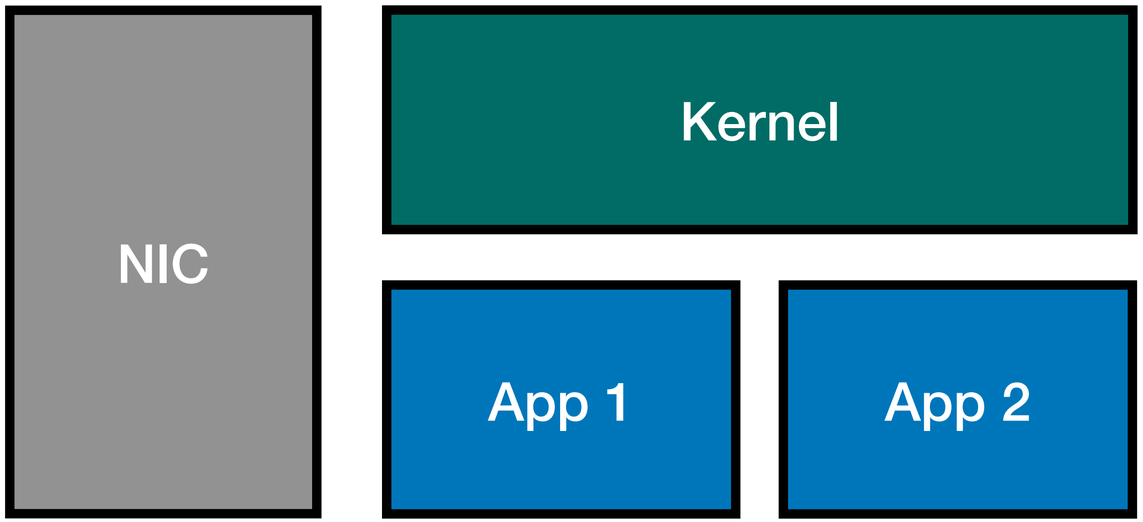
We Need Kernel Interposition over the Network Dataplane

Hugo Sadok, Zhipeng Zhao, Valerie Choung, Nirav Atre, Daniel S. Berger, James C. Hoe, Aurojit Panda, Justine Sherry

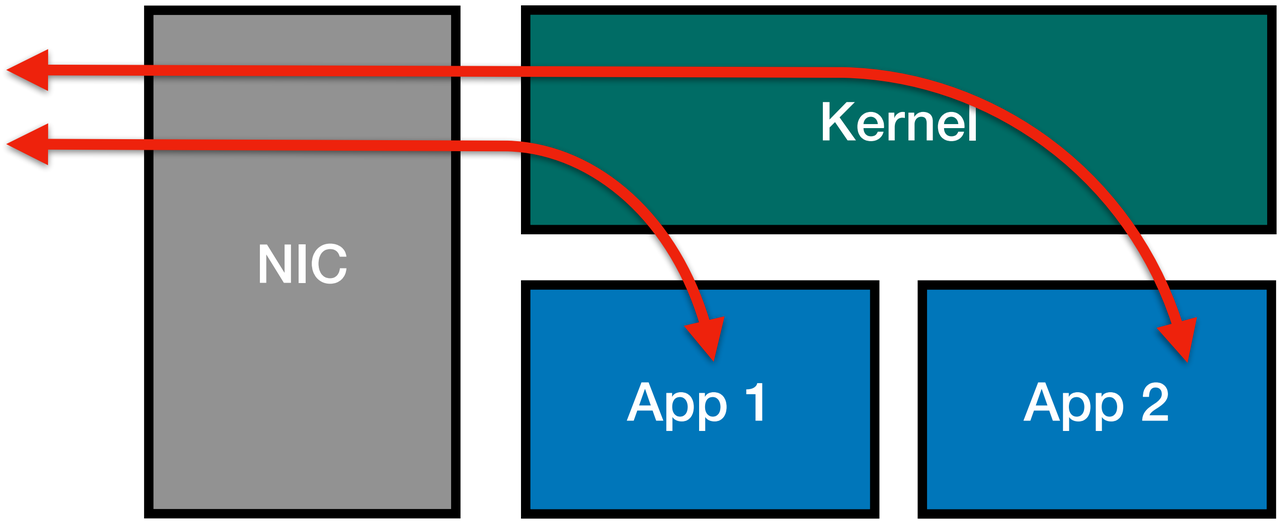
Traditional Kernel Network Stack

Most popular design today (e.g., Linux, Windows, BSD)



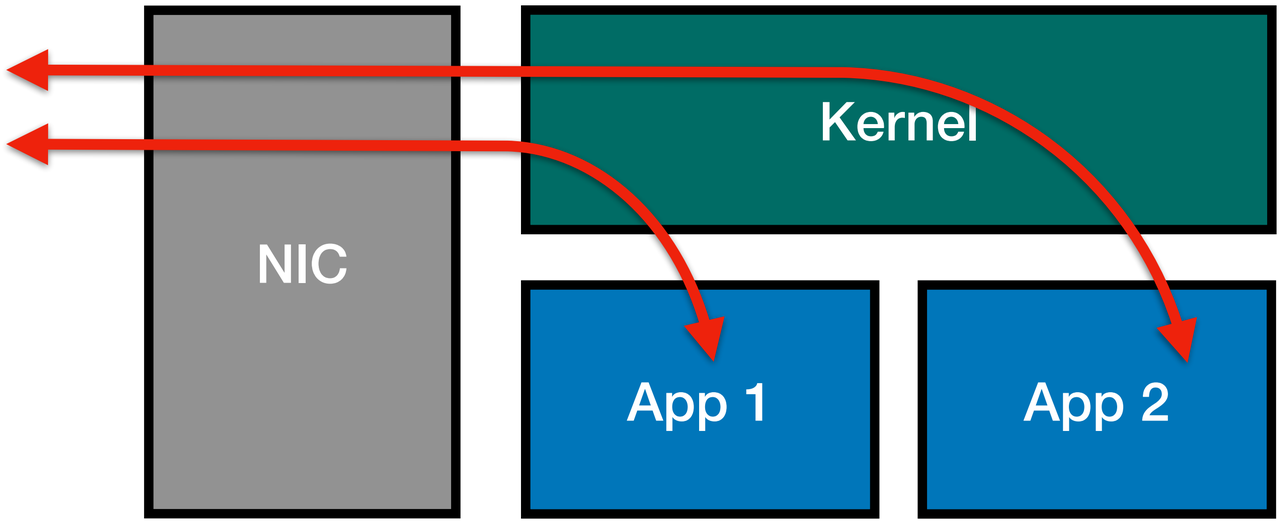
Traditional Kernel Network Stack

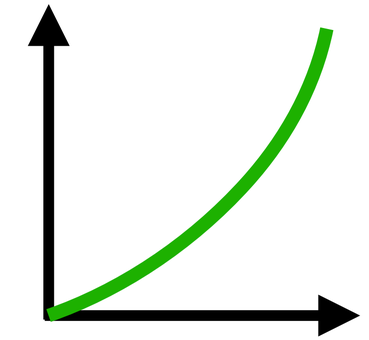
Most popular design today (e.g., Linux, Windows, BSD)



Traditional Kernel Network Stack

Most popular design today (e.g., Linux, Windows, BSD)

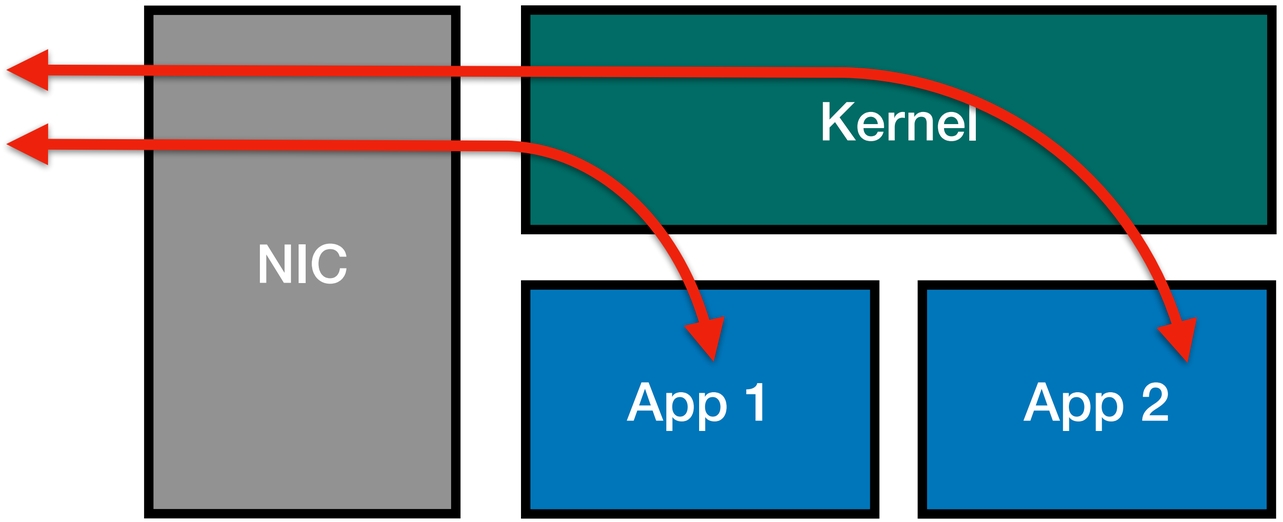


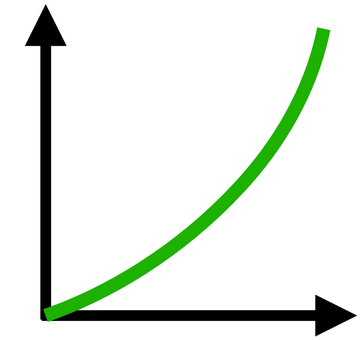


NIC Speeds

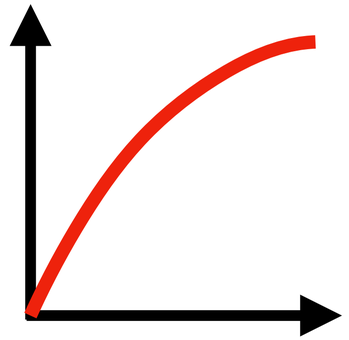
Traditional Kernel Network Stack

Most popular design today (e.g., Linux, Windows, BSD)





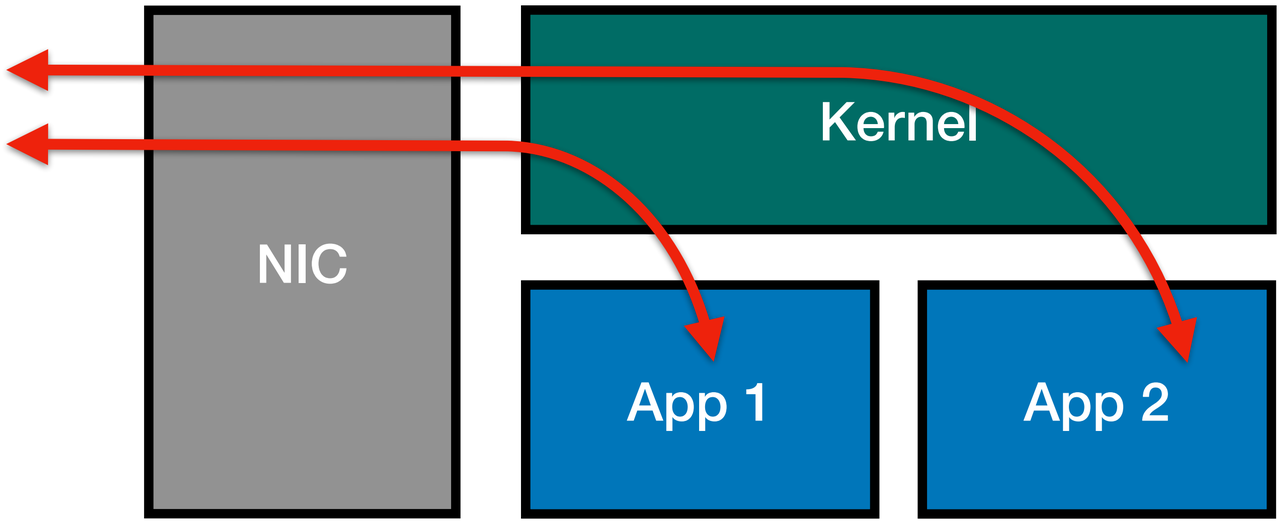
NIC Speeds

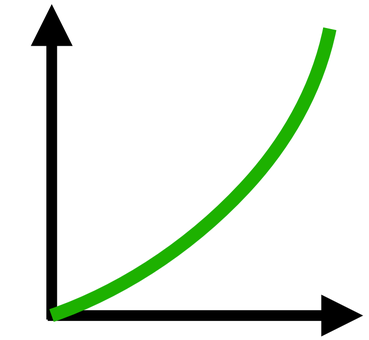


CPU Speeds

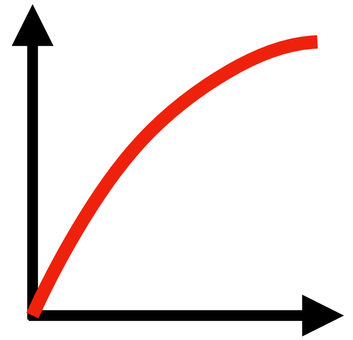
Traditional Kernel Network Stack

Most popular design today (e.g., Linux, Windows, BSD)





NIC Speeds



CPU Speeds

Low performance due to data movement:

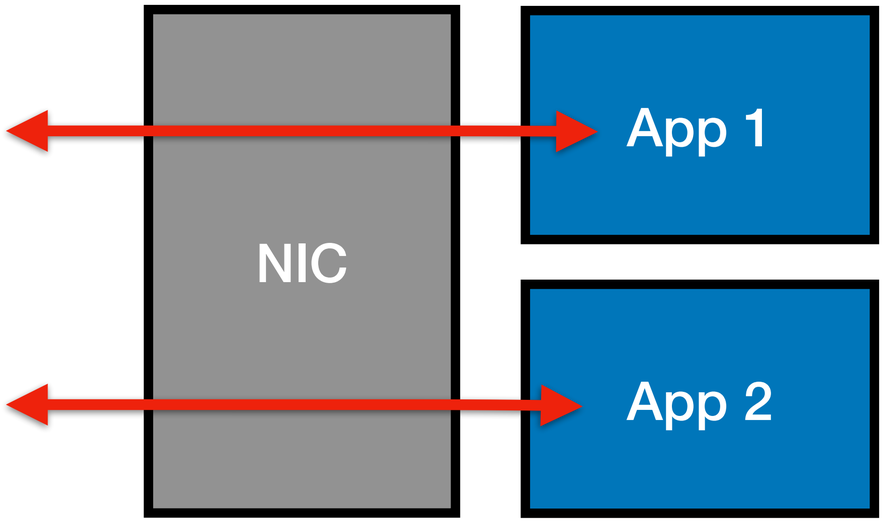
- Copies
- Syscalls

Kernel-Bypass Network Stack

Let applications interface directly with the NIC and implement the network stack in a library

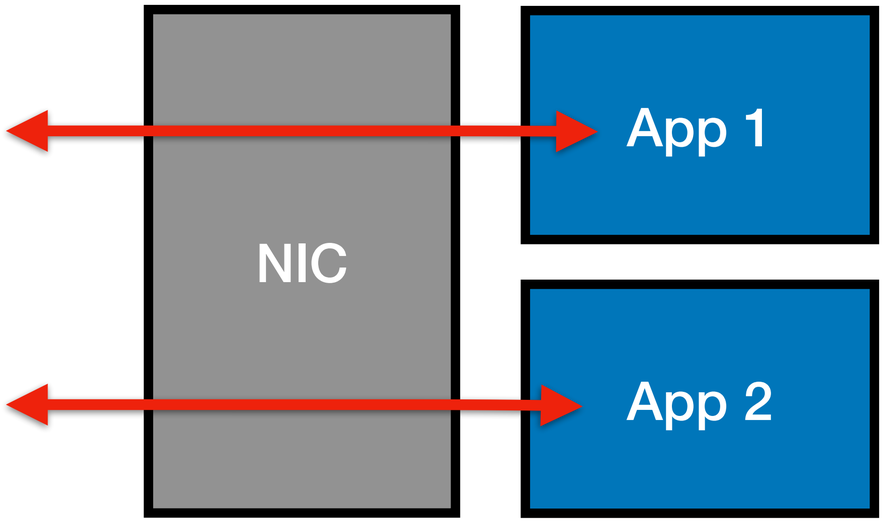
Kernel-Bypass Network Stack

Let applications interface directly with the NIC and implement the network stack in a library



Kernel-Bypass Network Stack

Let applications interface directly with the NIC and implement the network stack in a library

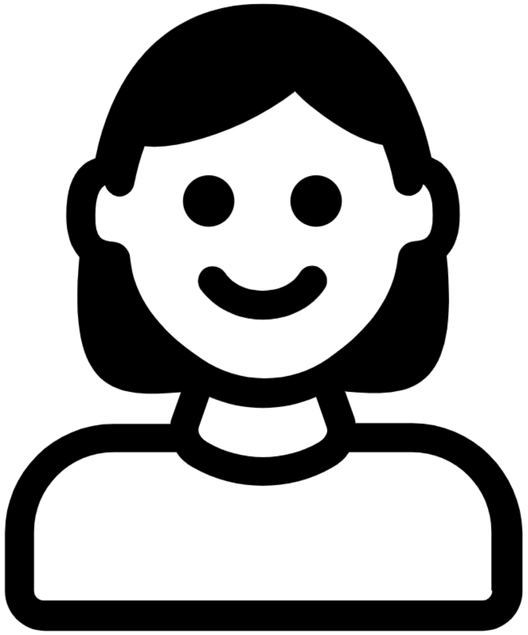


Avoids the additional data movement imposed by the kernel stack

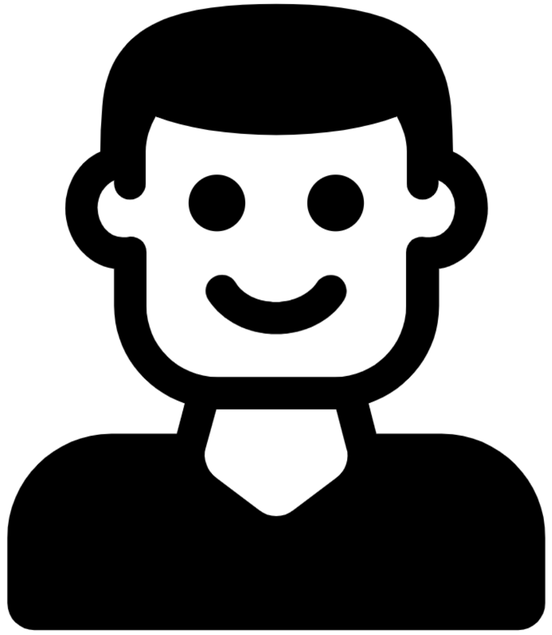
Kernel bypass leads to a maintenance and manageability nightmare for administrators!

Kernel bypass leads to a maintenance and manageability nightmare for administrators!

Main Issue:
Kernel bypass cannot fully replicate kernel functionality

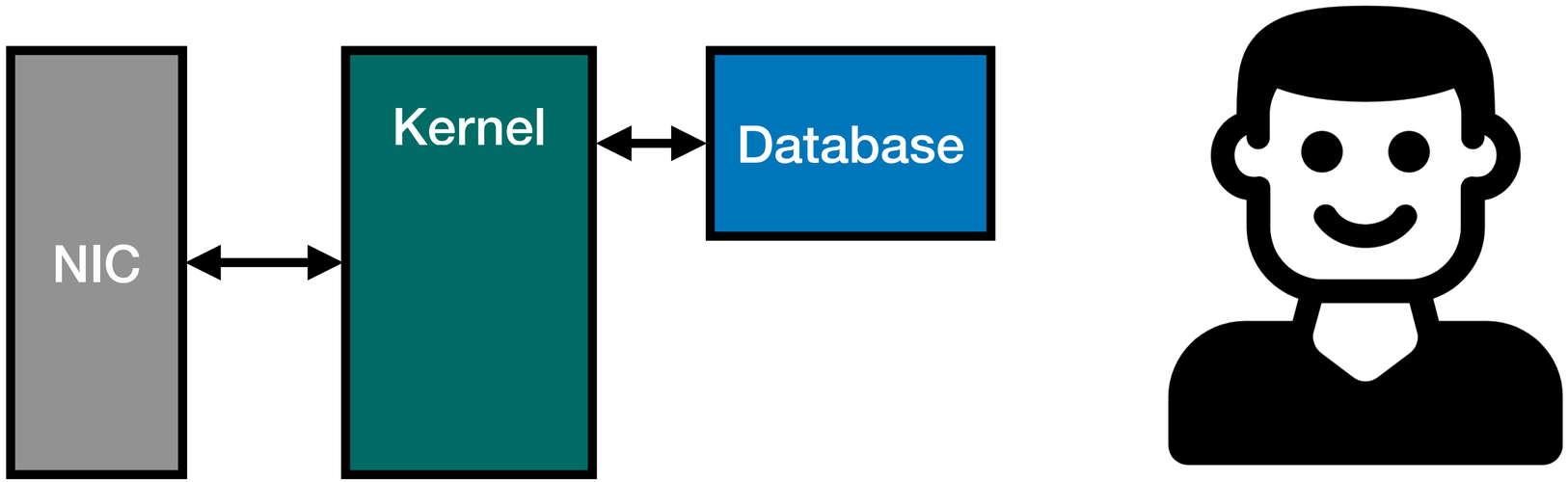


Alice
(Sysadmin)

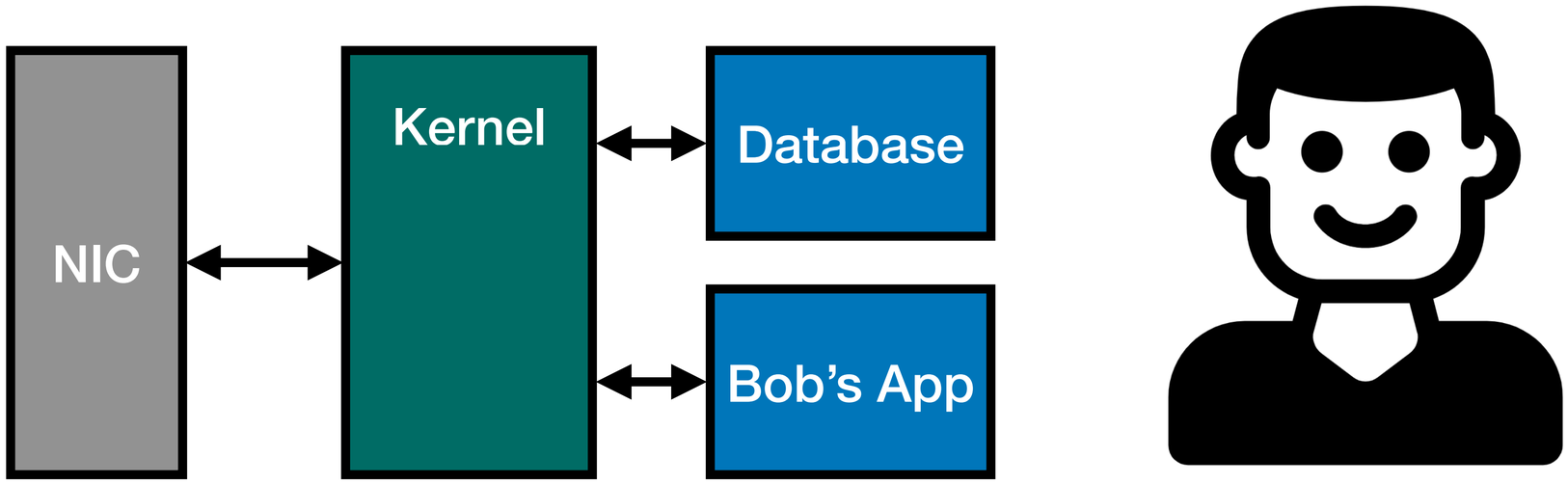


Bob
(Intern)

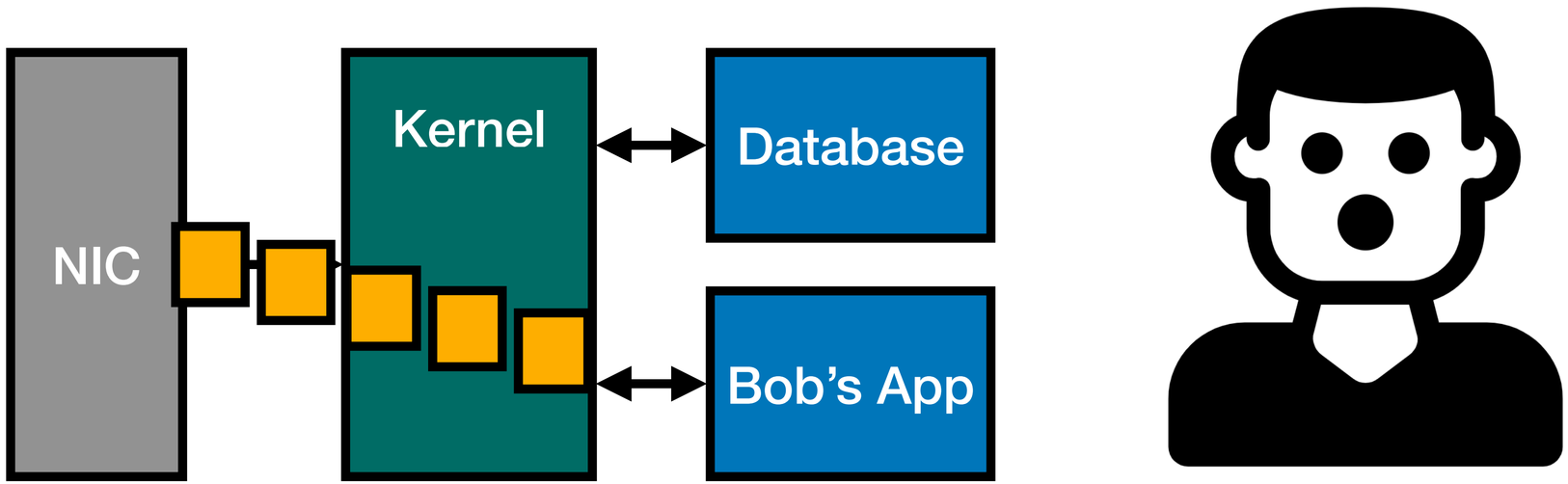
Policy Enforcement



Policy Enforcement



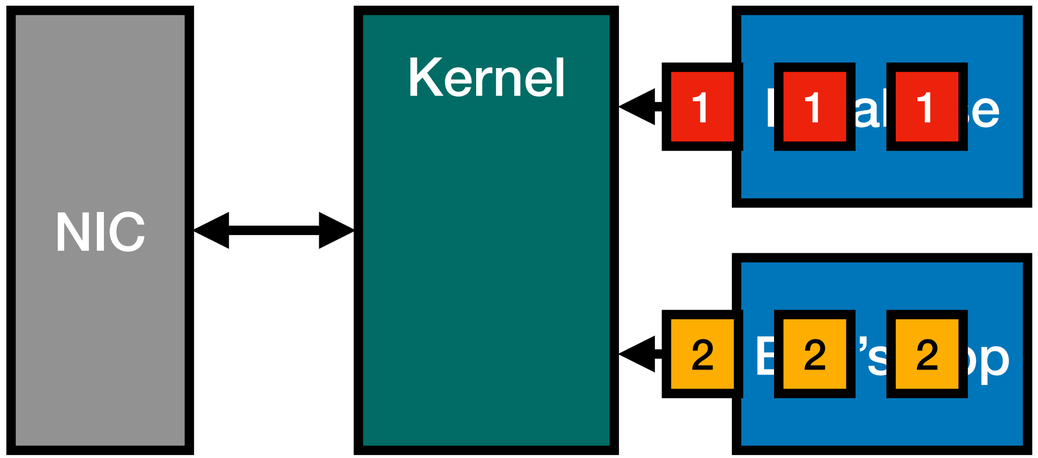
Policy Enforcement

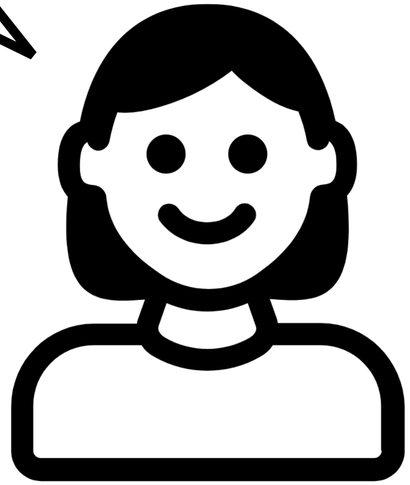


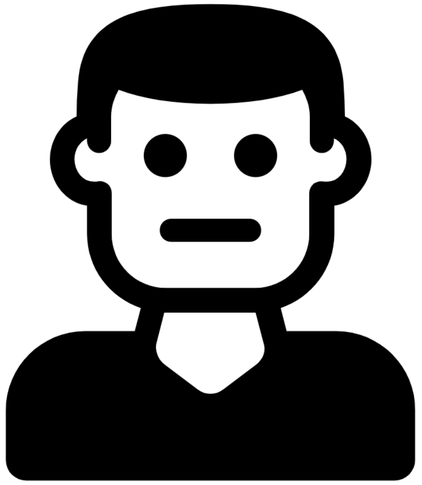
Bob's app is sending too much traffic, not leaving enough bandwidth for the database

Policy Enforcement

I want to prioritize traffic from the database over traffic from Bob's app.

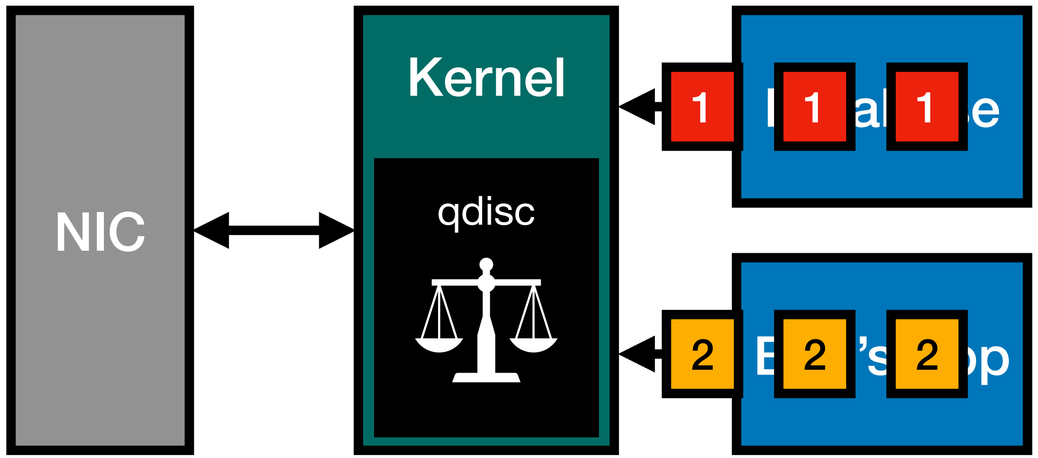


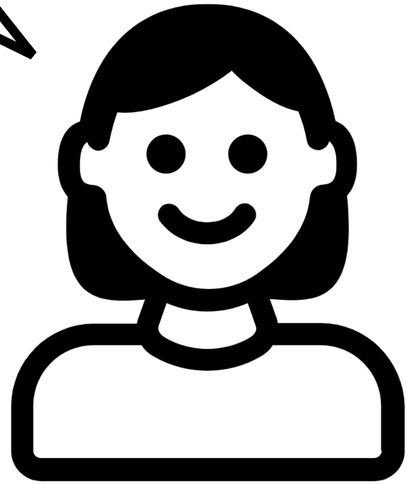


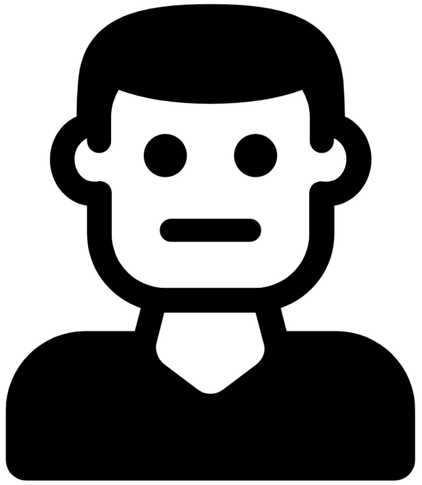


Policy Enforcement

I want to prioritize traffic from the database over traffic from Bob's app.



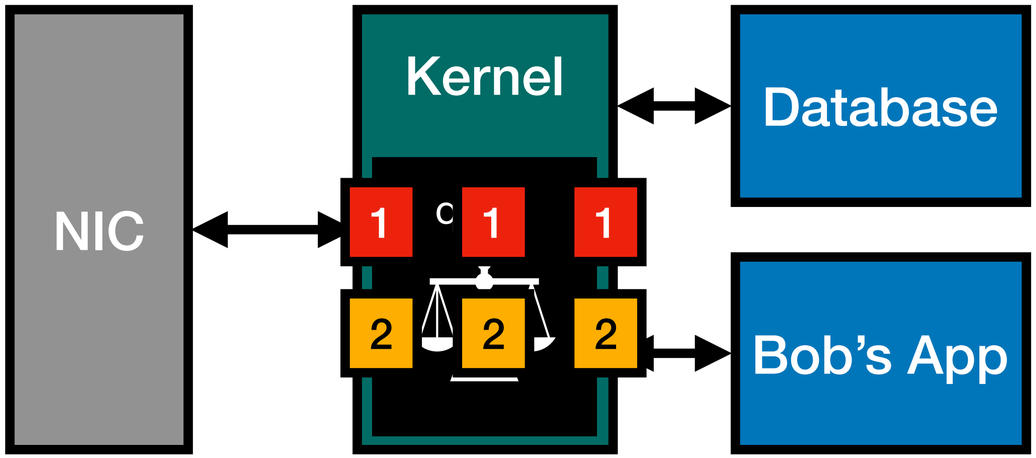


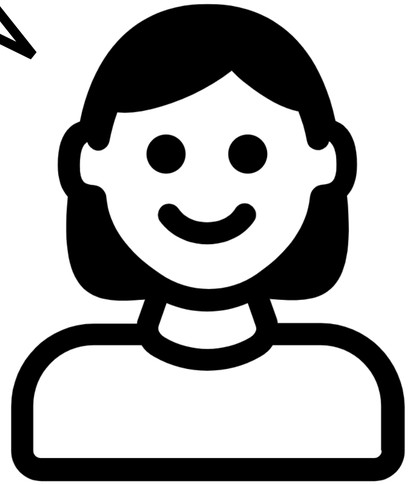


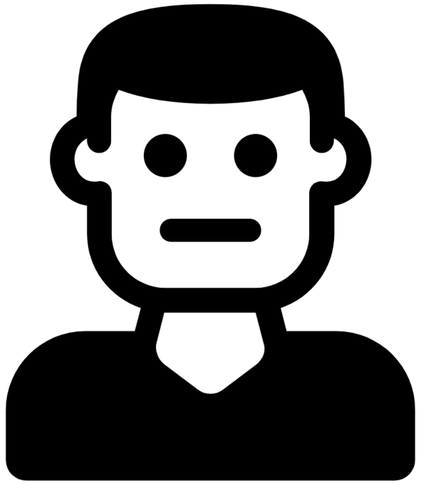
I can use `tc` to set queuing disciplines!

Policy Enforcement

I want to prioritize traffic from the database over traffic from Bob's app.



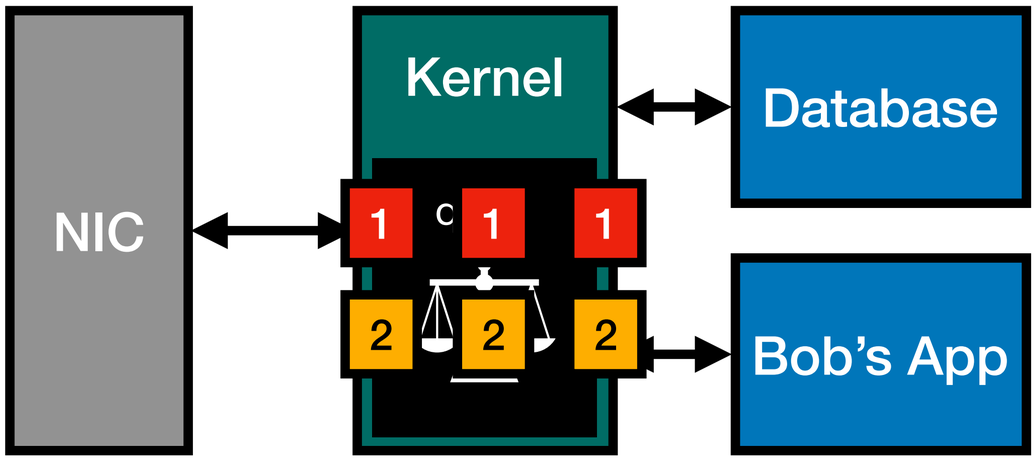


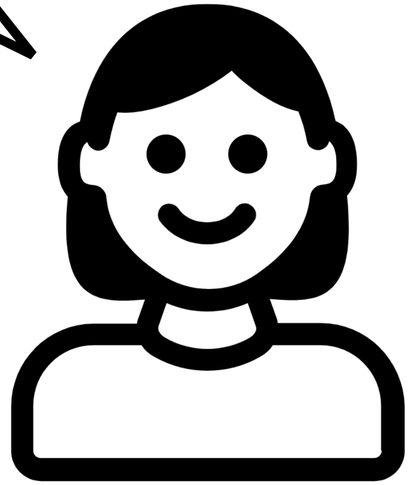


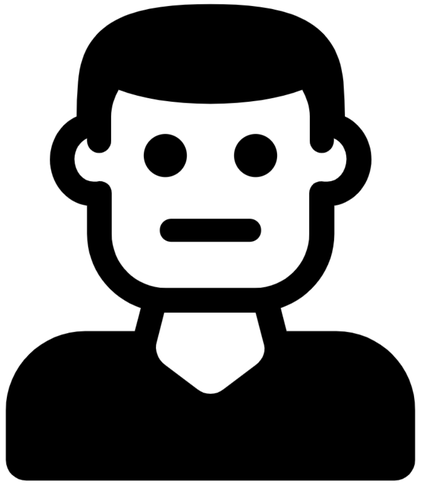
I can use `tc` *to set queuing disciplines!*

Policy Enforcement

I want to prioritize traffic from the database over traffic from Bob's app.



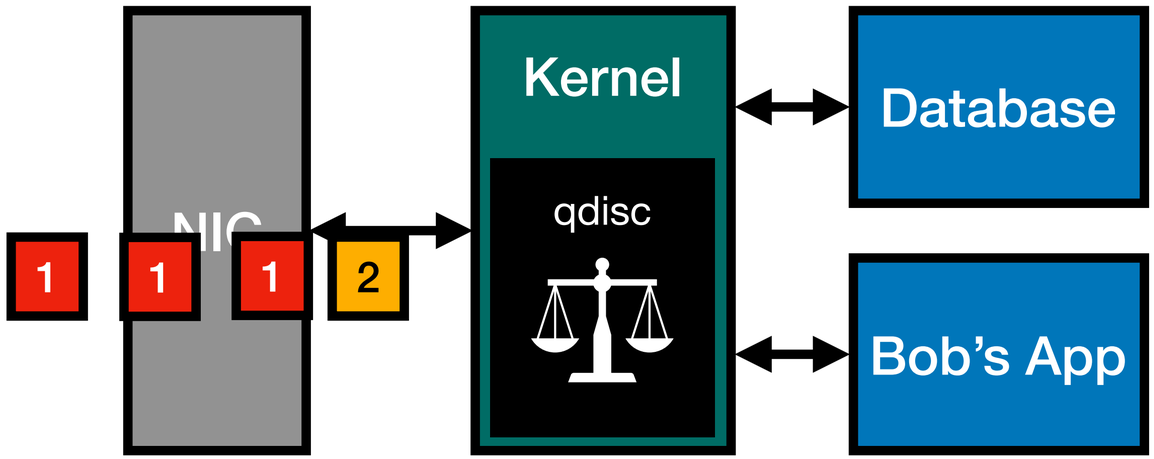


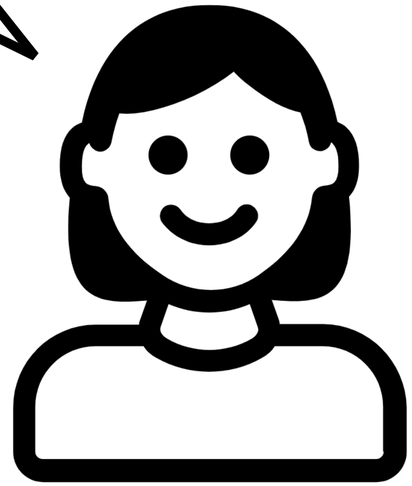


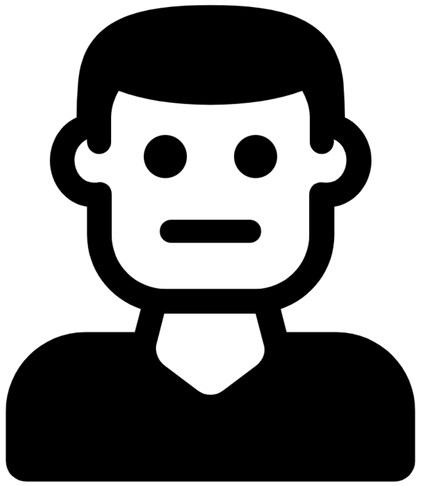
I can use `tc` *to set queuing disciplines!*

Policy Enforcement

I want to prioritize traffic from the database over traffic from Bob's app.

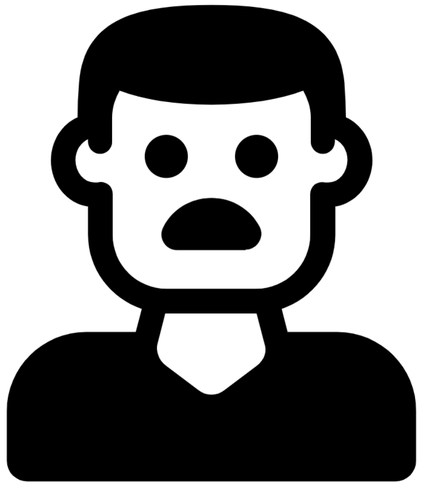


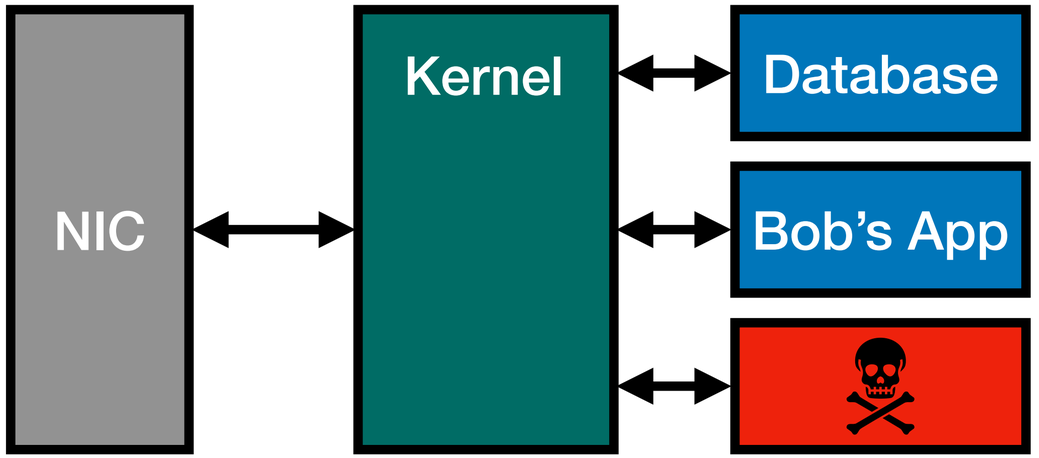




I can use `tc` to set queuing disciplines!

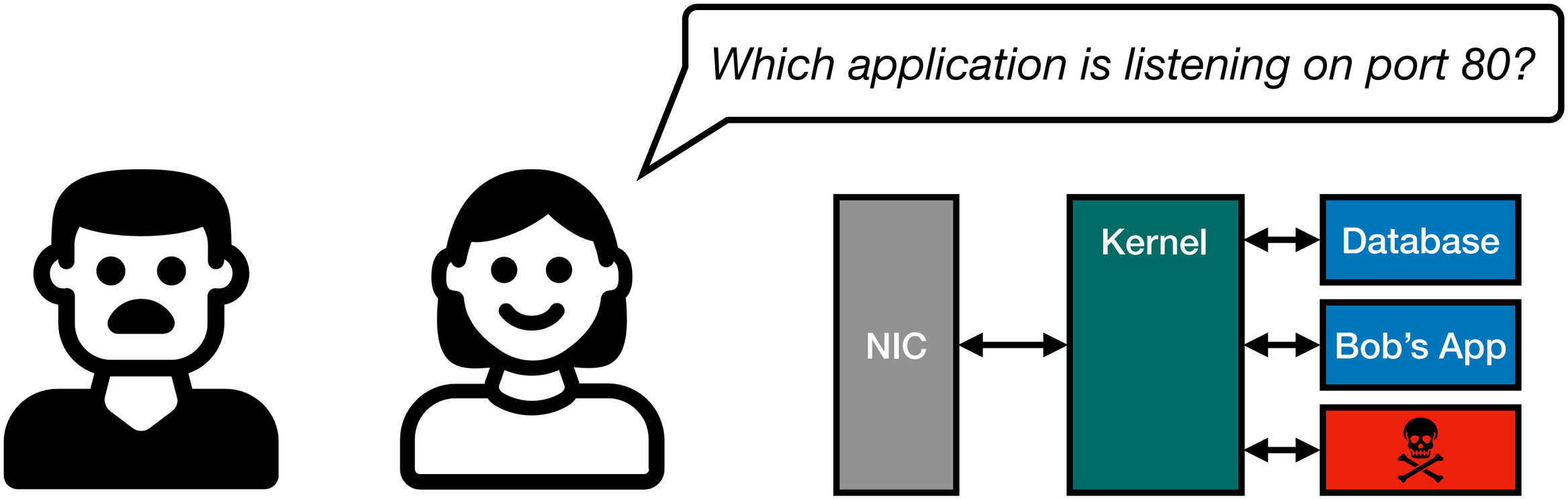
Debugging



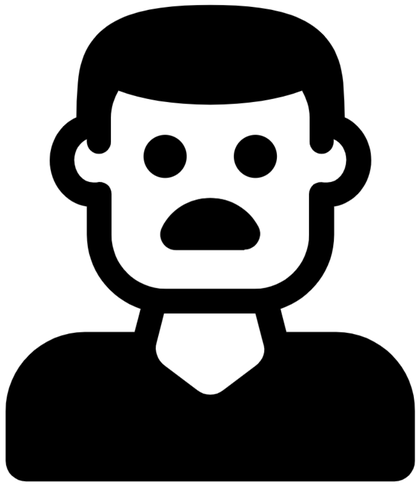


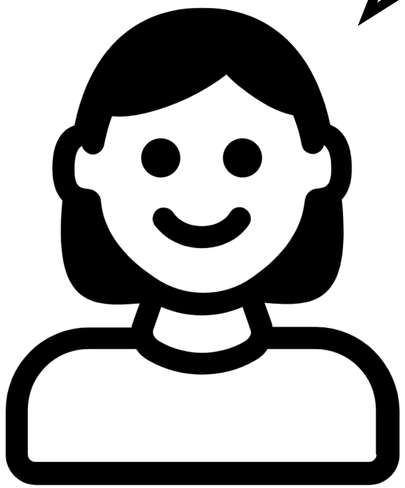
Bob set a weak password and the server got compromised with a malicious webserver

Debugging

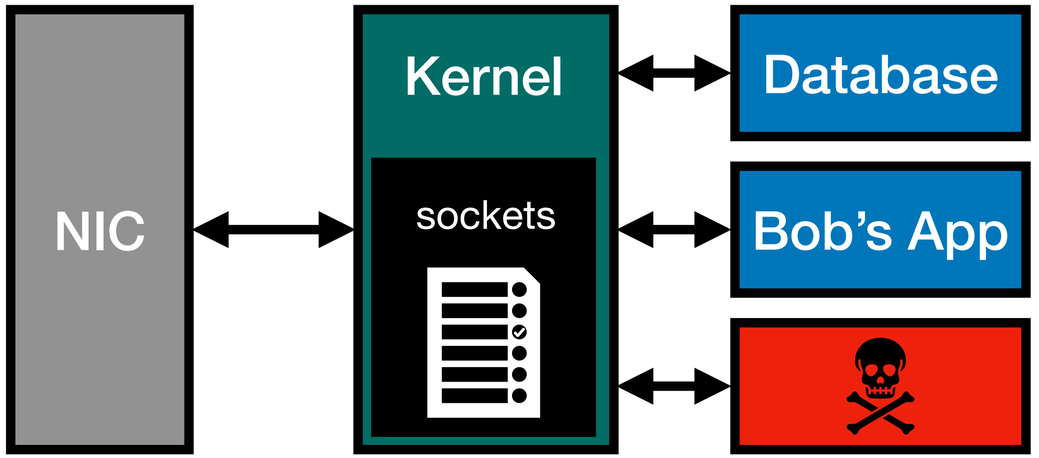

Which application is listening on port 80?
NIC
Kernel
Database
Bob's App

Debugging





Which application is listening on port 80?



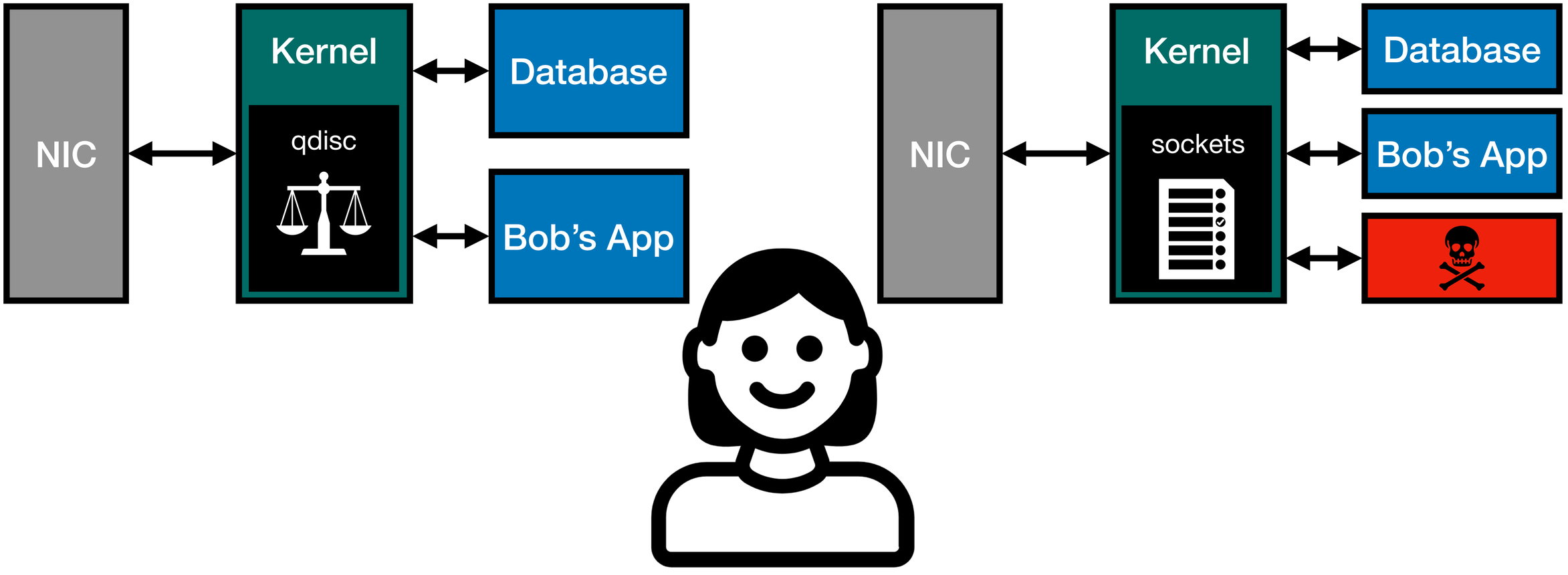
I can use tools like `netstat/ss` *to check all open sockets!*

Policy Enforcement

“I want to prioritize traffic from the database over traffic from Bob’s app”

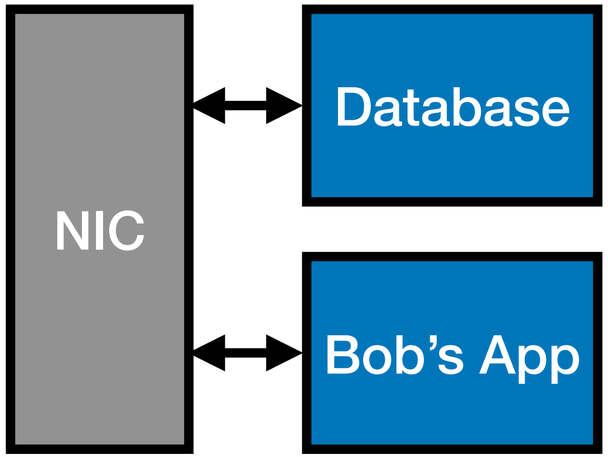
Debugging

“Which application is listening on port 80?”



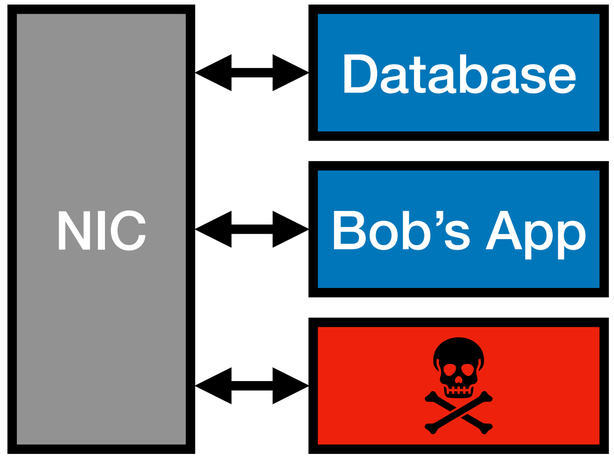
Policy Enforcement

“I want to prioritize traffic from the database over traffic from Bob’s app”



Debugging

“Which application is listening on port 80?”

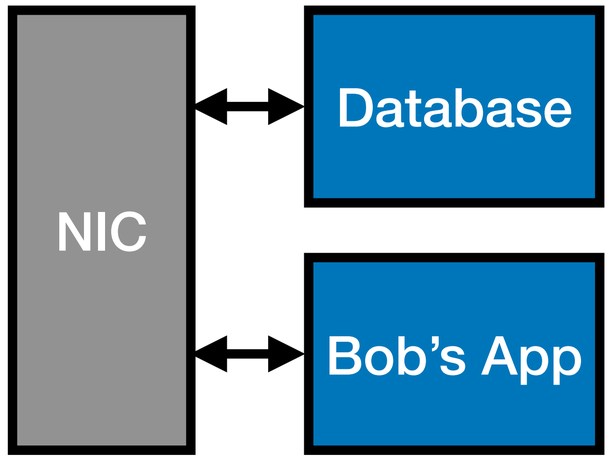


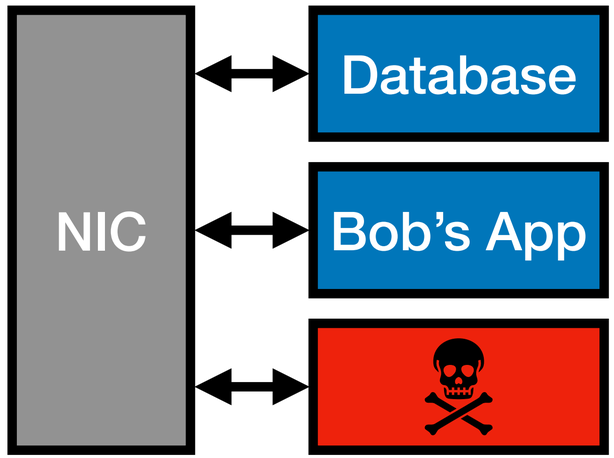
Policy Enforcement

“I want to prioritize traffic from the database over traffic from Bob’s app”

Debugging

“Which application is listening on port 80?”

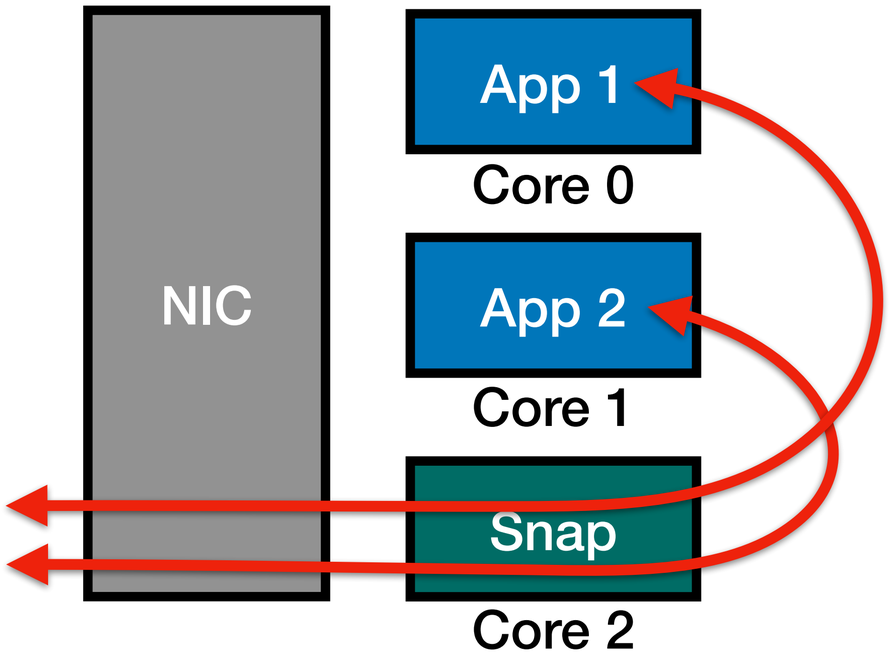




Problem: Lack of **kernel interposition**

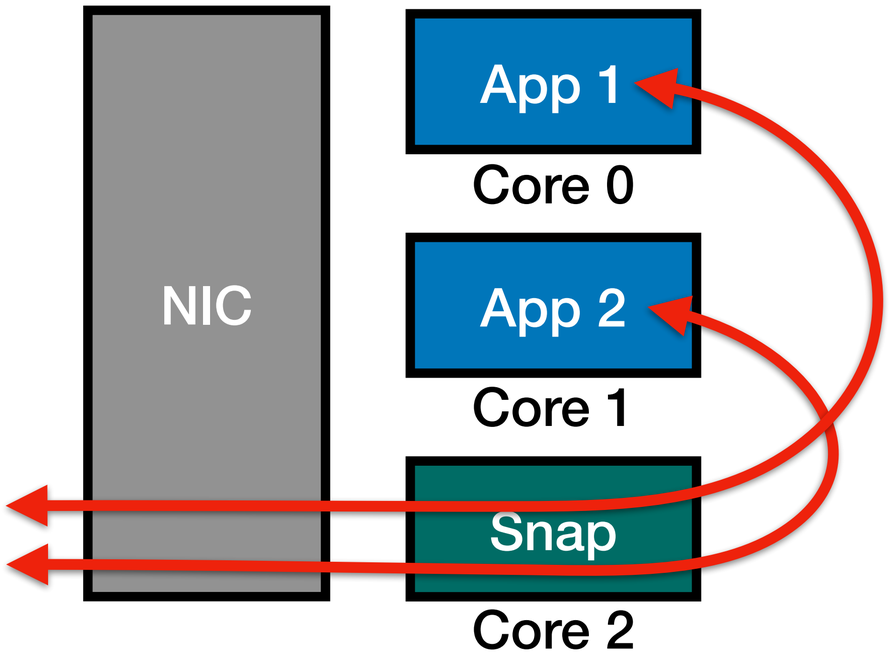
Existing Alternatives

Google Snap [SOSP ’19]



Existing Alternatives

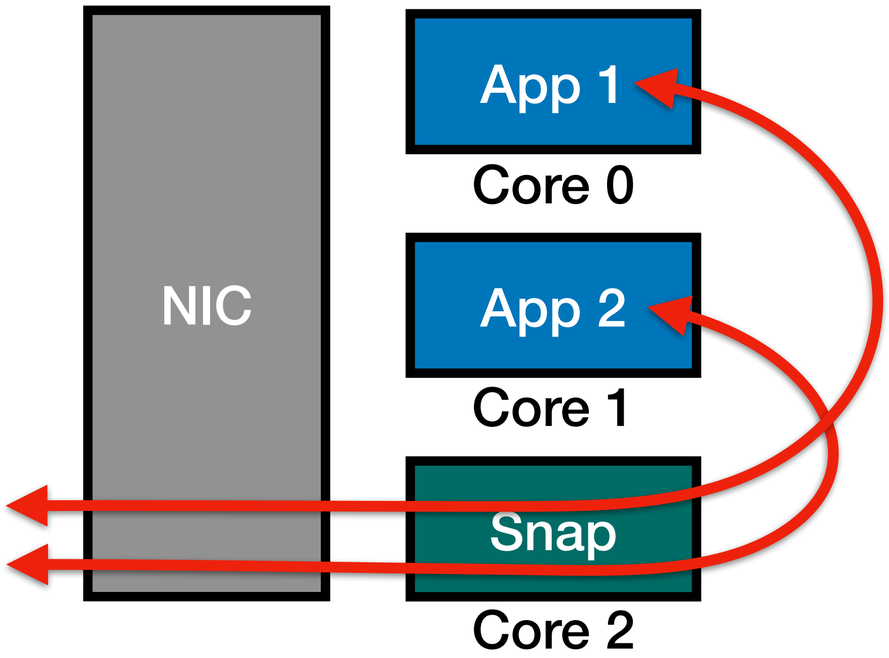
Google Snap [SOSP ’19]



Reintroduces data movement overheads

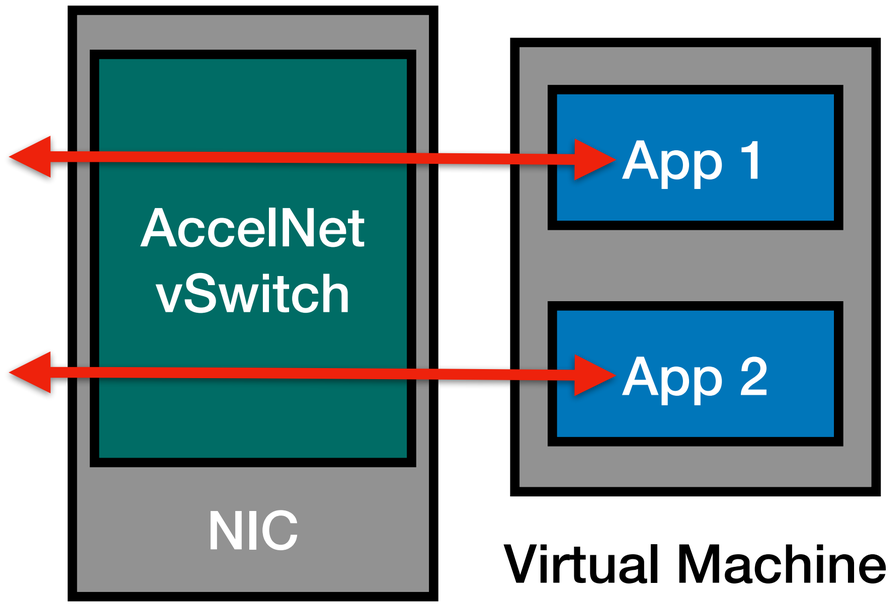
Existing Alternatives

Google Snap [SOSP ’19]



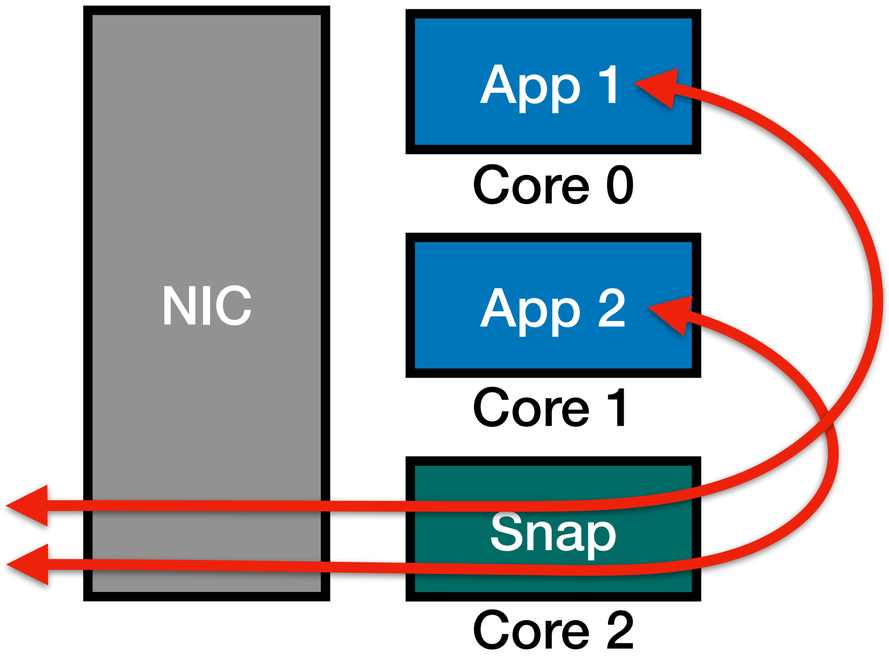
Reintroduces data movement overheads

Microsoft AccelNet [NSDI ’18]



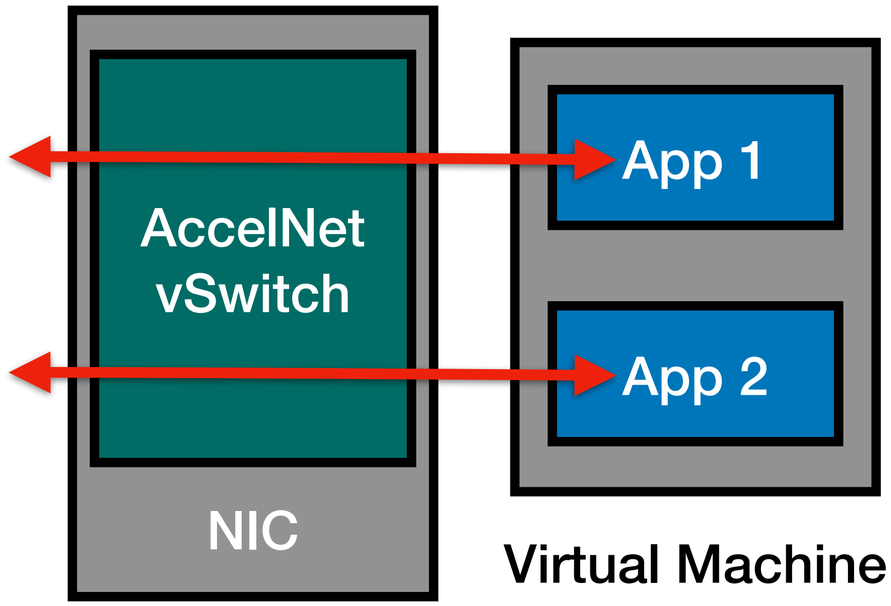
Existing Alternatives

Google Snap [SOSP ’19]



Reintroduces data movement overheads

Microsoft AccelNet [NSDI ’18]

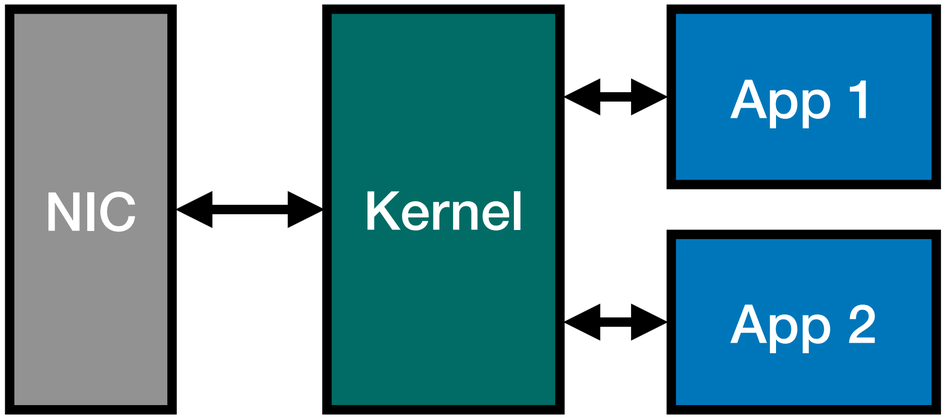


Control lies with the hypervisor/ no introspection into applications

So what do we *really* want?

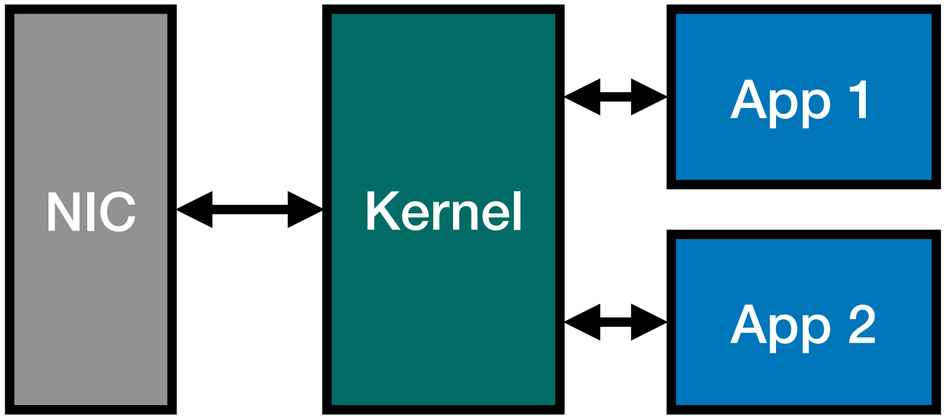
So what do we *really* want?

Logically like the kernel network stack

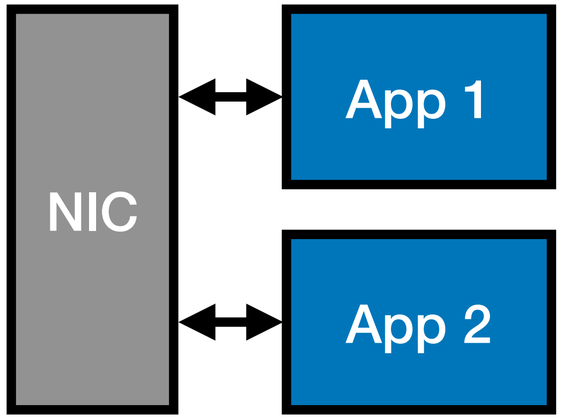


So what do we *really* want?

Logically like the kernel network stack

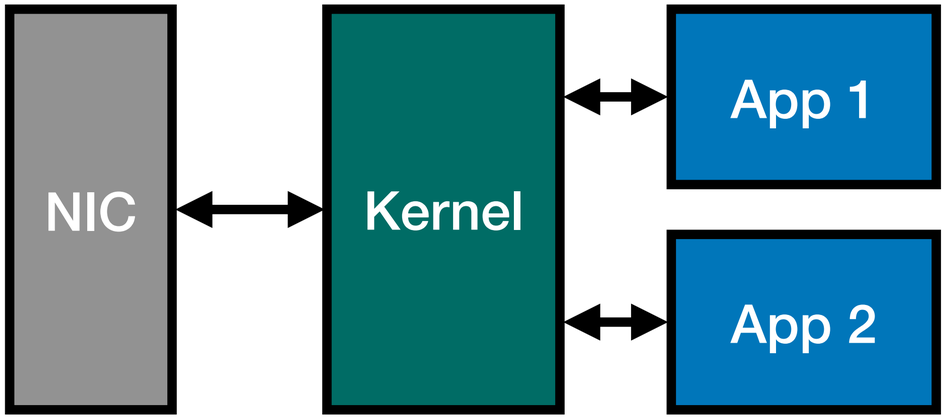


Physically like kernel bypass



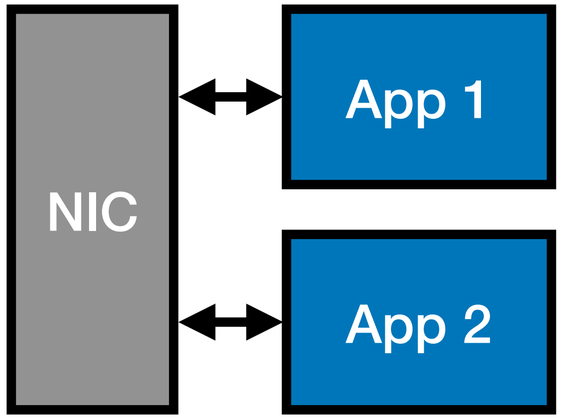
So what do we *really* want?

Logically like the kernel network stack



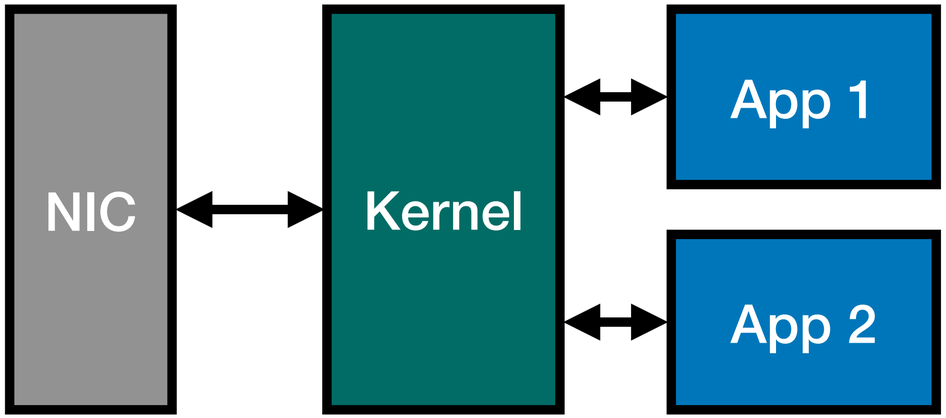
Kernel interposition

Physically like kernel bypass



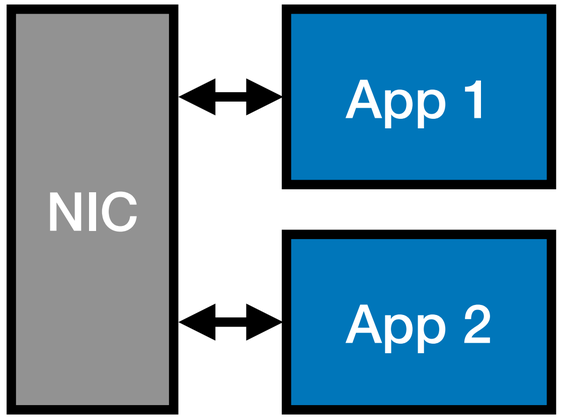
So what do we *really* want?

Logically like the kernel network stack



Kernel interposition

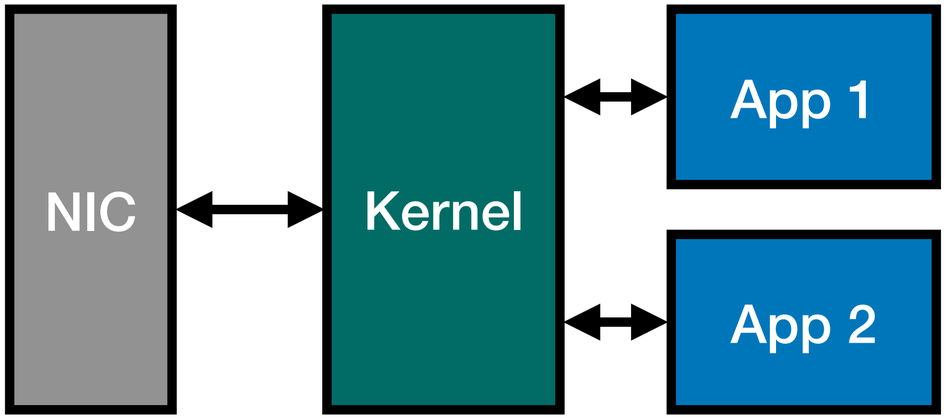
Physically like kernel bypass



On path

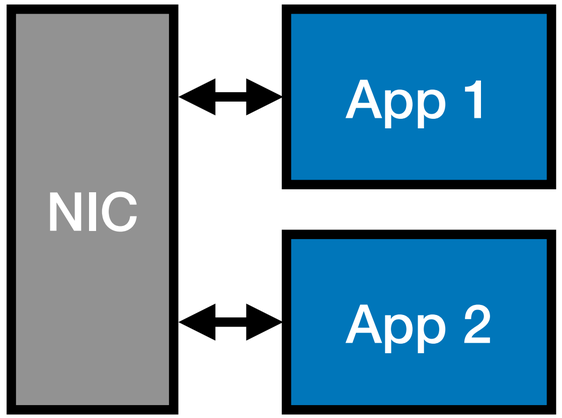
So what do we *really* want?

Logically like the kernel network stack



Kernel interposition

Physically like kernel bypass

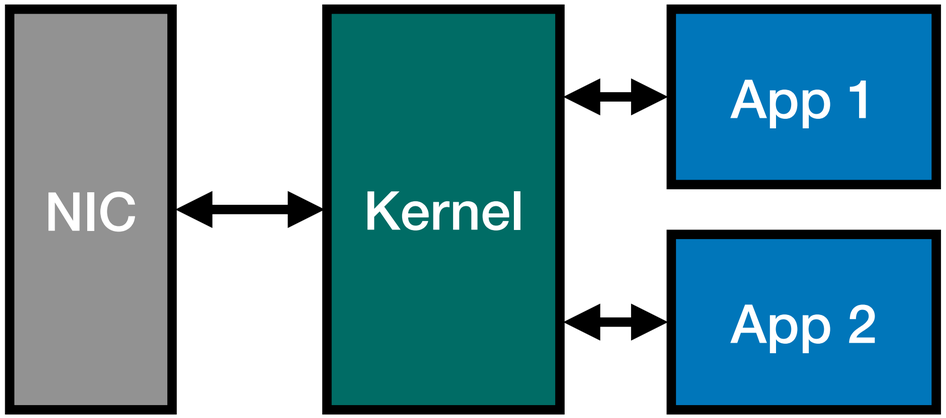


On path

KOPI: Kernel On-Path Interposition

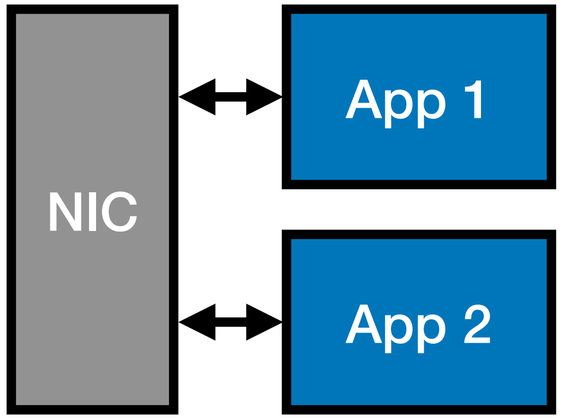
So what do we *really* want?

Logically like the kernel network stack



Kernel interposition

Physically like kernel bypass



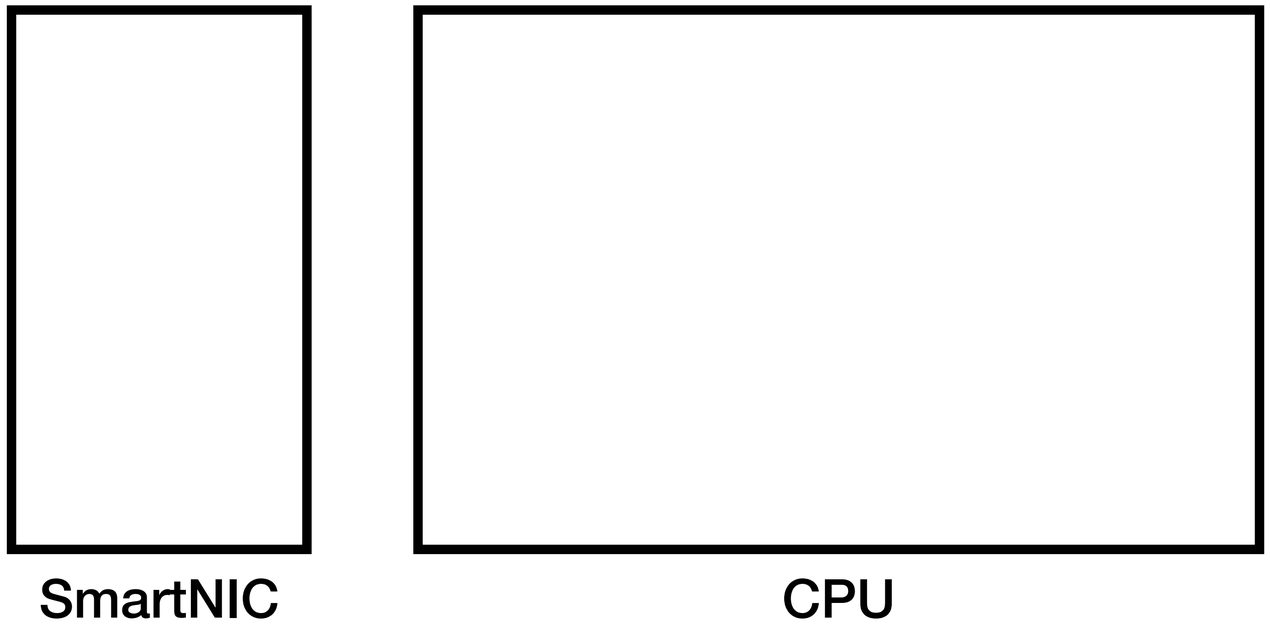
On path

KOPI: Kernel On-Path Interposition

Leverage programmable SmartNICs to implement a network dataplane that is both on path and logically controlled by the kernel

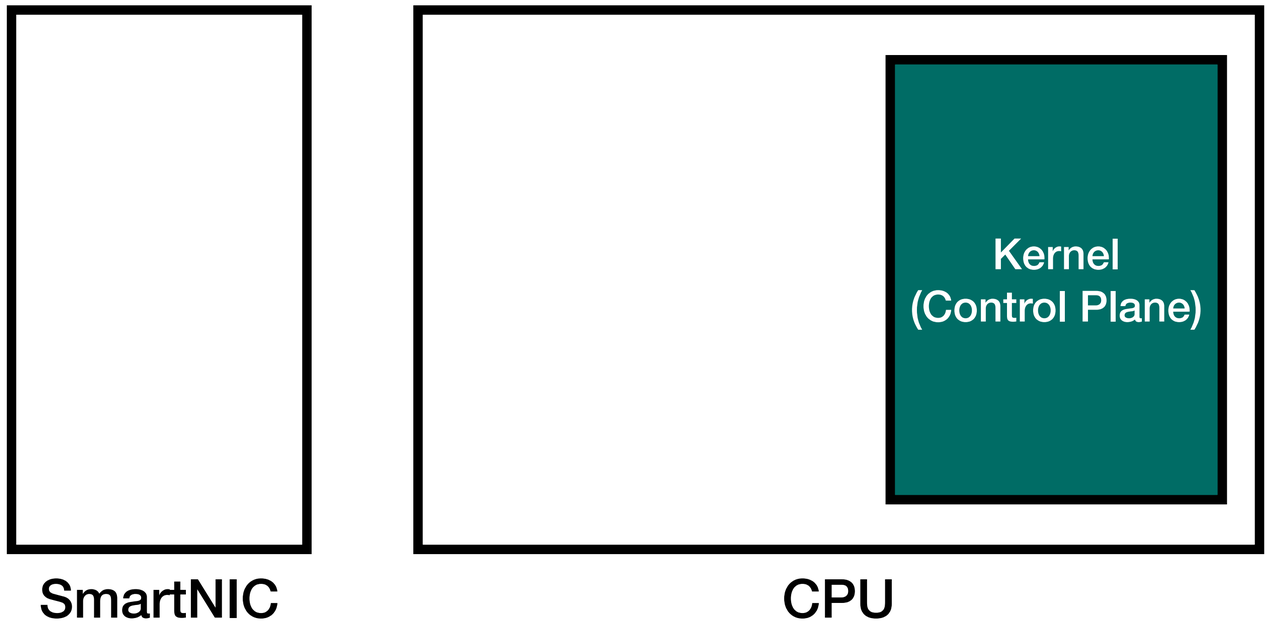
KOPI: Kernel On-Path Interposition

Leverage programmable SmartNICs to implement a network dataplane that is both on path and logically controlled by the kernel



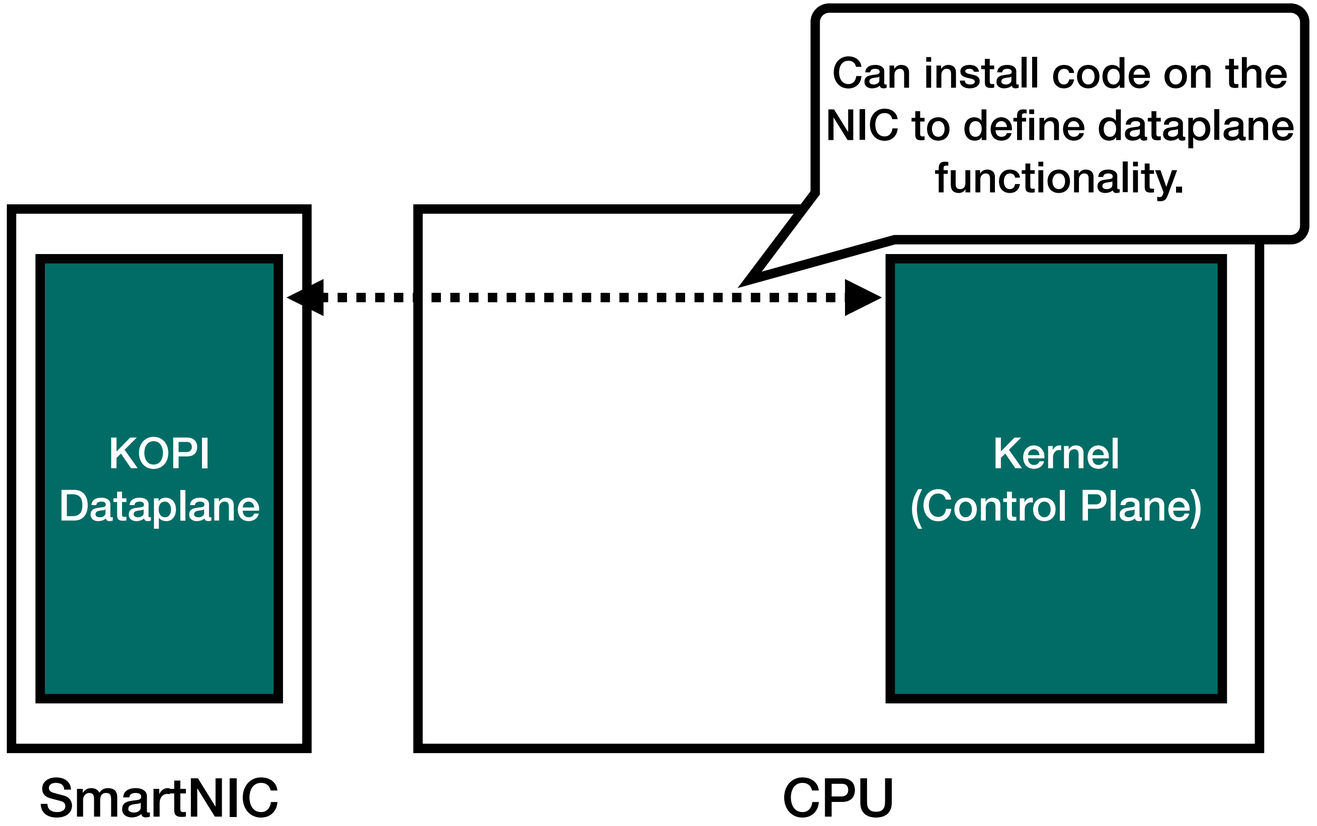
KOPI: Kernel On-Path Interposition

Leverage programmable SmartNICs to implement a network dataplane that is both on path and logically controlled by the kernel



KOPI: Kernel On-Path Interposition

Leverage programmable SmartNICs to implement a network dataplane that is both on path and logically controlled by the kernel

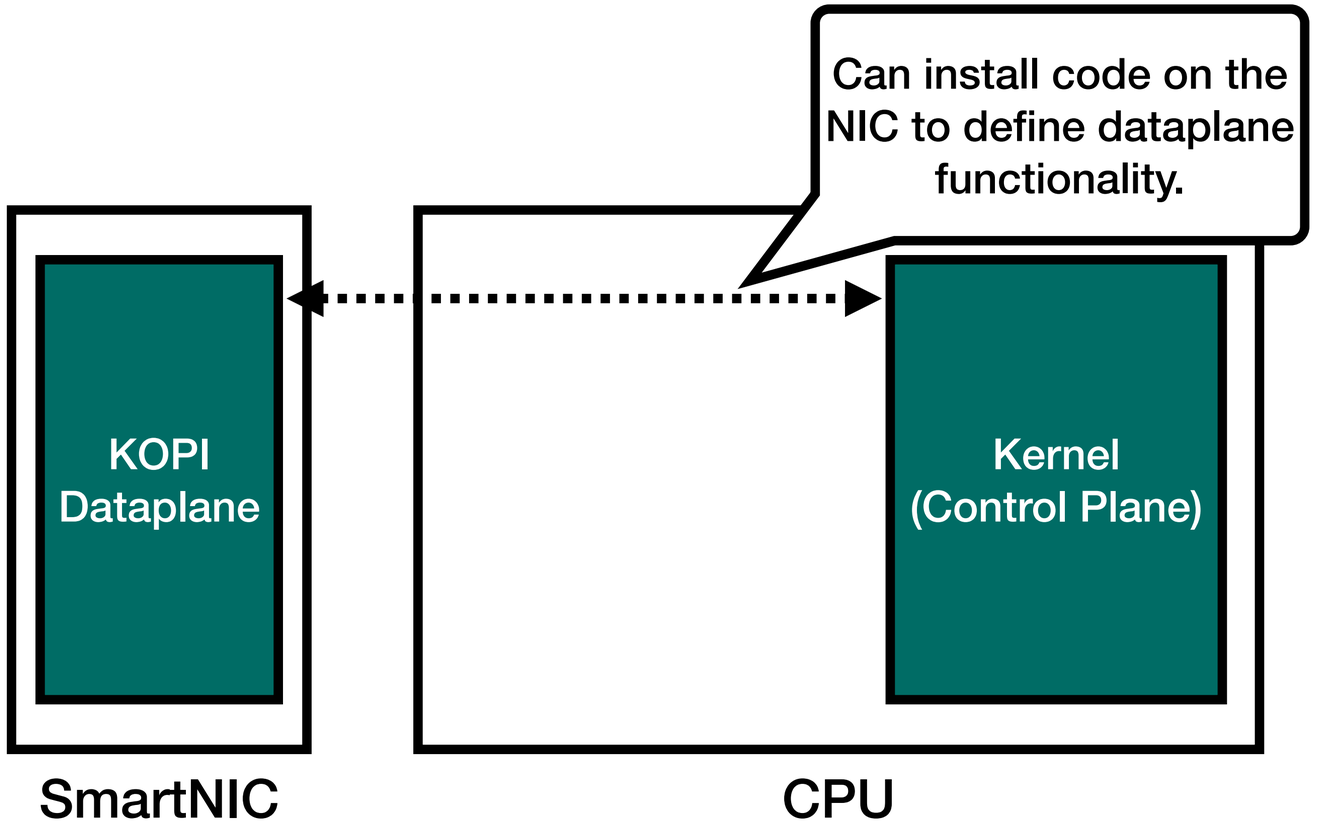


KOPI: Kernel On-Path Interposition

Leverage programmable SmartNICs to implement a network dataplane that is both on path and logically controlled by the kernel

Number of commits on Linux last year

net/netfilter: 377
net/sched: 249

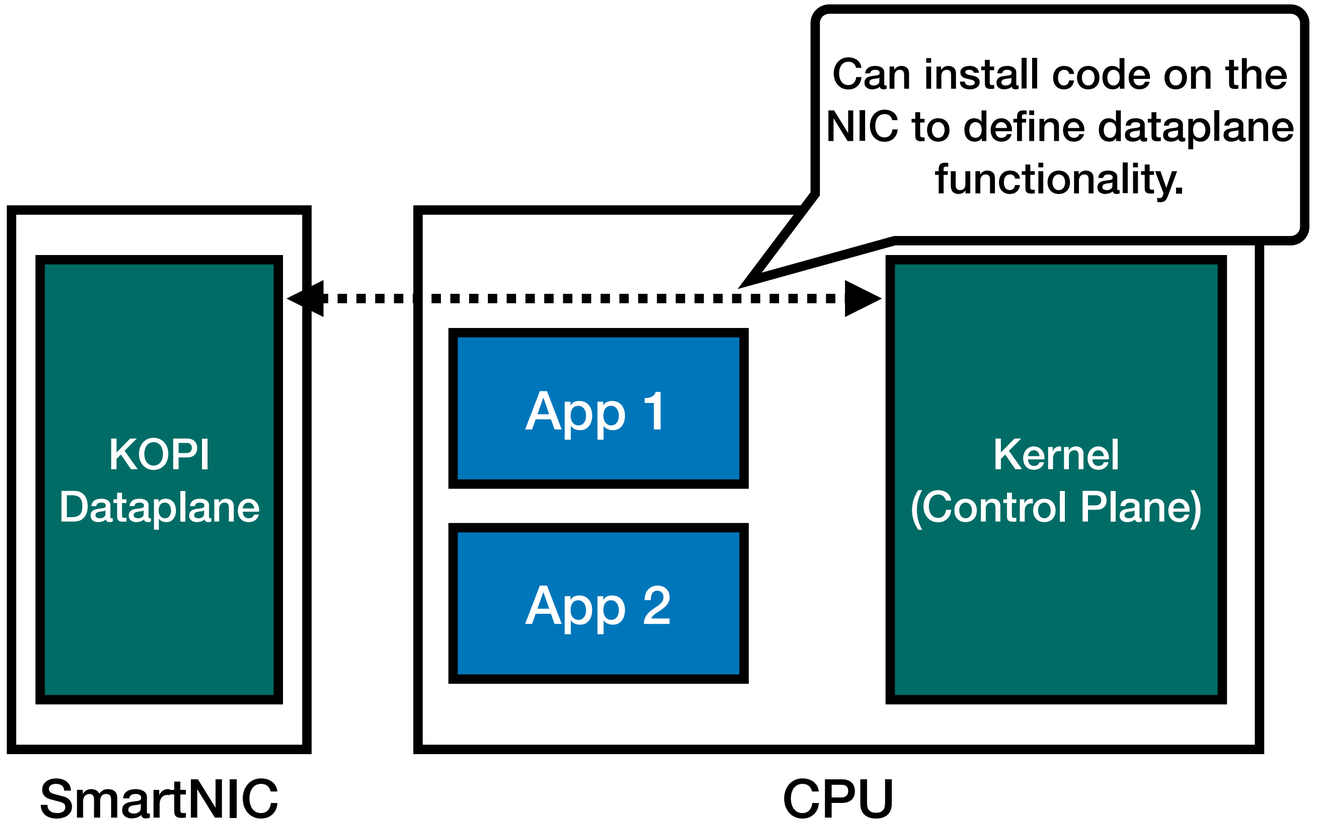


KOPI: Kernel On-Path Interposition

Leverage programmable SmartNICs to implement a network dataplane that is both on path and logically controlled by the kernel

Number of commits on Linux last year

net/netfilter: 377
net/sched: 249

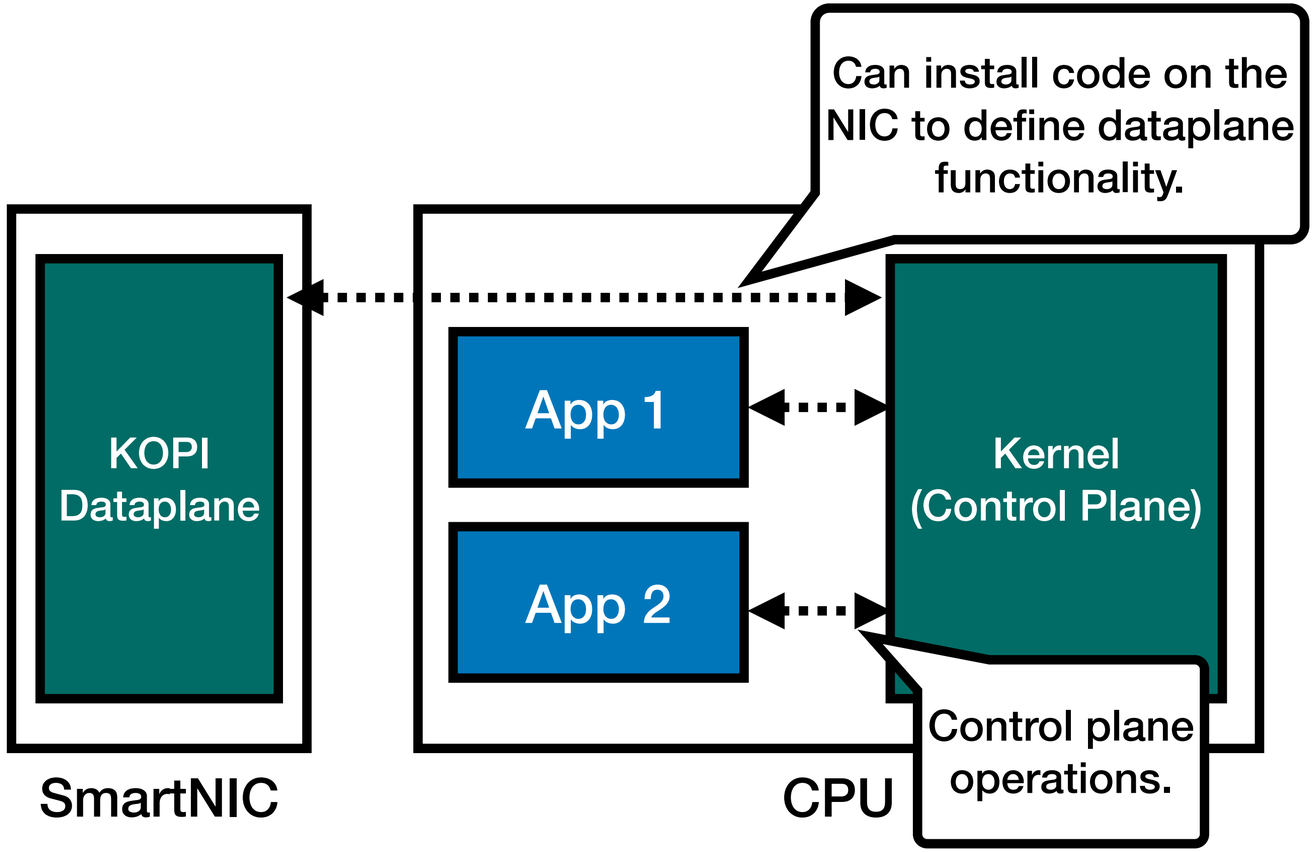


KOPI: Kernel On-Path Interposition

Leverage programmable SmartNICs to implement a network dataplane that is both on path and logically controlled by the kernel

Number of commits on Linux last year

net/netfilter: 377
net/sched: 249

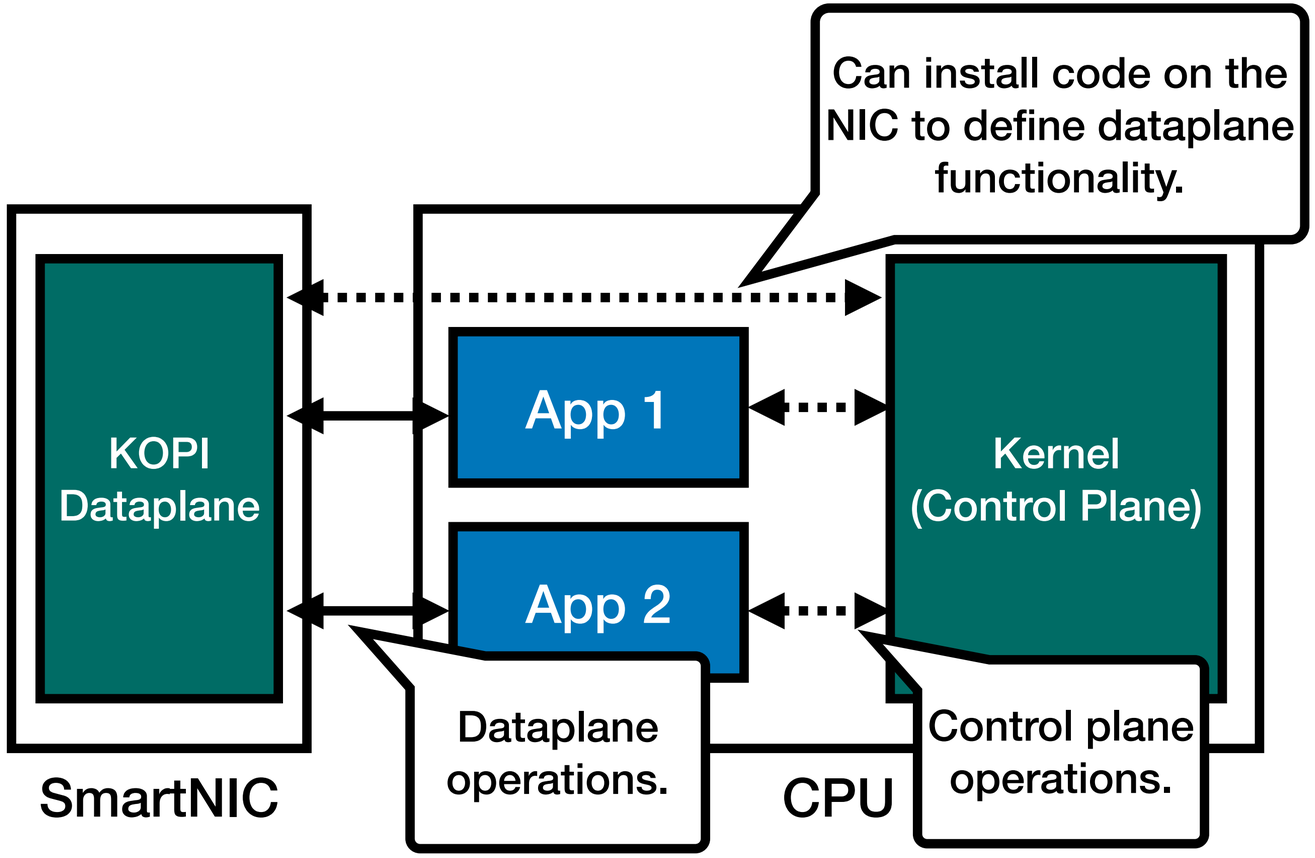


KOPI: Kernel On-Path Interposition

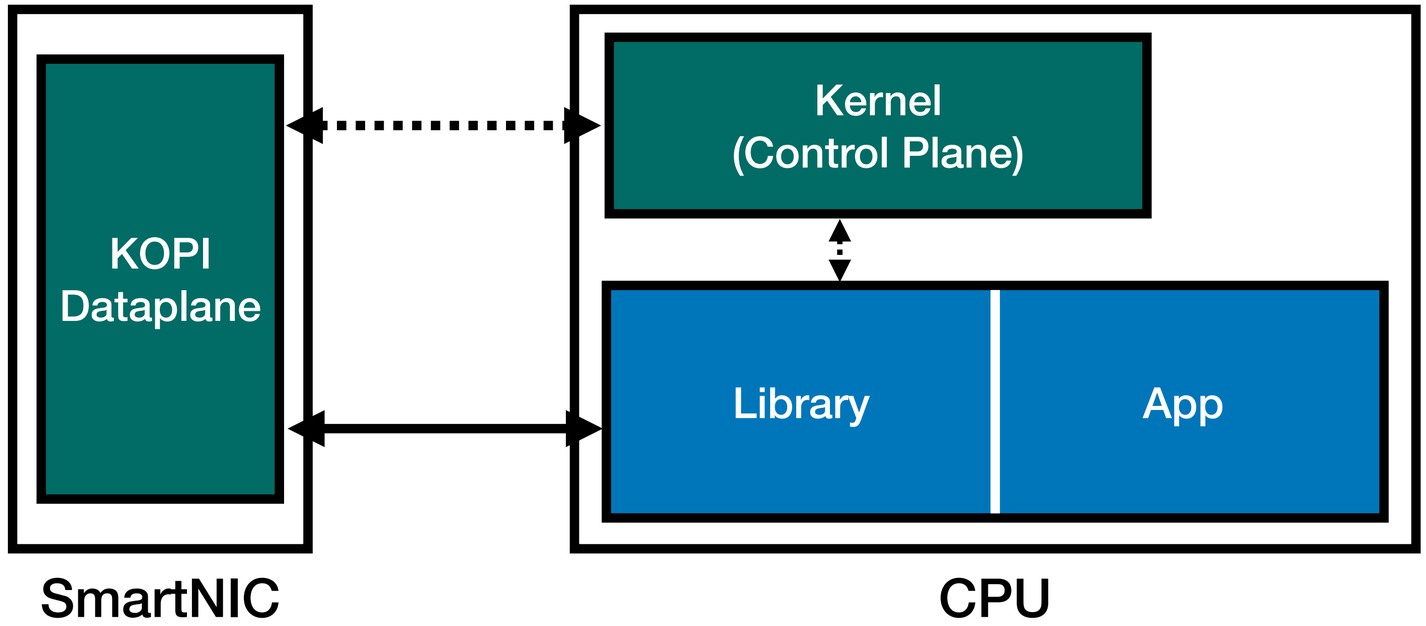
Leverage programmable SmartNICs to implement a network dataplane that is both on path and logically controlled by the kernel

Number of commits on Linux last year

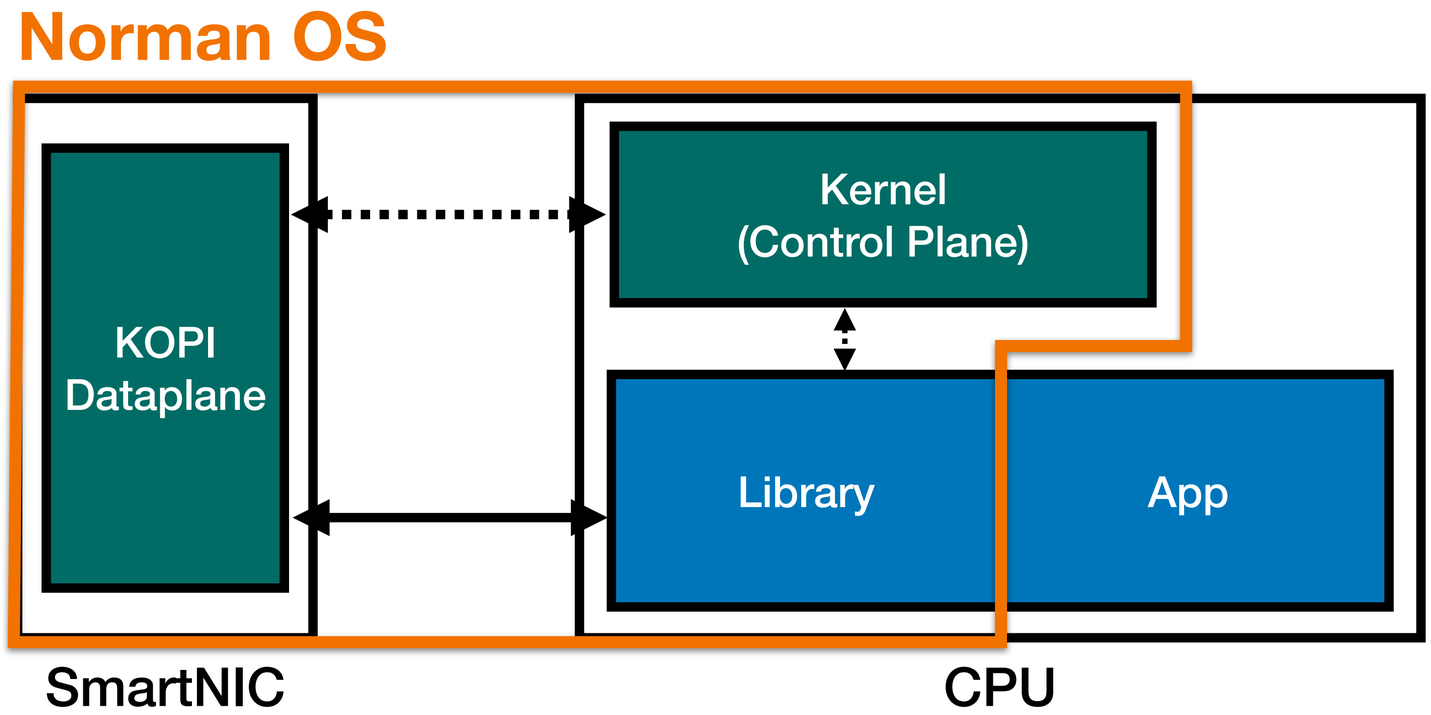
net/netfilter: 377
net/sched: 249



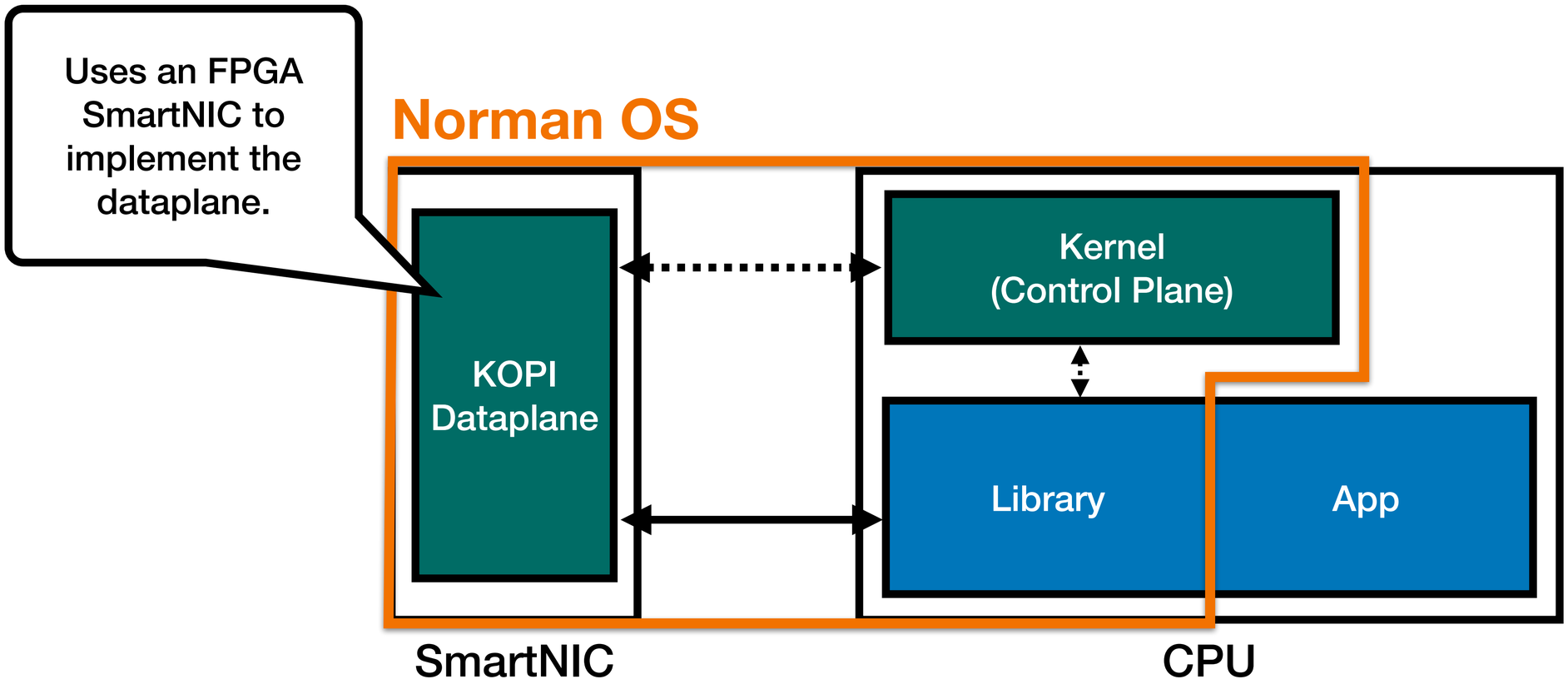
Overview of Norman: An OS that Implements KOPI



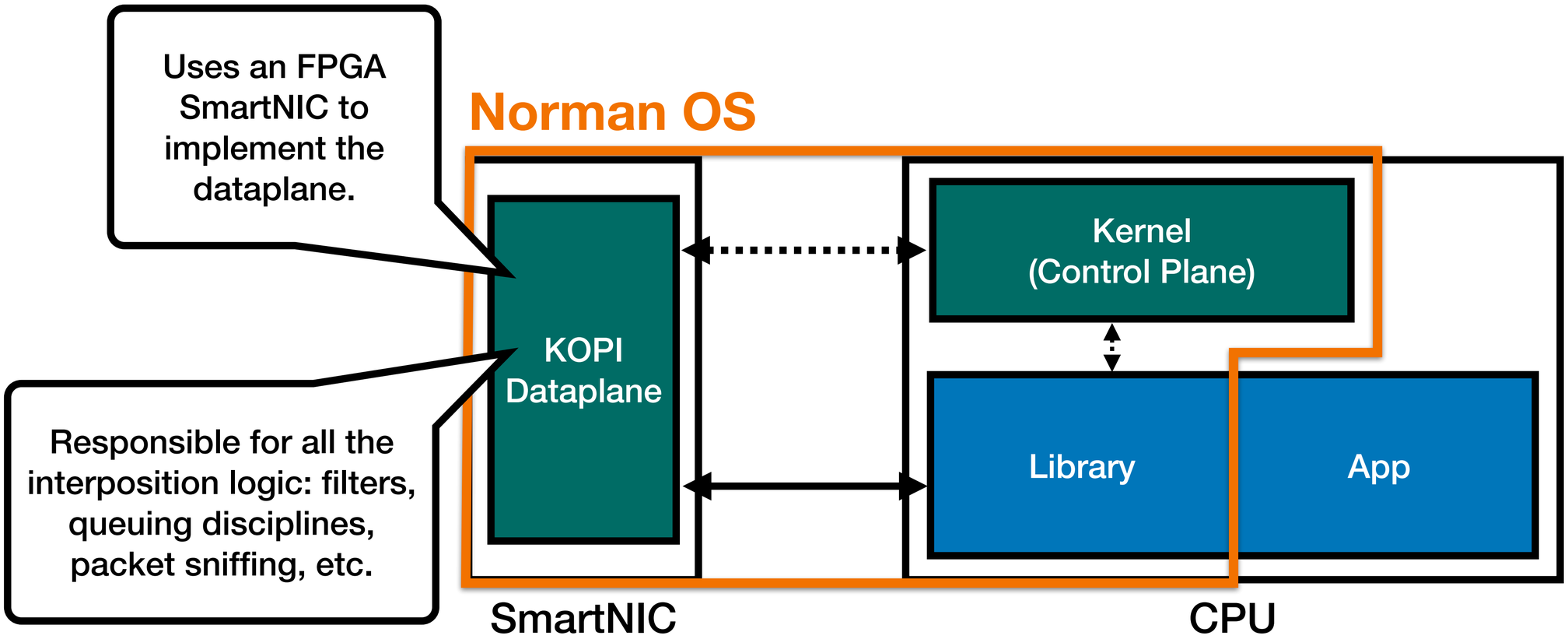
Overview of Norman: An OS that Implements KOPI



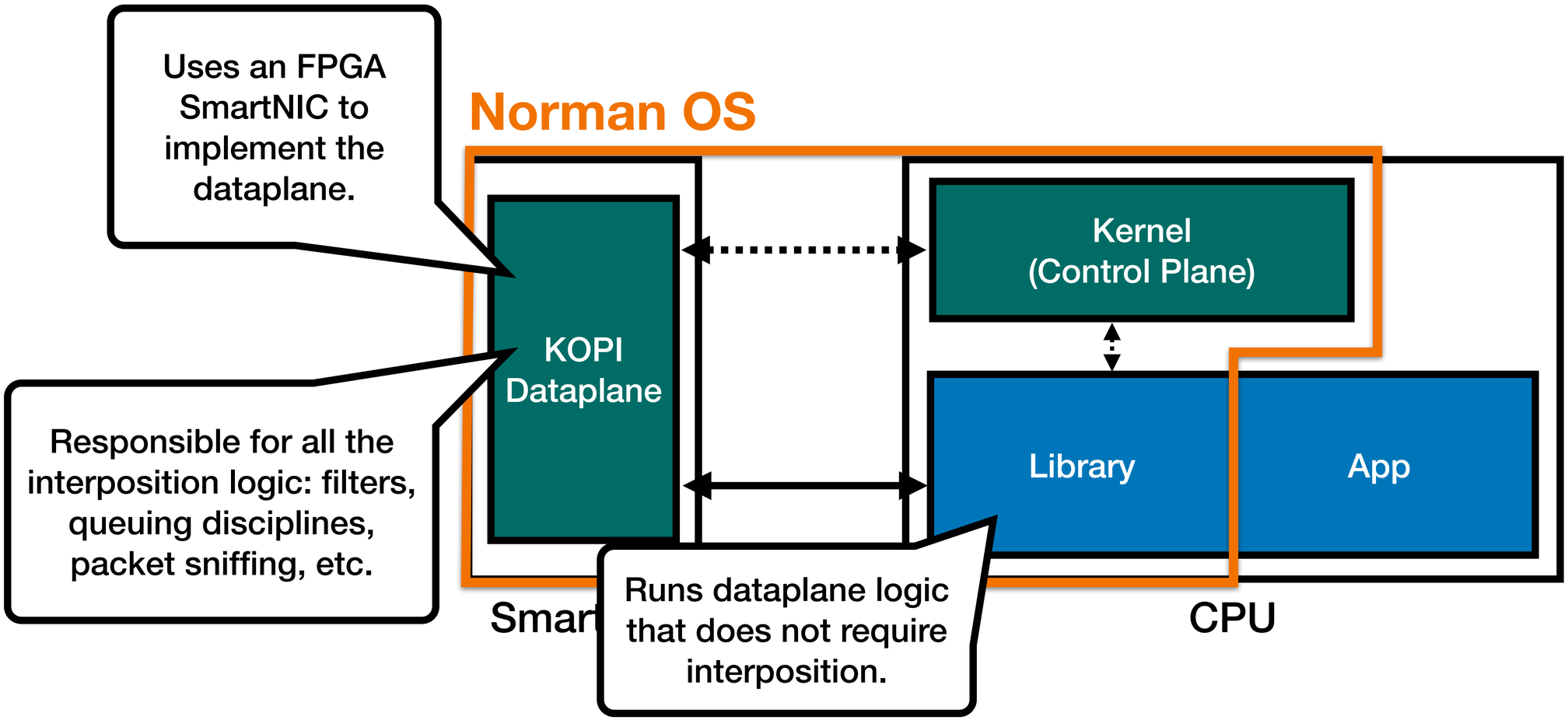
Overview of Norman: An OS that Implements KOPI



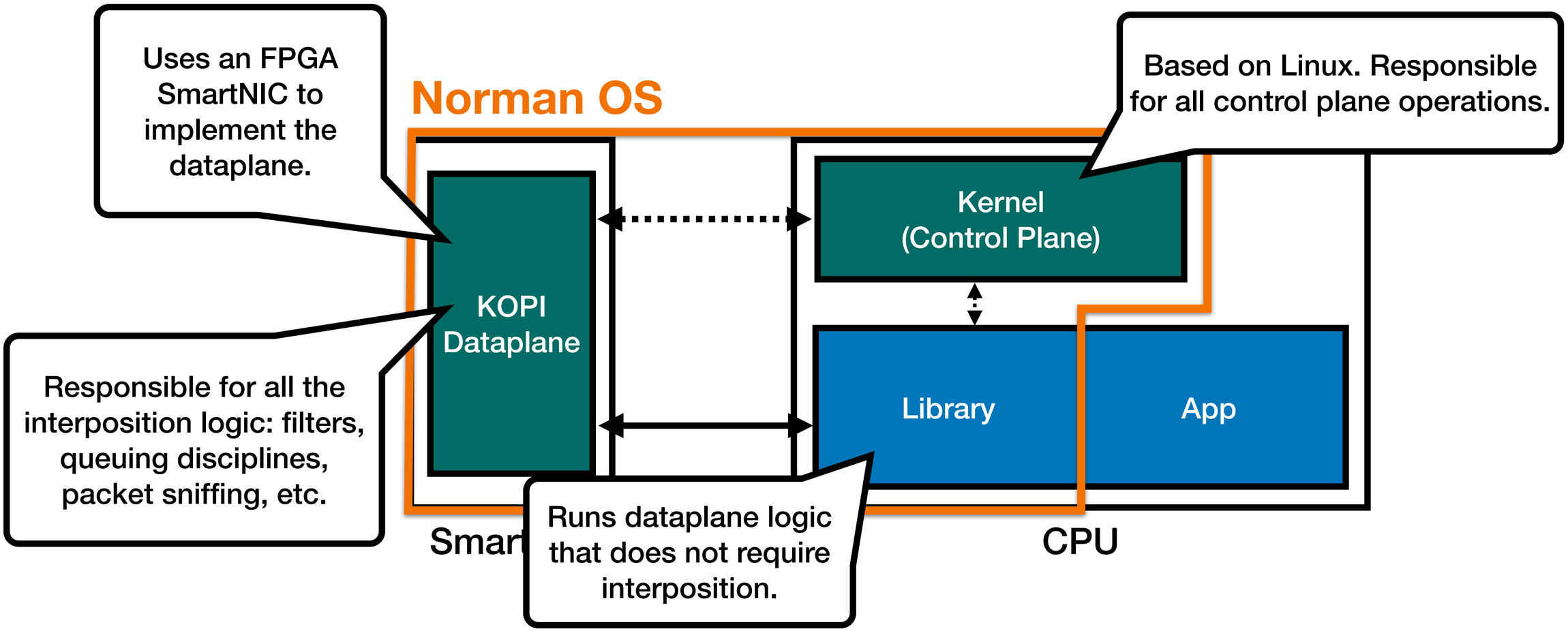
Overview of Norman: An OS that Implements KOPI



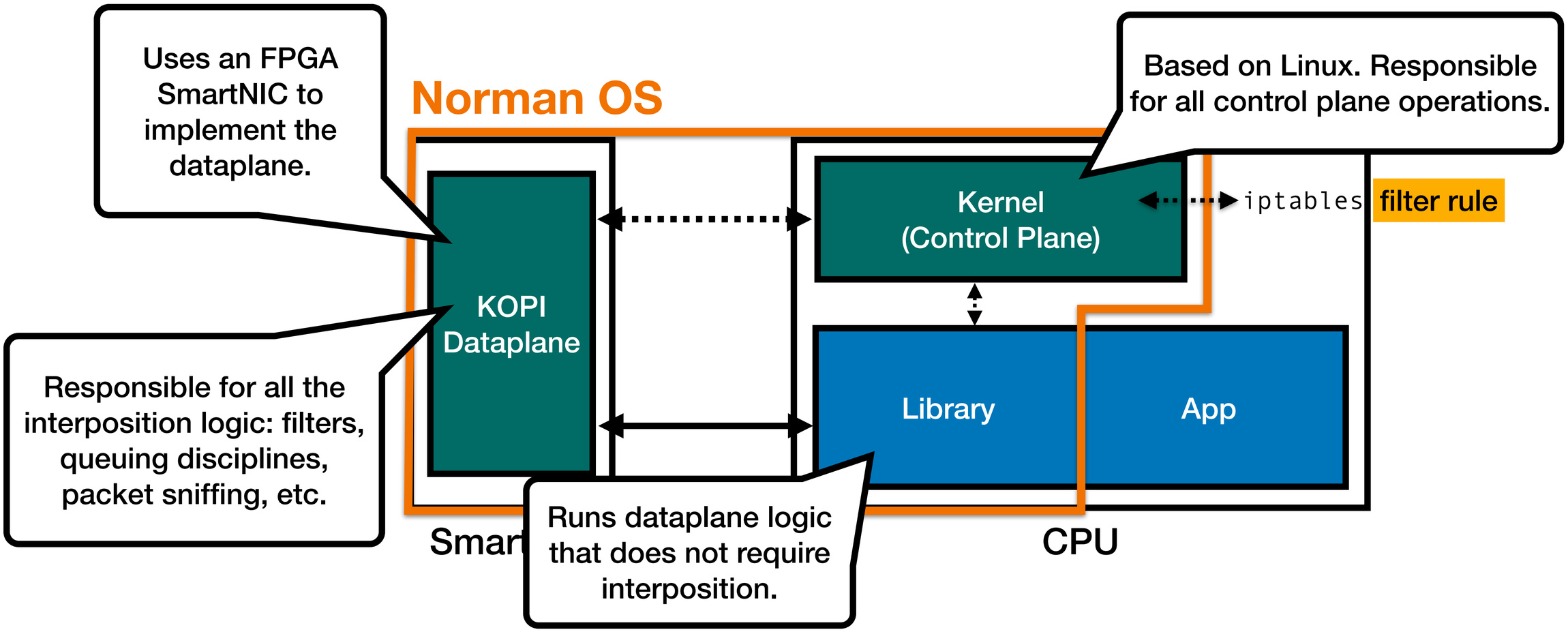
Overview of Norman: An OS that Implements KOPI



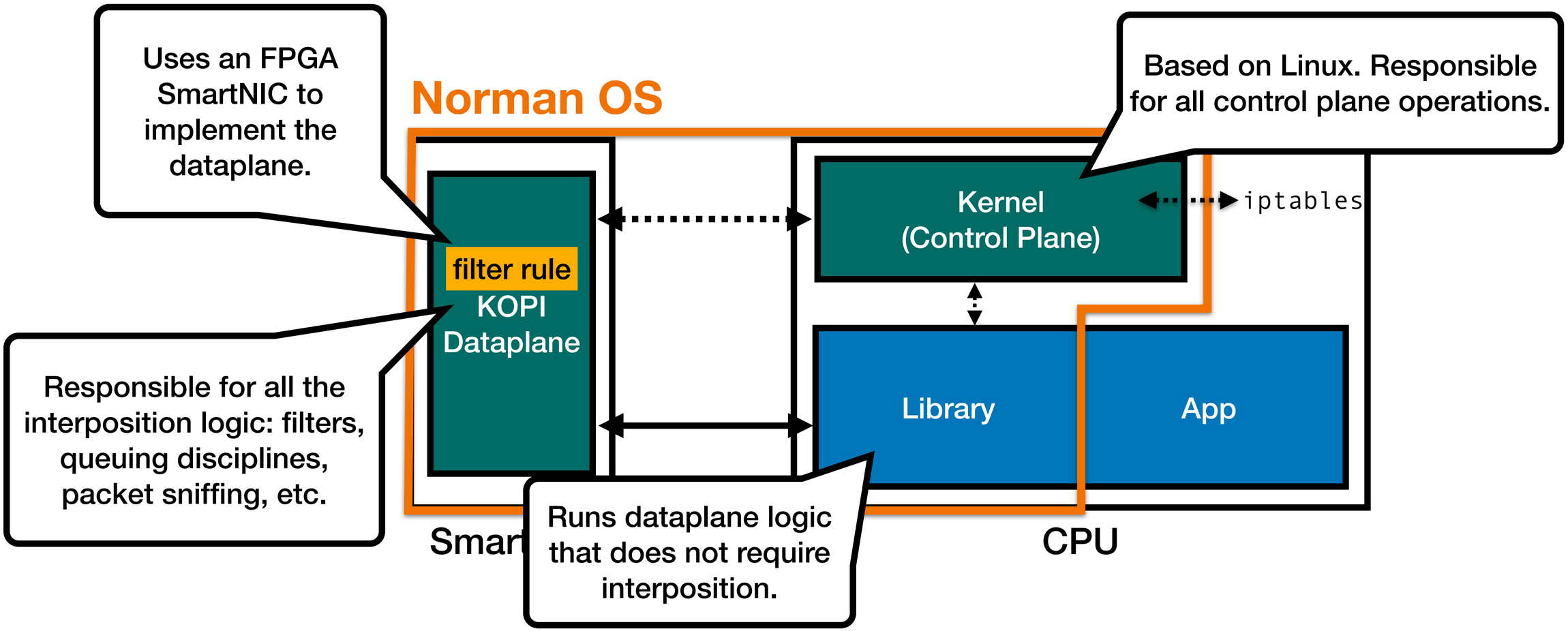
Overview of Norman: An OS that Implements KOPI



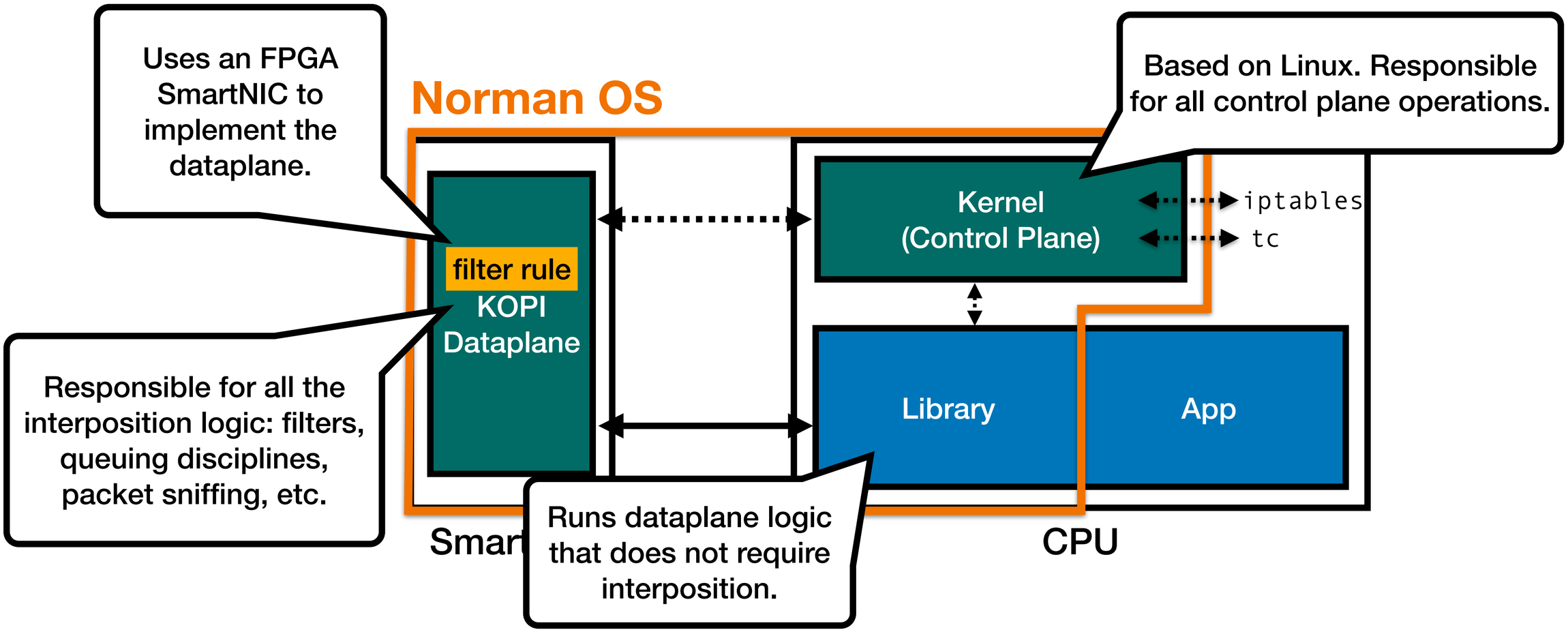
Overview of Norman: An OS that Implements KOPI



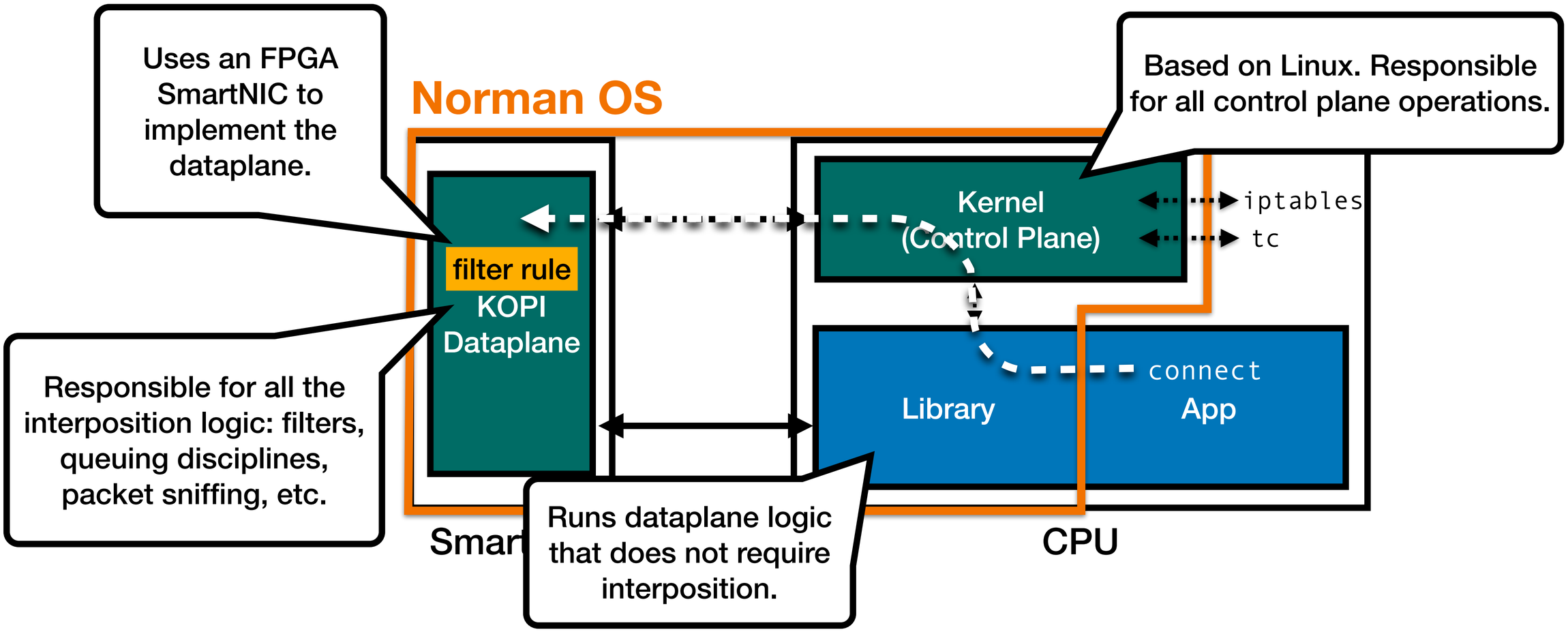
Overview of Norman: An OS that Implements KOPI



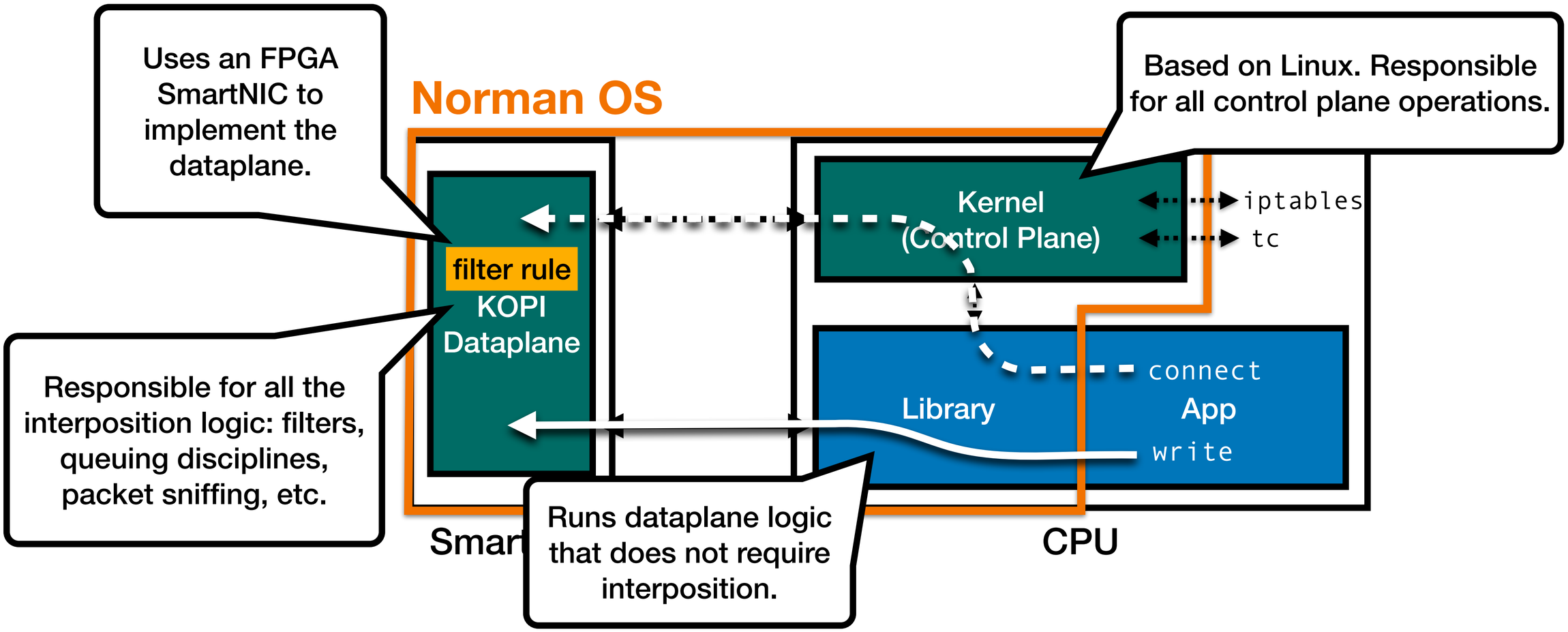
Overview of Norman: An OS that Implements KOPI



Overview of Norman: An OS that Implements KOPI



Overview of Norman: An OS that Implements KOPI



Open Challenges

Open Challenges

① Can Norman design scale to support enough connections?

Open Challenges

① Can Norman design scale to support enough connections?

② How to make an FPGA reconfigurable enough for our purposes?

Conclusion

Conclusion

Kernel interposition is **essential**

Conclusion

Kernel interposition is **essential**

The **KOPI architecture** gives a path to restore interposition without reintroducing overheads