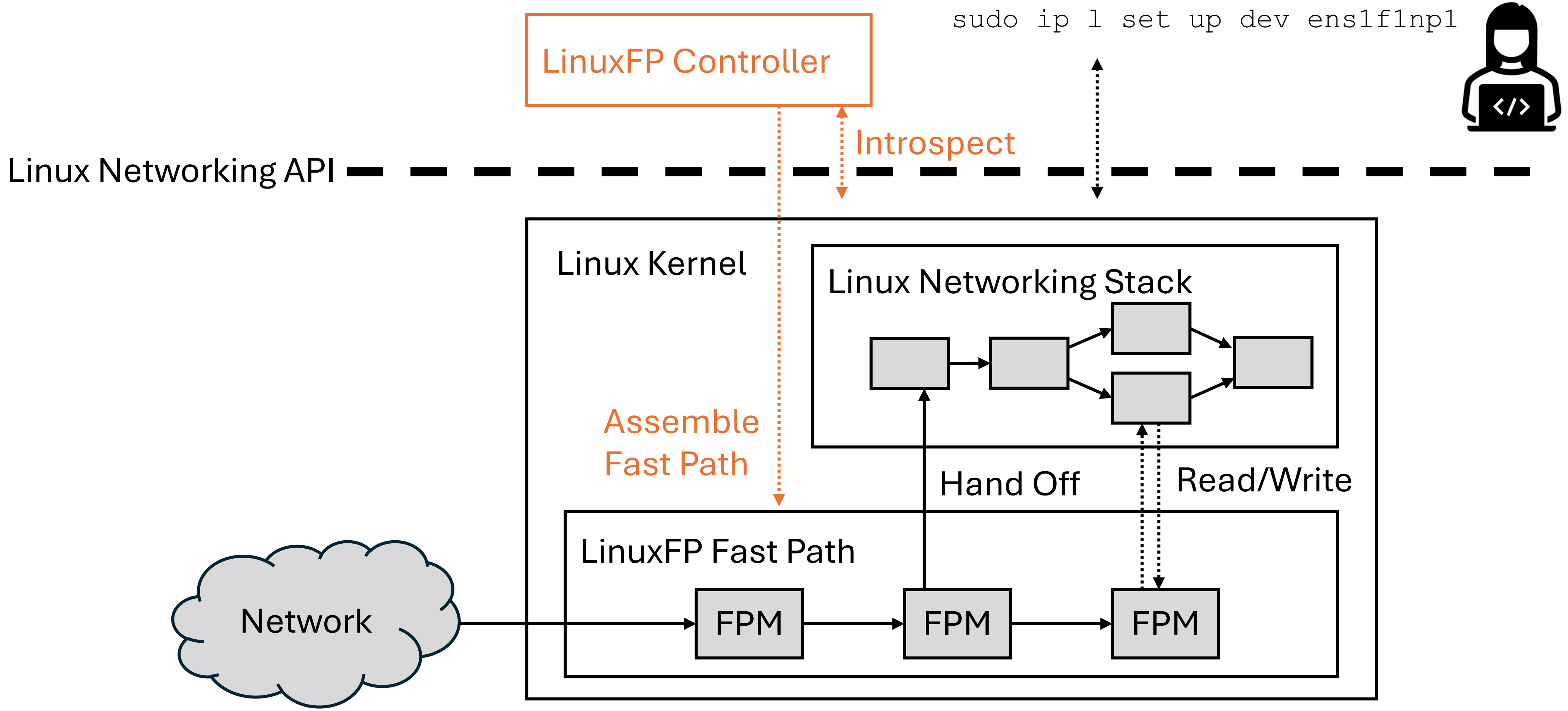
```
# This is my virtual router setup script
sudo ip a add 10.0.1.1/24 dev ens1f0np0
sudo ip a add 10.0.2.1/24 dev ens1f1np1
sudo ip l set up dev ens1f0np0
sudo ip l set up dev ens1f1np1
```



LinuxFP

What to Accelerate

```
# This is my virtual router setup script
sudo ip a add 10.0.1.1/24 dev ens1f0np0
sudo ip a add 10.0.2.1/24 dev ens1f1np1
sudo ip l set up dev ens1f0np0
sudo ip l set up dev ens1f1np1
```



LinuxFP Controller

Introspect

Linux Networking API

Linux Kernel

Linux Networking Stack

Assemble Fast Path

Hand Off

Read/Write

LinuxFP Fast Path

Network

FPM

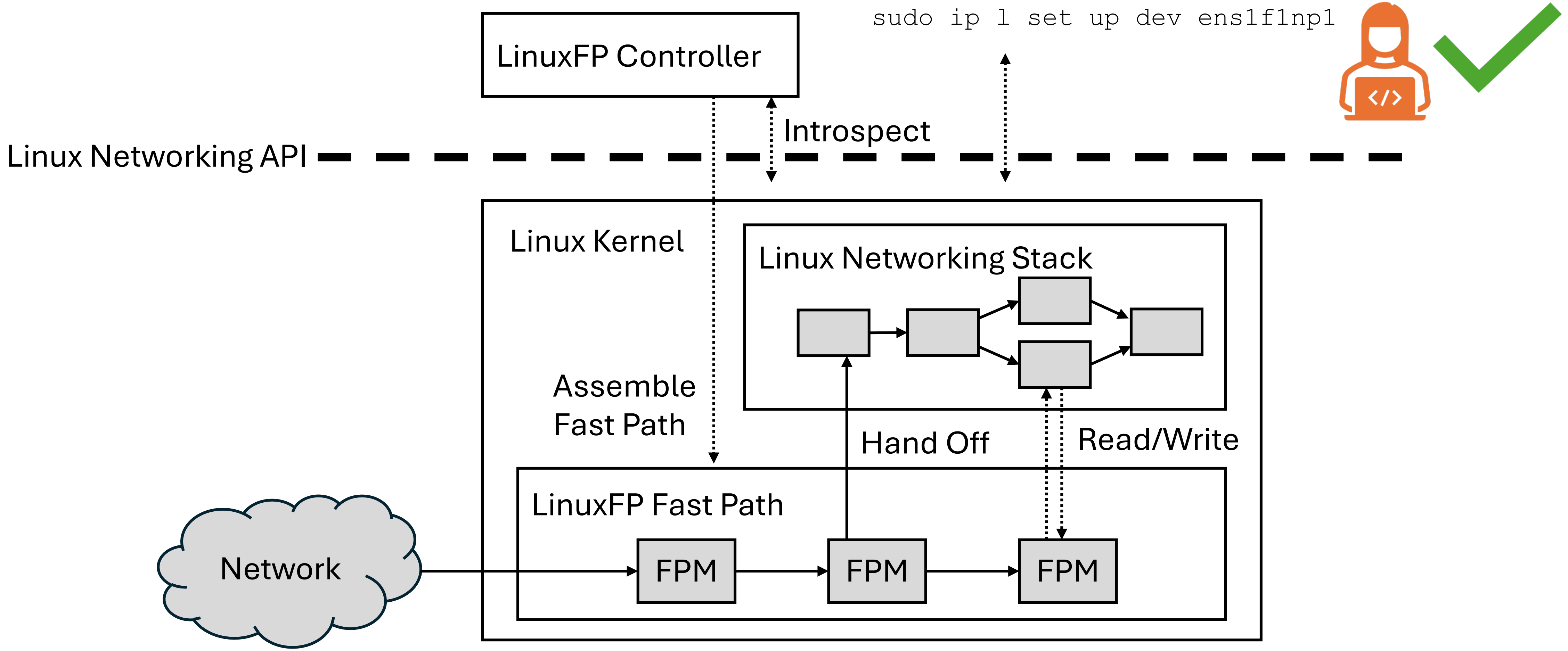
FPM

FPM

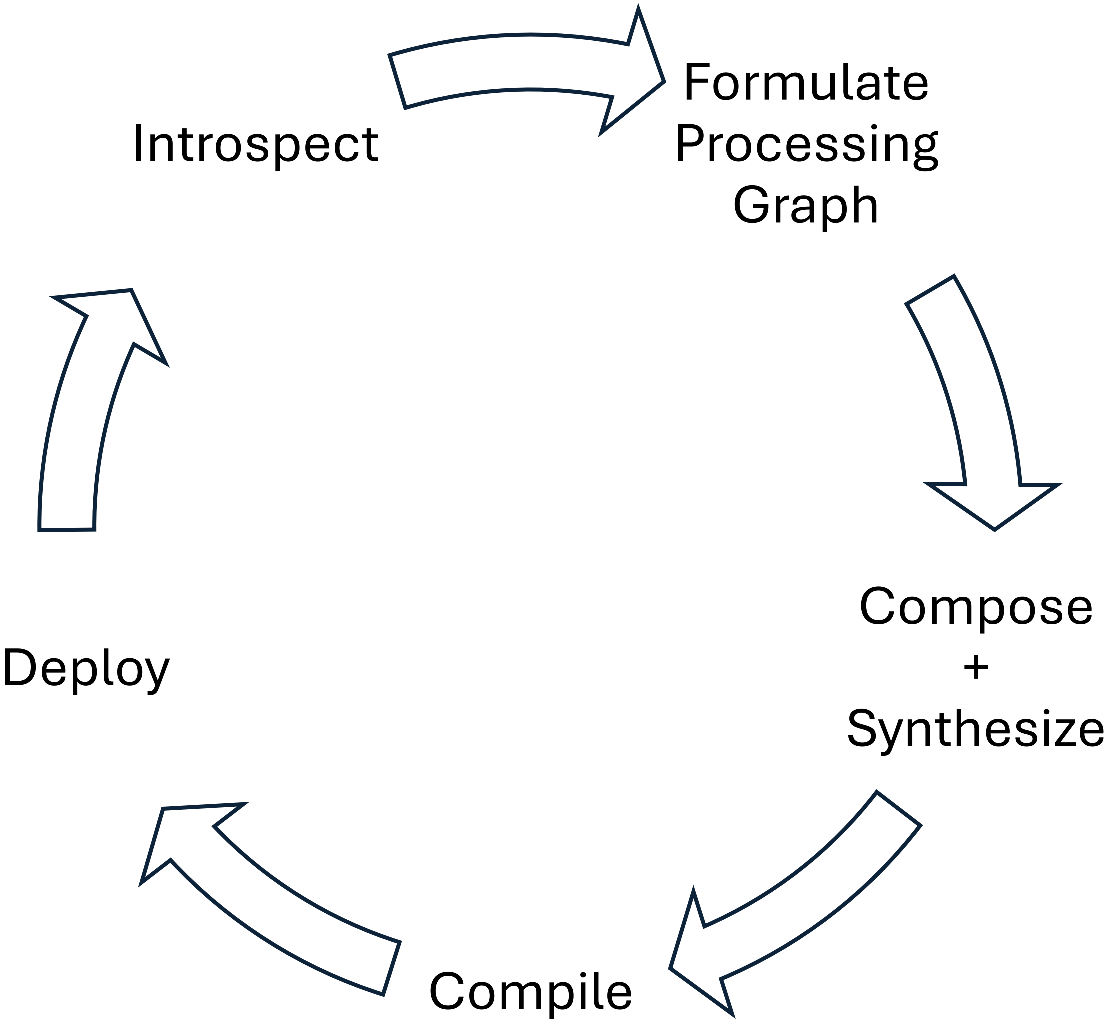
LinuxFP

What to Accelerate

```
# This is my virtual router setup script
sudo ip a add 10.0.1.1/24 dev ens1f0np0
sudo ip a add 10.0.2.1/24 dev ens1f1np1
sudo ip l set up dev ens1f0np0
sudo ip l set up dev ens1f1np1
```

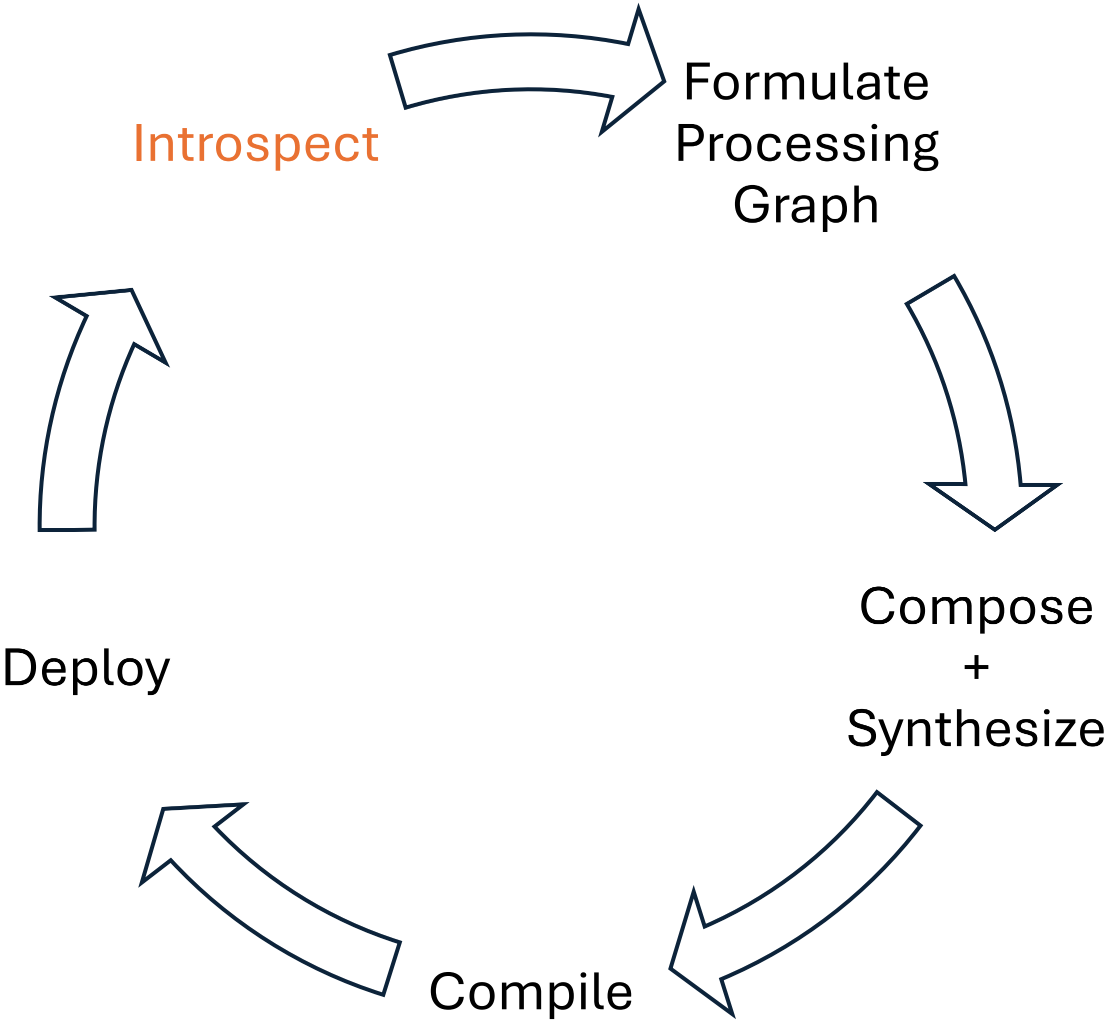


LinuxFP **Controller**



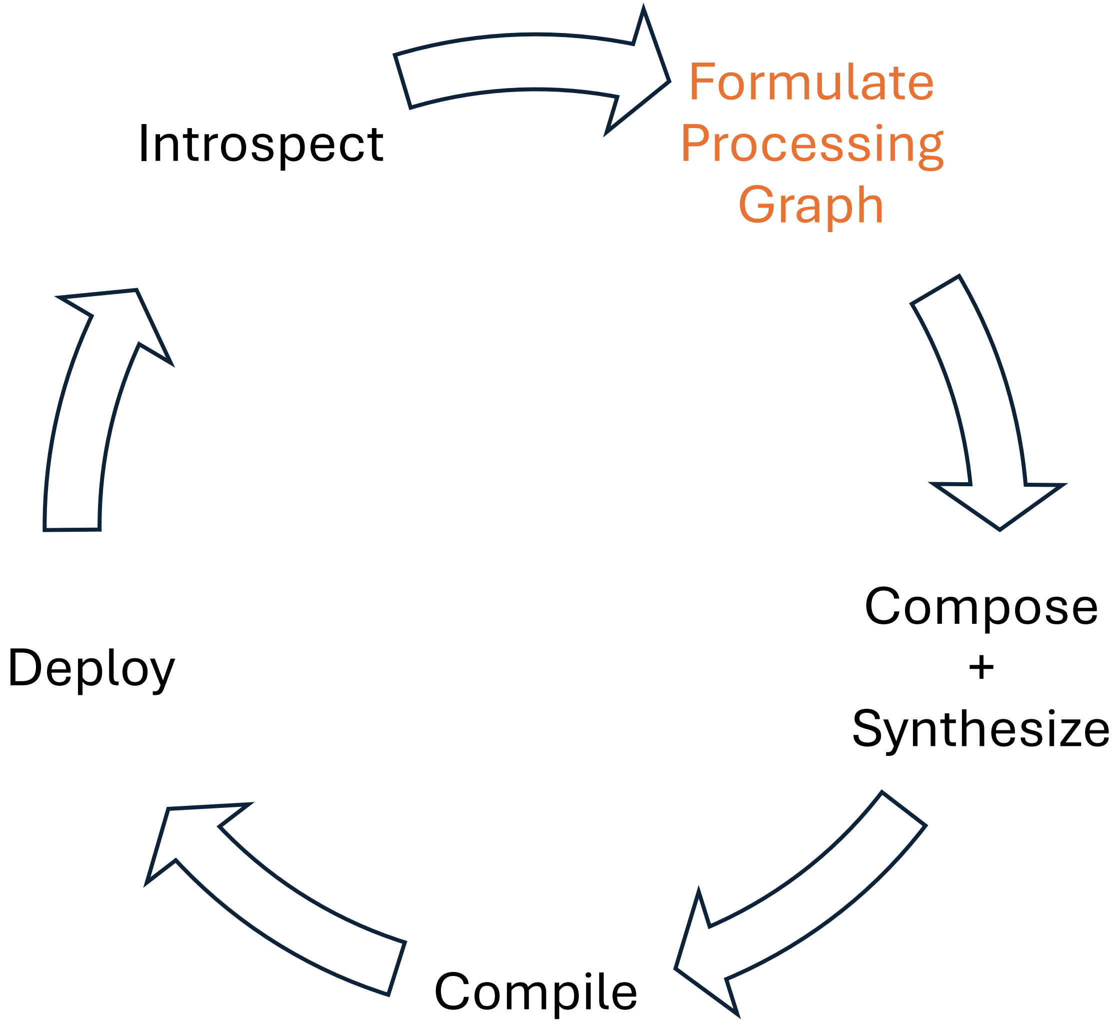
LinuxFP **Controller**

- **Service Inspector**
  - netlink



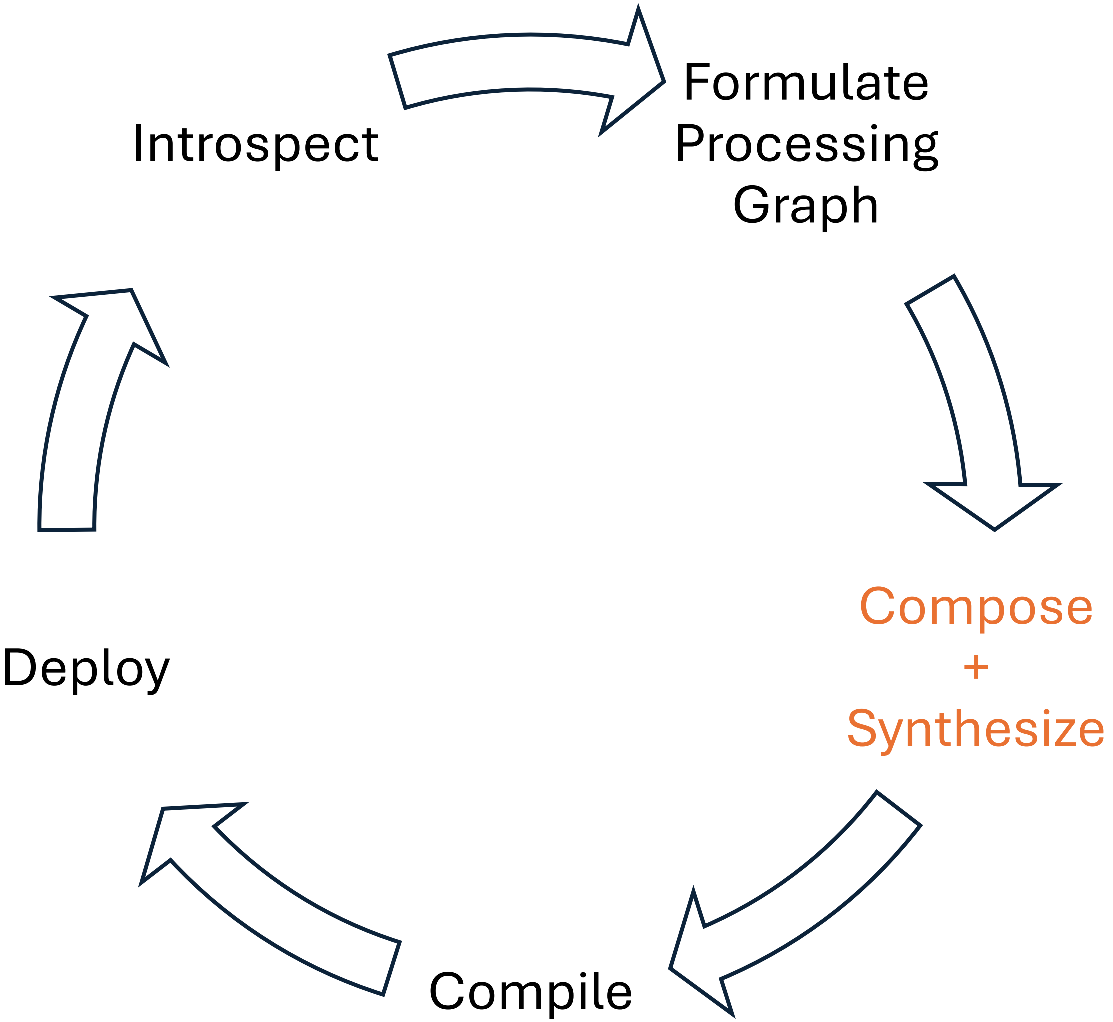
LinuxFP **Controller**

- Service Inspector
  - netlink
- **Topology Manager**
  - Maintain processing order
  - JSON graph, specific config



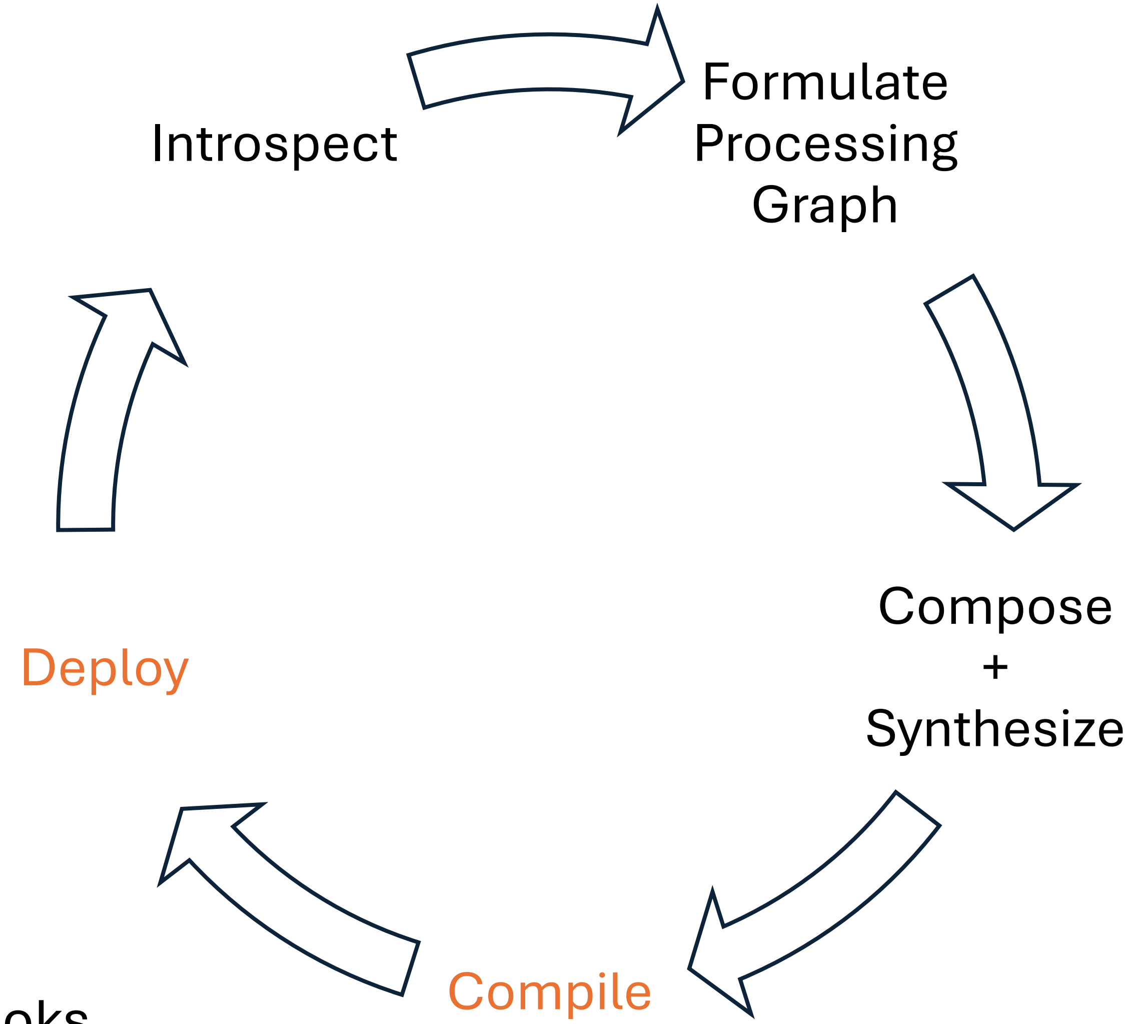
LinuxFP **Controller**

- Service Inspector
  - netlink
- Topology Manager
  - Maintain processing order
  - JSON graph, specific config
- **Fast Path Synthesizer**
  - Select FPM template
  - Parameterize using config
- **Compatibility Manager**



LinuxFP **Controller**

- Service Inspector
  - netlink
- Topology Manager
  - Maintain processing order
  - JSON graph, specific config
- Fast Path Synthesizer
  - Select FPM template
  - Parameterize using config
- Compatibility Manager
- **Fast Path Deployer**
  - Generate eBPF bytecode
  - Attach to TC (traffic control) or XDP hooks



What to accelerate?

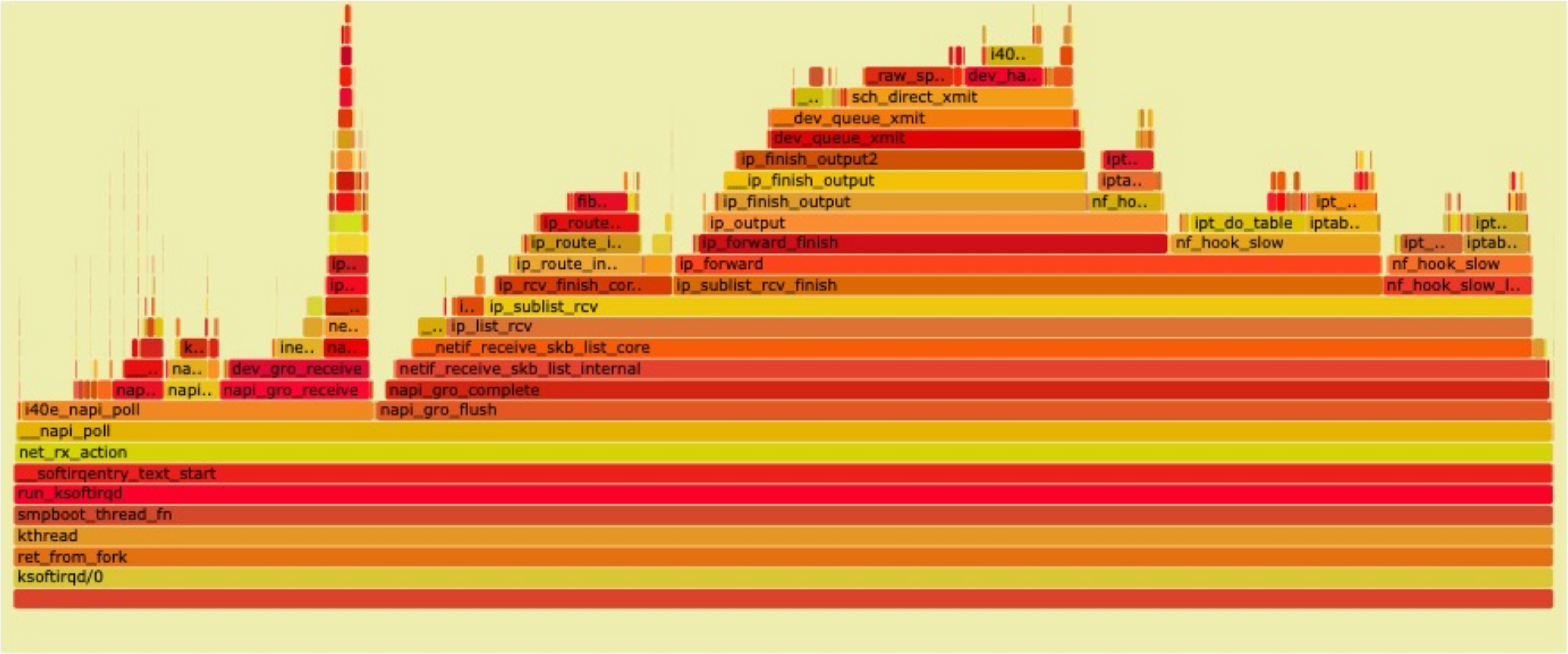
Fast path: narrow, common case

Slow path: wide, exceptional case

What to accelerate **for forwarding**?

What to accelerate **for forwarding**?

Depth Of Call Stack



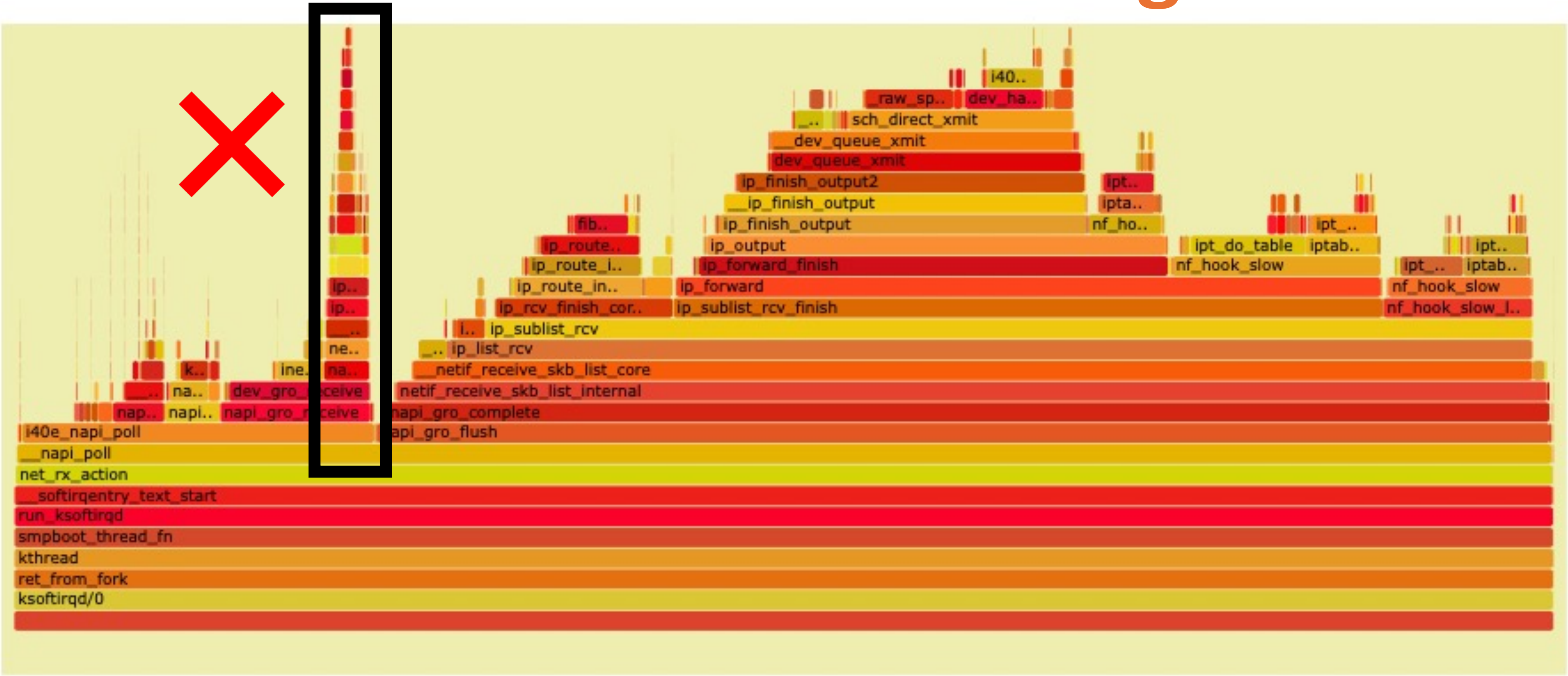
Trace Call Stacks

University of Colorado **Boulder**

PRINCETON UNIVERSITY

What to accelerate for forwarding?

Depth
Of
Call
Stack



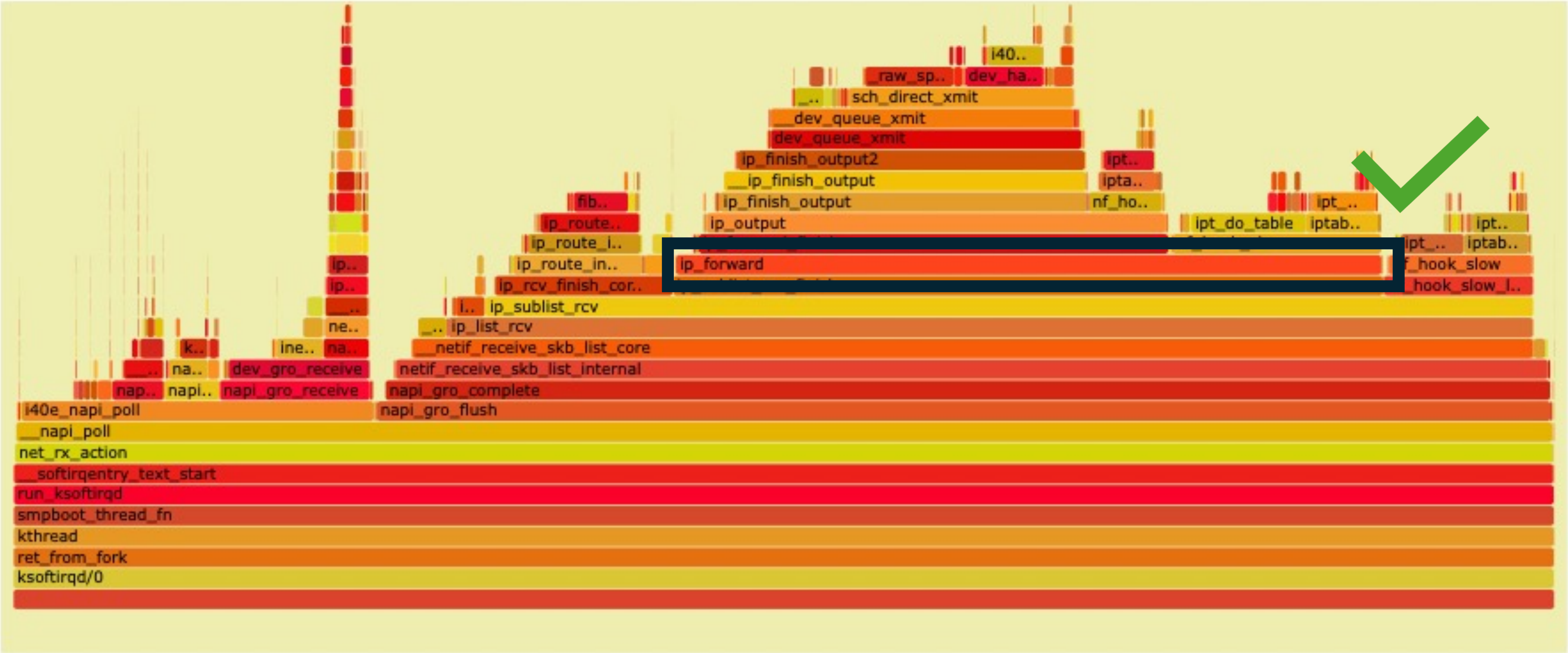
Trace Call Stacks

University of Colorado **Boulder**

PRINCETON UNIVERSITY

What to accelerate for forwarding?

Depth
Of
Call
Stack



Trace Call Stacks

University of Colorado **Boulder**

PRINCETON UNIVERSITY

Forwarding in LinuxFP

Fast Path

- Parsing
- Rewriting
- Forwarding information base (FIB) lookup
- Forwarding

Slow Path

- ARP handling
- IP (de)fragmentation

LinuxFP Evaluation

Enable fast packet processing

Maintain compatibility with the Linux networking API

University of Colorado **Boulder**

PRINCETON UNIVERSITY

LinuxFP Evaluation

Enable fast packet processing

Virtual Network Function: Router

Maintain compatibility with the Linux networking API

Virtual Network Functions: **Baselines**

- **Linux**

University of Colorado **Boulder**

PRINCETON UNIVERSITY

Virtual Network Functions: **Baselines**

- Linux
- **Polycube [0]**
  - Built on eBPF
  - Runs in Linux kernel
  - Fixed dataplane configured with custom CLI

[0] Miano, *et. al*. A Framework for eBPF-Based Network Functions in an Era of Microservices. *IEEE TNSM*, 18(1), 2021.
[1] FD.io: The Worlds' Secure Networking Dataplane, 2023. Retrieved October 20, 2023, https://fd.io.

Virtual Network Functions: Baselines

- Linux
- **Polycube [0]**
  - Built on eBPF
  - Runs in Linux kernel
  - Fixed dataplane configured with custom CLI
- **VPP [1]** (Vector Packet Processor)
  - Built on enabling technology of DPDK (kernel bypass)
  - Configured through custom CLI or custom API
  - Dedicated core(s), batching of packets

[0] Miano, *et. al*. A Framework for eBPF-Based Network Functions in an Era of Microservices. *IEEE TNSM*, 18(1), 2021.
[1] FD.io: The Worlds' Secure Networking Dataplane, 2023. Retrieved October 20, 2023, https://fd.io.

Virtual Network Functions: **Baselines**

- Linux
- Polycube [0]
  - Built on eBPF
  - Runs in Linux kernel
  - Fixed dataplane configured with **custom CLI**
- VPP [1] (Vector Packet Processor)
  - Built on enabling technology of DPDK (kernel bypass)
  - Configured through **custom CLI or custom API**
  - Dedicated core(s), batching of packets

[0] Miano, *et. al*. A Framework for eBPF-Based Network Functions in an Era of Microservices. *IEEE TNSM*, 18(1), 2021.
[1] FD.io: The Worlds' Secure Networking Dataplane, 2023. Retrieved October 20, 2023, https://fd.io.

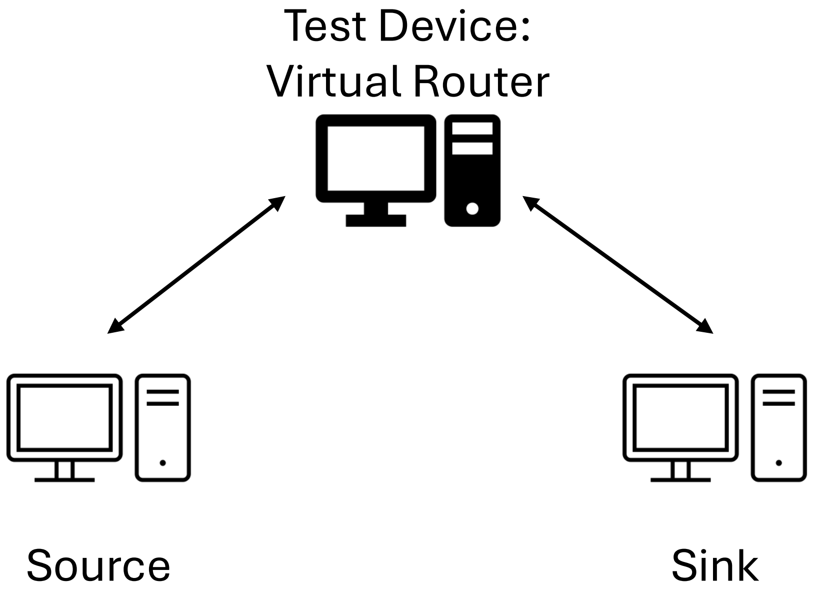
Virtual Network Functions: **Baselines**

- Linux
- Polycube [0]
  - **Built on eBPF**
  - Runs in Linux kernel
  - Fixed dataplane configured with custom CLI
- VPP [1] (Vector Packet Processor)
  - Built on enabling technology of DPDK (kernel bypass)
  - Configured through custom CLI or custom API
  - **Dedicated core(s), batching of packets**

[0] Miano, *et. al*. A Framework for eBPF-Based Network Functions in an Era of Microservices. *IEEE TNSM*, 18(1), 2021.
[1] FD.io: The Worlds’ Secure Networking Dataplane, 2023. Retrieved October 20, 2023, https://fd.io.

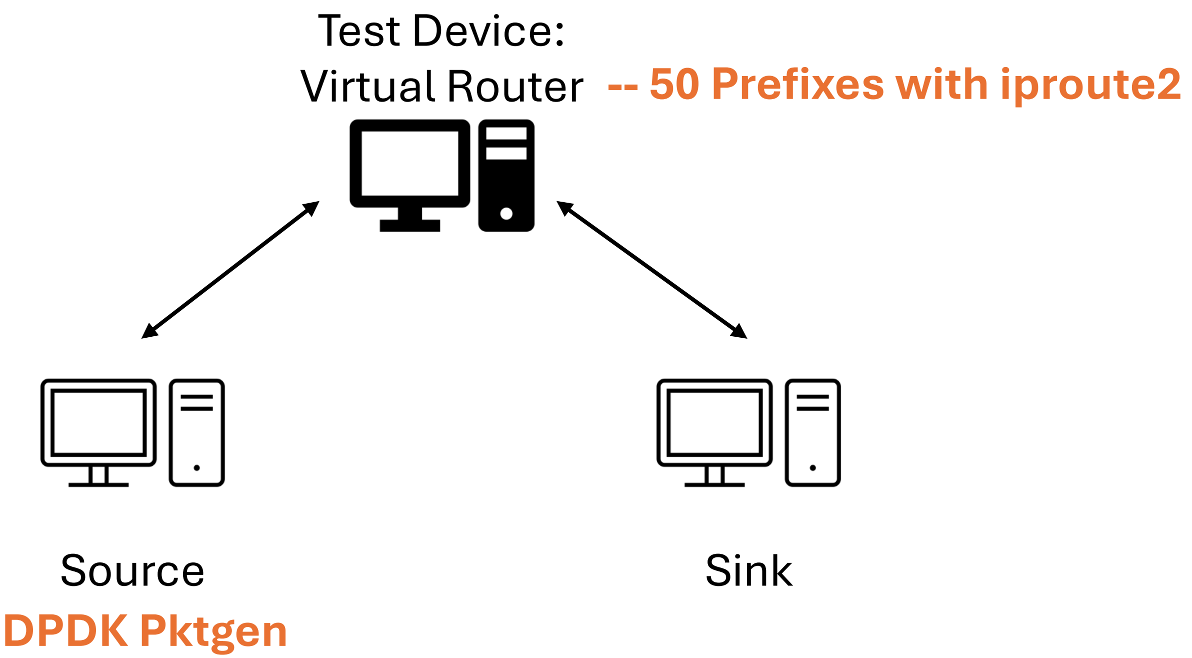
Virtual Network Functions: Virtual Router

Experimental Setup



Virtual Network Functions: Virtual Router

Experimental Setup

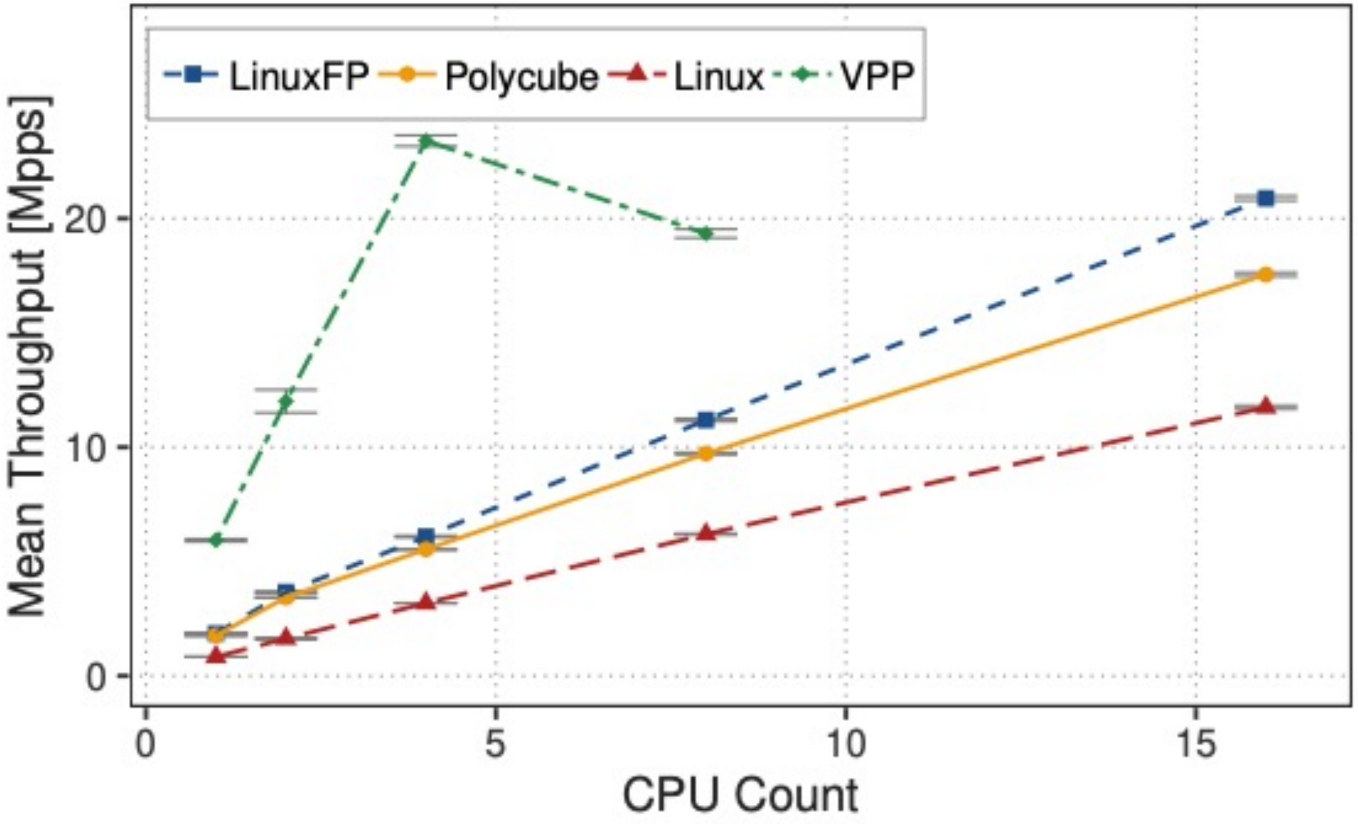


Virtual Network Functions: Virtual Router

Throughput: Number of Cores

Virtual Network Functions: Virtual Router

Throughput: Number of Cores

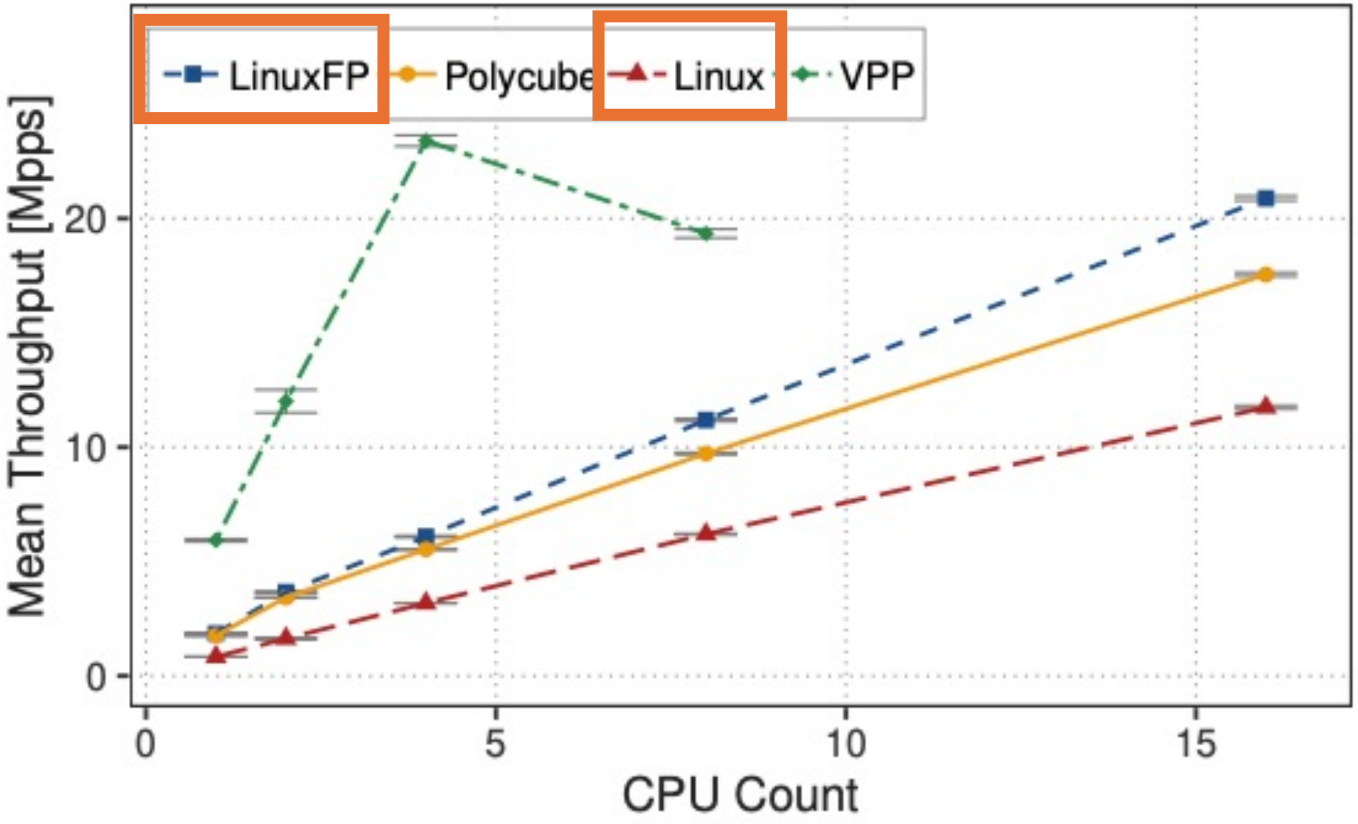


University of Colorado **Boulder**

PRINCETON UNIVERSITY

Virtual Network Functions: Virtual Router

Throughput: Number of Cores



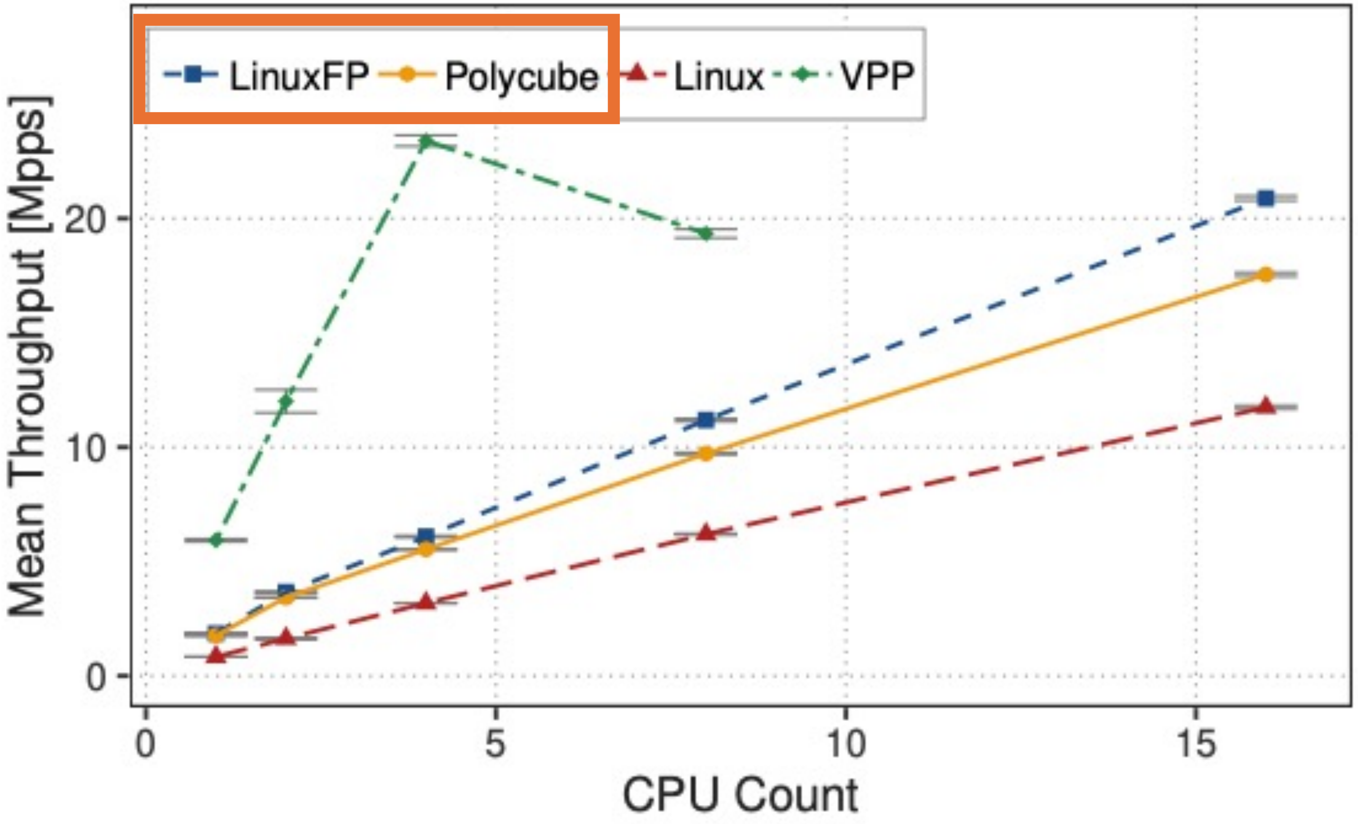
LinuxFP nearly doubles throughput compared to Linux

University of Colorado Boulder

PRINCETON UNIVERSITY

Virtual Network Functions: Virtual Router

Throughput: Number of Cores



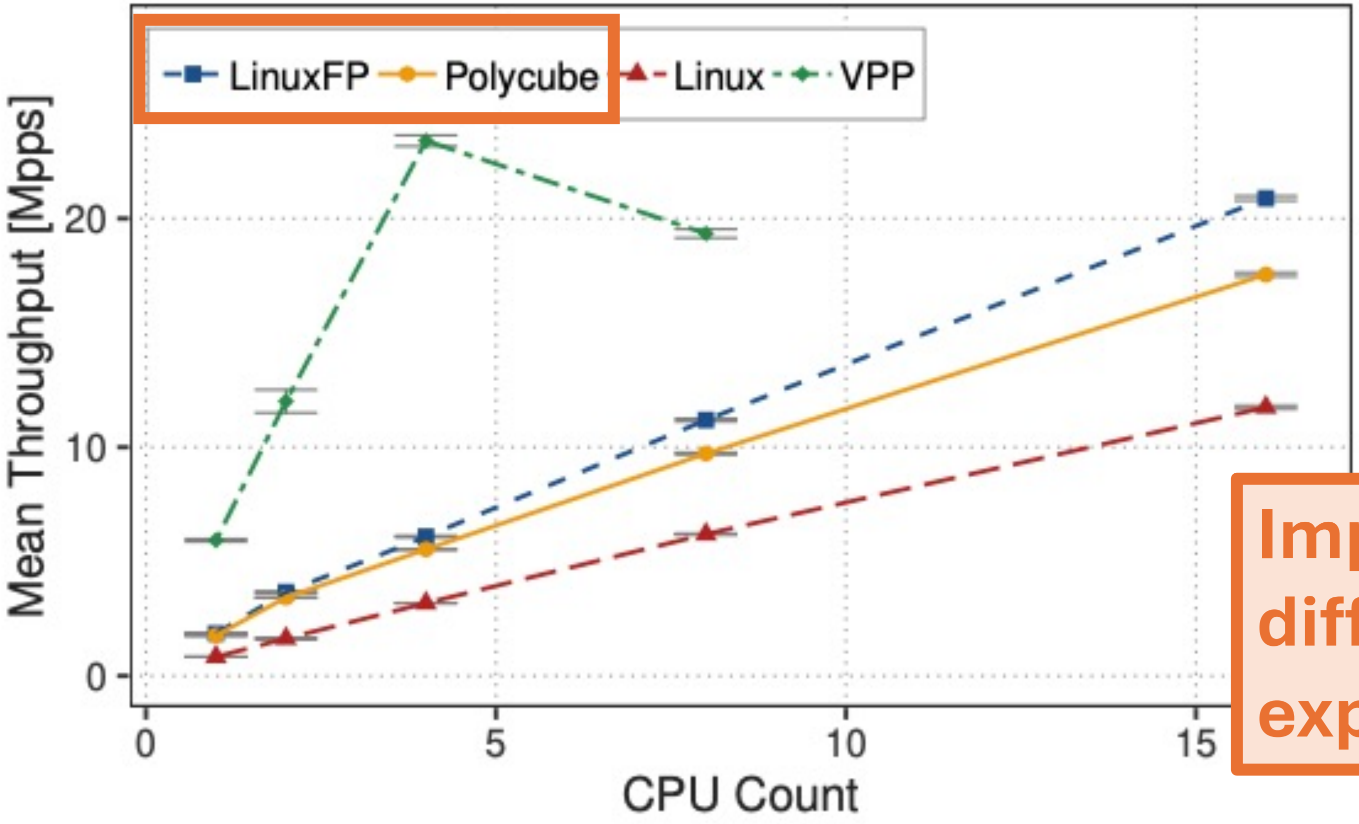
Comparable to Polycube, but LinuxFP keeps Linux API

University of Colorado **Boulder**

PRINCETON UNIVERSITY

Virtual Network Functions: Virtual Router

Throughput: Number of Cores

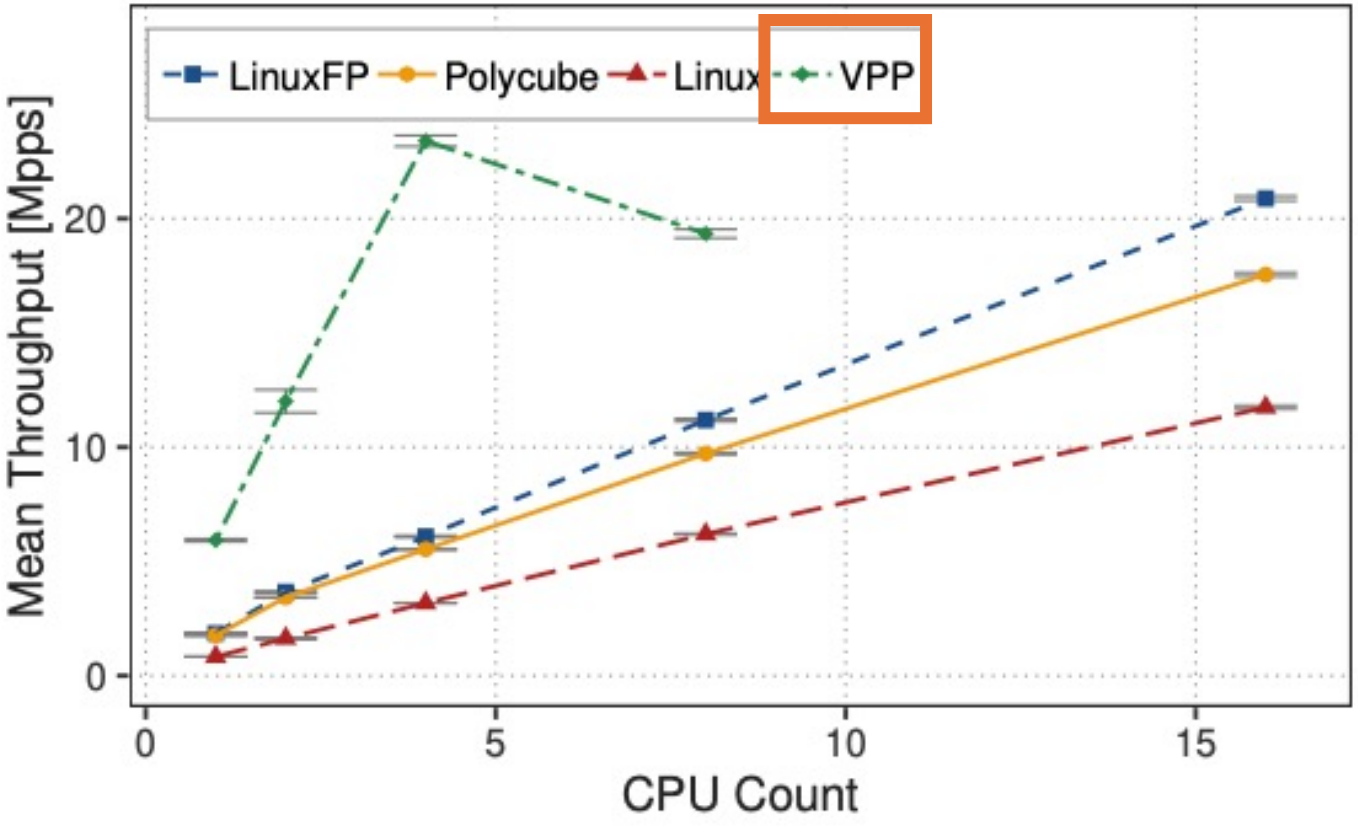


Comparable to Polycube, but LinuxFP keeps Linux API

Implementation differences explored in paper

Virtual Network Functions: Virtual Router

Throughput: Number of Cores



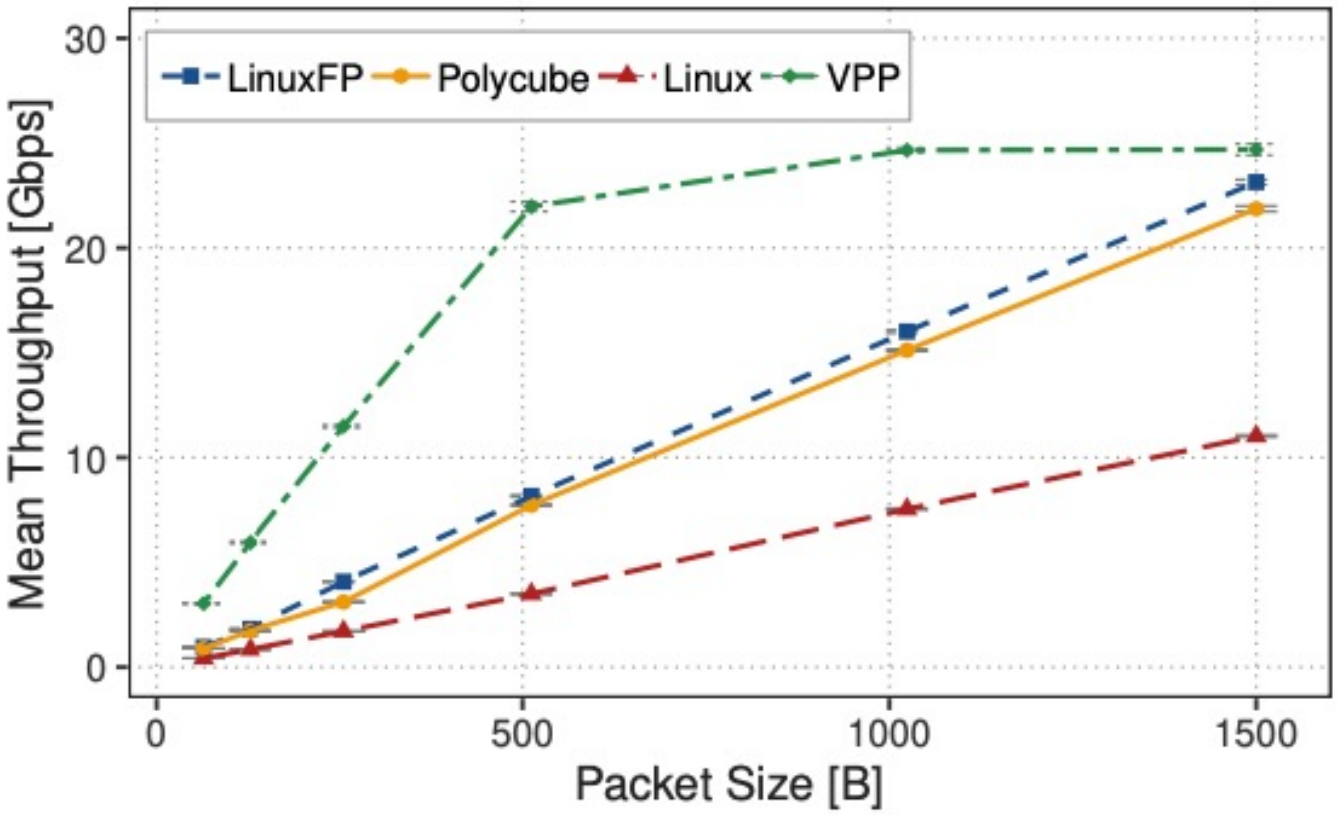
Vector processing (batching), Dedicated cores

Virtual Network Functions: Virtual Router

Single Core Throughput: Packet Size

Virtual Network Functions: Virtual Router

Single Core Throughput: Packet Size

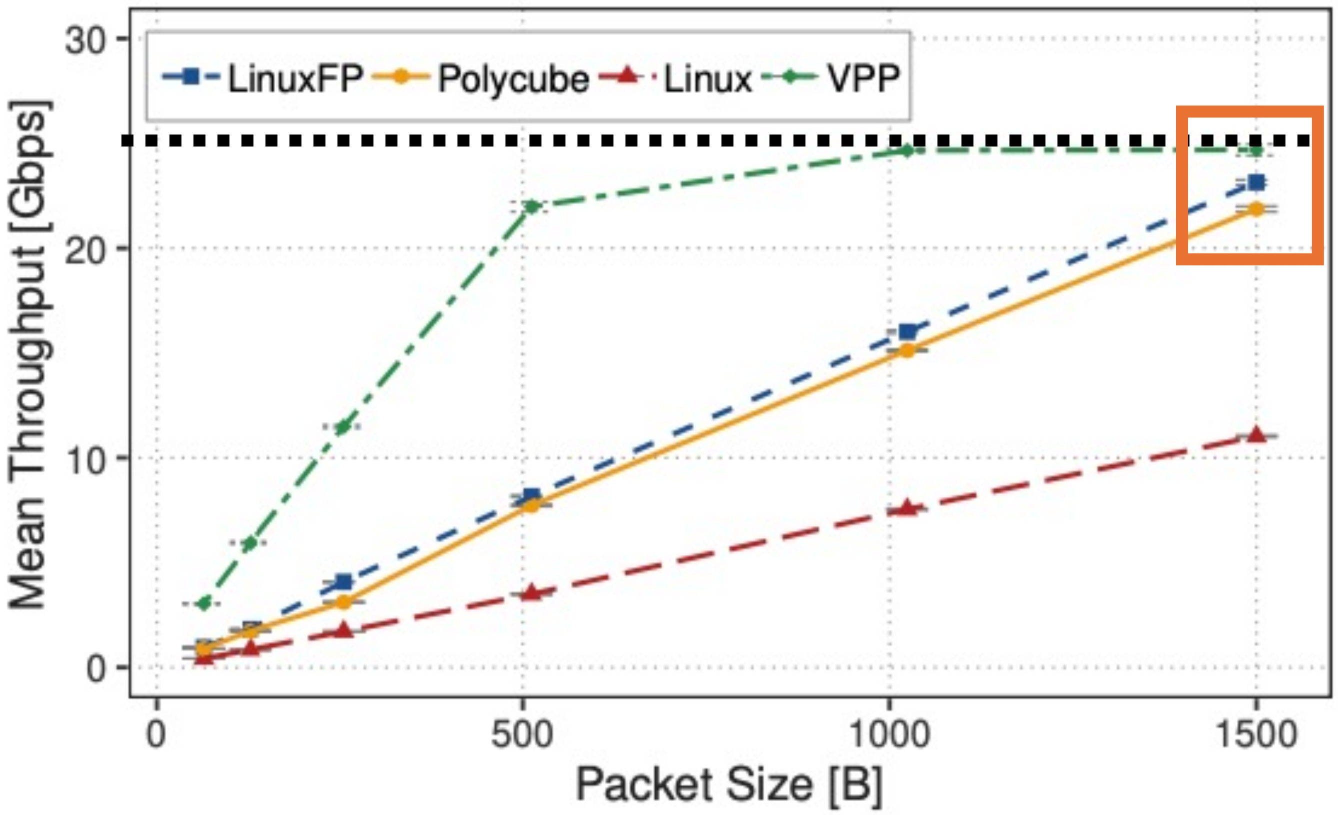


University of Colorado Boulder

PRINCETON UNIVERSITY

Virtual Network Functions: Virtual Router

Single Core Throughput: Packet Size

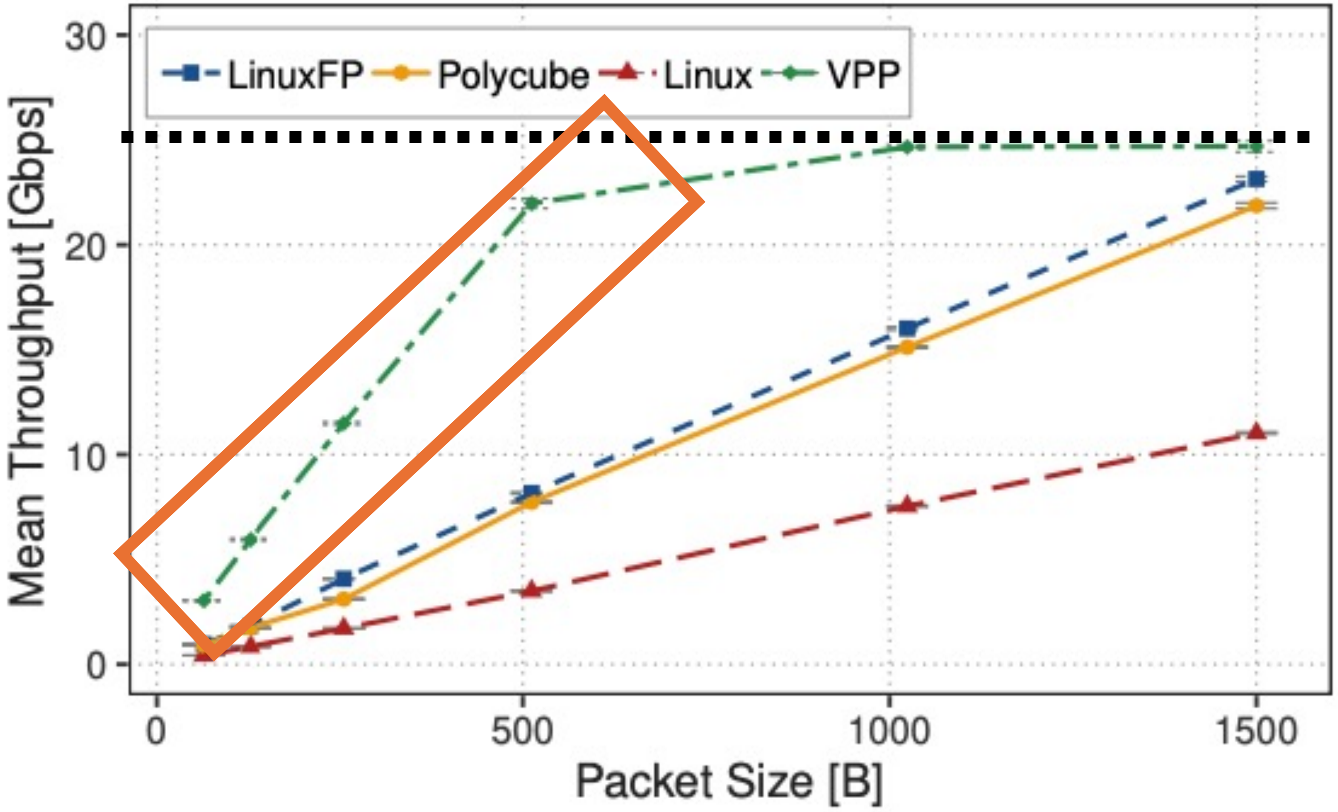


University of Colorado **Boulder**

PRINCETON UNIVERSITY

Virtual Network Functions: Virtual Router

Single Core Throughput: Packet Size



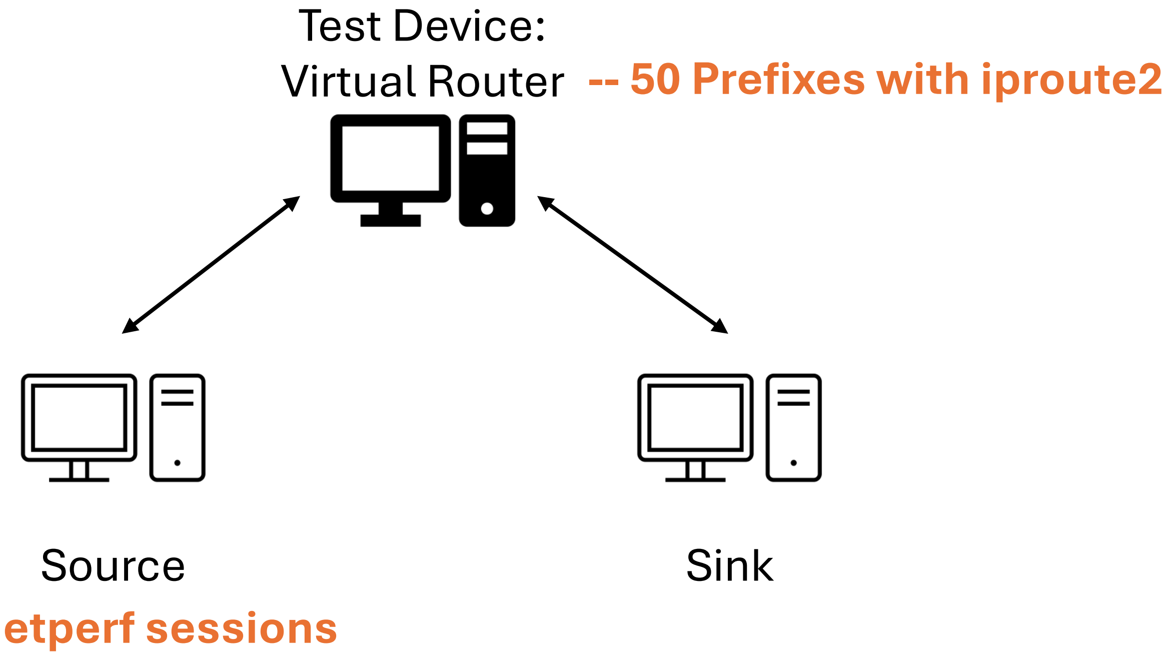
University of Colorado Boulder

PRINCETON UNIVERSITY

Virtual Network Functions: Virtual Router

Single Core Latency

Experimental Setup



Virtual Network Functions: Virtual Router

Single Core Latency

System	Average	99th Percentile	Standard Deviation
Linux	326.872 µS	512.378 µS	109.265 µS
Polycube	145.792 µS	269.772 µS	60.204 µS
VPP	85.604 µS	182.265 µS	32.011 µS
LinuxFP	151.675 µS	279.407 µS	76.798 µS

Virtual Network Functions: Virtual Router

Single Core Latency

System	Average	99th Percentile	Standard Deviation
Linux	326.872 µS	512.378 µS	109.265 µS
Polycube	145.792 µS	269.772 µS	60.204 µS
VPP	85.604 µS	182.265 µS	32.011 µS
LinuxFP	151.675 µS	279.407 µS	76.798 µS

Less than half of average latency of Linux

LinuxFP Evaluation

Enable fast packet processing

Virtual Network Functions: Router

Maintain compatibility with the Linux networking API

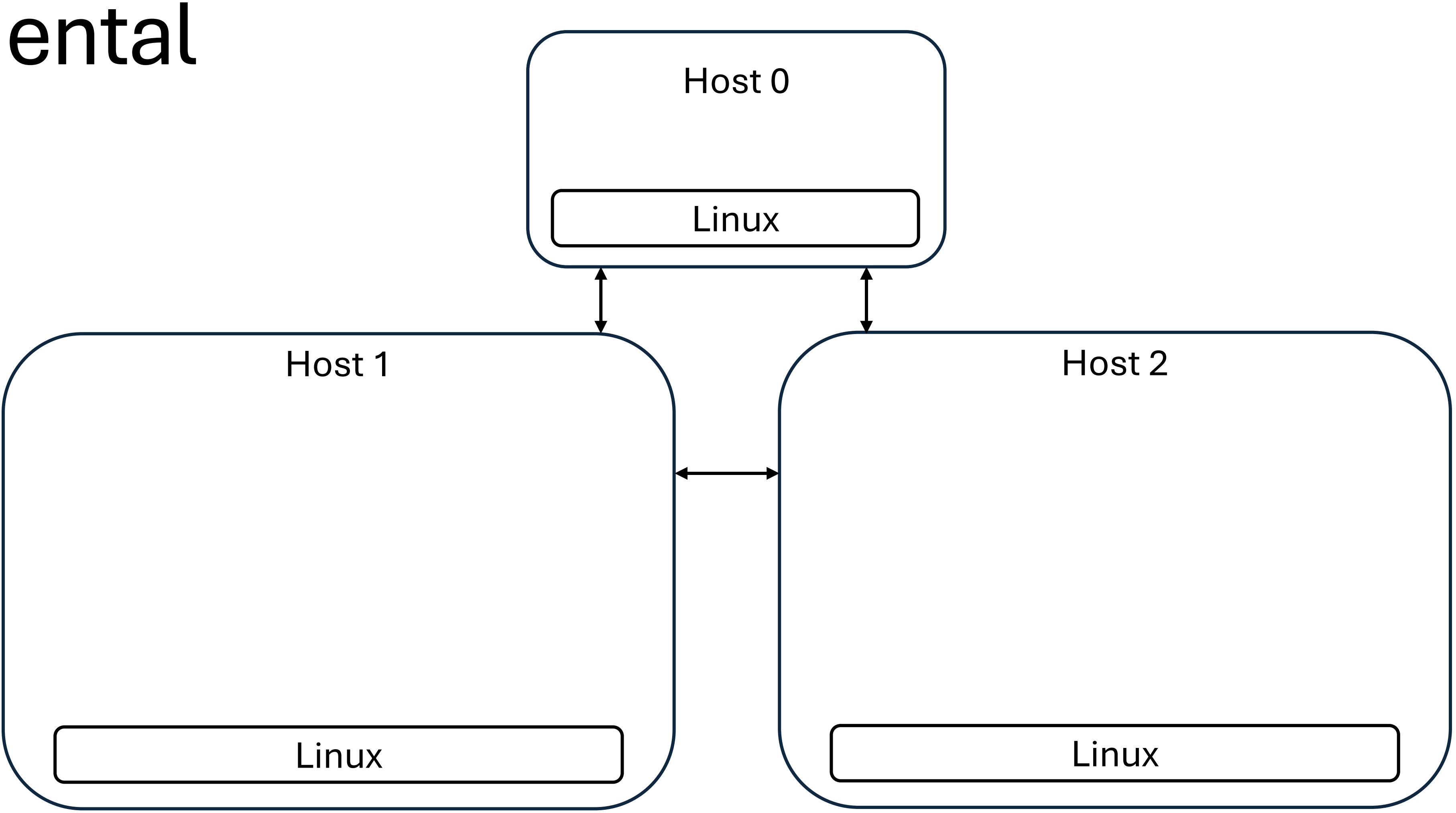
LinuxFP Evaluation

Enable fast packet processing

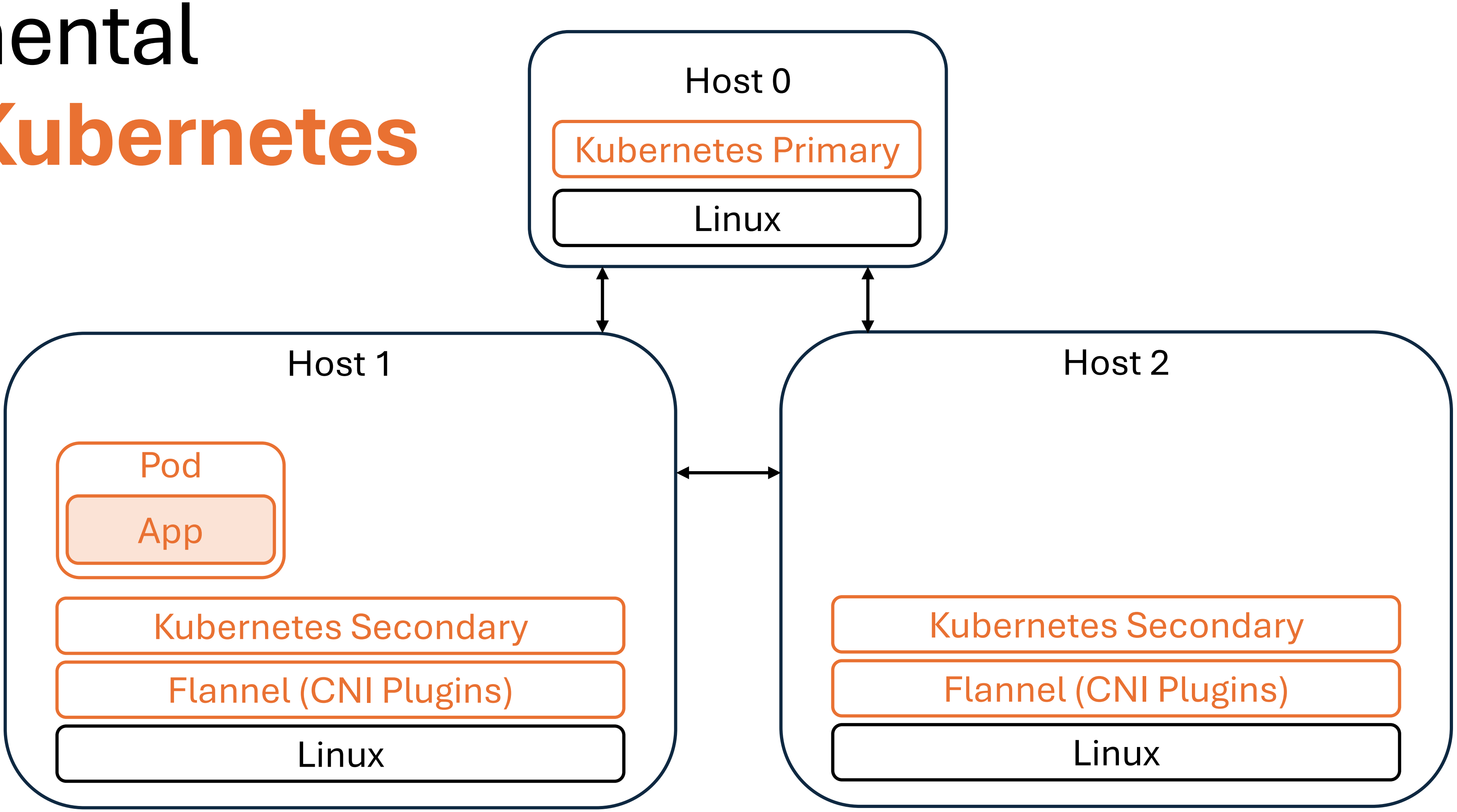
Maintain compatibility with the Linux networking API

Pod-to-Pod Networking with Kubernetes and Flannel

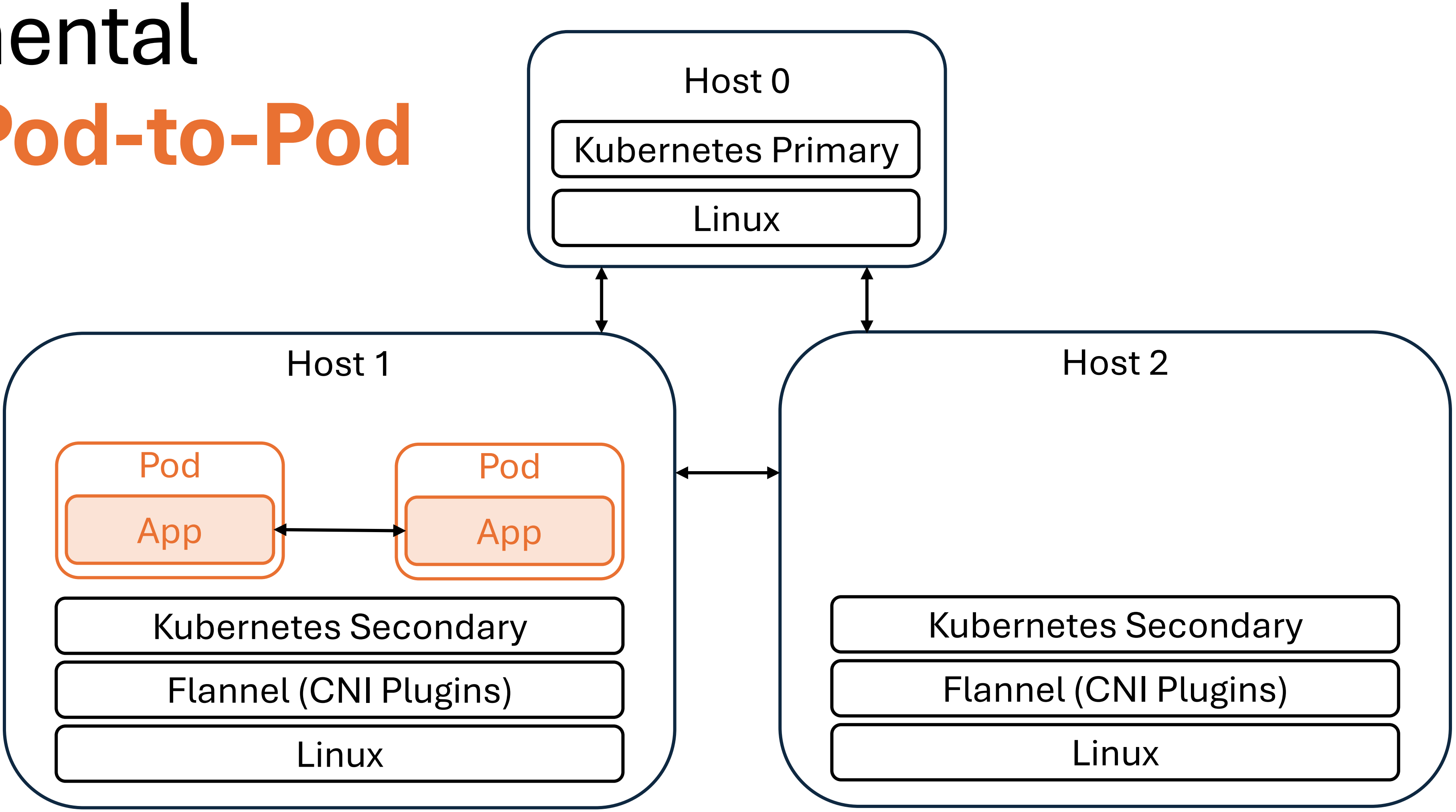
Experimental Setup



Experimental Setup: Kubernetes



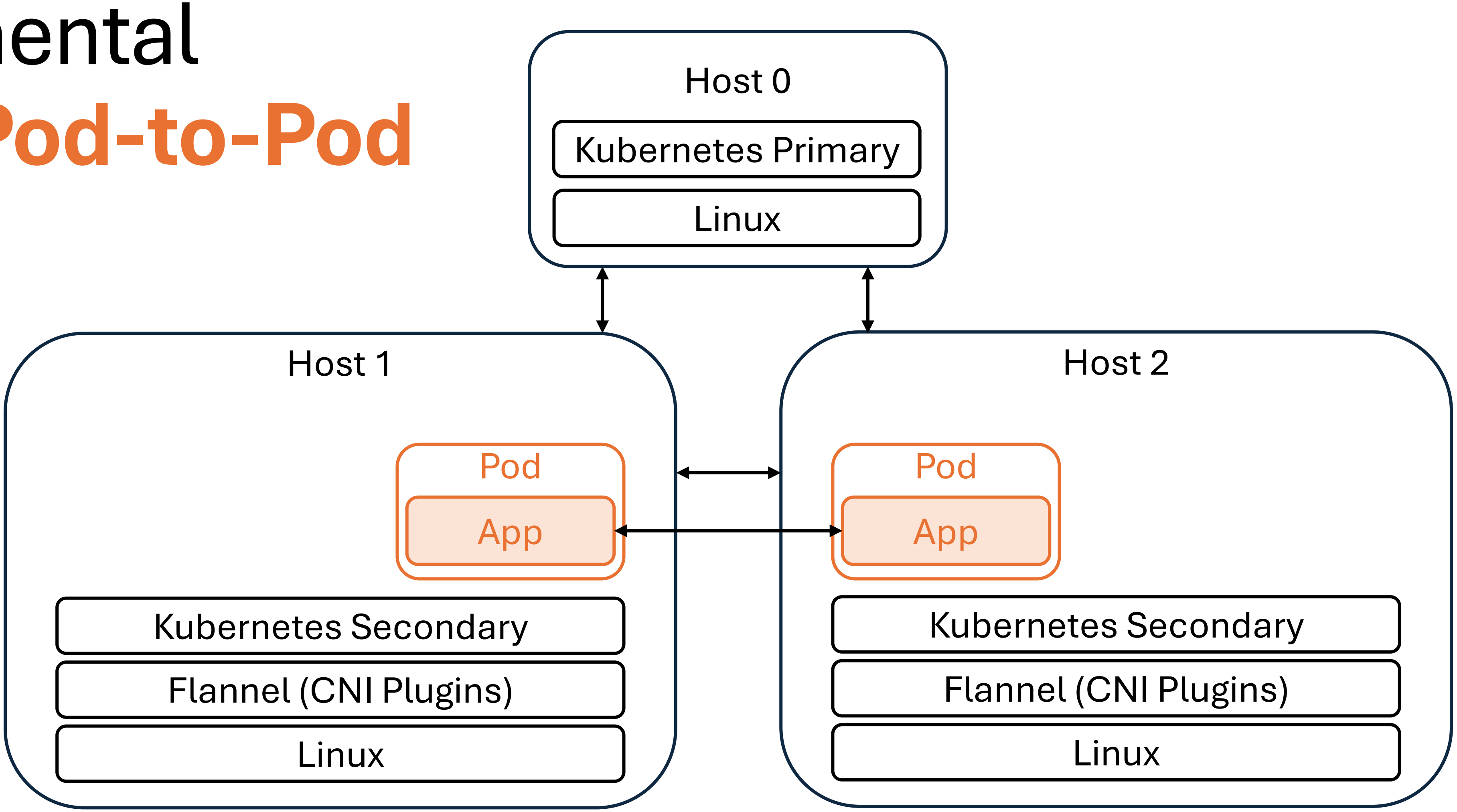
Experimental Setup: Pod-to-Pod



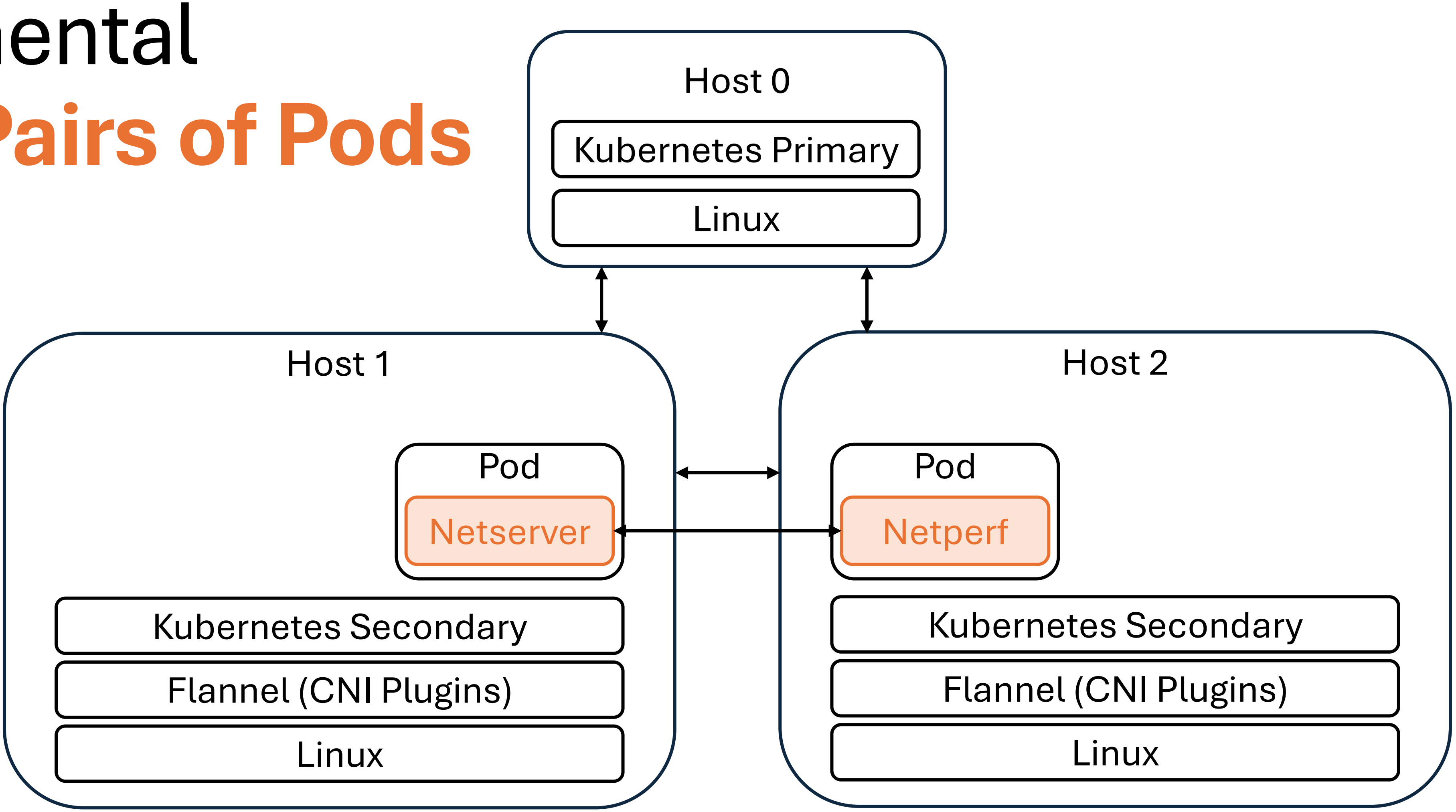
University of Colorado **Boulder**

PRINCETON UNIVERSITY

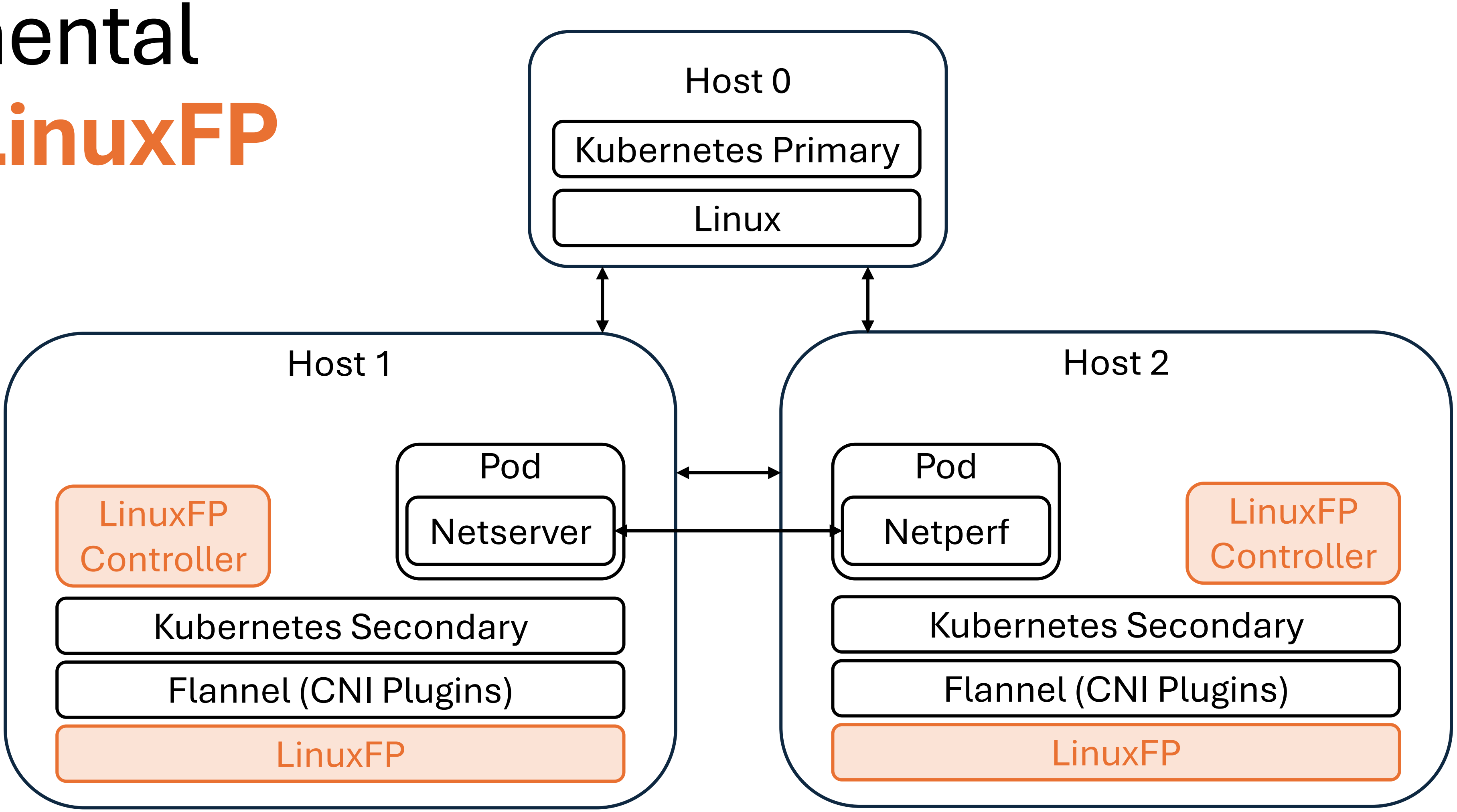
Experimental Setup: Pod-to-Pod



Experimental Setup: Pairs of Pods



Experimental Setup: LinuxFP

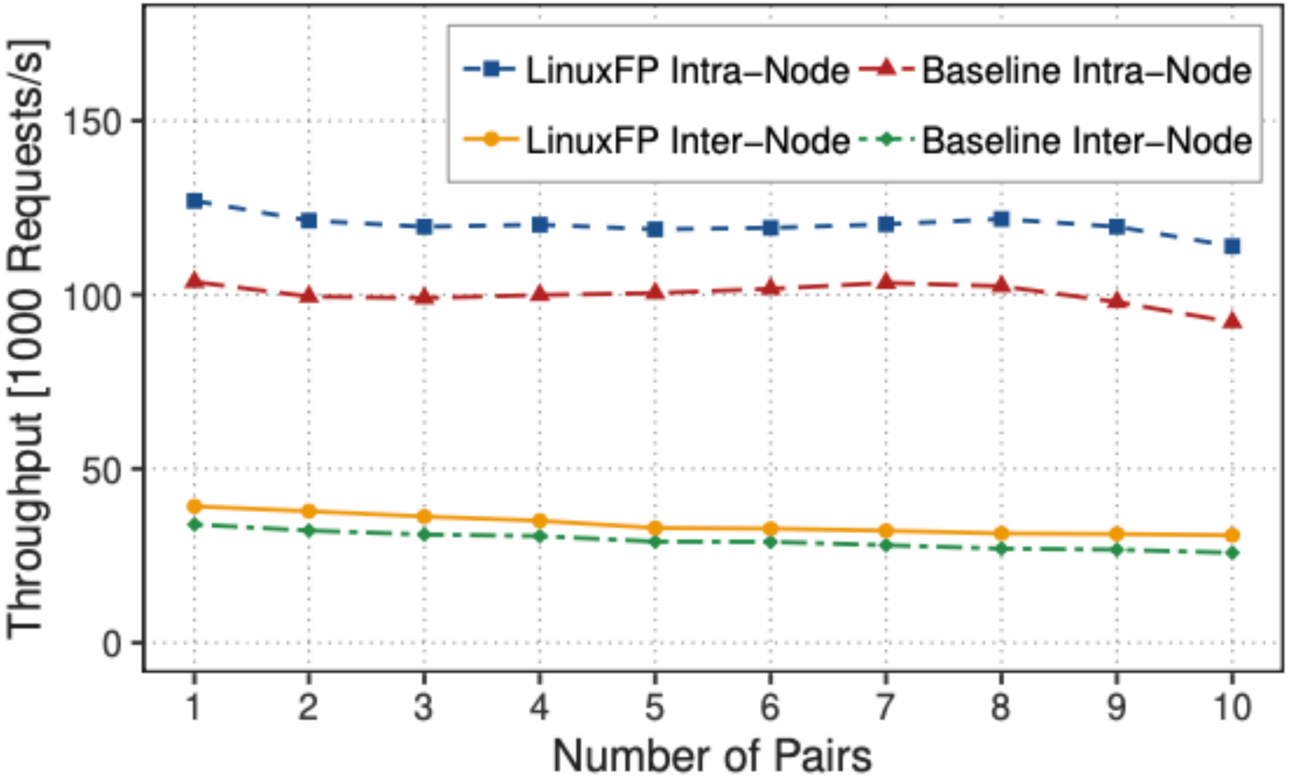


University of Colorado **Boulder**

PRINCETON UNIVERSITY

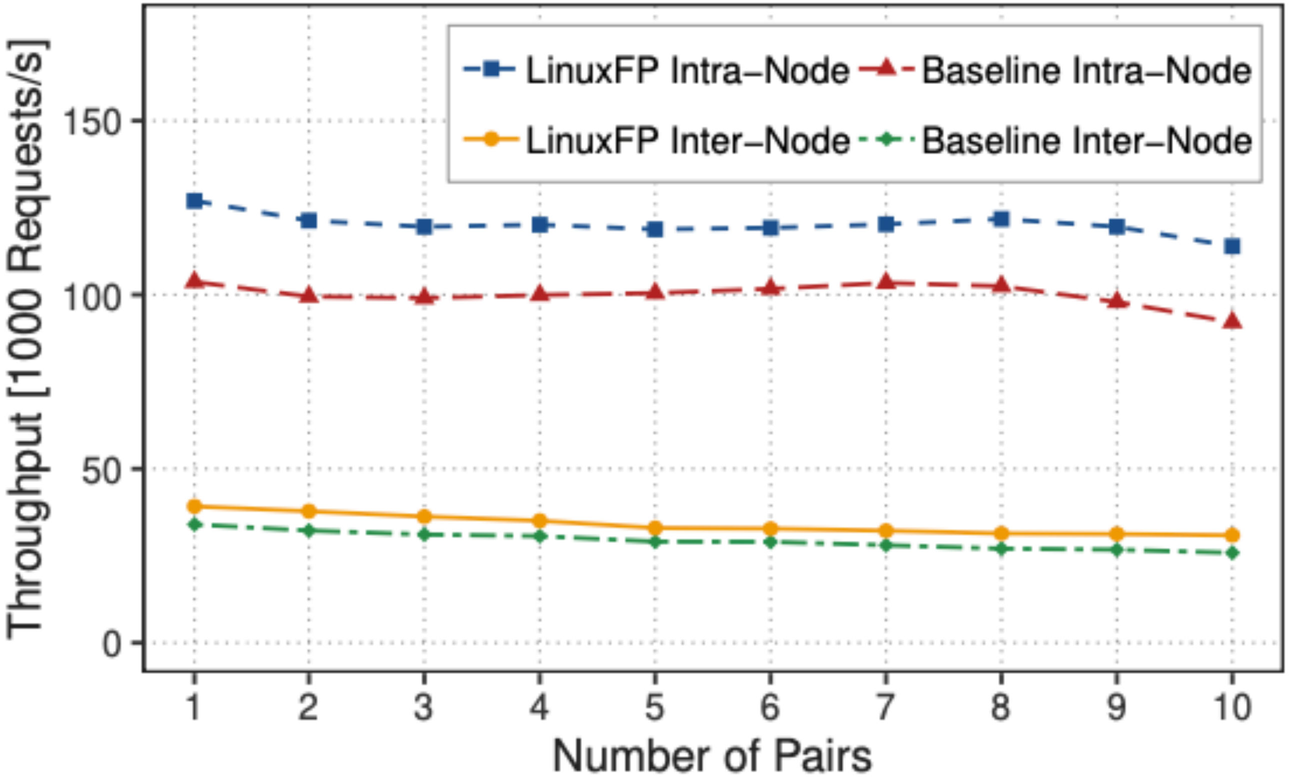
Kubernetes Pod-to-Pod

Throughput: Pairs of Pods



Kubernetes Pod-to-Pod

Throughput: Pairs of Pods



120% (intra) of Linux tput

116% (inter) of Linux tput

University of Colorado **Boulder**

PRINCETON UNIVERSITY

LinuxFP

- **Transparently** enables **accelerated** packet processing while:
  - Maintaining compatibility with the Linux networking API
  - Maintaining access to the breadth of the Linux networking stack

Questions?

ICDCS 2024

Jersey City, New Jersey USA

Erika Hunhoff (erika.hunhoff@colorado.edu)

Thank you!

- Marcelo Abranches
- Rohan Eswara
- Oliver Michel
- Eric Keller

- LinuxFP is available at: github.com/mcabranches/tna
- BPF Kernel Helper functions available at: github.com/mcabranches/linux

University of Colorado **Boulder** PRINCETON UNIVERSITY