

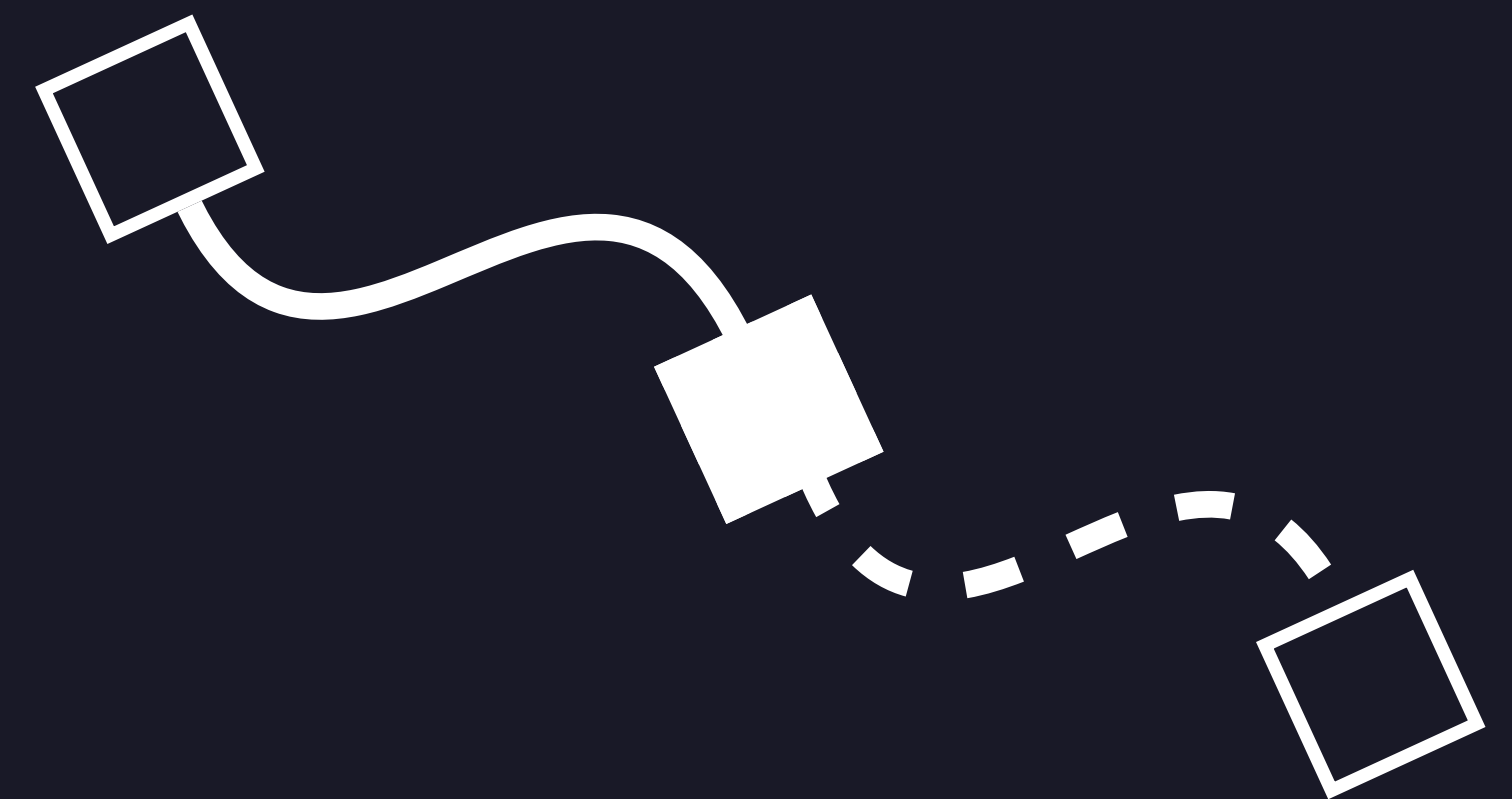
Enhancing security in industrial control systems through programmable kernel-level microsegmentation

UniGe - Computer Science
Software Security and Engineering

Milo Galli

Advisor - Enrico Russo, Giacomo Longo

Examiner - Giovanni Lagorio



Outline

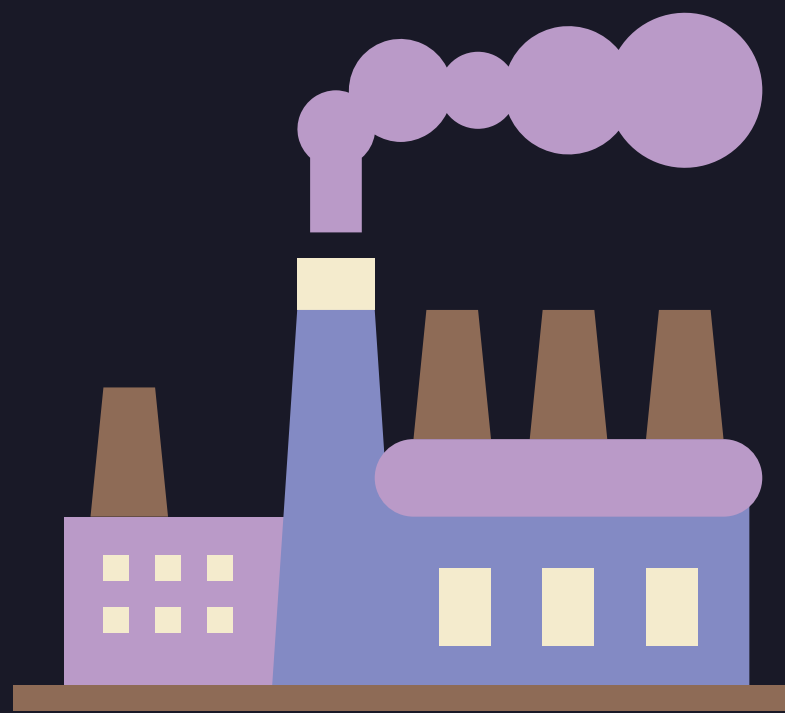
- A bit of background
- Problem's analysis
- Constraints check for an ideal solution
- Explore the proposed solution

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

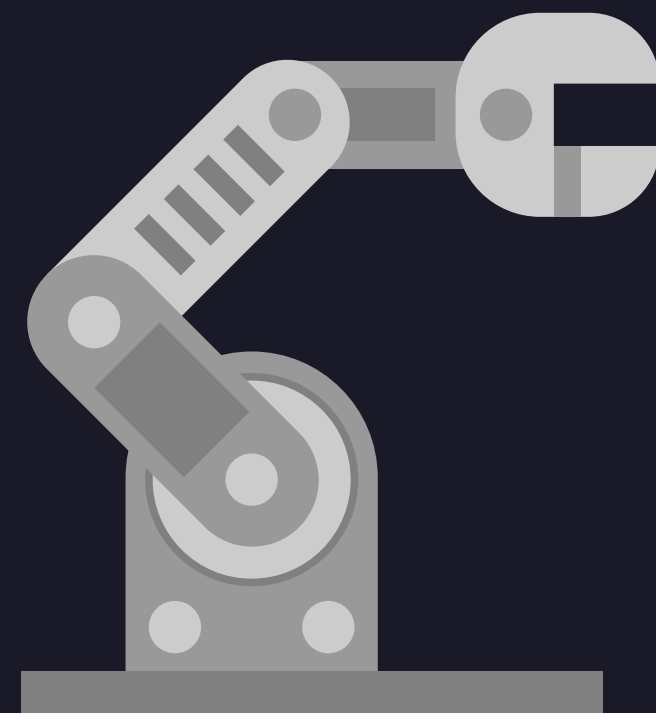
Operational Technology (OT) systems

“...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

Legacy
Systems

Critical
Assets

Standardized
Protocols

**By design
internal network communication
is unrestricted, unfiltered
and omnidirectional**

Unrestricted

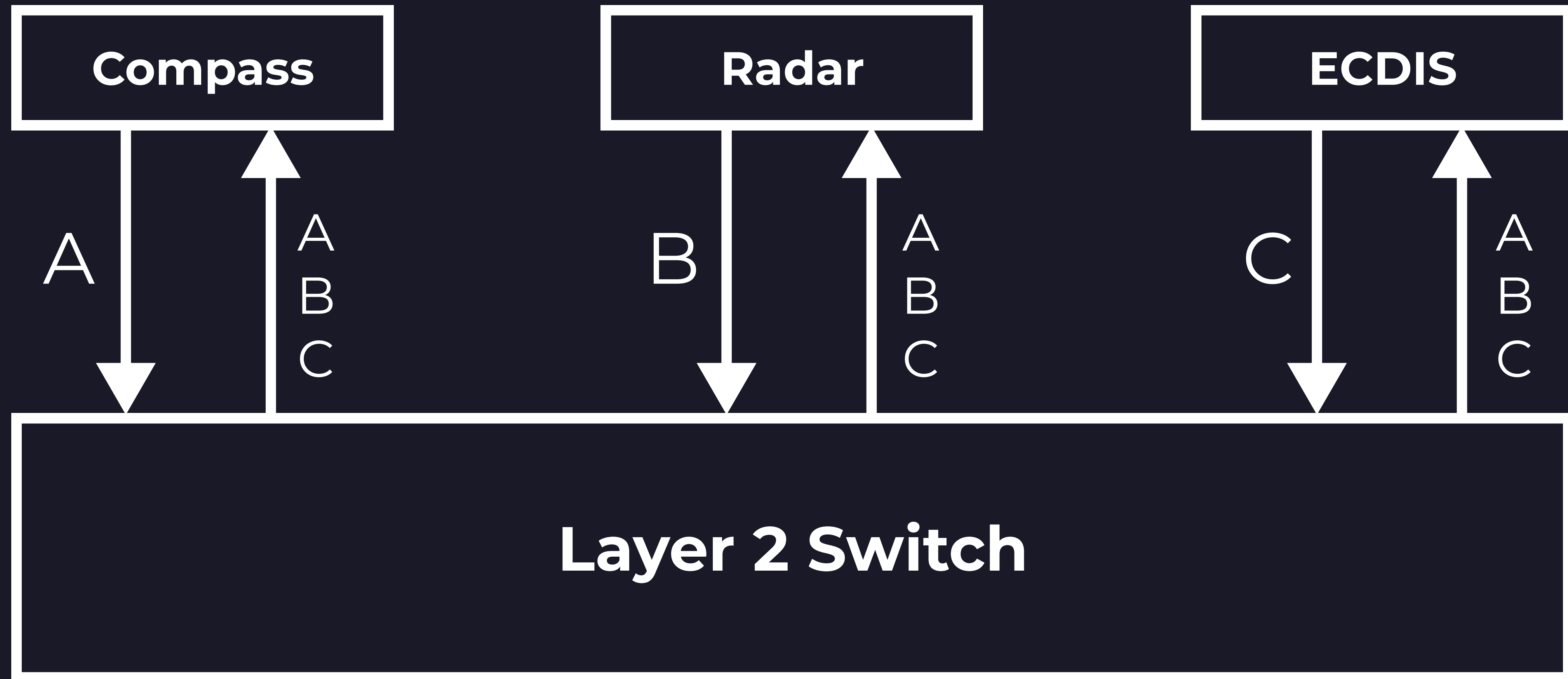
Everyone can send
any kind of message

Omnidirectional

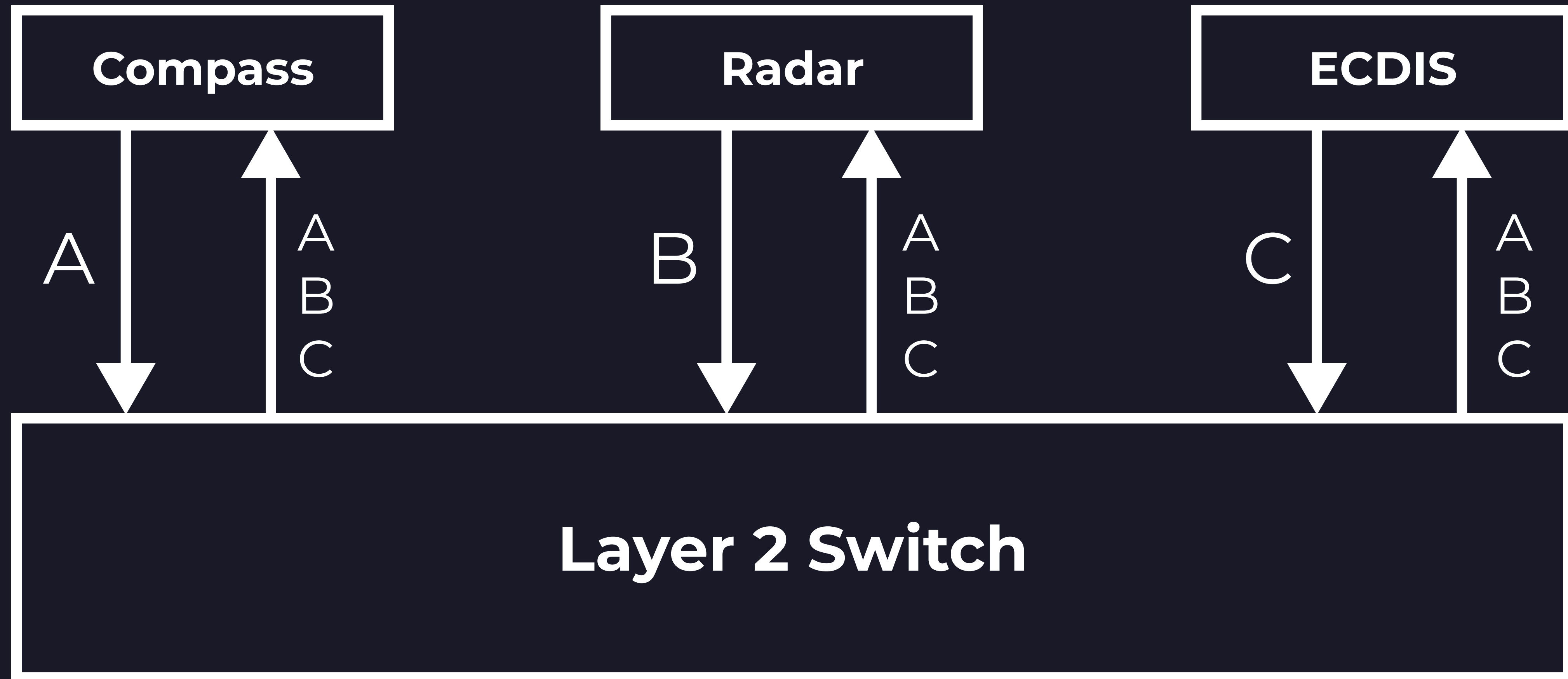
Everyone speaks
to everybody

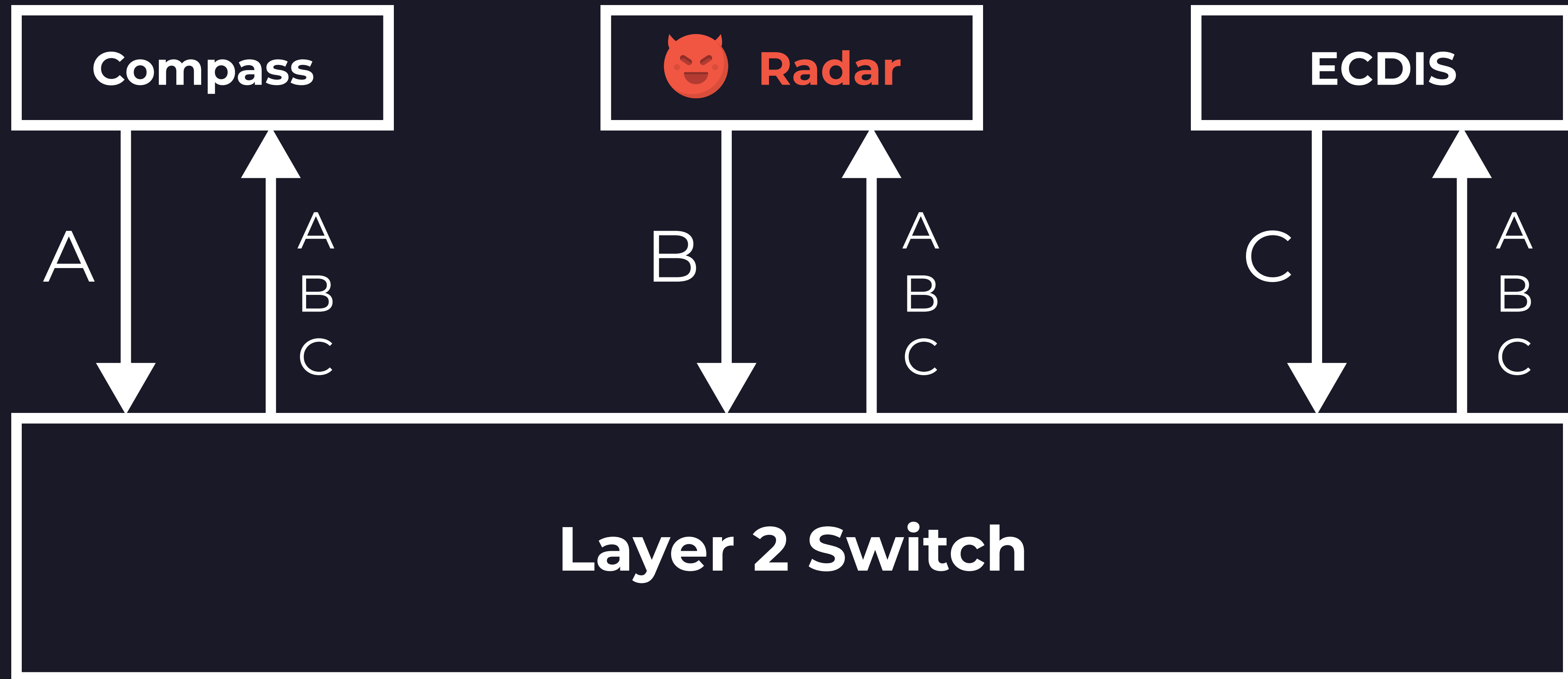
Unfiltered

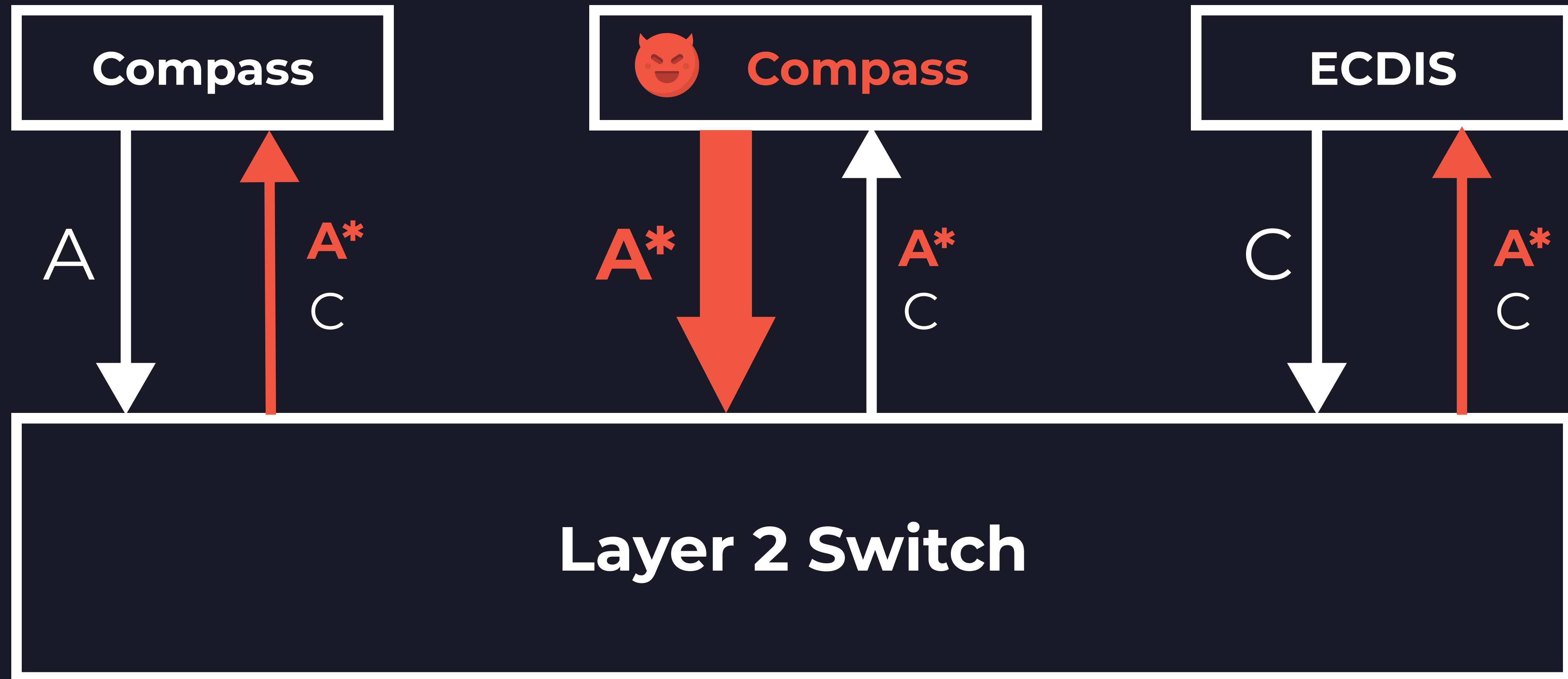
Everyone can receive
any kind of message











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

The Solution's Constraints

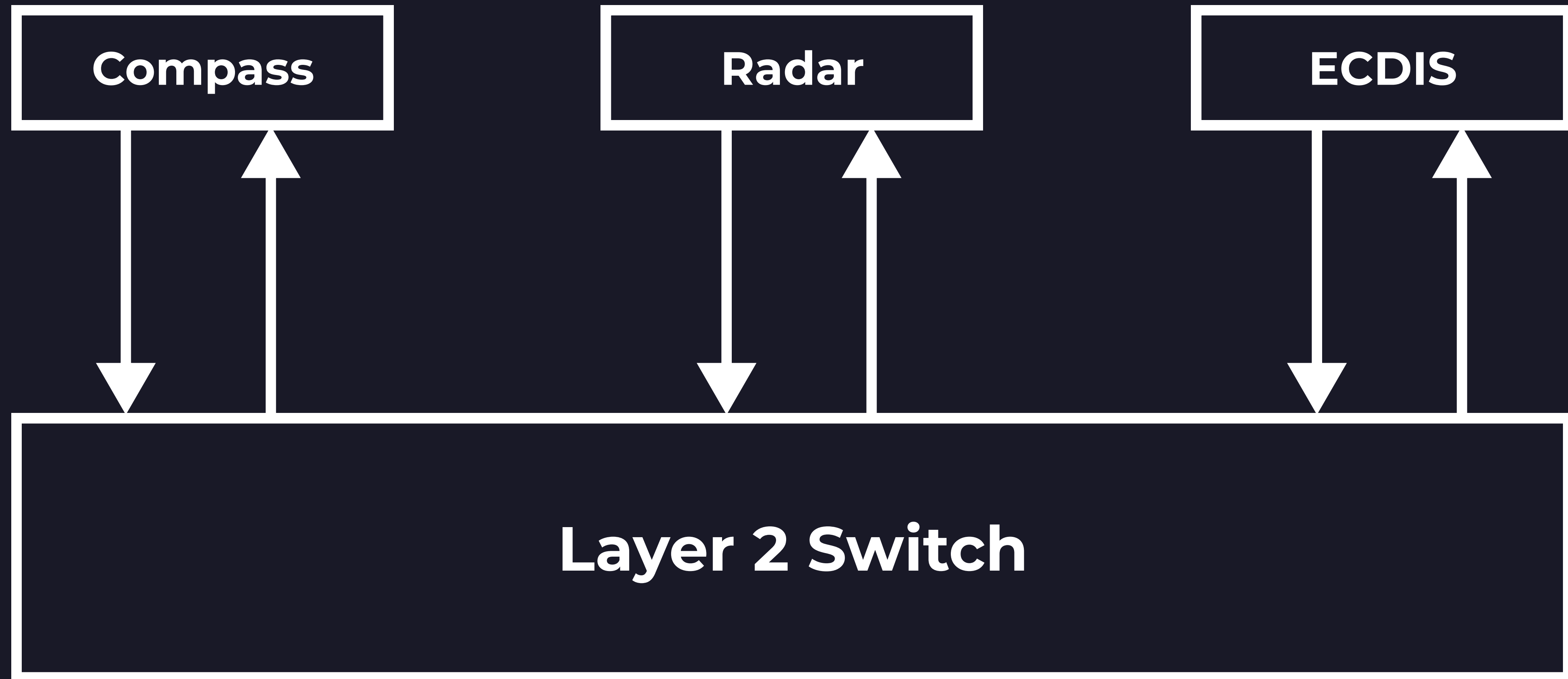
Transparency and Negligible Overhead

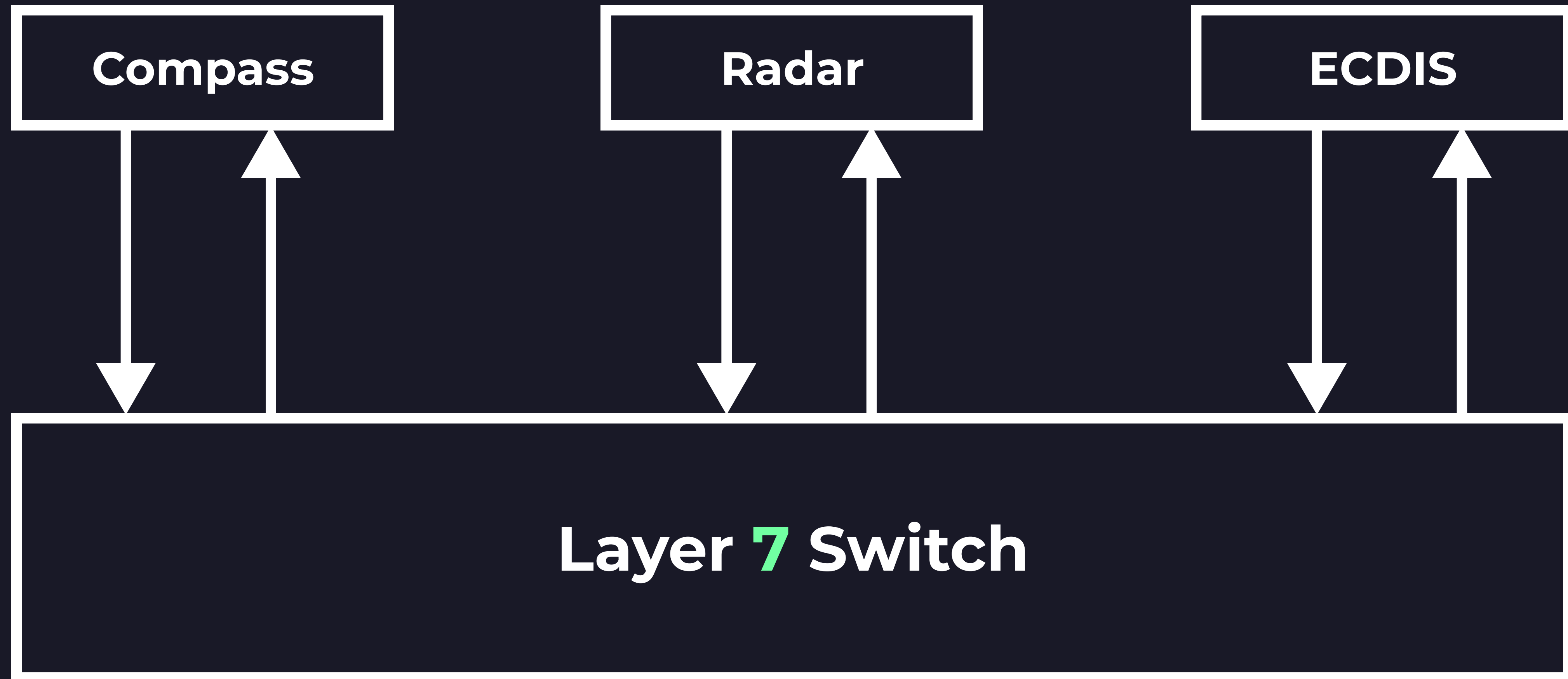
Our solution

An Application Layer Switch that leverages Kernel Level technologies to handle network traffic

Software Defined Networking

Using a software solution
to solve an hardware problem





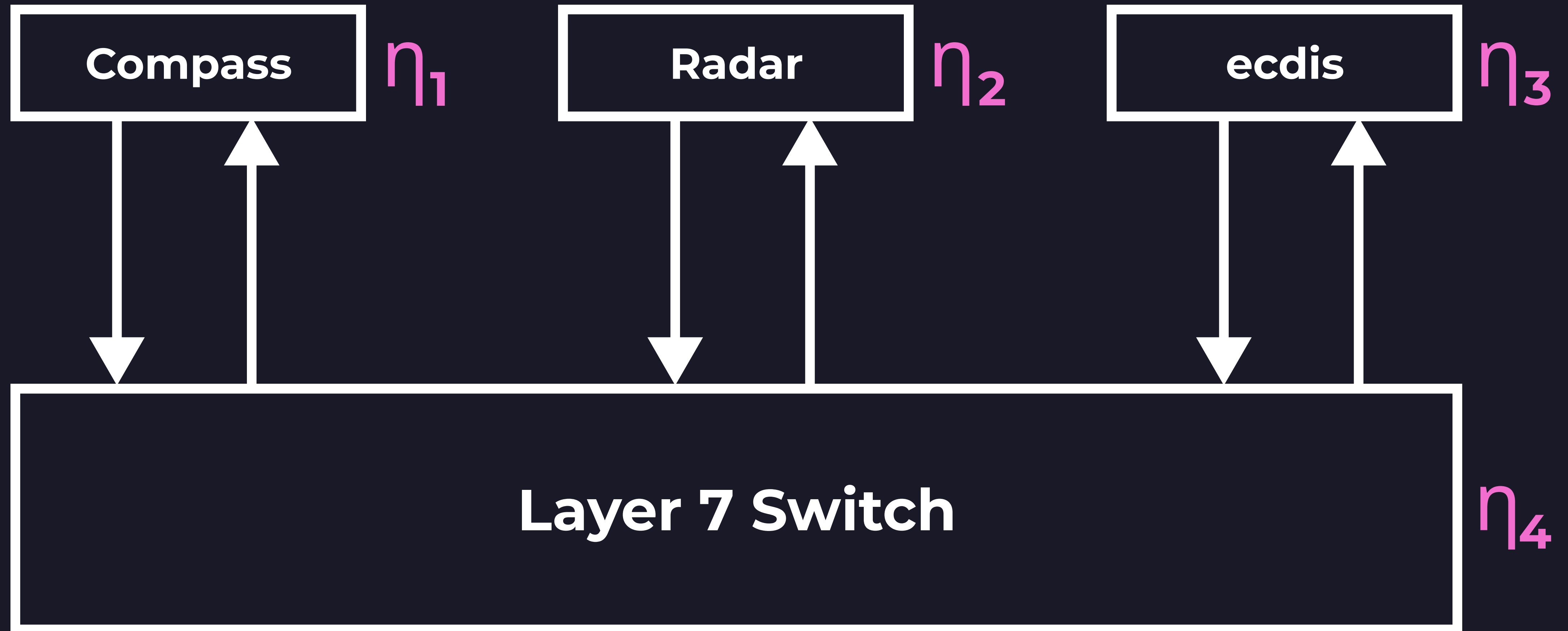
The Solution's Constraints

Transparency and Negligible Overhead

No changes to the system's behaviour
No need for new communication standards
No need for additional hardware/software solutions

$$0 < \eta \leq 1$$

Failure Probability
of a System's Component

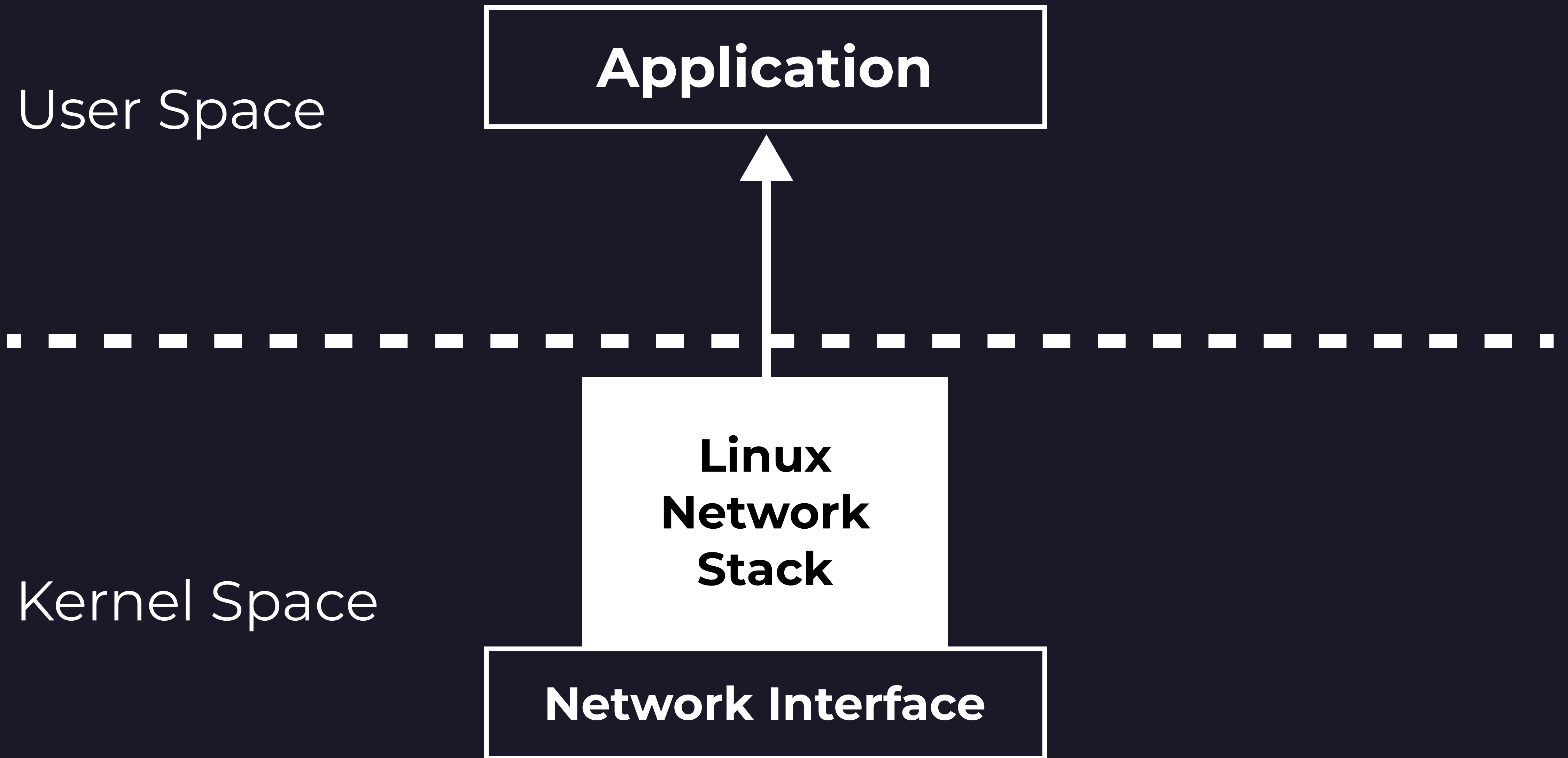


$$P(\text{System Works}) = (1 - \eta_1)(1 - \eta_2) \cdots (1 - \eta_n)$$

$$P(\text{System Fails}) = 1 - P(\text{System Works})$$

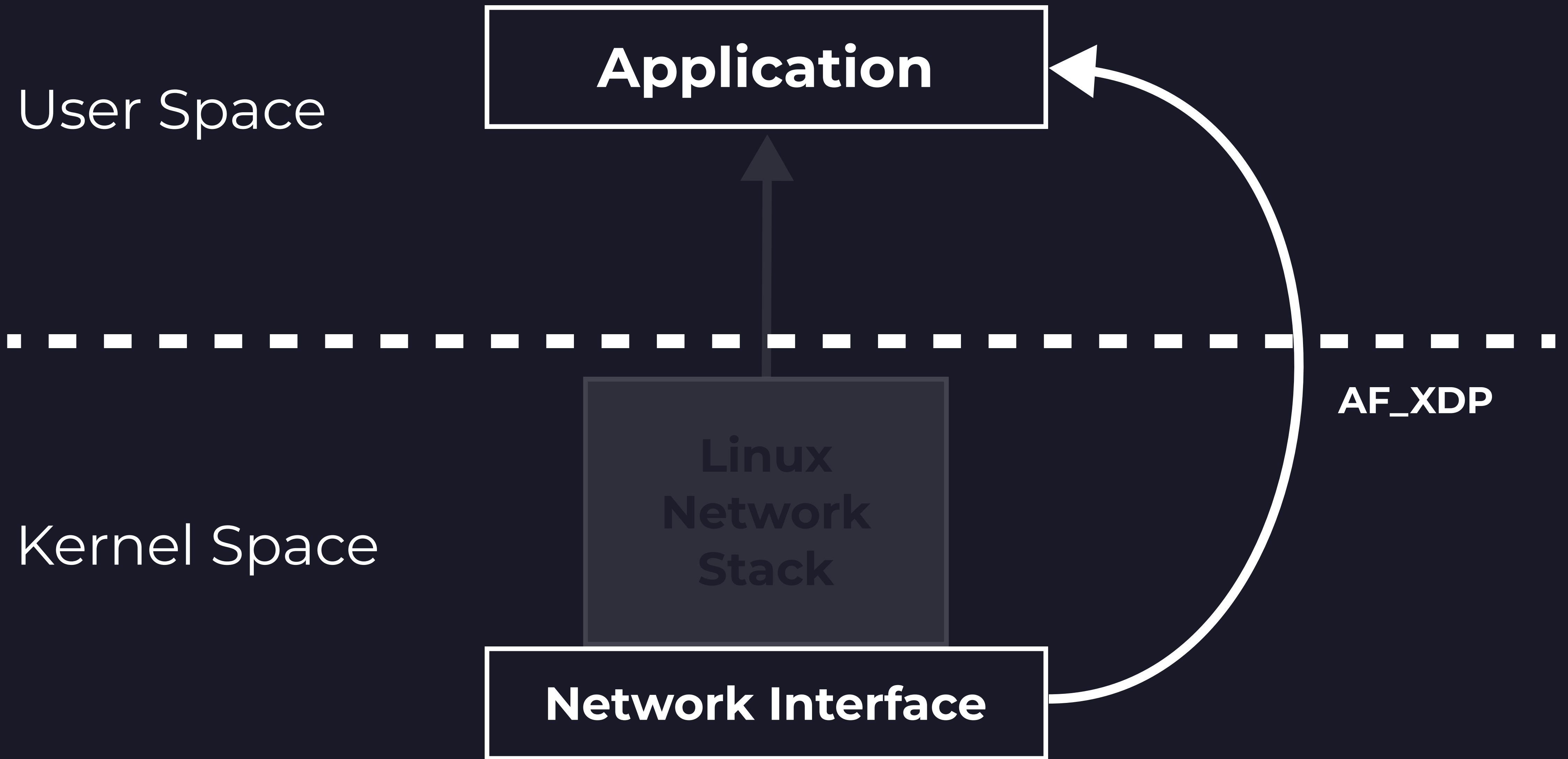
The Solution's Constraints

Transparency and Negligible Overhead

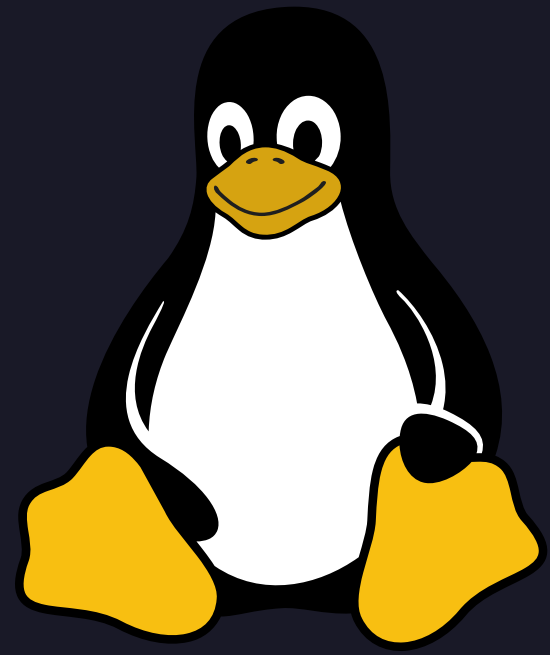




Extended Berkeley Packet Filter (EBPf)
and
the new Address Family Express Data Path (AF_XDP)
for fast network packet processing



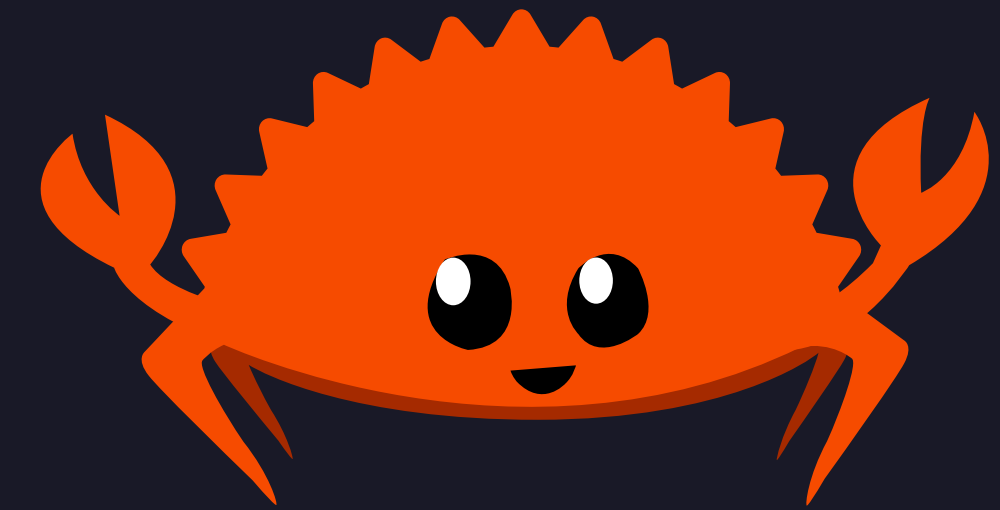
The implementation



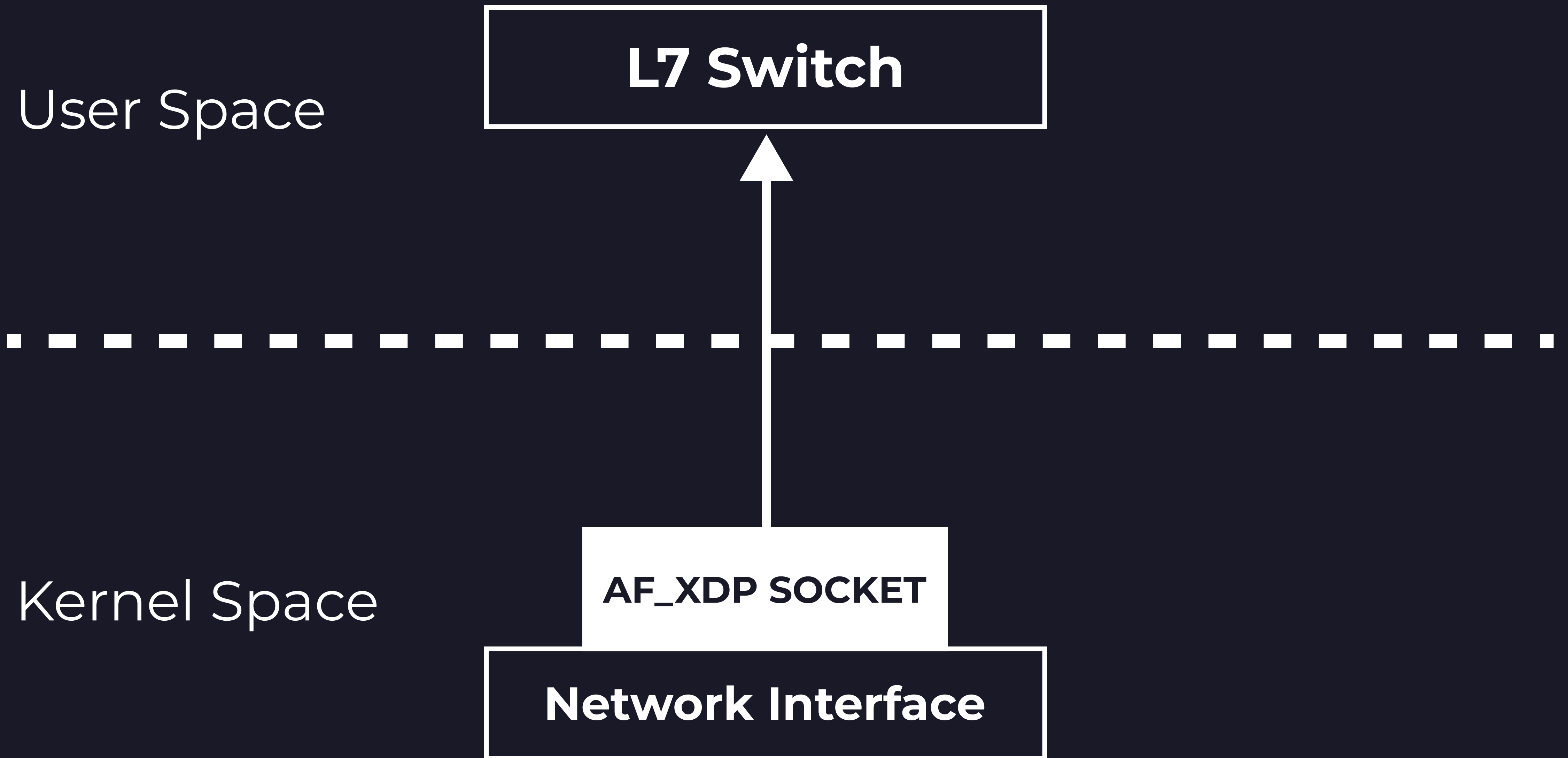
Linux



EBPF



Rust



L7 Switch

NMEA

National Marine Electronics Association

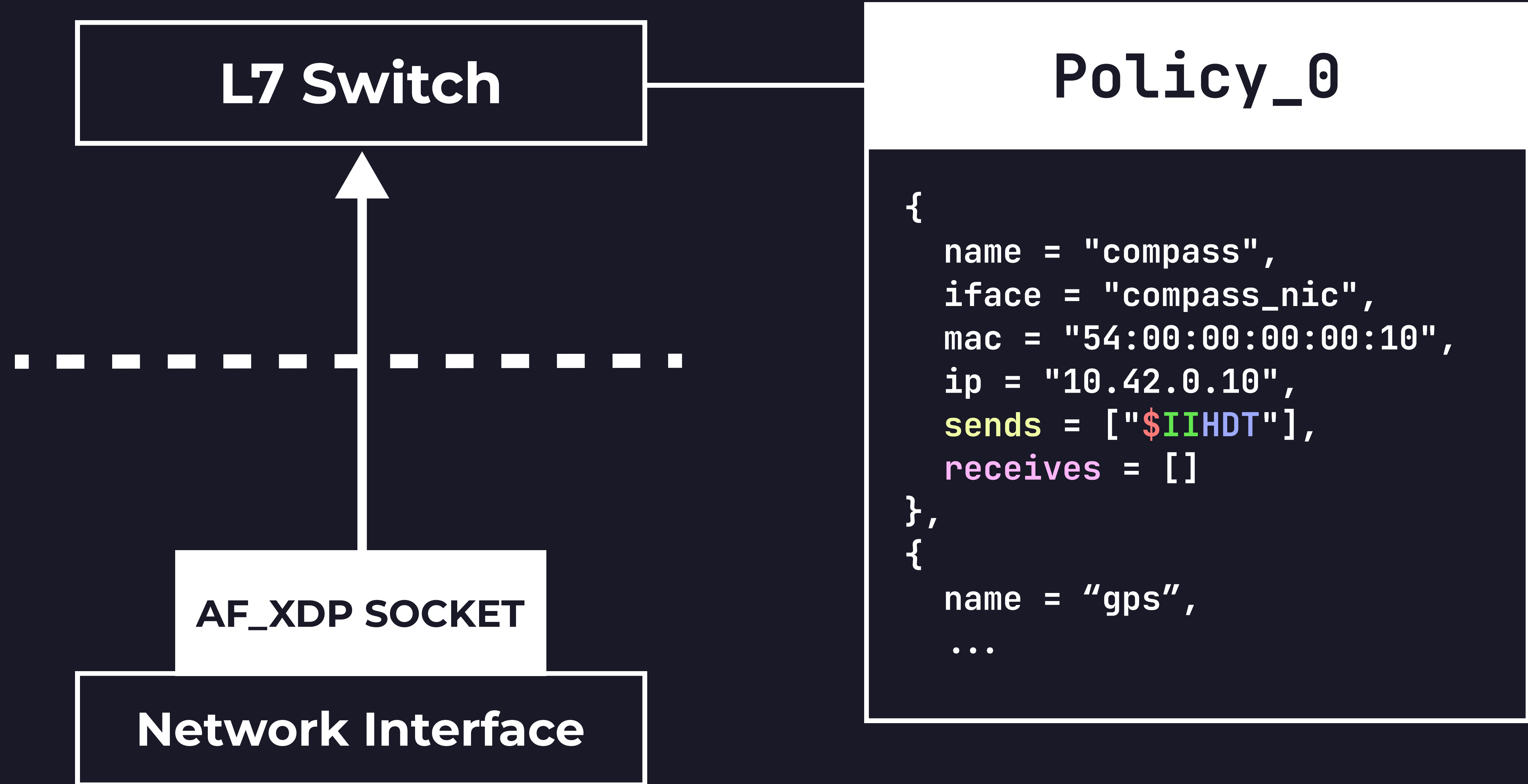
\$GPAAM,A,A,0.10,N,WPTNME*32

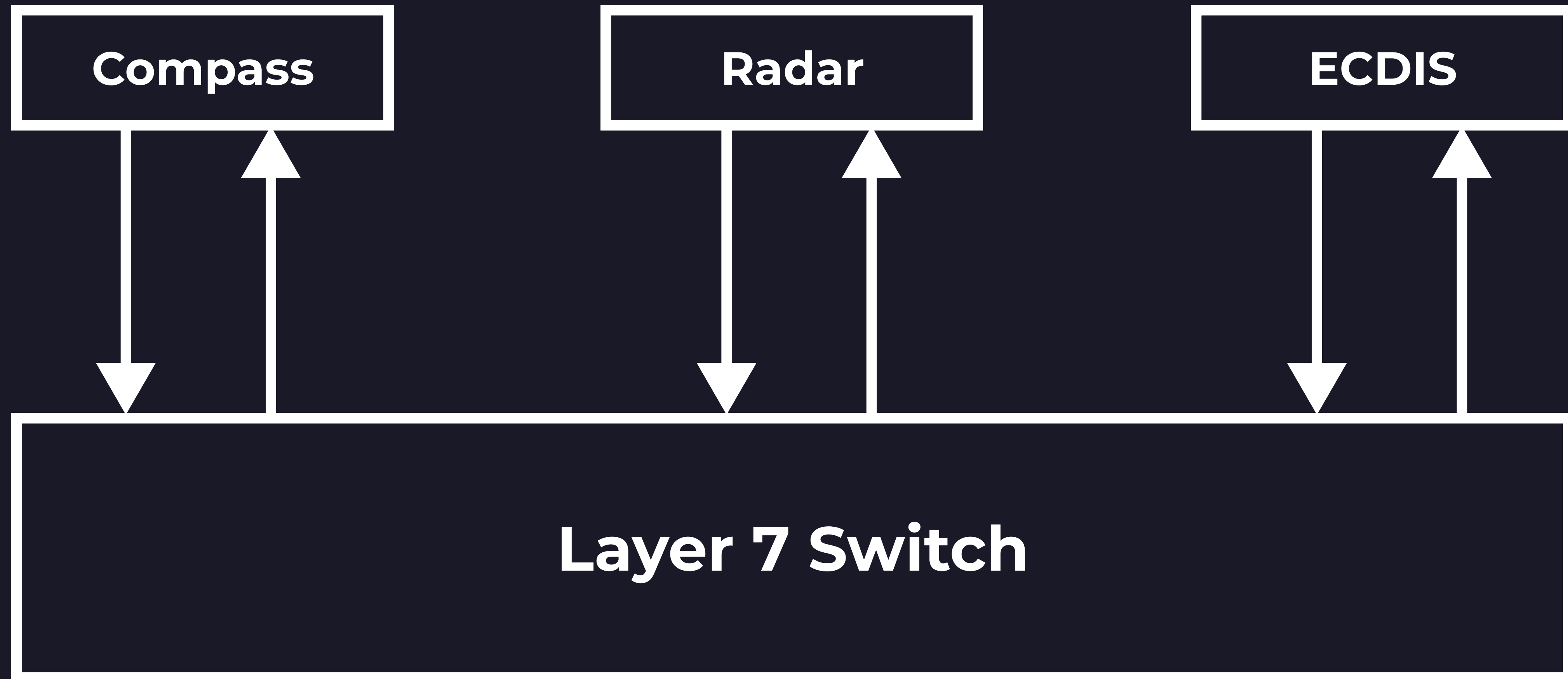
\$SDDBT,7.8,f,2.4,M,1.3,F*0D

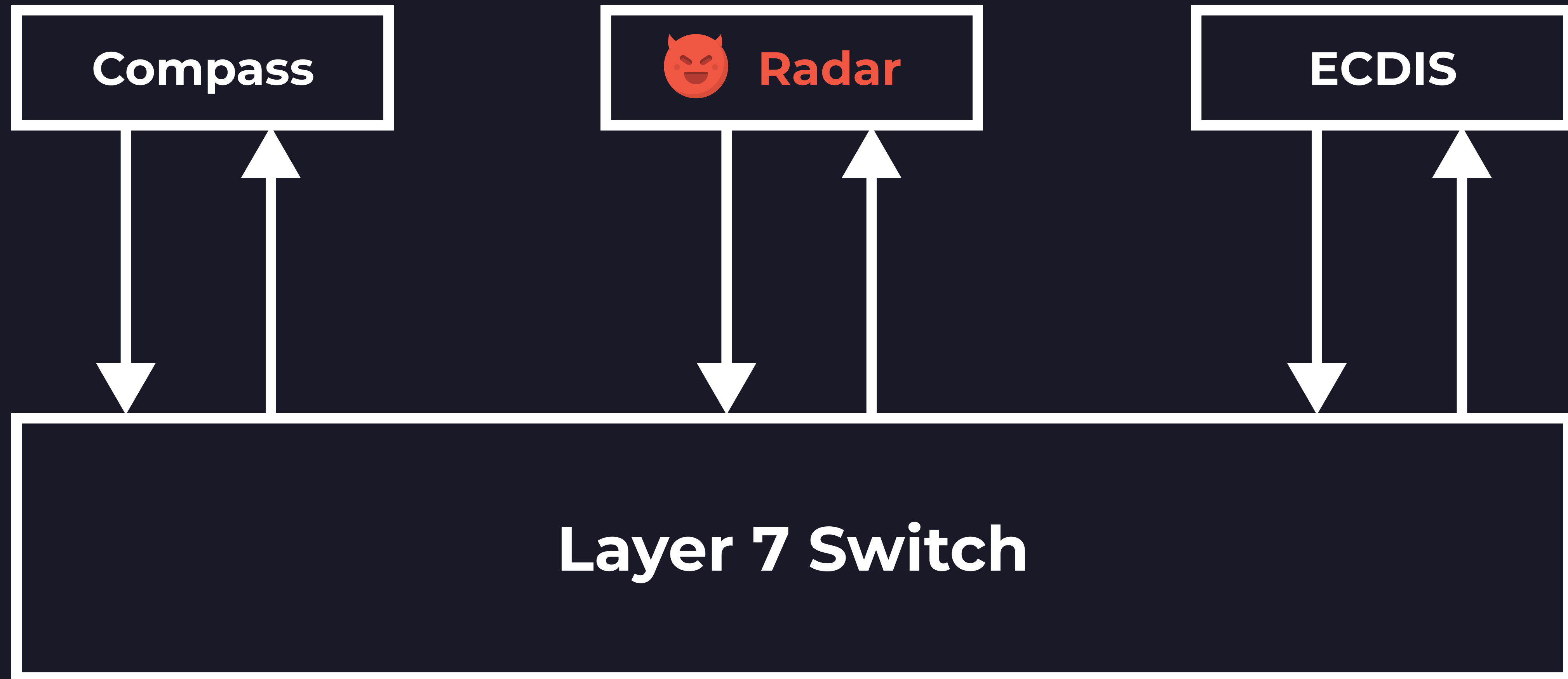
\$GPDTM,W84,C*52

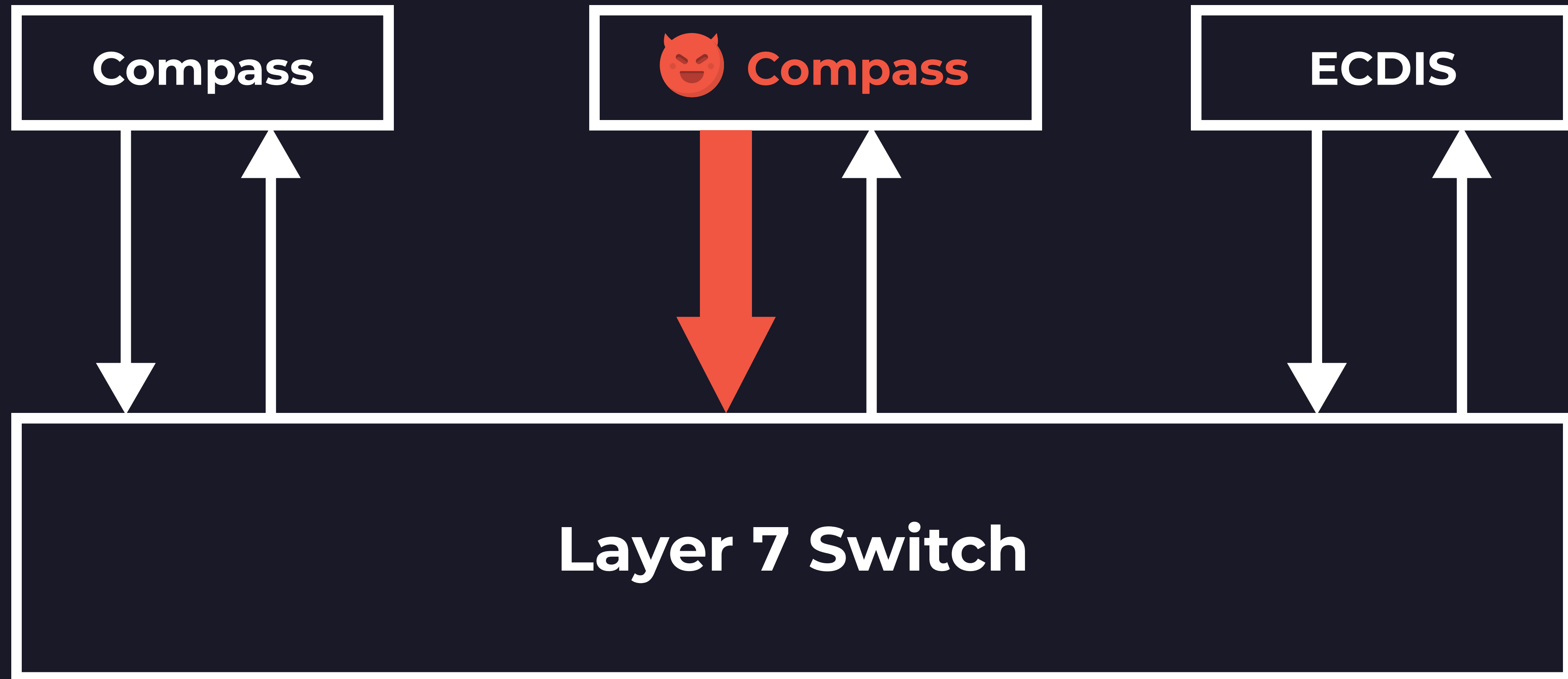
AF_XDP SOCKET

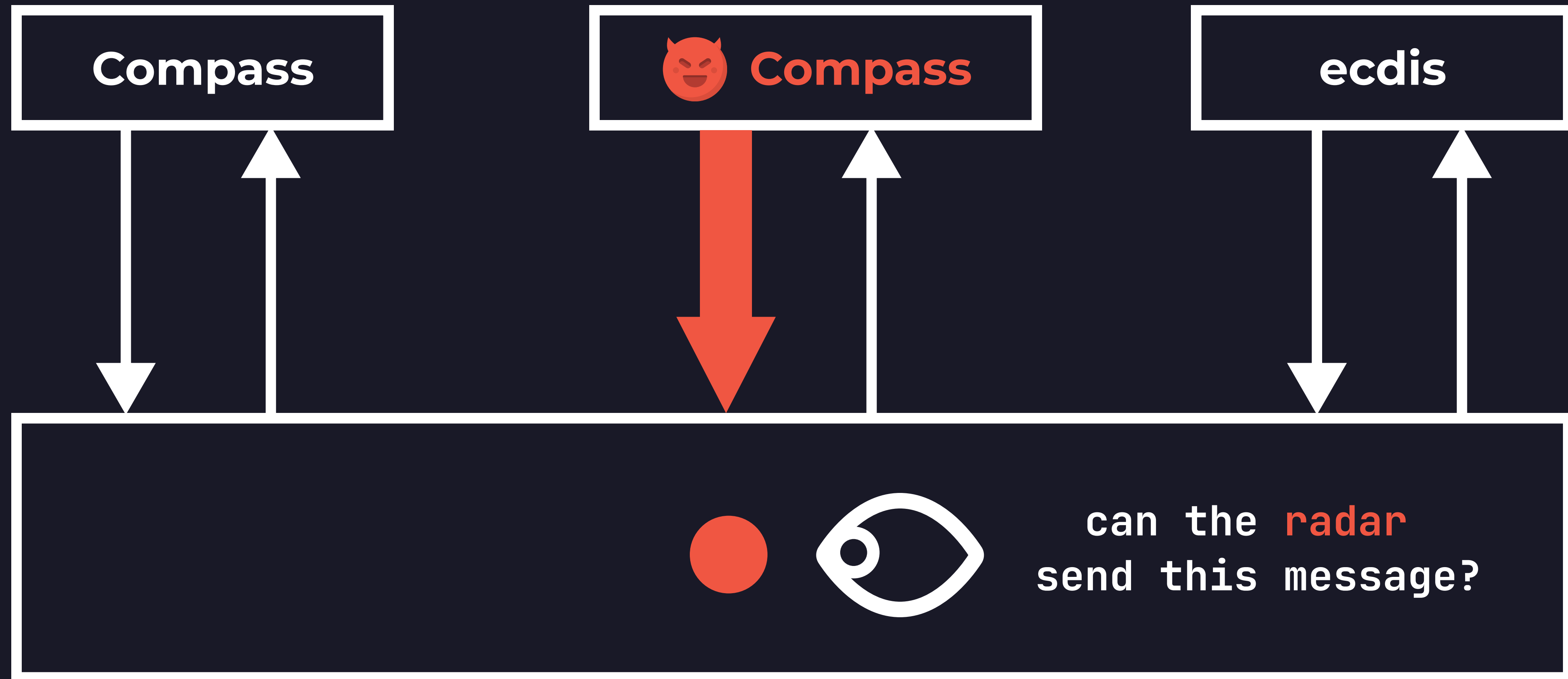
Network Interface

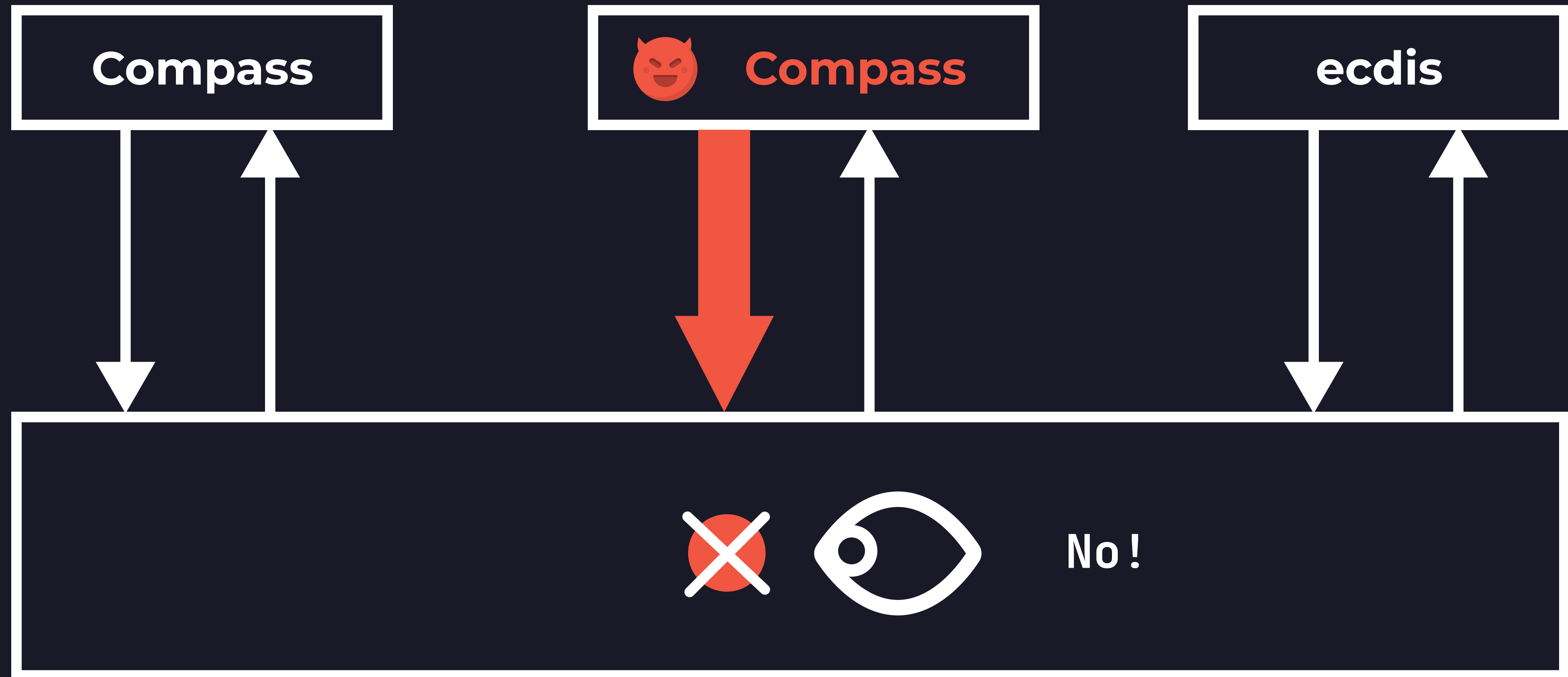


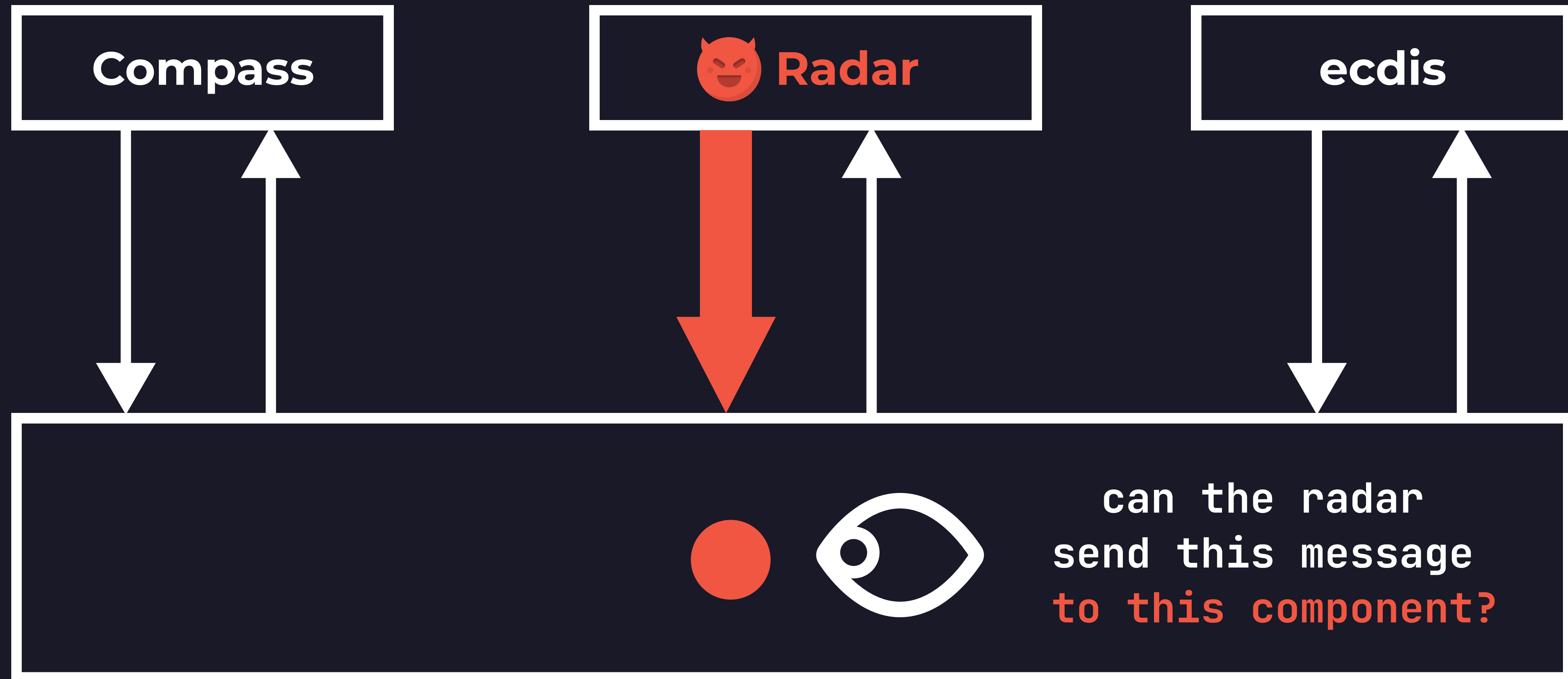


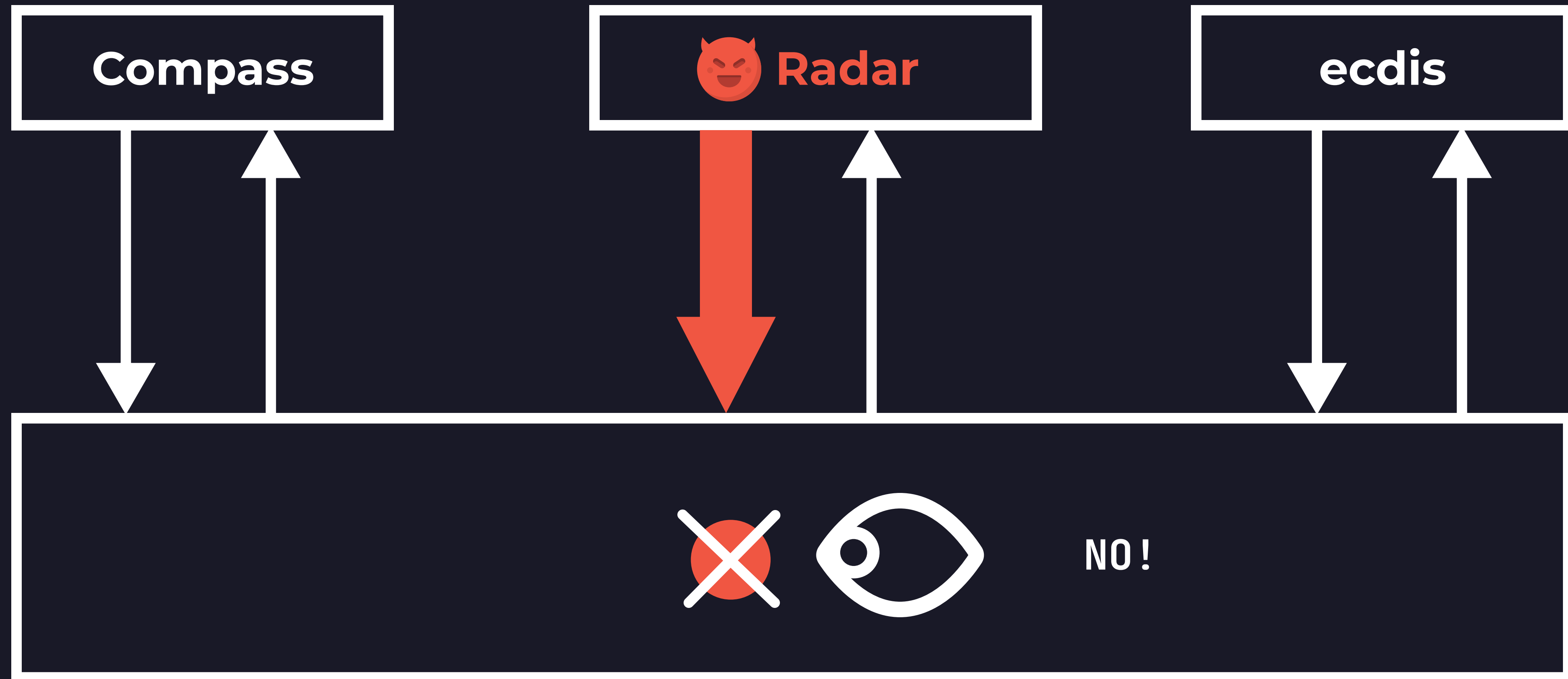












Before

Internal network
communication
is unrestricted, unfiltered
and omnidirectional

After

Internal network
communication
is restricted, filtered,
information flow
is directable

Before

Internal network
communication
is unrestricted, unfiltered
and omnidirectional

After

Internal network
communication
is restricted, filtered,
information flow
is directable,
**modular and centrally
manageable**

Does it work?

**Testing the
software correctness
is trivial... what about performance?**

Linux Network Stack Baseline

Standard datagram size of

1460 B

Bitrate of

1000 Mbit/s

Testing in Linux Network Namespaces

mean over 500 runs per role

Role	Bitrate [Mbit/s]	Datagram Size [B]
SENDER	995	1460 (Standard UDP)
RECEIVER	995	1460 (Standard UDP)
SENDER	4793	8972 (Jumbo Frames)
RECEIVER	4793	8972 (Jumbo Frames)

Recap

Analyzed OT Systems' vulnerabilities

Checked the solution's constraints

Shown the solution's implementation

Assessed the solution's validity and performance

Does it work?

Does it work?

Yes!

Questions?

