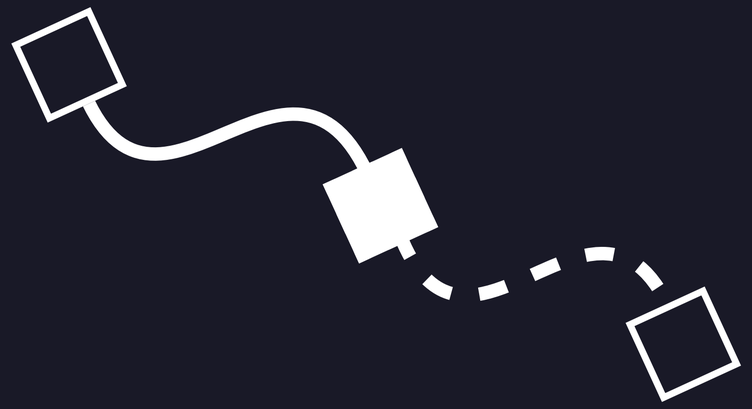
Enhancing security in industrial control systems through programmable kernel-level microsegmentation

UniGe - Computer Science
Software Security and Engineering
Milo Galli
Advisor - Enrico Russo
Examiner - Giovanni Lagorio

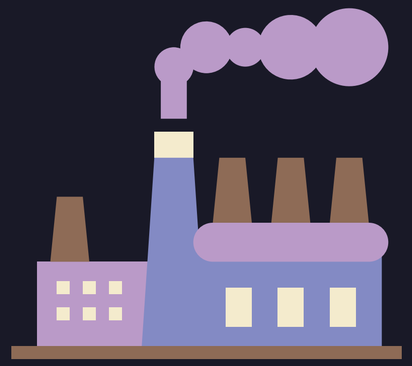


Enhancing security in industrial control systems through programmable kernel-level microsegmentation

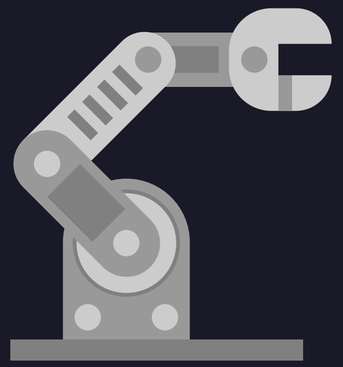
Operational Technology (OT) systems

“...OT systems are hardware and software solutions that monitor and control physical devices, processes, and infrastructure in industrial environments...”

Operational Technology (OT) systems



Machine Industry



Automation Systems

Transportation Systems

Maritime OT systems

Main characteristics

Legacy Systems

Big Assets

Standardized Protocols

NMEA

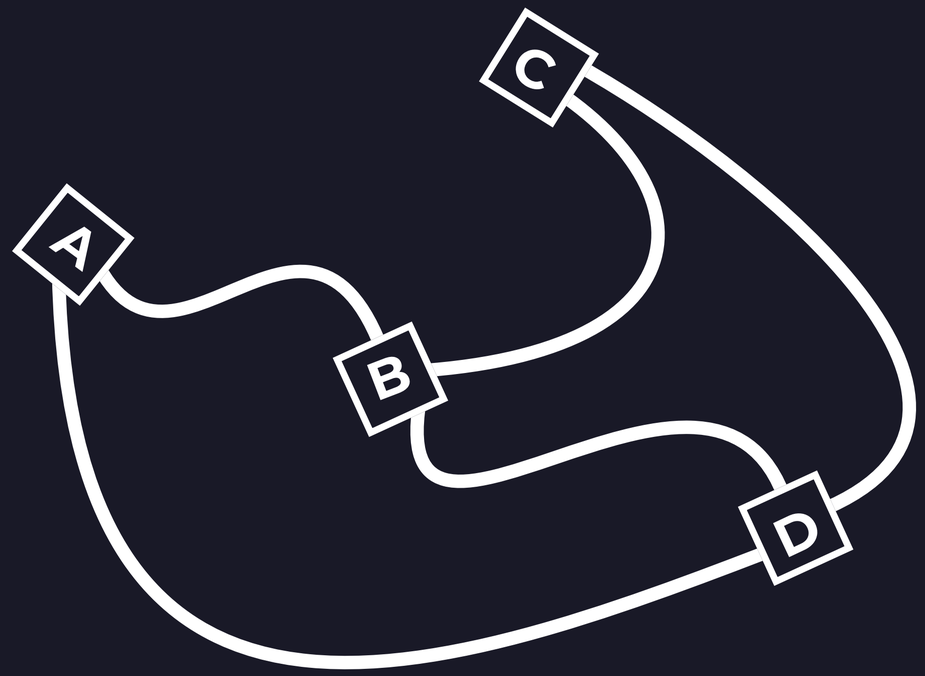
National Marine Electronics Association

e.g. $GPAAM,A,A,0.10,N,WPTNME*32

The problem

Internal network communication is unrestricted, unfiltered and omnidirectional

The problem



The Solution’s Costraints

Retrocompatibility, Transparency,
No-Overhead

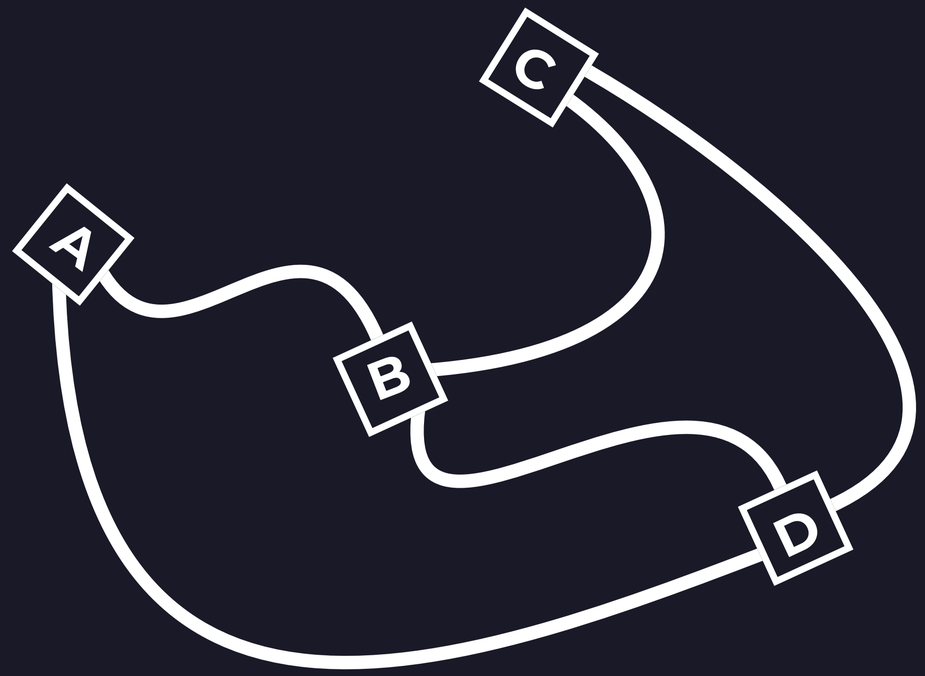
Enhancing security in industrial control systems

through programmable kernel-level microsegmentation

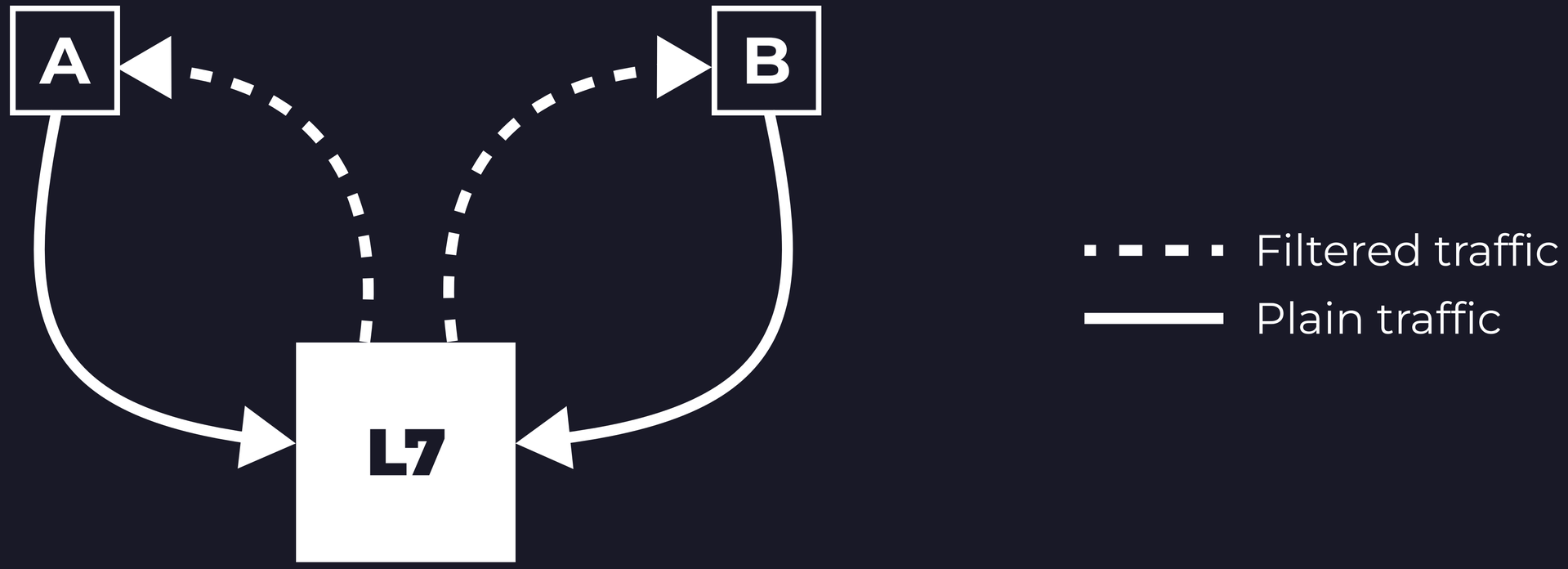
Our solution

An Application Layer Switch that leverages Kernel-Level technologies to filter, analyse and drop network traffic enabling Software Defined Networking

The problem



Our solution



Why this solution for retrocompatibility?

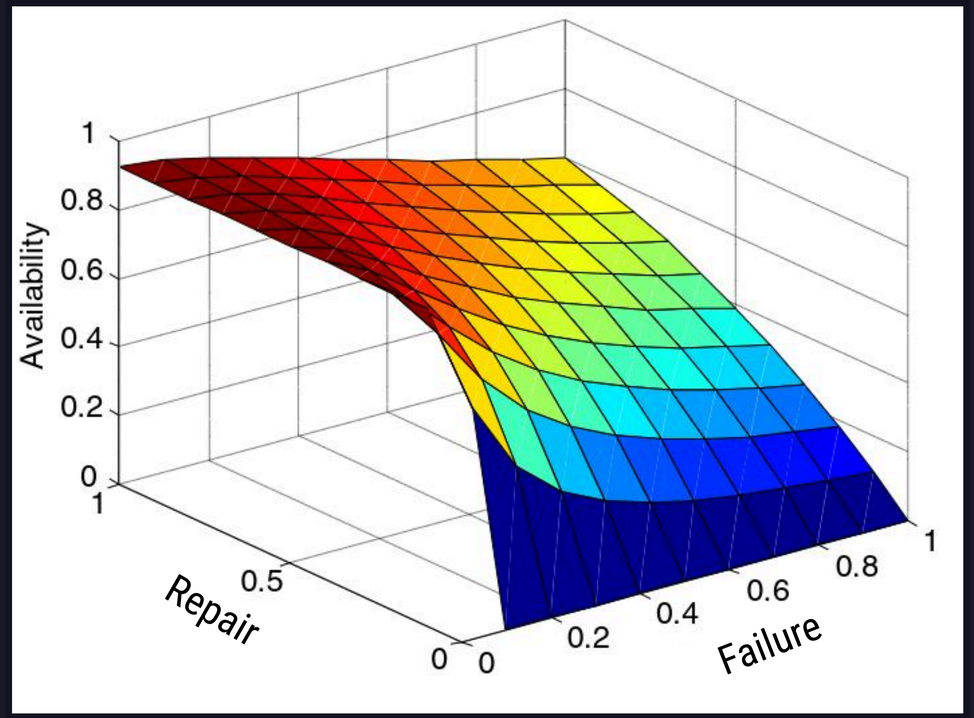
The only dependency
is the Linux Kernel (> v3.15)

“Linux can run on anything”

@Action Retro

Why this solution for Transparency ?

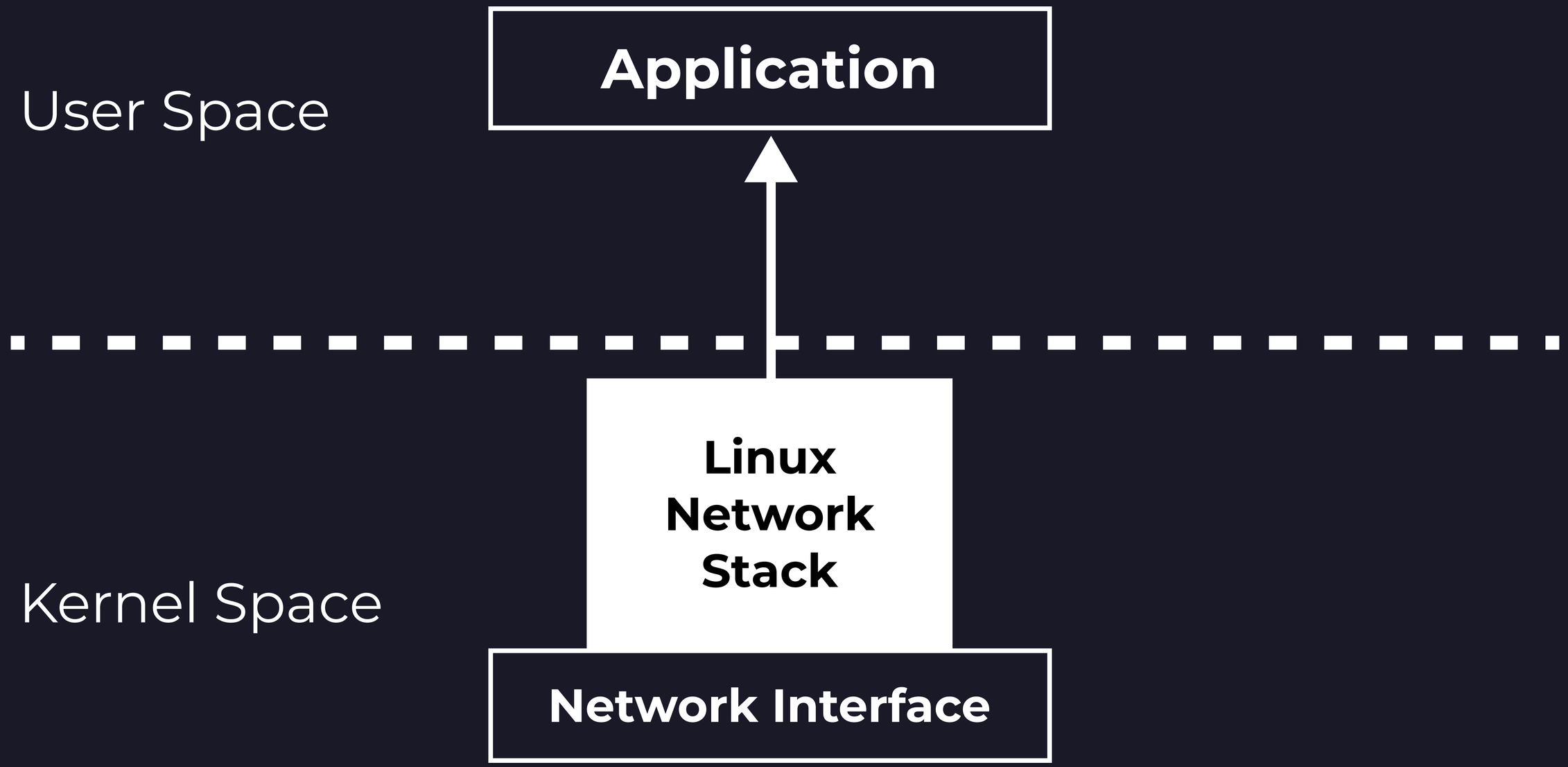
The only dependency
is the Linux Kernel (> v3.15)



“Reliability Analysis of Multi-Hardware–Software System with Failure Interaction”
Journal of Computational and Cognitive Engineering
2023, Vol. 2(1) 38–46

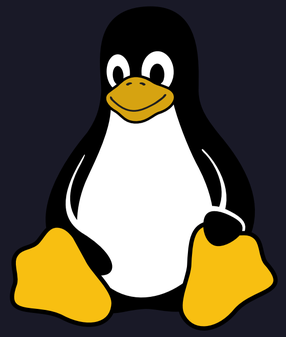
Why this solution doesn’t introduce overhead?

The only dependency
is the Linux Kernel (> v3.15)


User Space
Application
Linux
Network
Stack
Kernel Space
Network Interface

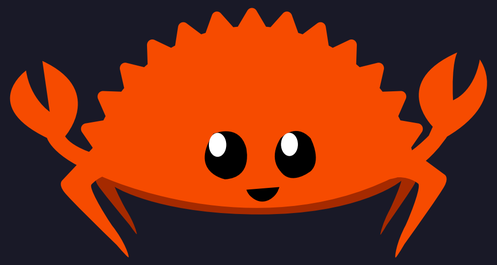
Extended Berkeley Packet Filter (EBPF)
and
the new Address Family Express Data Path (AF_XDP)

The implementation

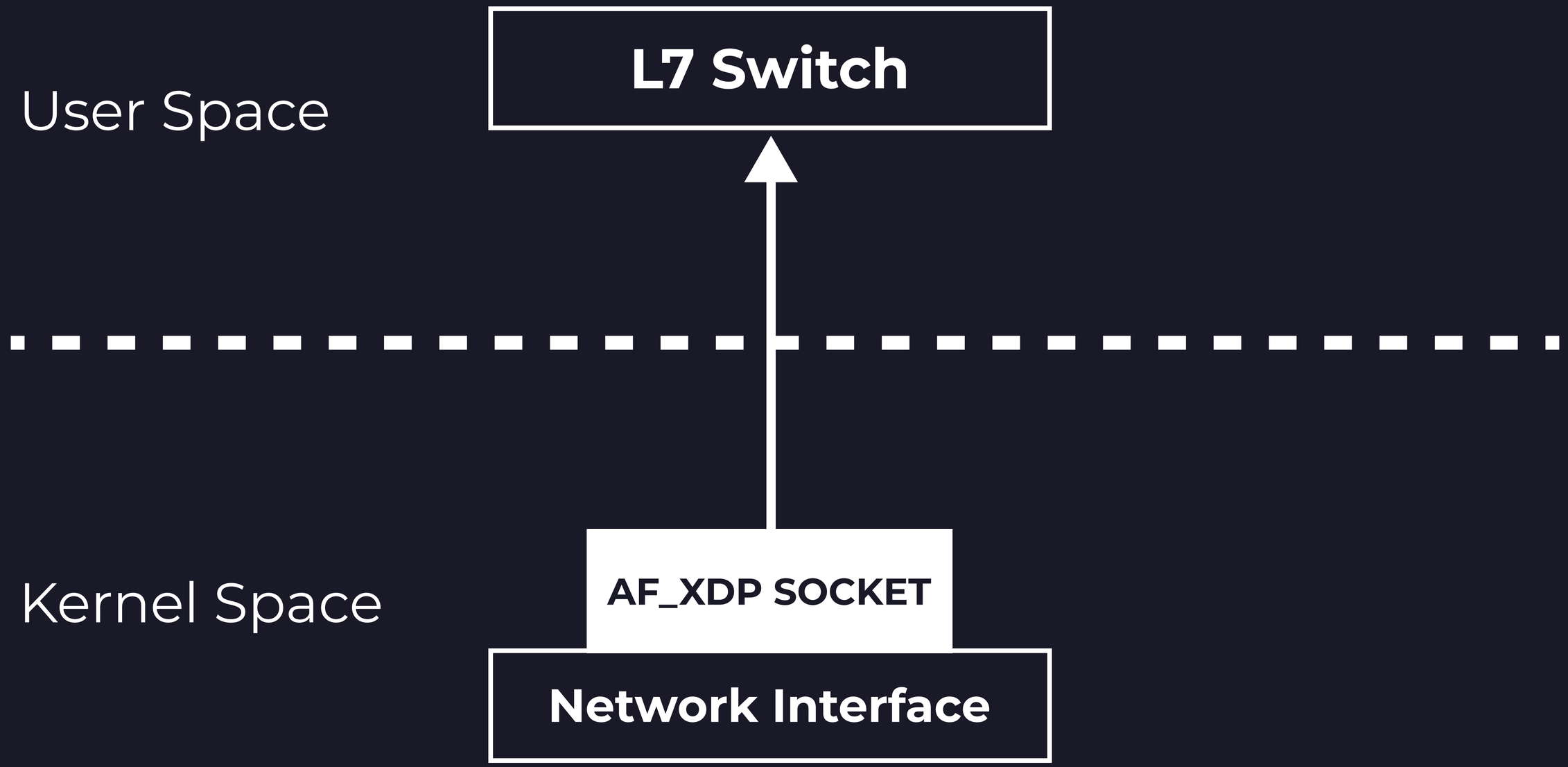


Linux

EBPF

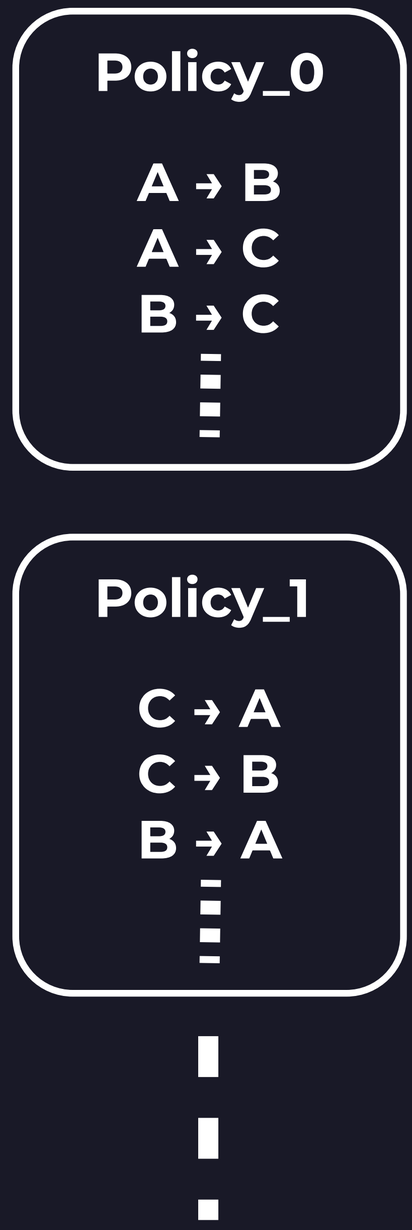


Rust


User Space
L7 Switch
Kernel Space
AF_XDP SOCKET
Network Interface

User Space

L7 Switch



Does it work ?

Virtualising the enviroment... yes!

Linux Network Stack Baseline

Datagram size of

1460 B

Bitrate of

1000 Mbit/s

Unicast Transmission

Using Linux Network Namespaces

Role	Bitrate [Mbit/s]	Datagram Size [B]
SENDER	**1000**	**1460**
RECEIVER	**1000**	**1460**
SENDER	**4800**	**8972**
RECEIVER	**4800**	**8972**

Multicast Transmission

Using Linux Network Namespaces

Role	Bitrate [Mbit/s]	Datagram Size [B]
SENDER	**1000**	**1460**
RECEIVERs	**1000**	**1460**
SENDER	**4800**	**8972**
RECEIVER_1	**4800**	**8972**
RECEIVER_nth	**500**	**8972**

On a real ship... testing is needed

Questions?