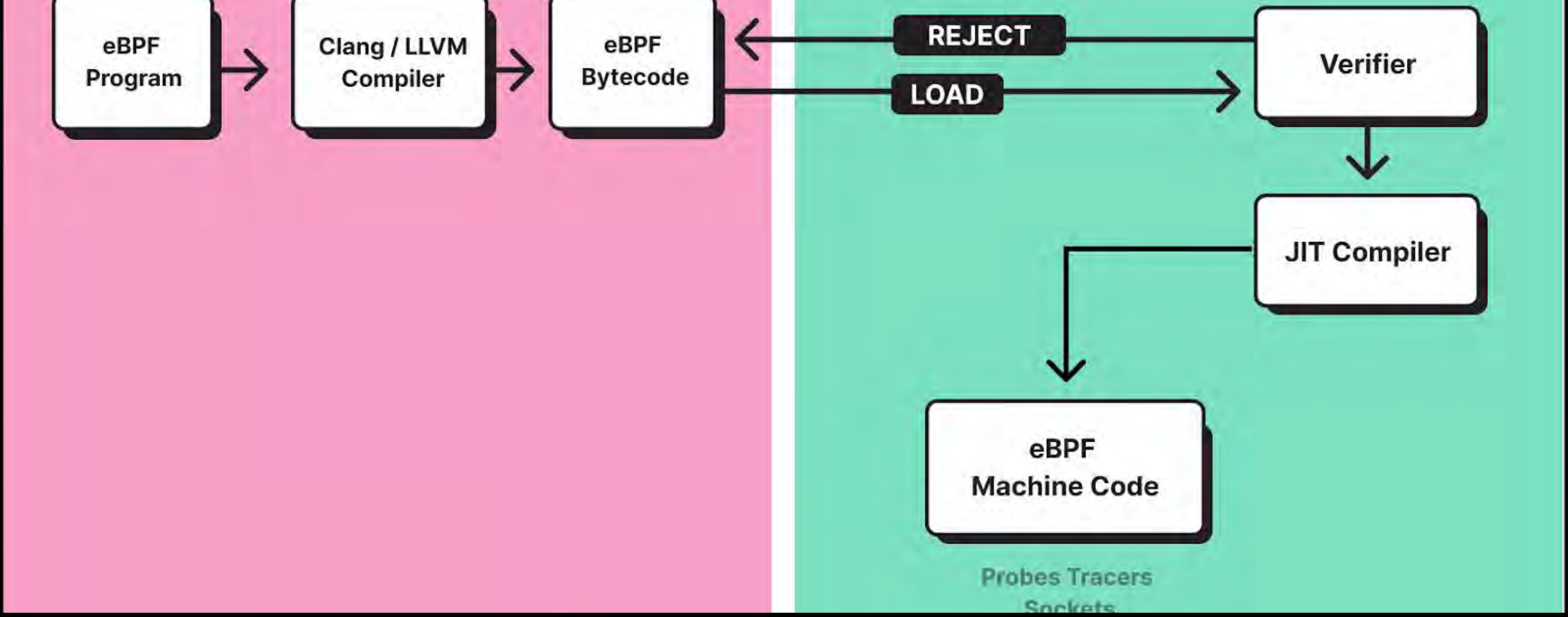



# Observability in Kubernetes: Unlocking the Power of eBPF

Explore how extended Berkeley Packet Filter (eBPF) empowers observability, security, and performance in Kubernetes-based container systems.

# Who am I?

**Alejandro Mercado Peña**

Automation, DevOps and Chaos engineer

Observability is my thing.

Technical Content Writer and International Speaker

Dad & Cats

Currently Living in México.

CDF Ambassador

eBPF enthusiast

My blog:

https://medium.com/@alexmarket

Linkedin

https://www.linkedin.com/in/alexmarket/

# Observability is another Data Problem

# Introduction to Kubernetes

### High-level API and Components

Kubernetes provides a high-level API and a set of components that abstract away low-level system details.

### Hides System-level Complexity

Application developers don't need to know about IP tables, cgroups, namespaces, seccomp, or container runtimes.

### Relies on Linux Functionalities

Though Kubernetes abstracts away many details, it heavily relies on core Linux functionalities underneath.

**Kubernetes provides a high-level, abstracted view of container orchestration, but its inner workings are deeply rooted in the powerful functionalities of the Linux operating system.**

# What is BPF?

- **eBPF: A Mini-VM in the Linux Kernel**

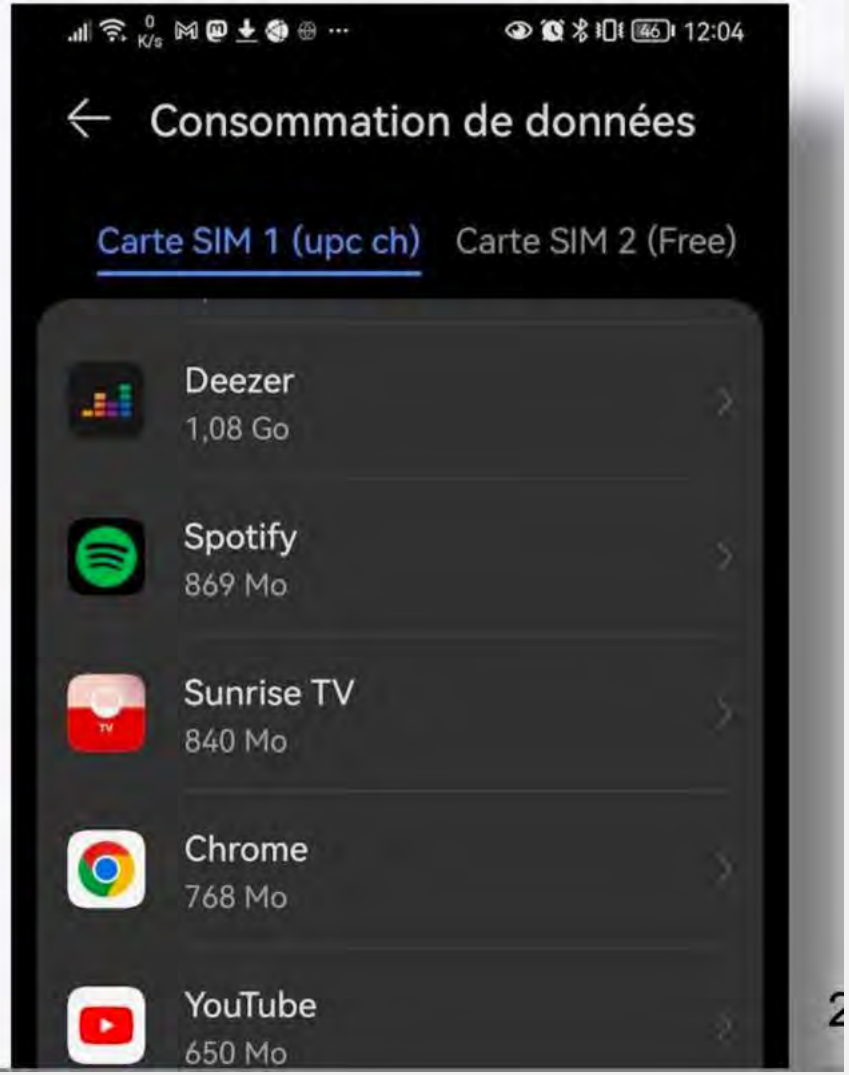
  BPF is a mini-virtual machine that resides within the Linux kernel, allowing for the execution of BPF programs.

- **Event-Driven Functionality**

  BPF programs are event-driven, meaning they are executed when specific events occur within the kernel.

- **Kernel Object Attachment**

  BPF programs can be attached to various kernel objects, such as sockets, kprobes, tracepoints, and network schedulers, enabling interaction with the kernel.




# Have you ever used eBPF?

## eBPF is low-visibility but ubiquitous

- Load balancing & DDoS protection on major websites
- Data stats app on Android
- Kubernetes Networks
- systemd

NETFLIX FACEBOOK Google Microsoft

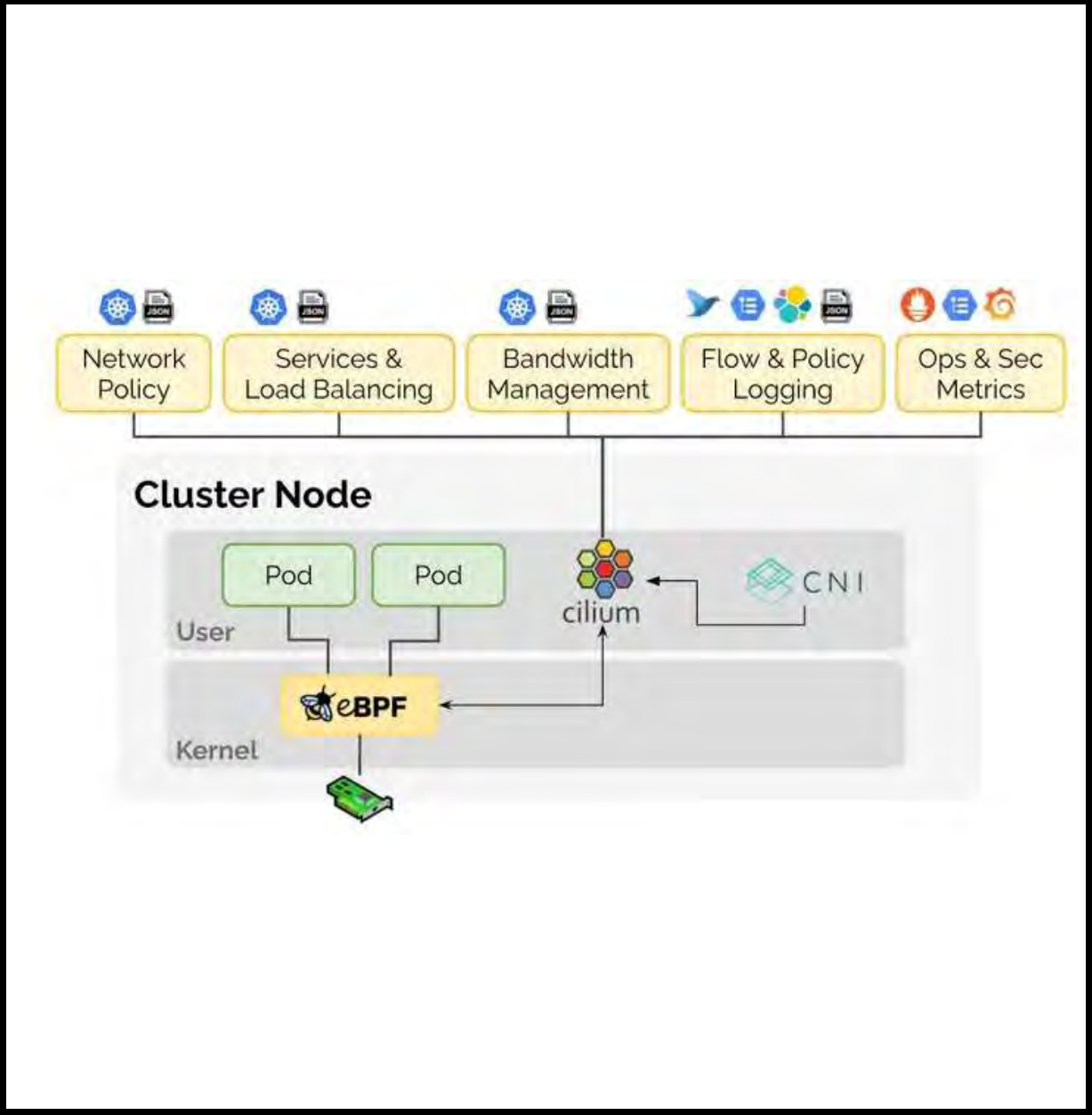
# The Kubernetes Ecosystem and eBPF

With Kubernetes:

- Network: Cilium
- Service mesh : Kuma, Cilium Service Mesh
- System security: Falco, Tetragon, Tracee
- Observability: Pixie

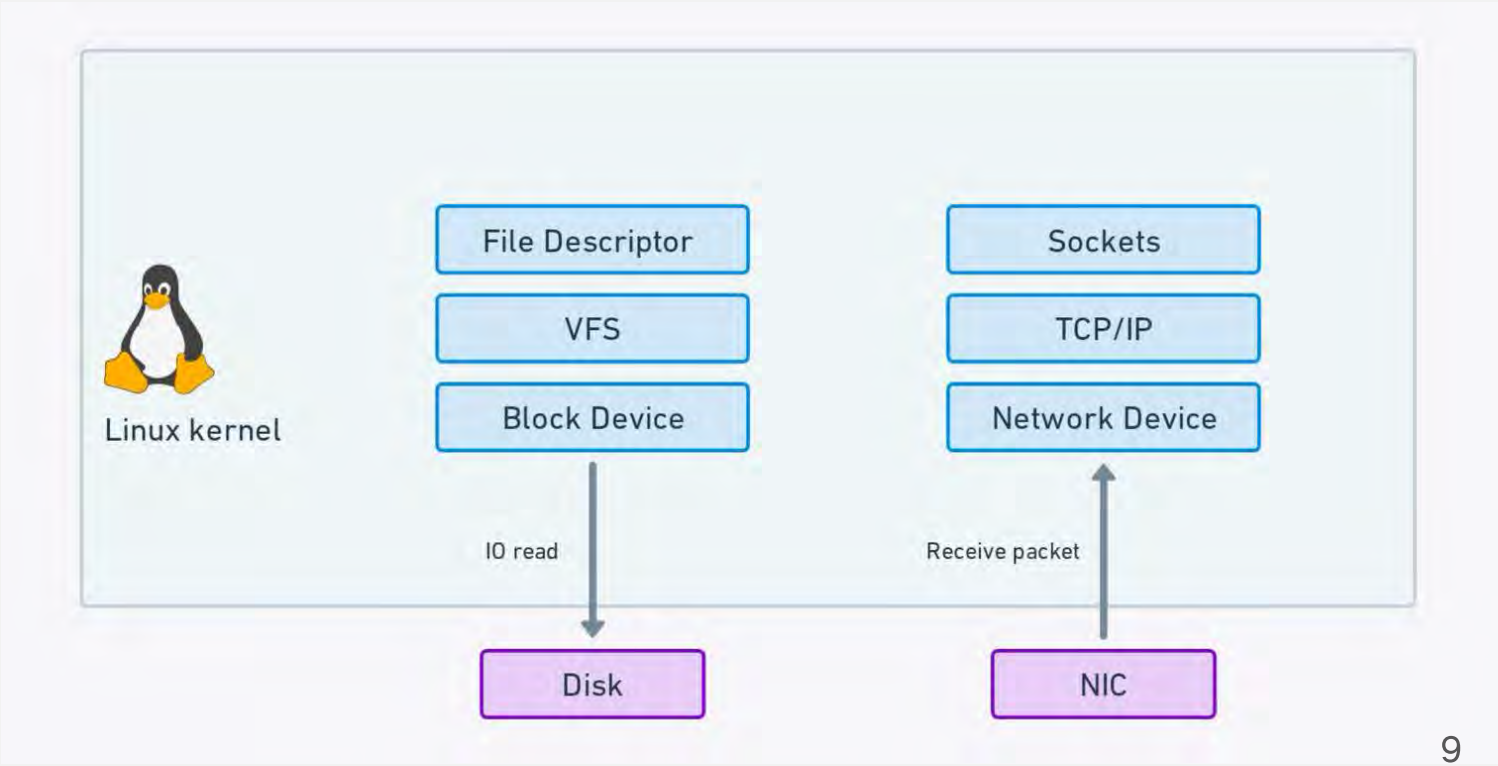
# The Power of eBPF

eBPF (extended Berkeley Packet Filter) is a powerful Linux kernel technology that extends the capabilities of the classic BPF. It allows attaching programs to various kernel objects, including kprobes, tracepoints, and network schedulers. This gives developers and system administrators unprecedented visibility and control over the system's behavior.

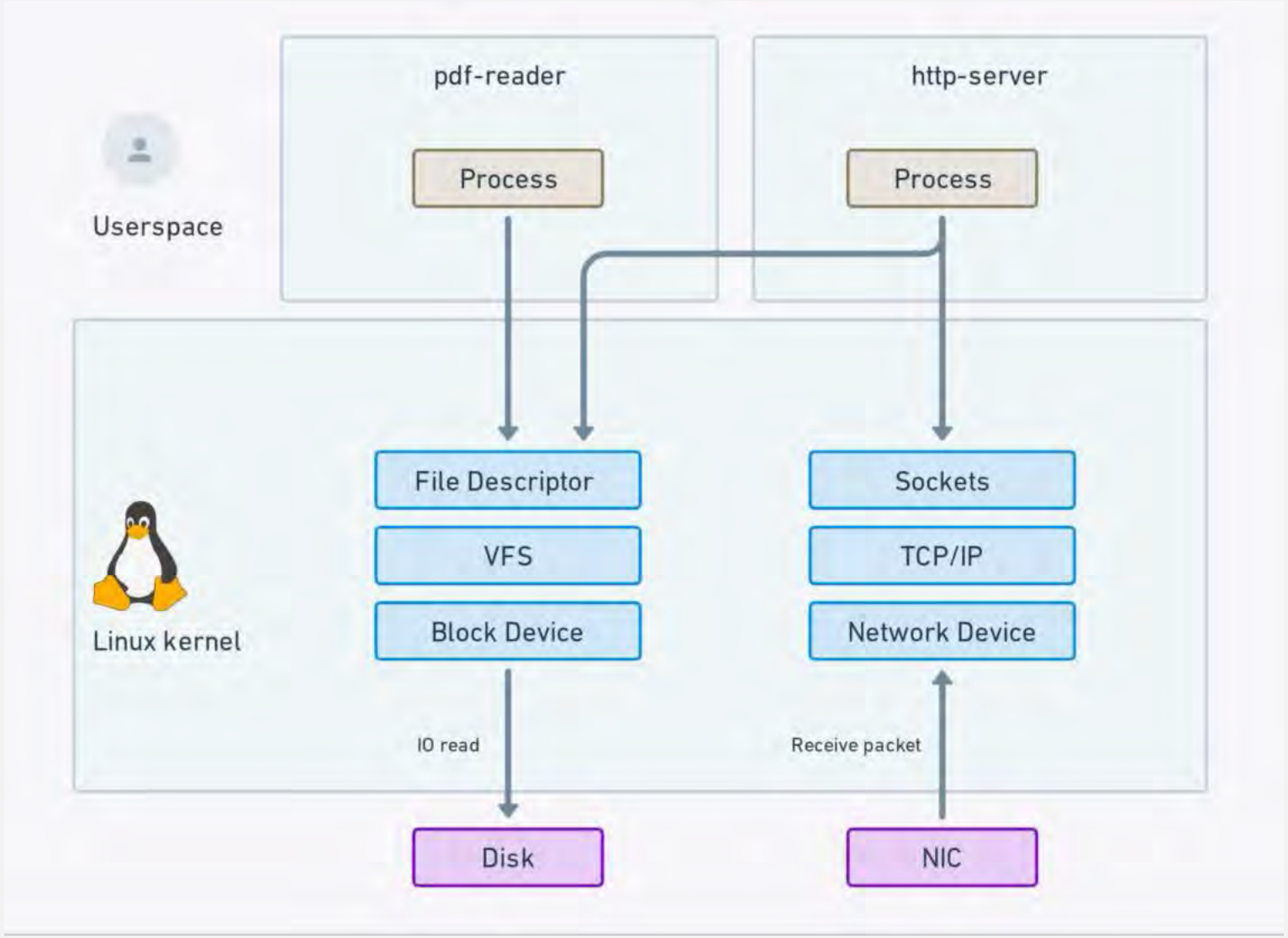


# Kernel Space

- Very privileged
- Manages access to physical resources
  - Ex. memory, disk, network
- Exposes these resources in abstraction forms
  - Ex. files, sockets, processes
- Critical component



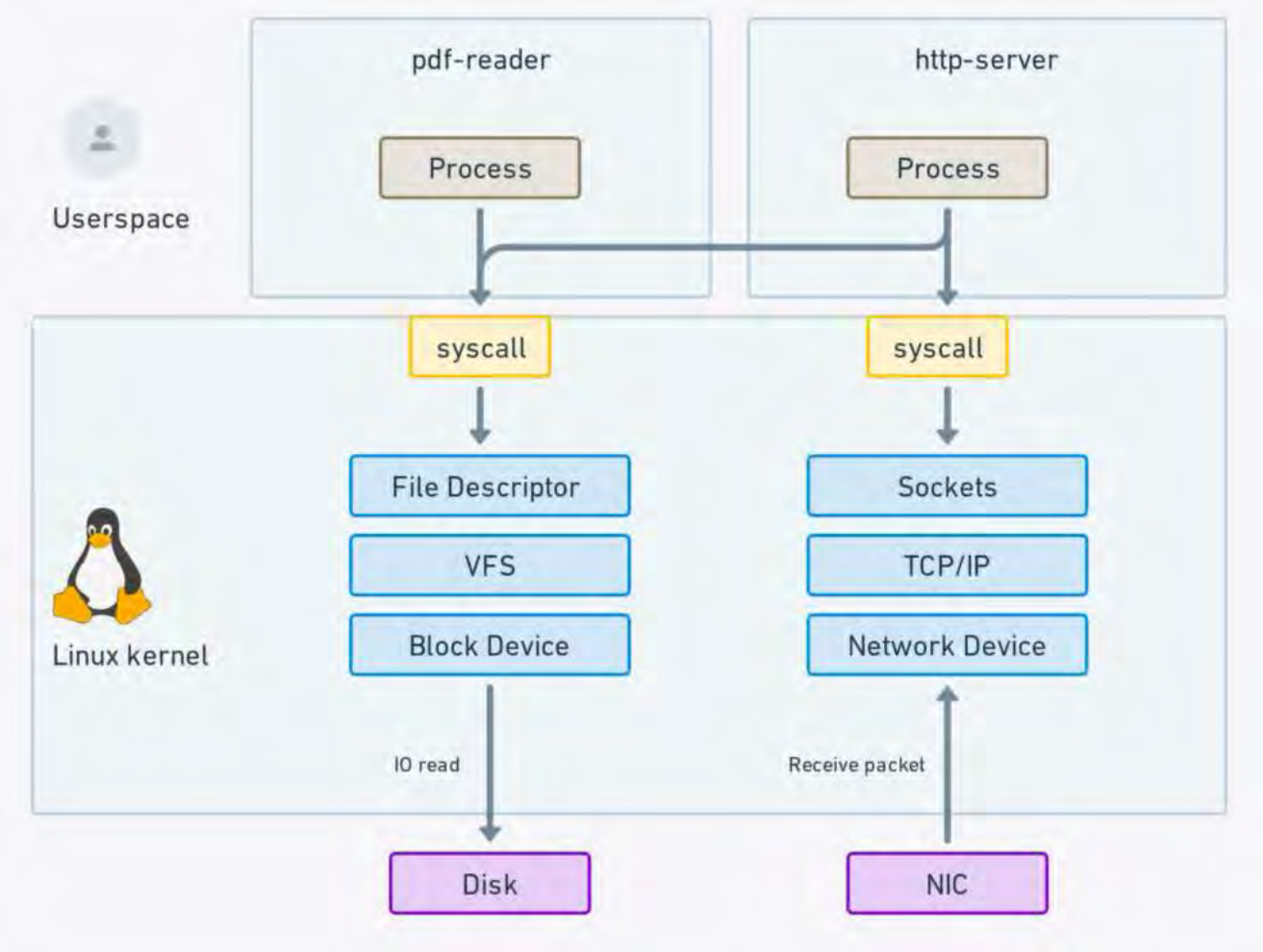
# Kernel Space and Userspace




## Userspace

- Application process domain
- All access to resources must go through the kernel

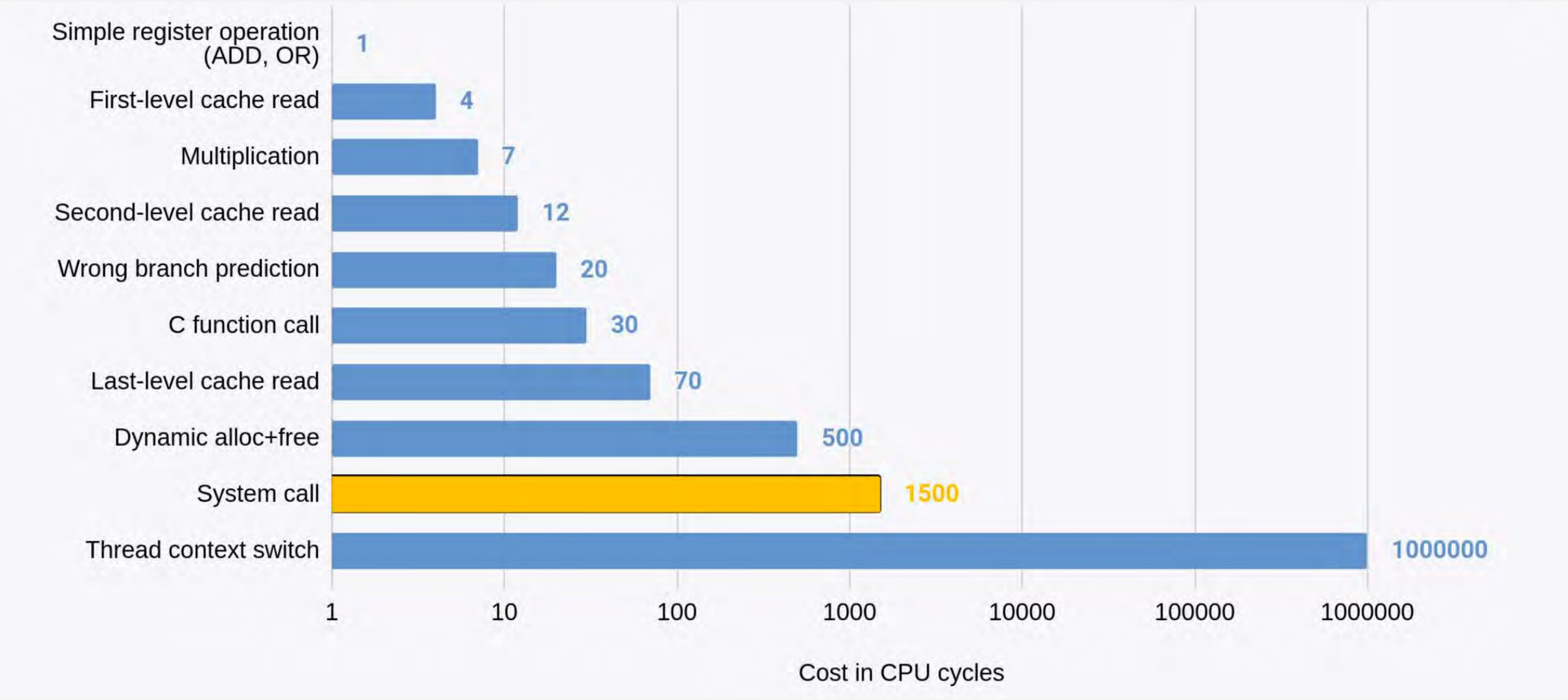
# System Calls




## Syscalls

- Main interface between kernel and userspace
- Asks the kernel to perform a task
  - Open a file
  - Read what was received on a network socket
  - ...
- Very common and quite expensive

# Cost of System Calls




http://ithare.com/infographics-operation-costs-in-cpu-clock-cycles/

# Kernel and userspace

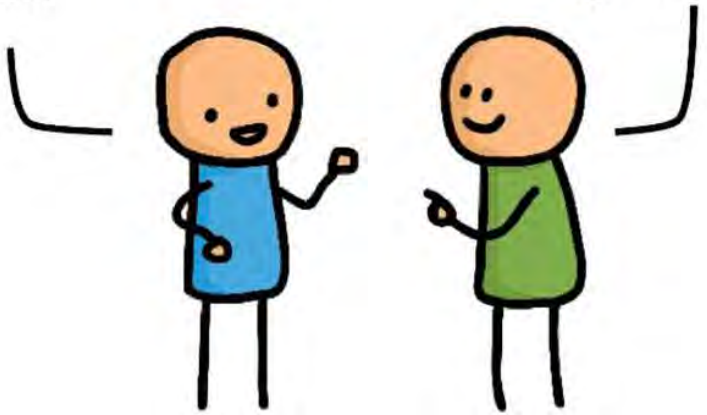
- **Applications may want new kernel features**
  - New network protocol
  - New load balancing algorithm
  - Traffic redirection for a sidecar container
  - .....
  - Generally 2 options:
  - Ask the kernel to send everything to the application
    - Ex. all Ethernet traffic to implement a new protocol
    - Very expensive
  - Implement in the kernel...

Application Developer:

i want this new feature to observe my app

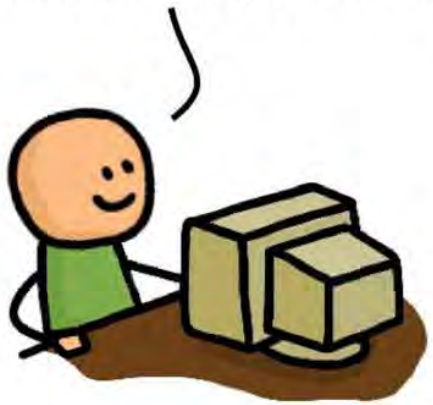
Hey kernel developer! Please add this new feature to the Linux kernel

OK! Just give me a year to convince the entire community that this is good for everyone.



1 year later...

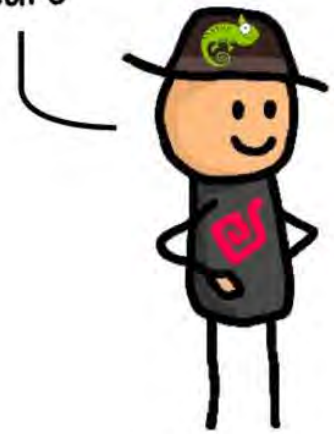
i'm done. The upstream kernel now supports this.



But i need this in my Linux distro

5 year later...

Good news. Our Linux distribution now ships a kernel with your required feature



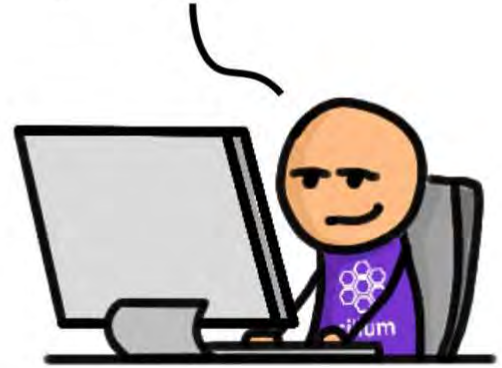
OK but my requirements have changed since...

Application Developer:

i want this new feature to observe my app

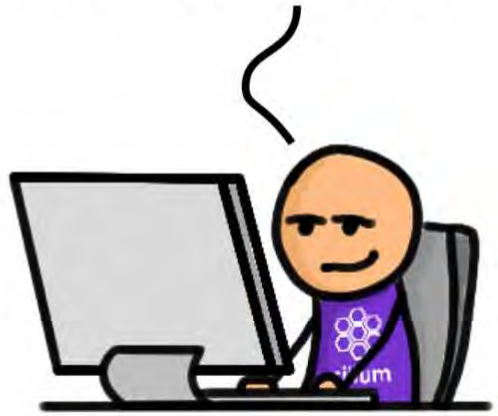
eBPF Developer:

OK! The kernel can't do this so let me quickly solve this with eBPF.



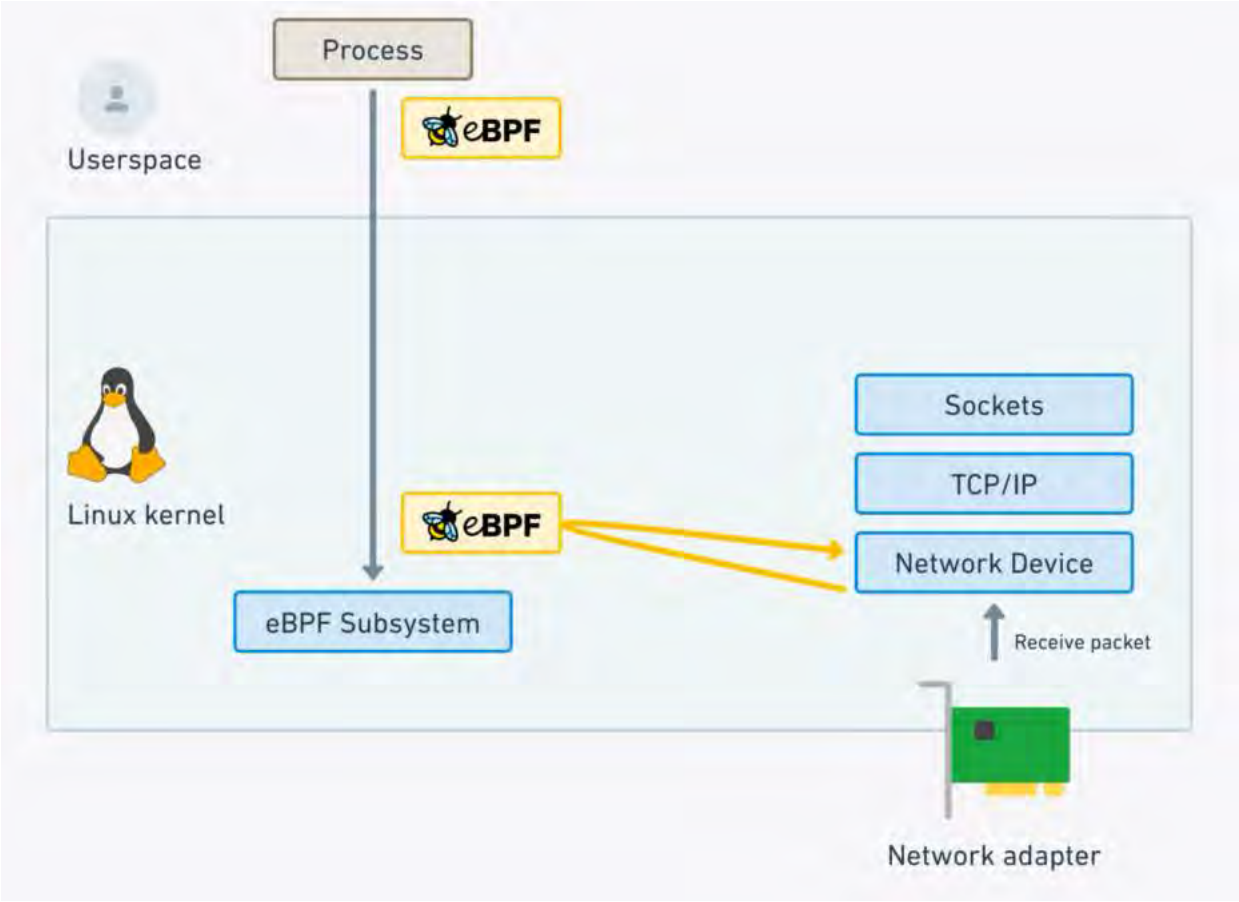
A couple of days later...

Here is a release of our eBPF project that has this feature now. BTW, you don't have to reboot your machine.



Made by Vadim Shchekoldin

# Program the kernel




- ● Program loaded into the kernel
- ● Attached to events
- ○ Receiving packets
- ○ Calling kernel functions
- ○ ...
- ● Executed for each event

# “eBPF is dangerous”

# NOT really

- Bug in the verifier => access to all memory
- Code executed in a highly privileged context, that of the kernel

## More

- By default, admin privileges are required to load a program _(ツ)_/
- ...but how you use it can
  - Ex. attach to all kernel functions

It's also easy to block, for example, the entire network or all syscalls.

Harder to crash the kernel by mistake, but possible

# Existing eBPF Use Cases in Kubernetes

### Cilium: Dynamic Network Control and Visibility

Cilium uses eBPF to generate and apply network rules dynamically, without modifying the Linux kernel.

### Weave Scope: Tracking TCP Connections

Weave Scope employs eBPF to accurately track TCP connections between containers.

### Limiting syscalls with seccomp-bpf

Seccomp-bpf allows applying custom filters in the form of BPF programs to limit the set of syscalls an application can use.

**Kubernetes tools are increasingly leveraging the power of eBPF to enhance networking, monitoring, and security capabilities within the container-based ecosystem.**

# Cilium: Dynamic Network Control and Visibility

## Cilium Overview

Cilium is a networking project that heavily utilizes eBPF to provide dynamic network control and visibility for container-based systems.

## eBPF-Powered Networking

Cilium leverages eBPF to translate network policy definitions into BPF programs that are loaded into the kernel and attached to containers' virtual ethernet devices. These programs are executed on each packet, applying the necessary rules.

## Flexible Attachment Points

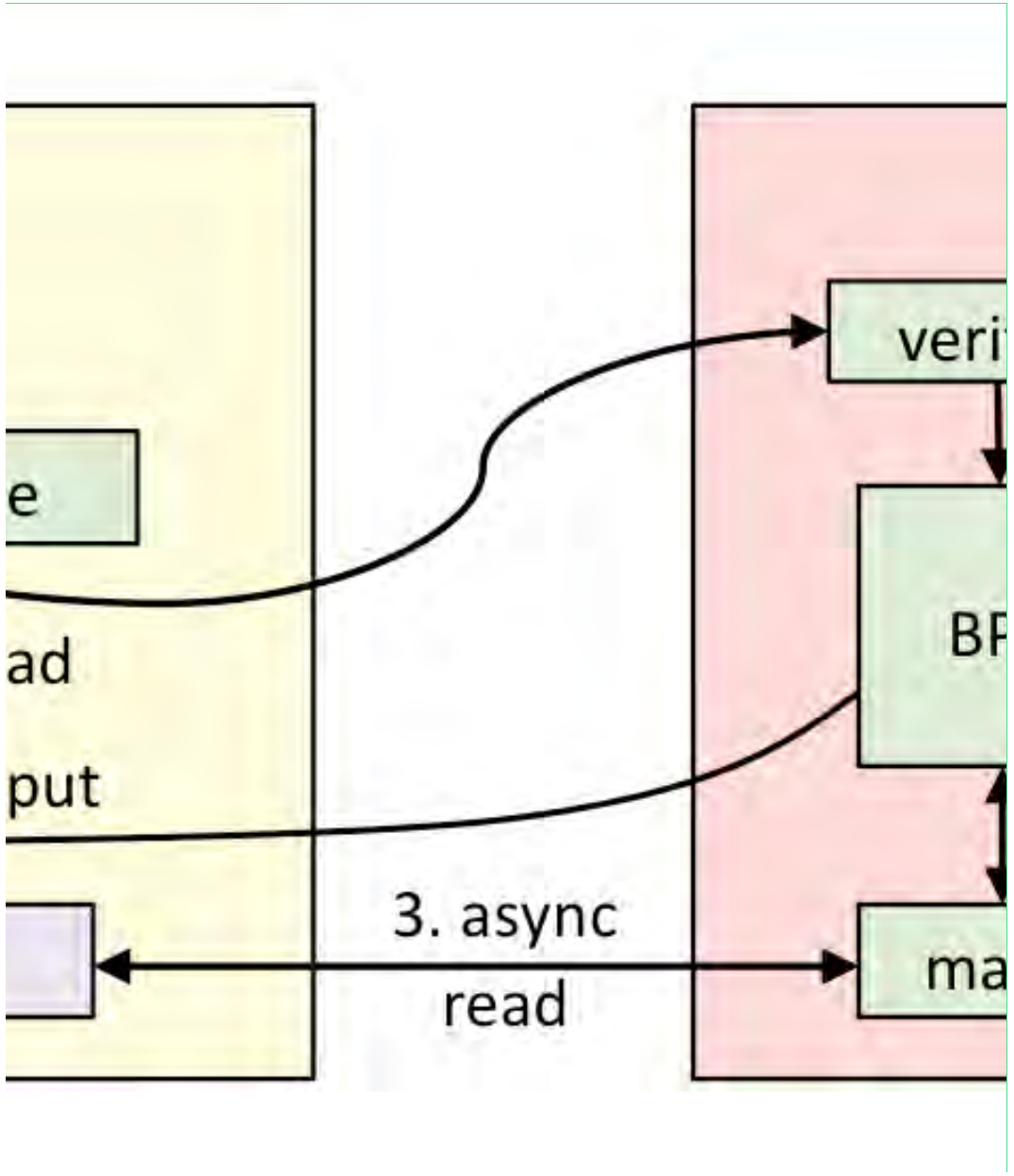
Depending on the network rules, Cilium can attach the BPF programs using tc (traffic control) or XDP (eXpress Data Path), allowing it to apply the rules at the lowest and most performant point in the networking stack.

## Dynamic Policy Application

By using eBPF, Cilium can dynamically generate and apply network rules without modifying the Linux kernel, providing flexible and efficient network control for container workloads.

# Weave Scope: Tracking TCP Connections

Weave Scope, an open-source tool for monitoring and visualizing container-based systems, employs eBPF technology to accurately track TCP connections between containers. By attaching BPF programs to kernel-level probes, Weave Scope can observe and report on the network activity within a Kubernetes cluster, providing valuable insights for troubleshooting and performance optimization.



# Potential eBPF Use Cases in Kubernetes

### Pod and Container Level Network Statistics

Collect detailed per-pod and per-container network statistics using eBPF programs attached to cgroups.

### Application-Applied LSM

Use eBPF-based Landlock LSM to allow unprivileged applications to build their own sandboxes and restrict their own actions.

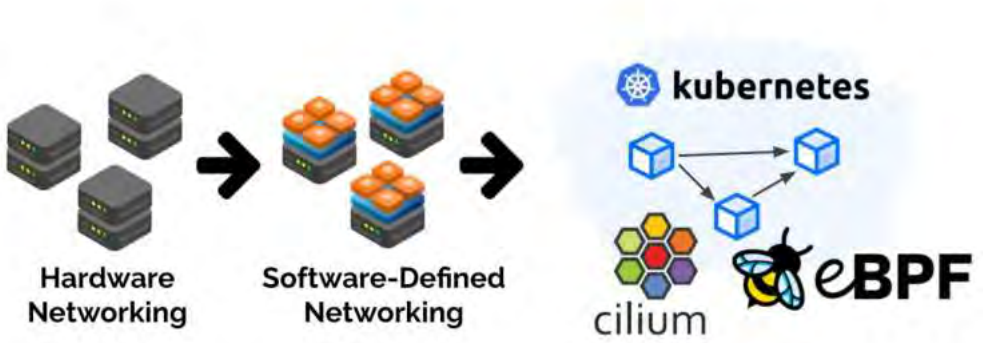
### Auditing kubectl-exec Sessions

Attach eBPF programs to record and log the exact sequence of commands executed during kubectl exec sessions.

**As eBPF continues to evolve, we can expect to see even more innovative applications in the Kubernetes ecosystem, unlocking new possibilities for observability, security, and performance within container-based systems.**

Kubernetes and eBPF have emerged as a powerful combination, unlocking new possibilities for observability, security, and performance within container-based systems. eBPF's ability to dynamically attach to kernel objects and execute custom programs provides unparalleled visibility and control, enabling Kubernetes tools to enhance networking, monitoring, and auditing capabilities.

**As eBPF continues to evolve, we can expect to see even more innovative applications that push the boundaries of what's possible in the Kubernetes ecosystem.**

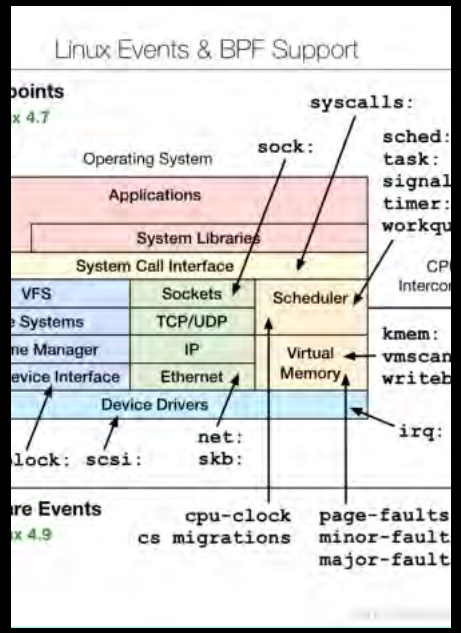


# OpenTelemetry and eBPF

OpenTelemetry

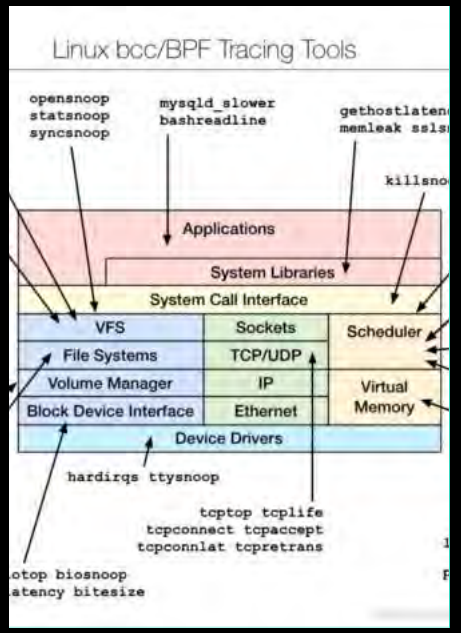

eBPF

# Learning More About eBPF




## Comprehensive Reading List

A curated list of recommended readings to dive deeper into eBPF

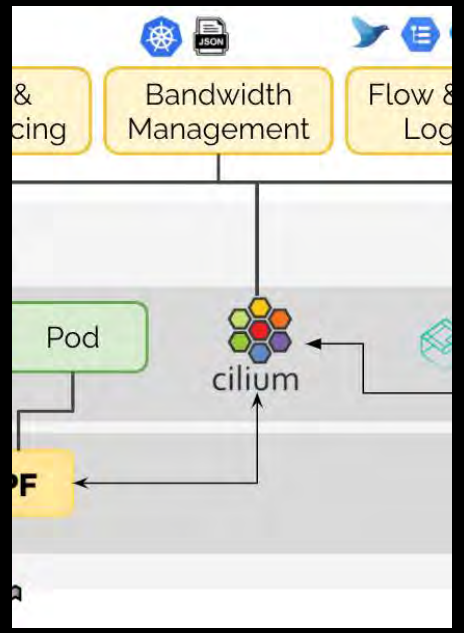



## BCC (BPF Compiler Collection)

A collection of eBPF-based tools and examples for working with eBPF

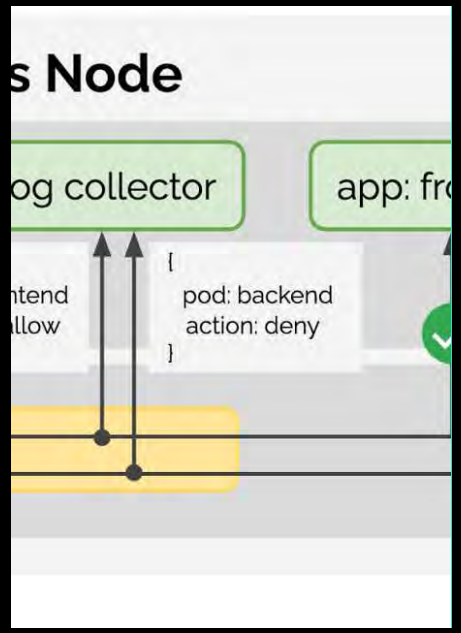
## BPF Tracing and More

A video by Brendan Gregg on the capabilities and use cases of eBPF




## Cilium and eBPF

A video on how the Cilium project leverages eBPF for networking and security




## eBPF in Kubernetes

A video on using eBPF in the Kubernetes ecosystem


# https://medium.com/@alexmarket

Alejandro Mercado Peña

Contact me @ https://www.linkedin.com/in/alexmarket/