Kernel: to be or not to be?

Yiying Zhang

About Me and WukLab

WukLab: Building next-gen datacenter systems

- Fields
  - Operating systems
  - Distributed systems
  - Datacenter networking
  - Computer architecture

"I see myself as a generalist -- I am attracted to the biggest problem I can find, regardless of area"

About Me and WukLab

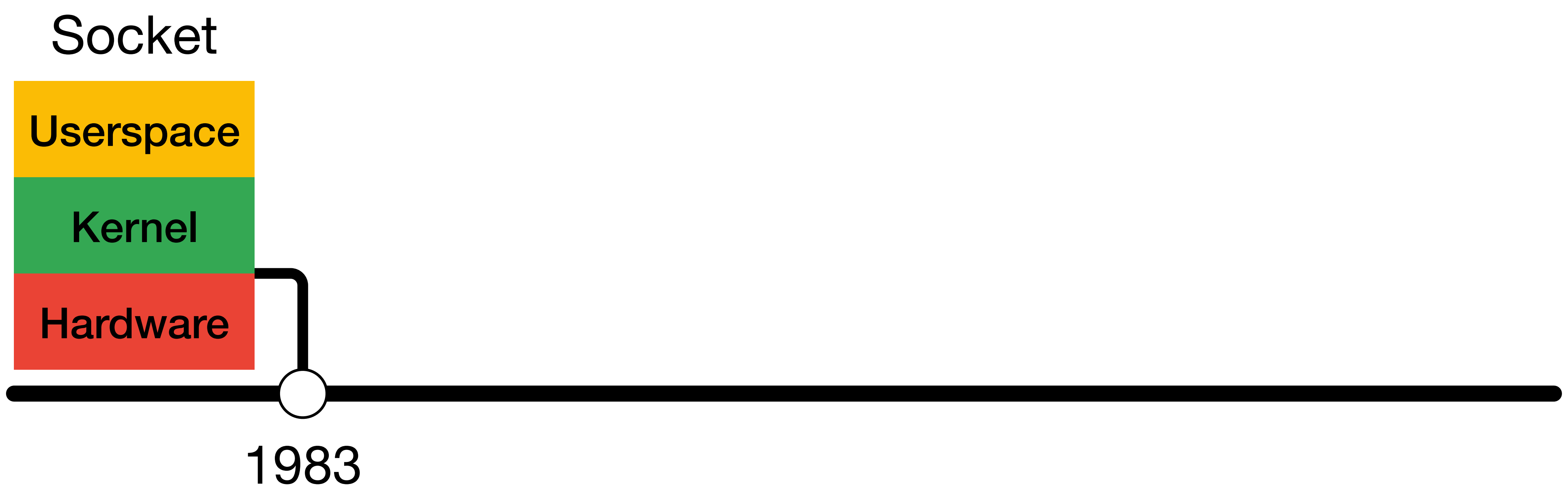
WukLab: Building next-gen datacenter systems

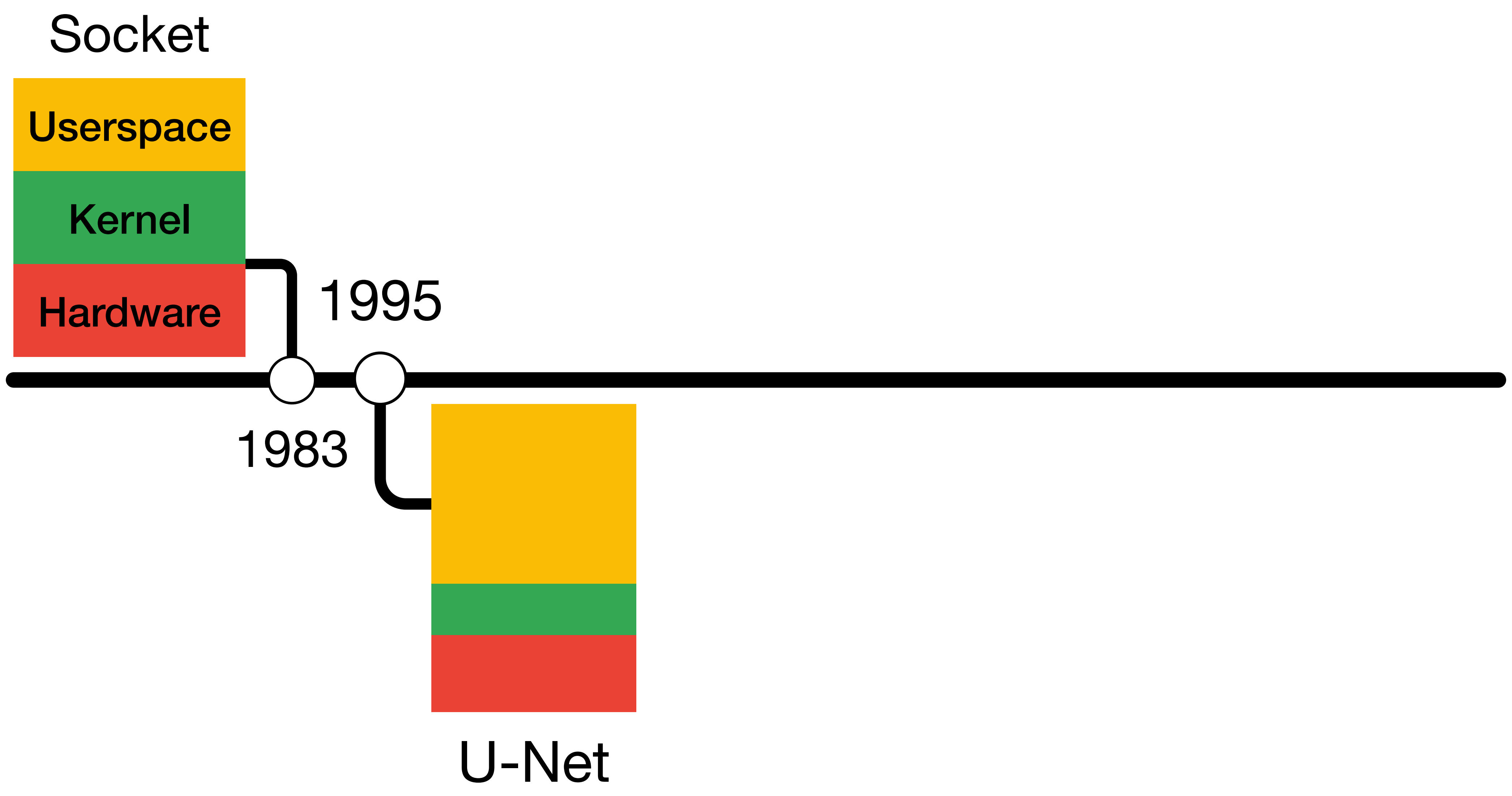
- Fields
  - Operating systems
  - Distributed systems
  - Datacenter networking
  - Computer architecture

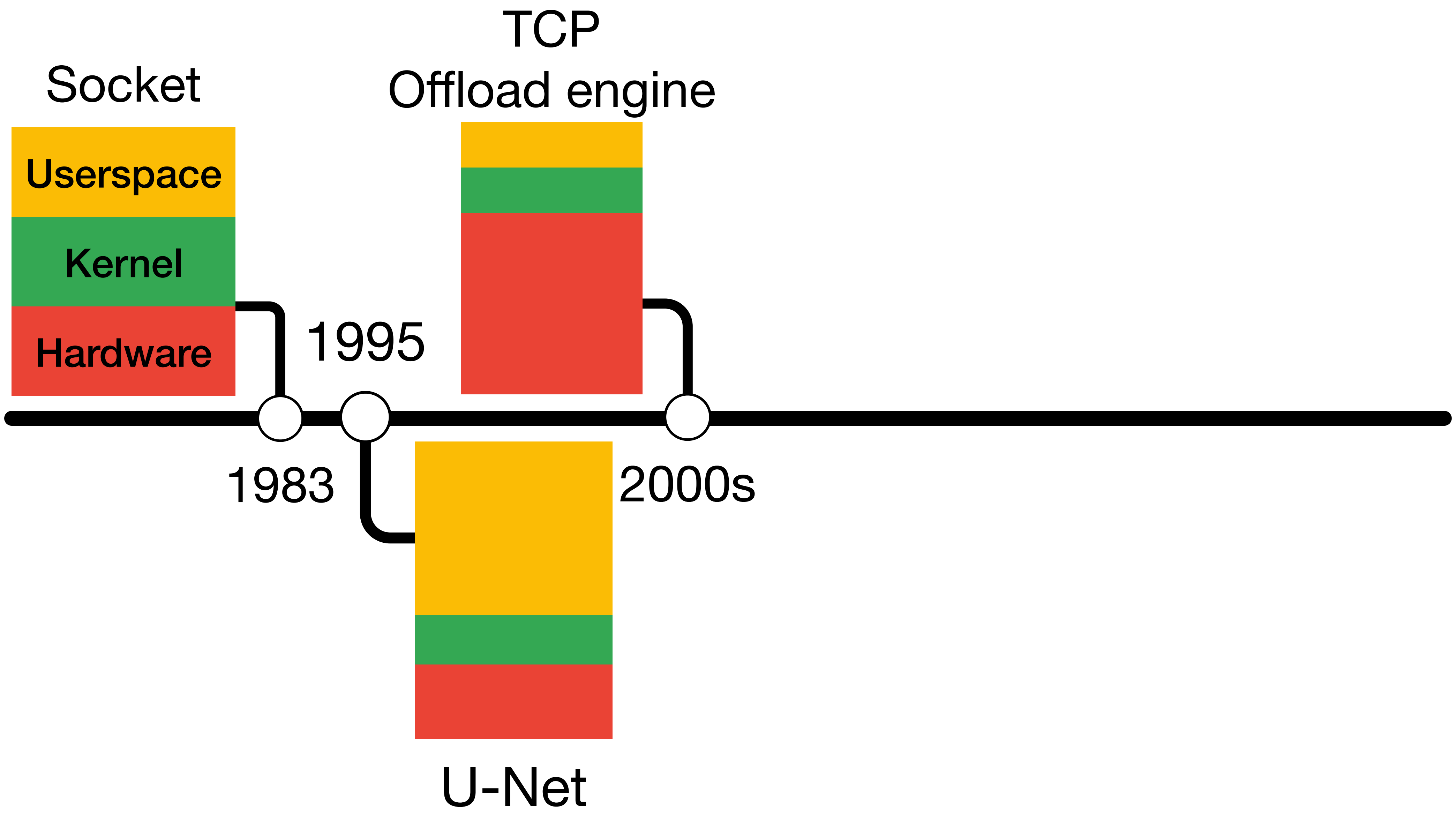
“I see myself as a generalist -- I am attracted to the biggest problem I can find, regardless of area”

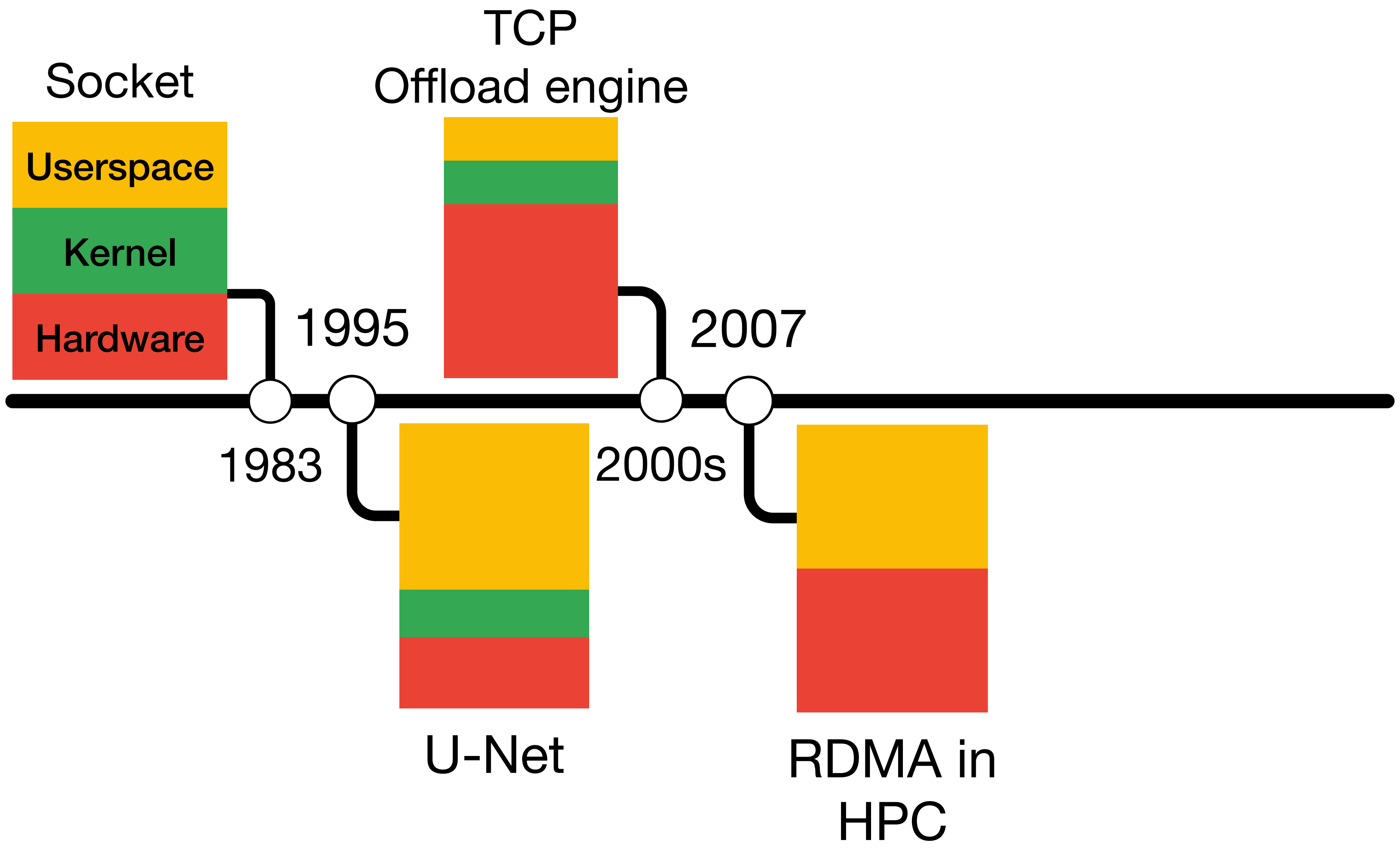
But most of our works are in kernel

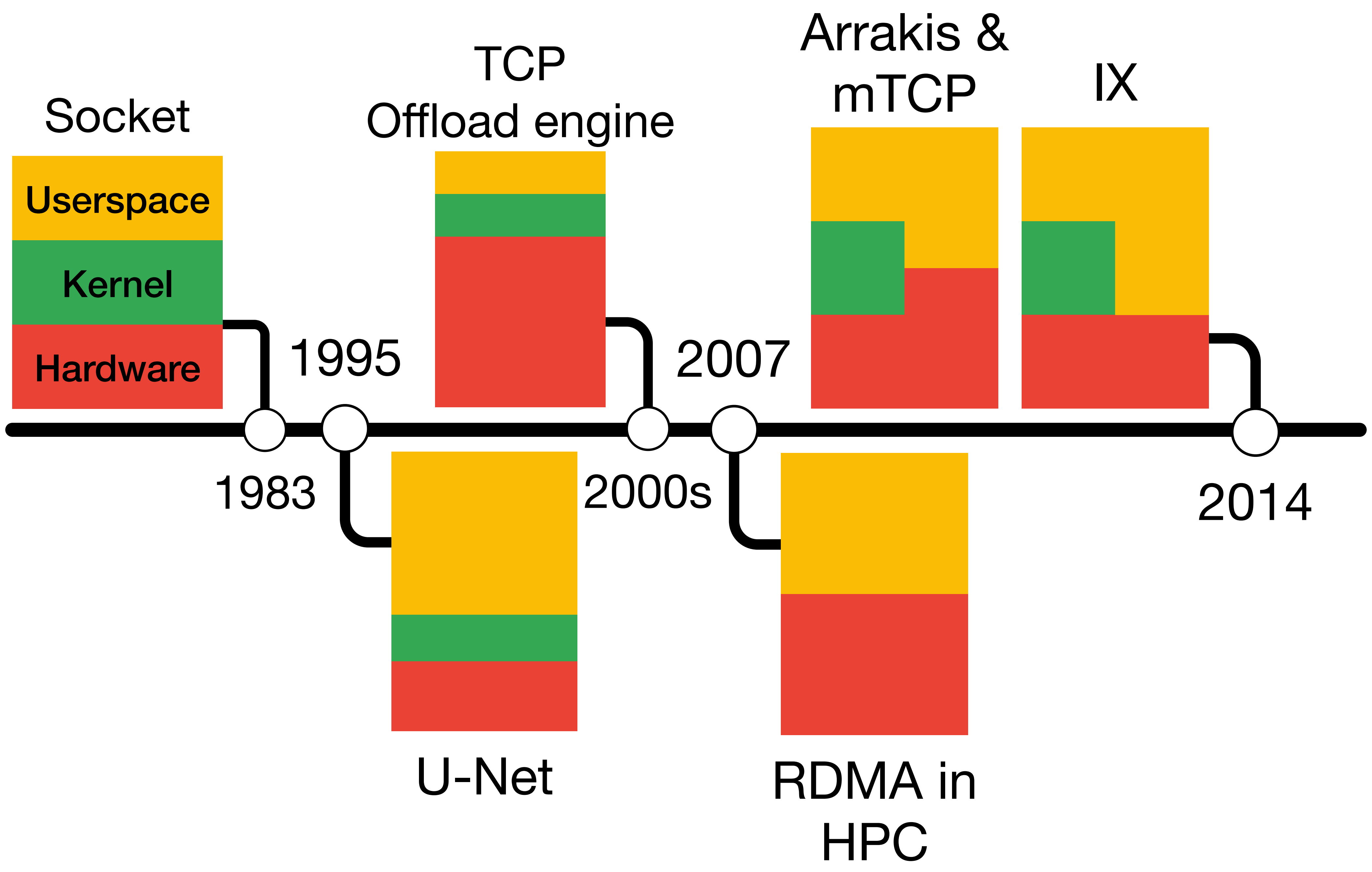
What should go into kernel?

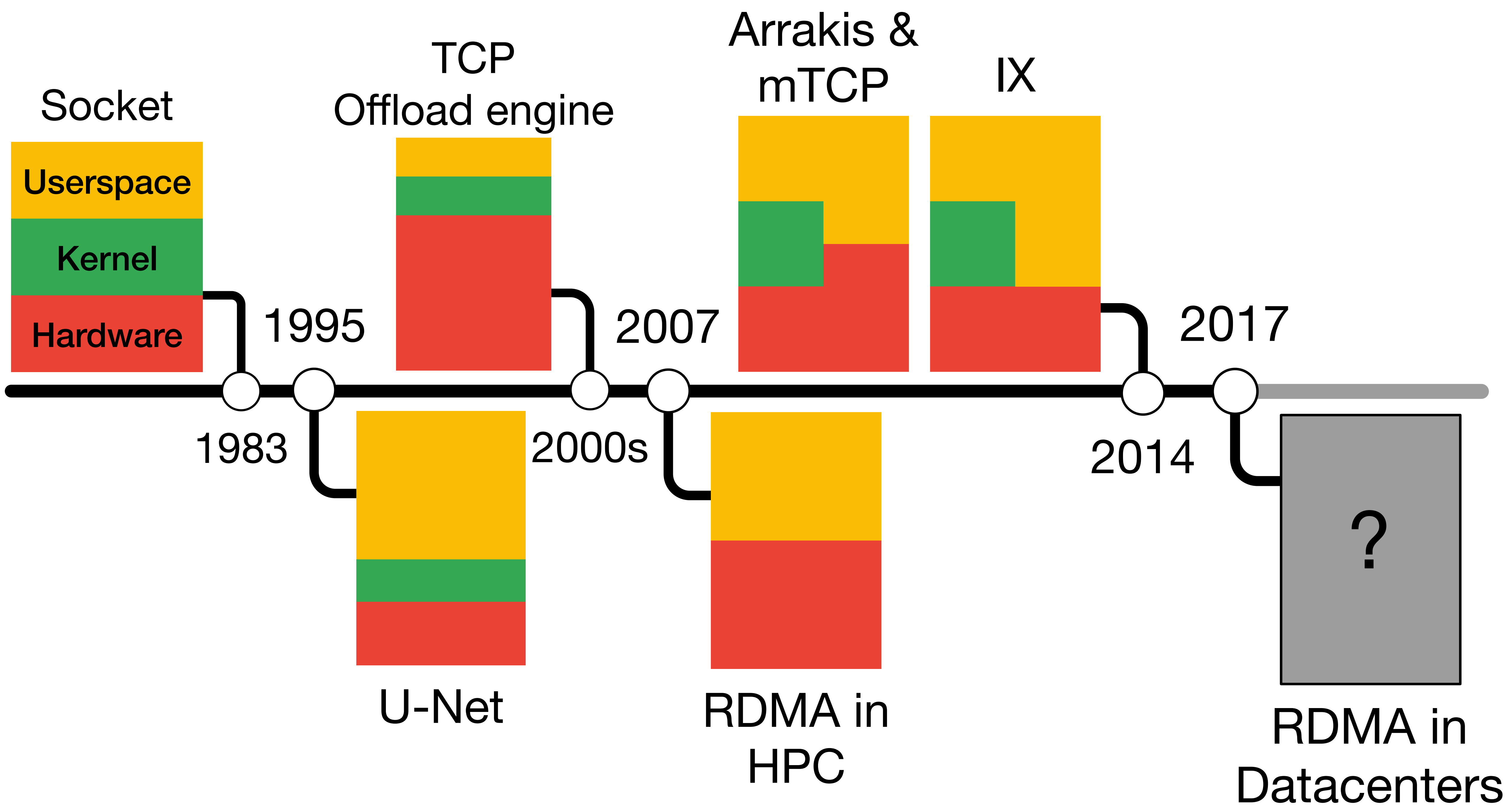

Socket
Userspace
Kernel
Hardware
1983

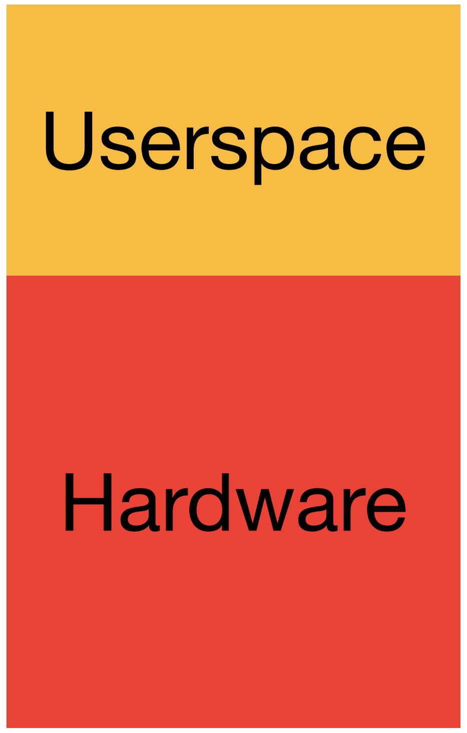

Socket
Userspace
Kernel
Hardware
1995
1983
U-Net


TCP
Offload engine
Socket
Userspace
Kernel
Hardware
1995
1983
2000s
U-Net


Socket
Userspace
Kernel
Hardware
1983
1995
TCP
Offload engine
2000s
2007
U-Net
RDMA in
HPC

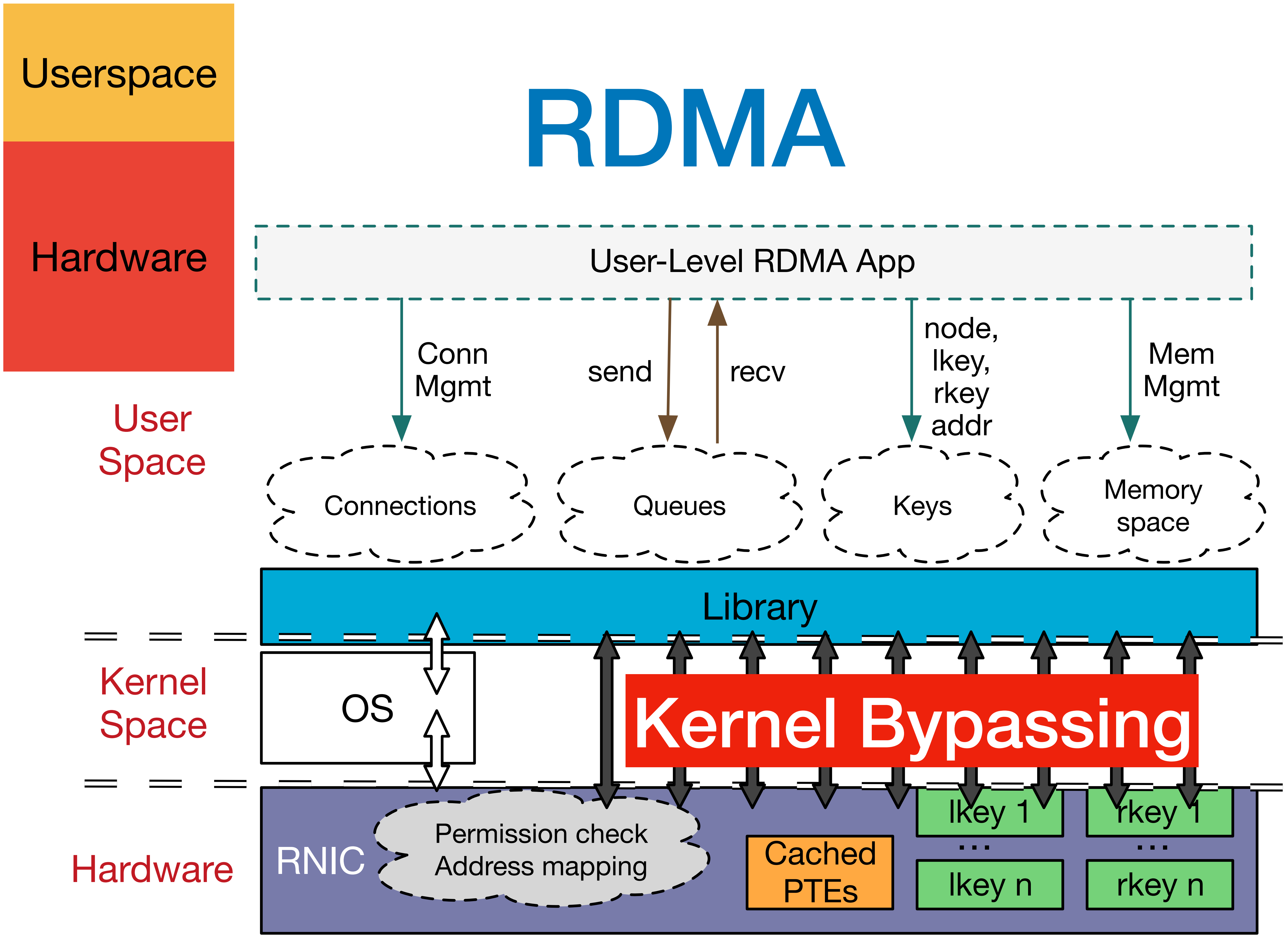

Socket
Userspace
Kernel
Hardware
1983
1995
U-Net
TCP
Offload engine
2000s
2007
RDMA in
HPC
Arrakis &
mTCP
IX
2014


Socket
Userspace
Kernel
Hardware
1983
1995
U-Net
TCP
Offload engine
2000s
2007
RDMA in
HPC
Arrakis &
mTCP
IX
2014
2017
?
RDMA in
Datacenters

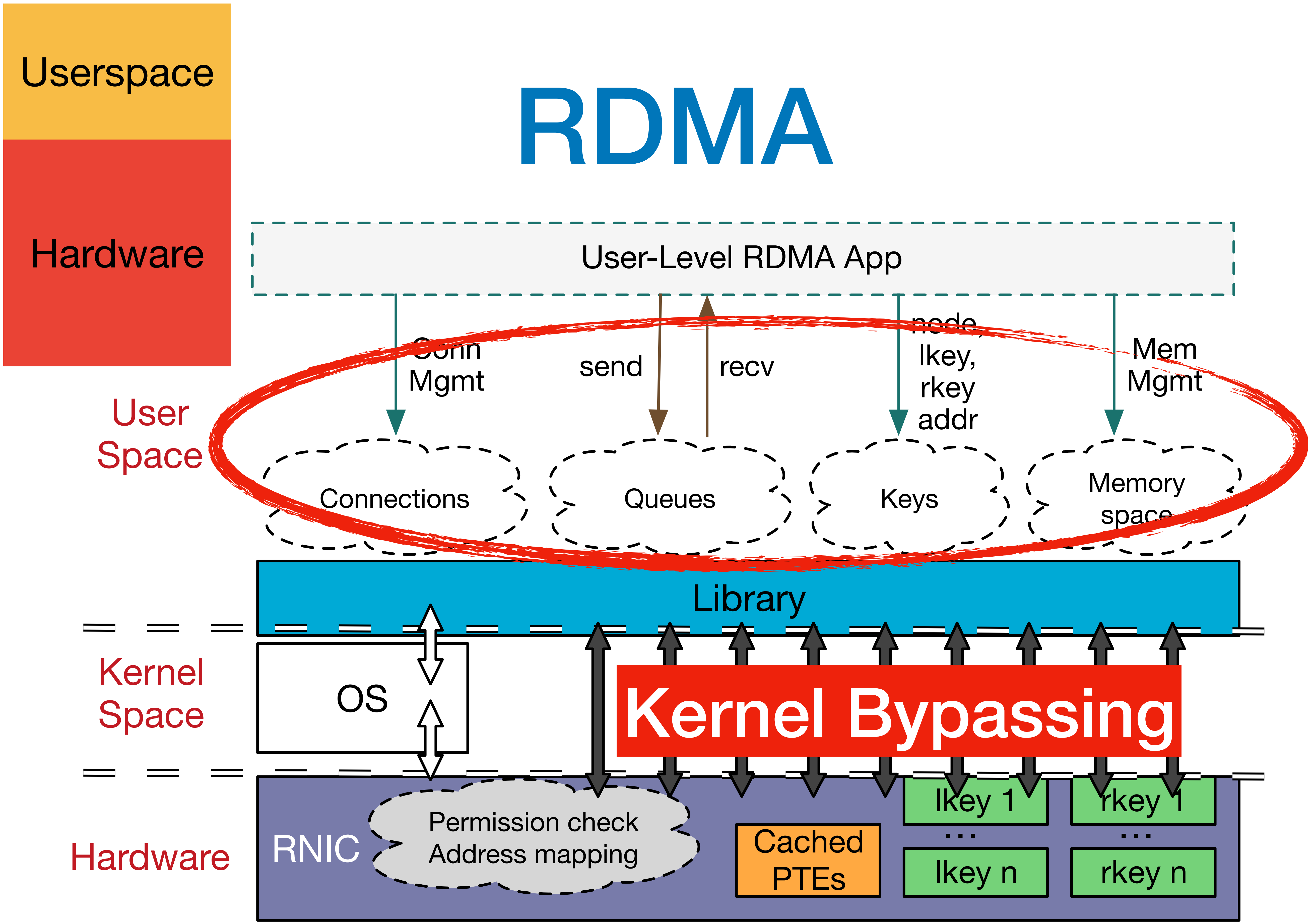


RDMA

RDMA

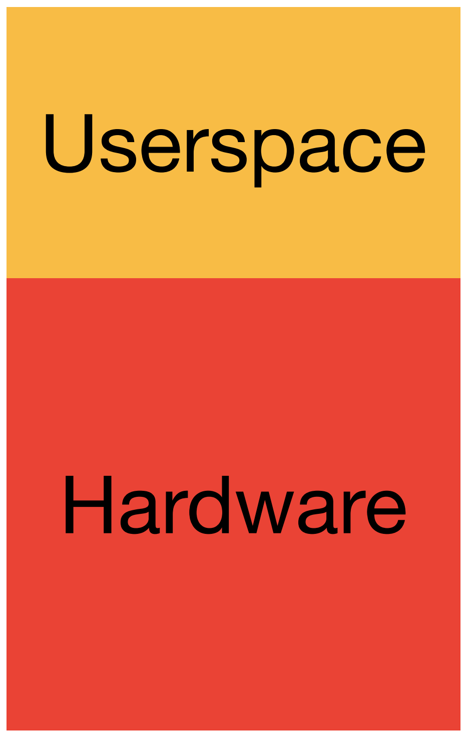

Userspace
Hardware
User-Level RDMA App
Conn Mgmt
send
recv
node, lkey, rkey addr
Mem Mgmt
User Space
Connections
Queues
Keys
Memory space
Library
Kernel Space
OS
Kernel Bypassing
Hardware
RNIC
Permission check
Address mapping
Cached PTEs
lkey 1
…
lkey n
rkey 1
…
rkey n

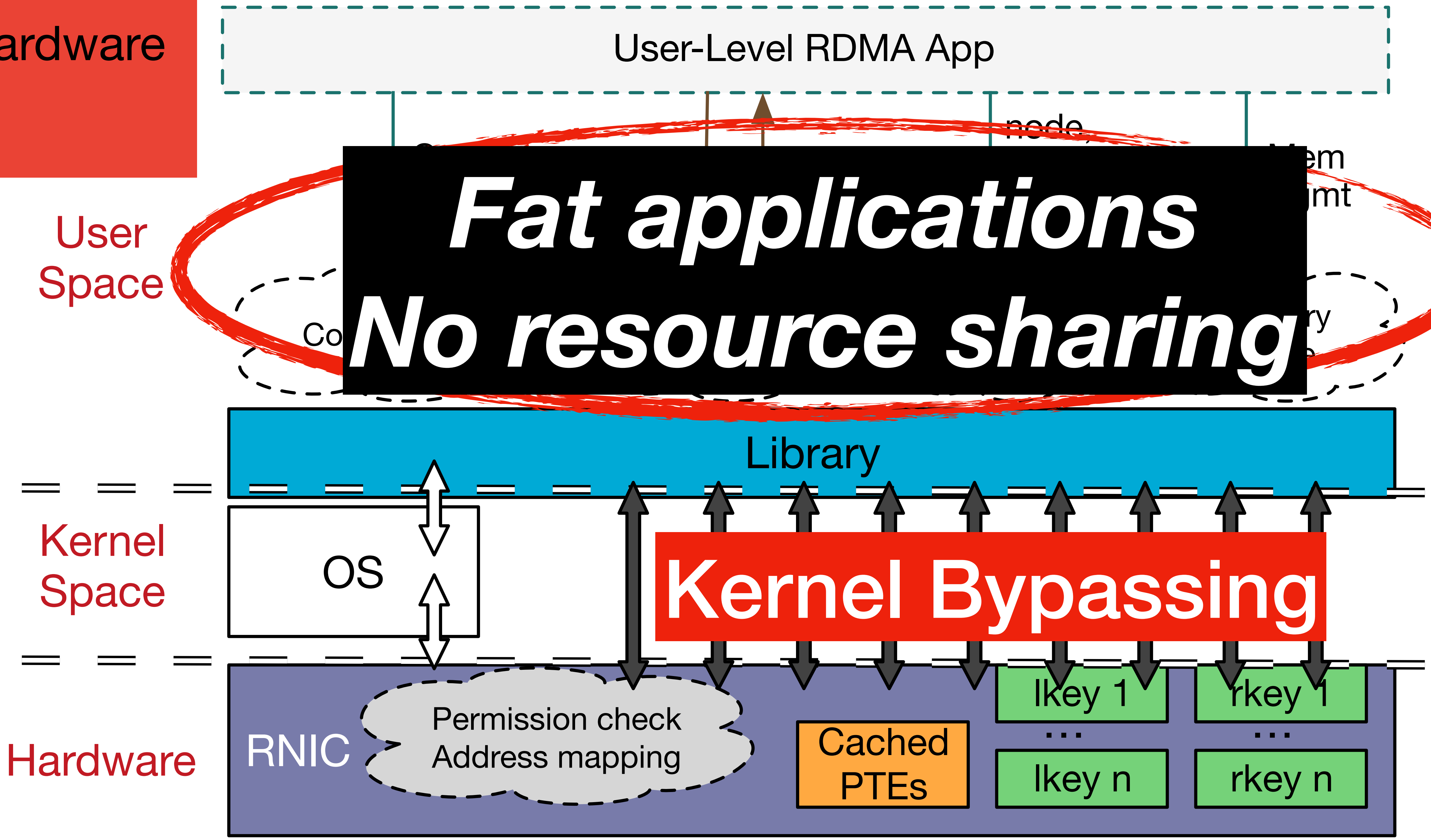
RDMA



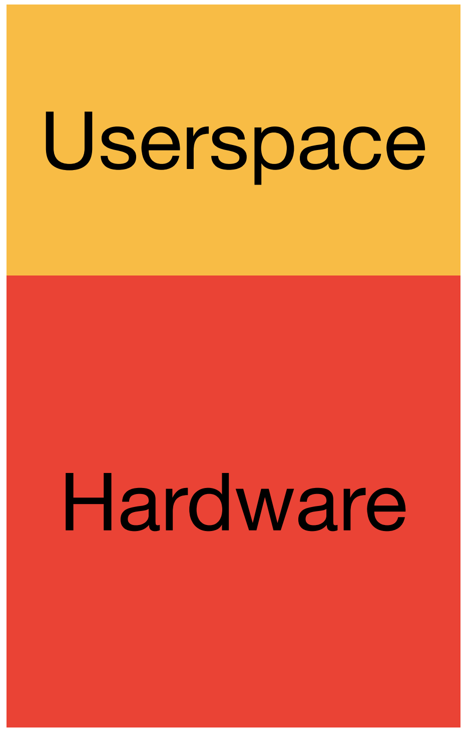
Userspace
Hardware
User-Level RDMA App
Conn Mgmt
send
recv
node, lkey, rkey addr
Mem Mgmt
User Space
Connections
Queues
Keys
Memory space
Library
Kernel Space
OS
Kernel Bypassing
Hardware
RNIC
Permission check
Address mapping
Cached PTEs
lkey 1
...
lkey n
rkey 1
...
rkey n

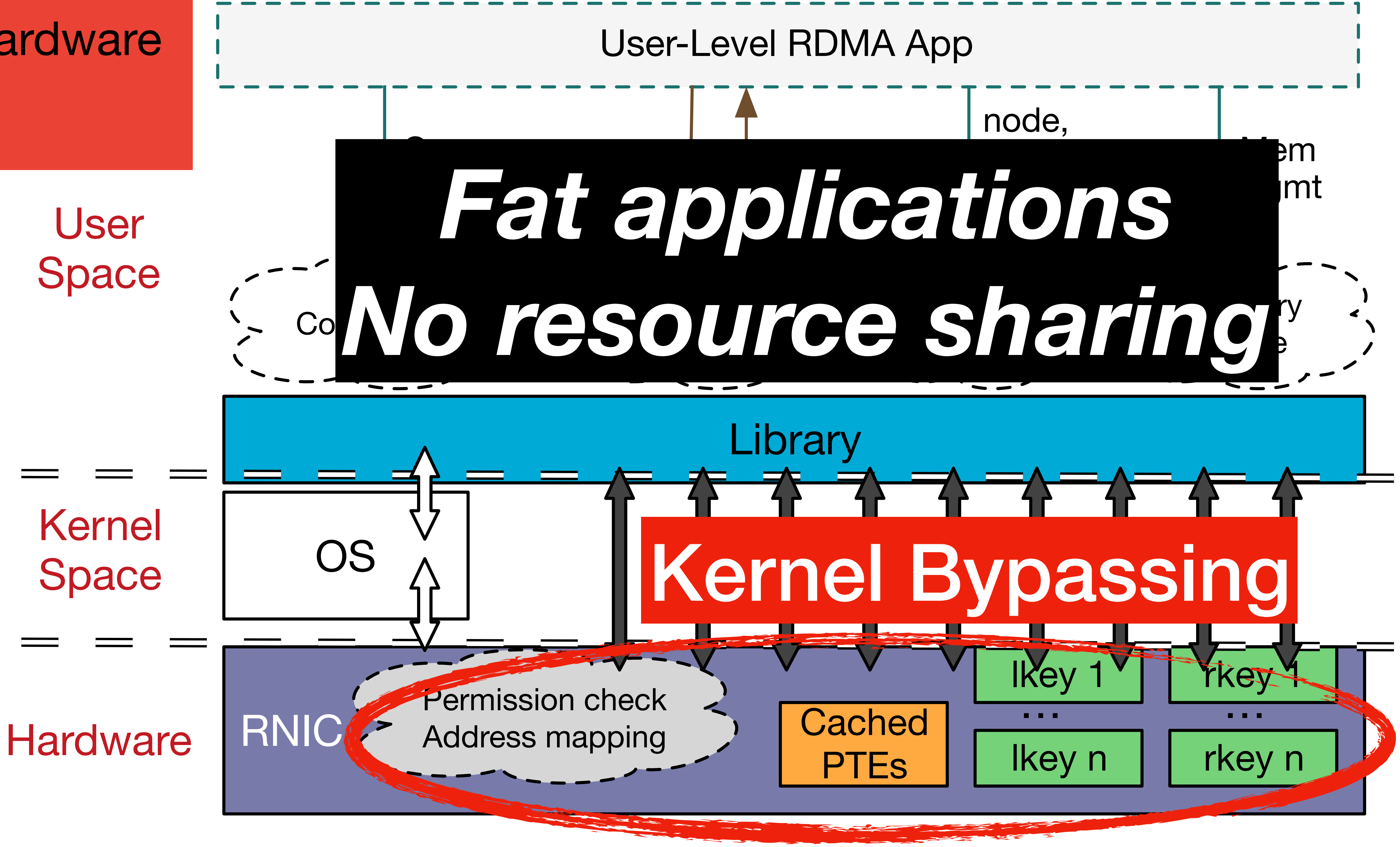
RDMA


Userspace
Hardware

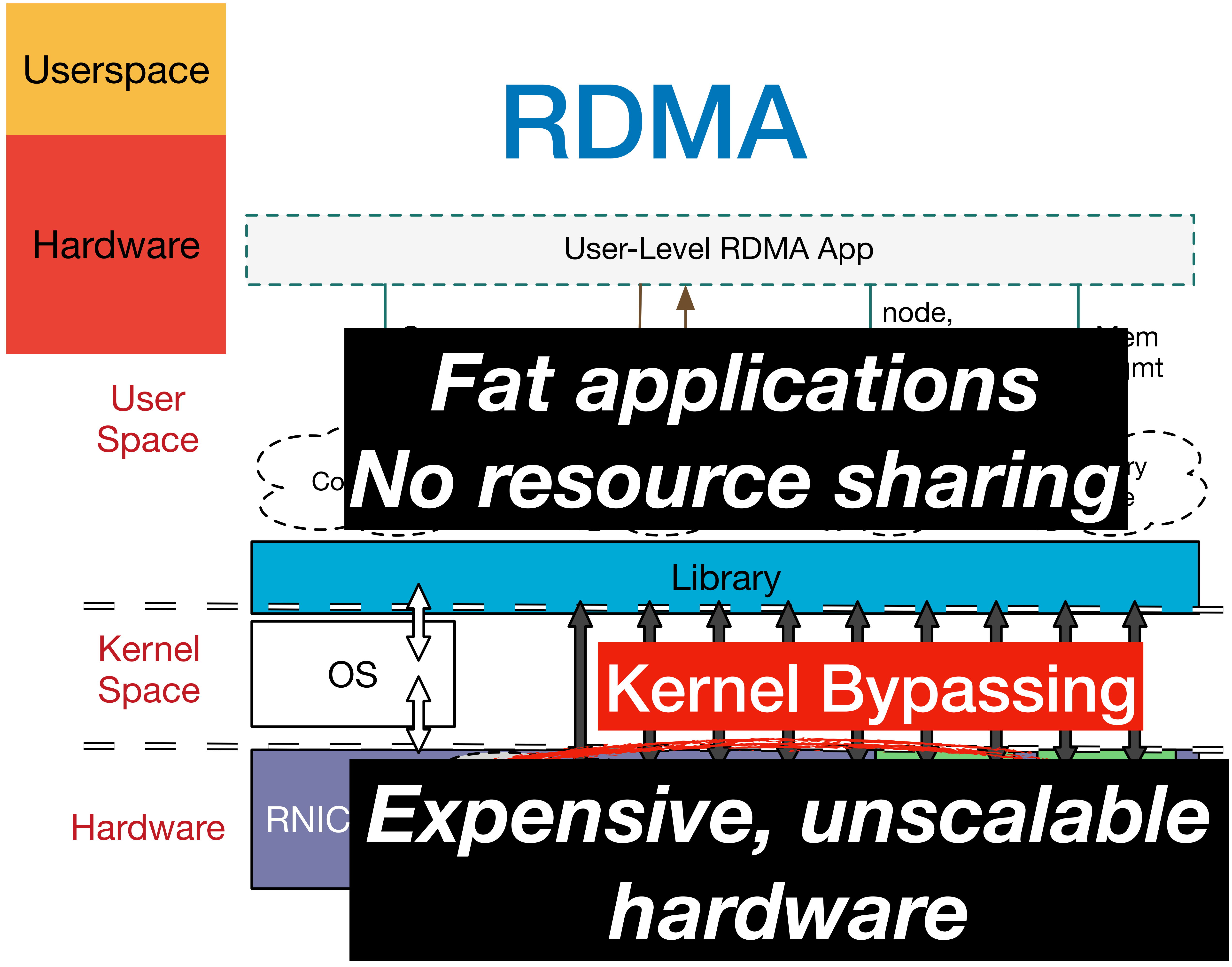

User-Level RDMA App
node
Mem
mgmt
User
Space
Co
ry
Fat applications
No resource sharing
Library
Kernel
Space
OS
Kernel Bypassing
Hardware
RNIC
Permission check
Address mapping
Cached
PTEs
lkey 1
…
lkey n
rkey 1
…
rkey n

RDMA


Userspace
Hardware


User-Level RDMA App
node,
Mem
mgmt
Fat applications
No resource sharing
User Space
Co
ry
e
Library
Kernel Space
OS
Kernel Bypassing
Hardware
RNIC
Permission check
Address mapping
Cached PTEs
lkey 1
lkey n
rkey 1
rkey n

RDMA



Fat applications

No resource sharing

Kernel Bypassing

Expensive, unscalable hardware

Are we removing too much from kernel?

Without Kernel

High-level abstraction

Resource sharing

Protection

Flexible QoS management

Without Kernel

Resource sharing

Protection

Flexible QoS management

Without Kernel

Resource sharing

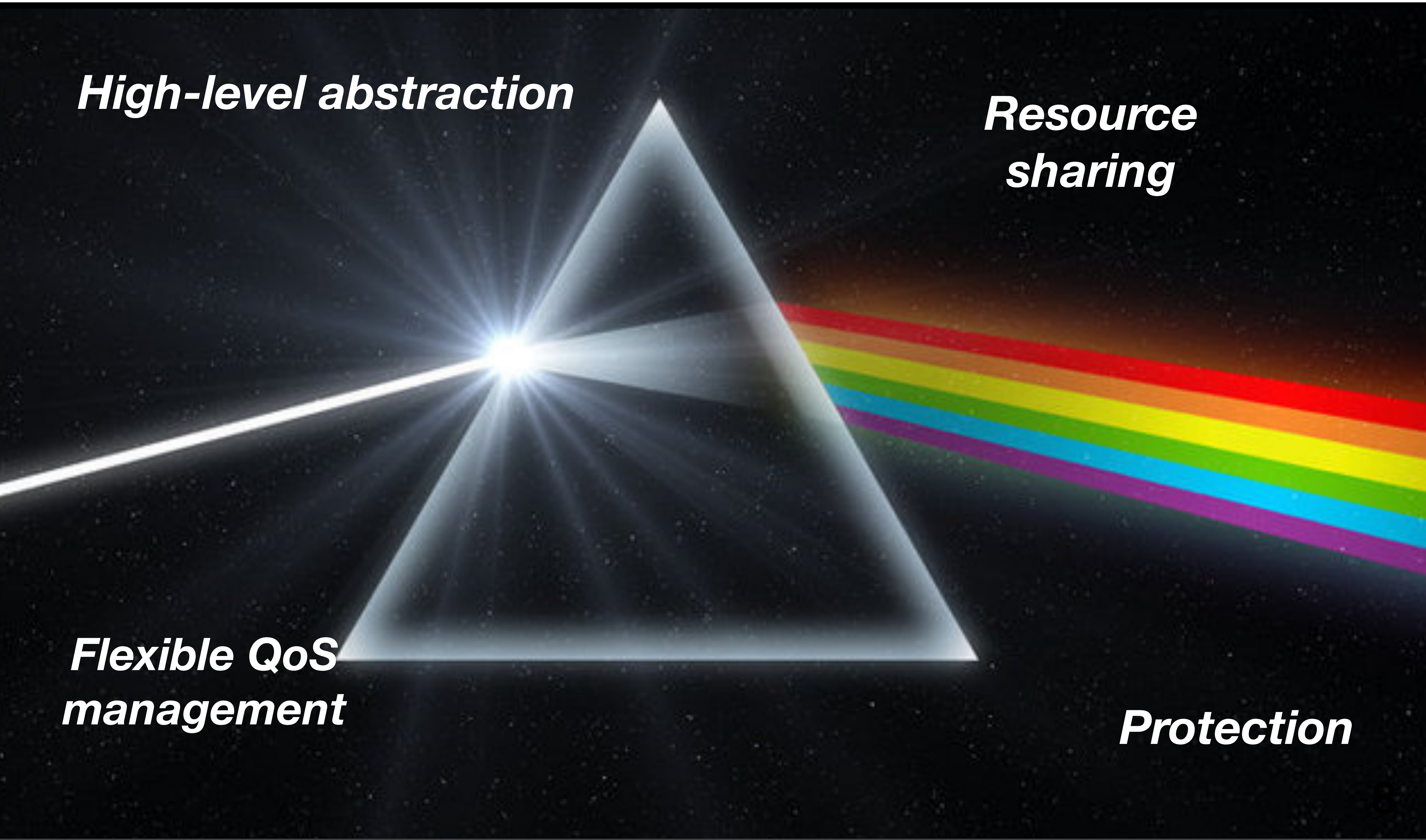
Flexible QoS management

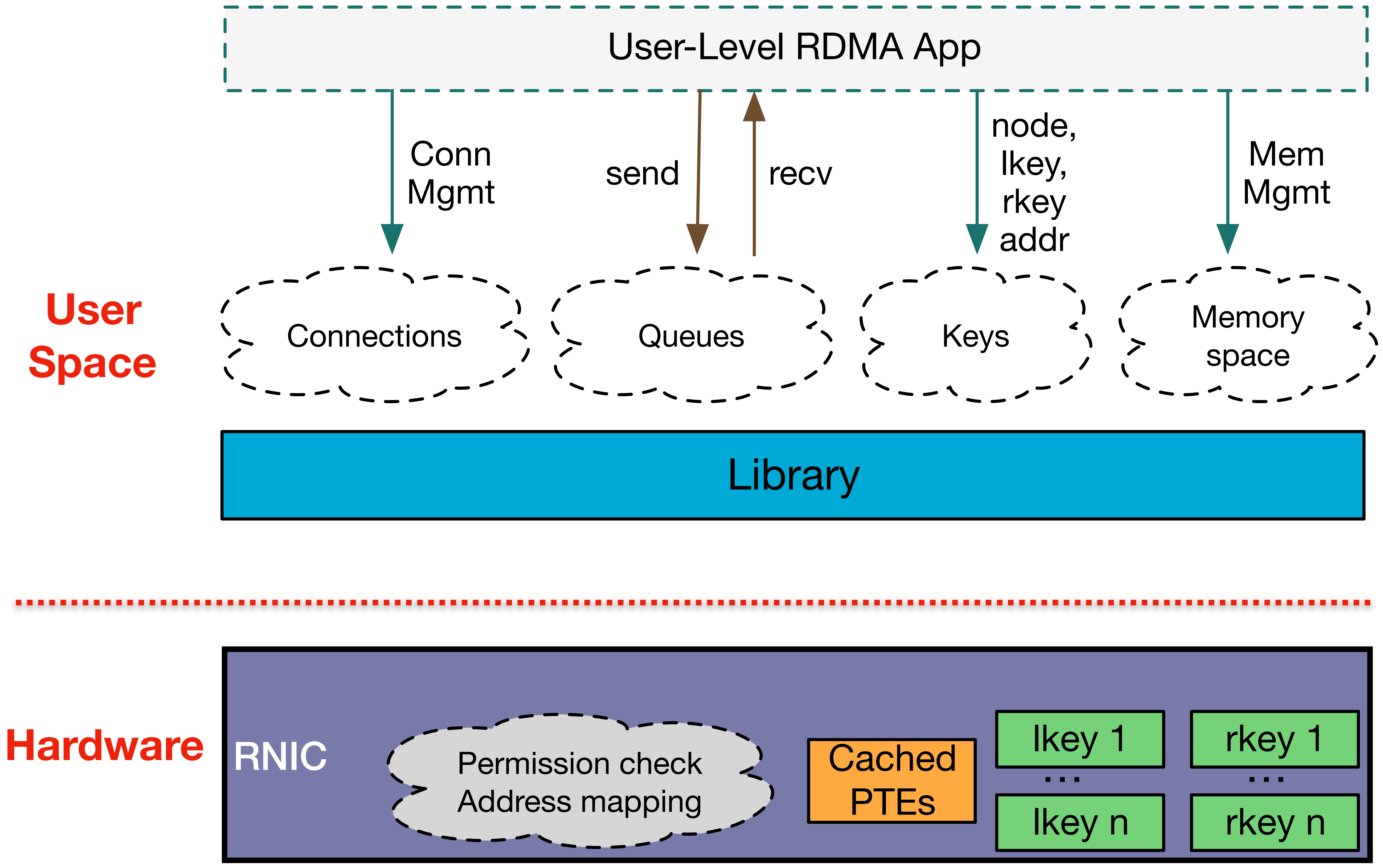
Without Kernel

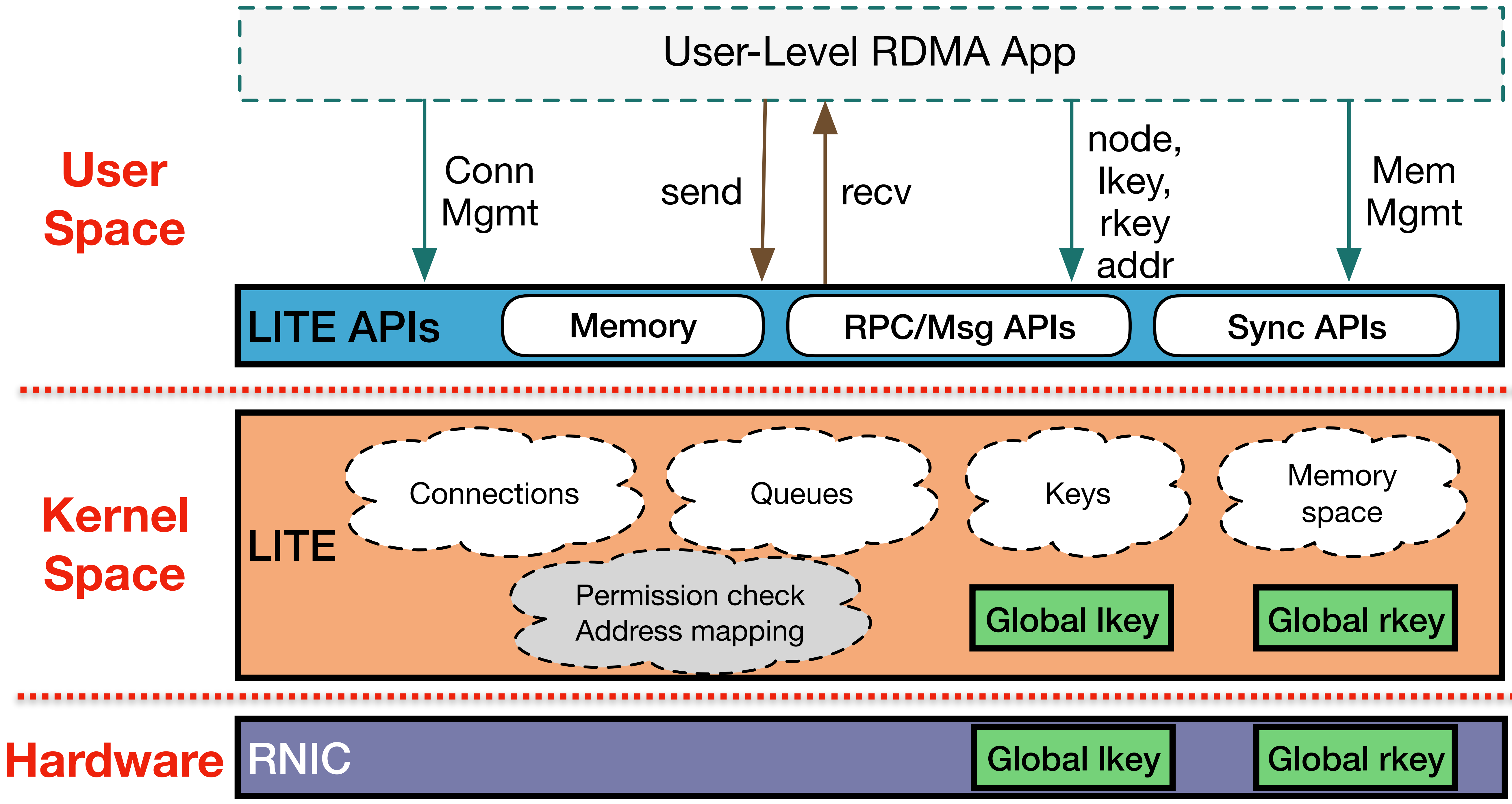
Flexible QoS management

Without Kernel

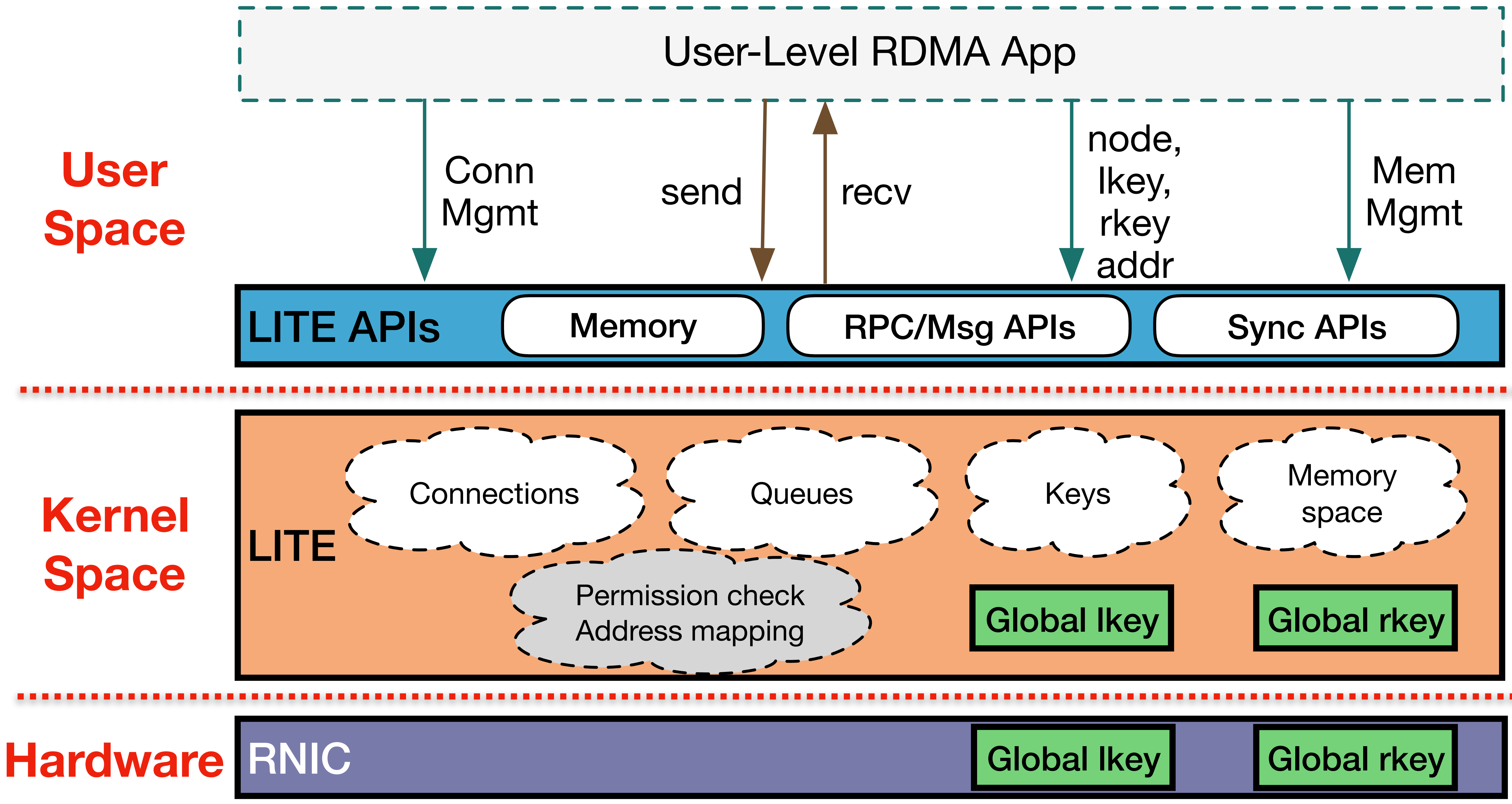
LITE - Kernel-Level Indirection for RDMA [SOSP'17]



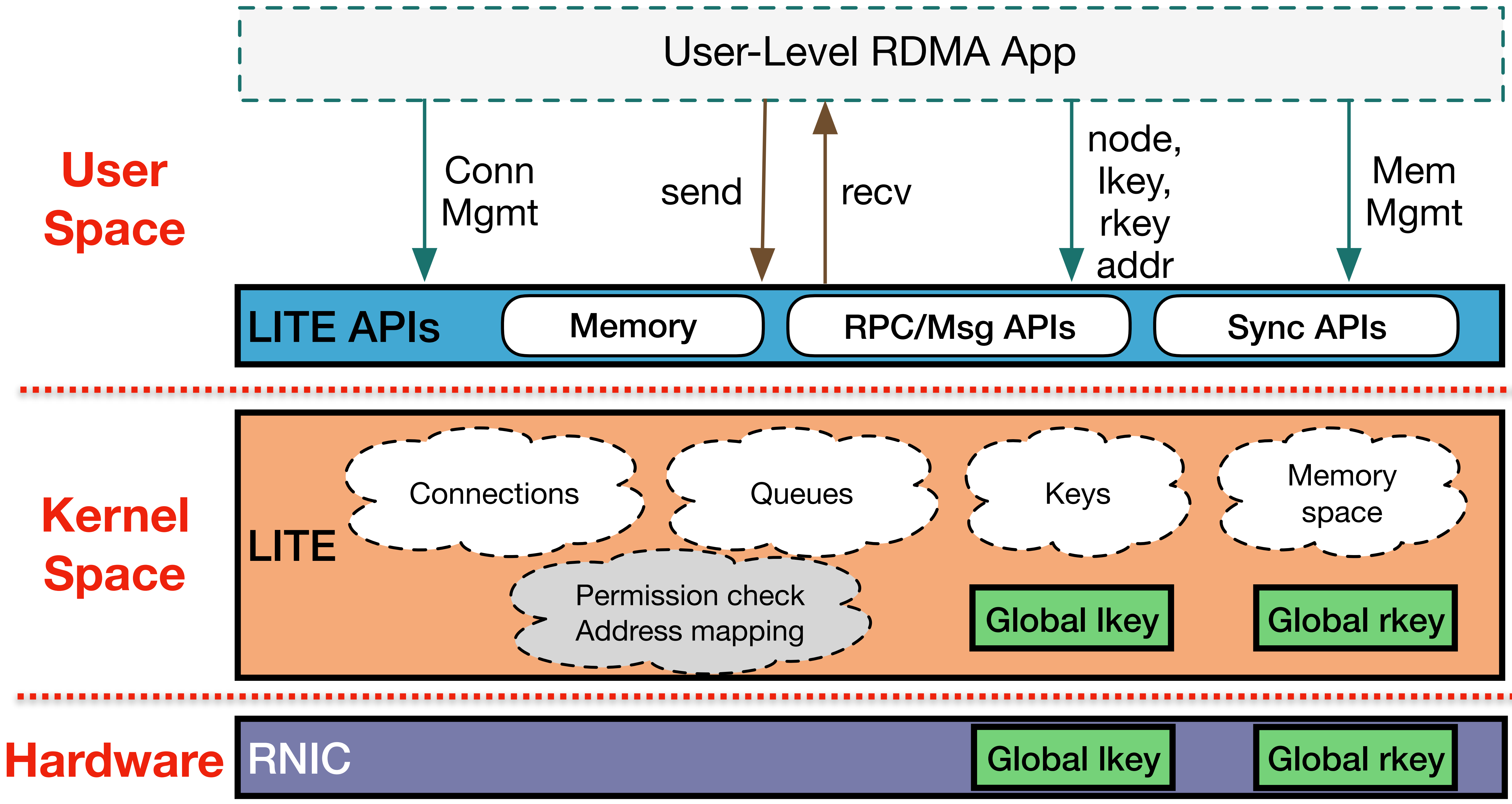

User-Level RDMA App
Conn Mgmt
send
recv
node, lkey, rkey addr
Mem Mgmt
User Space
Connections
Queues
Keys
Memory space
Library
Hardware
RNIC
Permission check
Address mapping
Cached PTEs
lkey 1
…
lkey n
rkey 1
…
rkey n


User-Level RDMA App
User Space
Conn Mgmt
send
recv
node, lkey, rkey addr
Mem Mgmt
LITE APIs
Memory
RPC/Msg APIs
Sync APIs
Kernel Space
LITE
Connections
Queues
Keys
Memory space
Permission check Address mapping
Global lkey
Global rkey
Hardware
RNIC
Global lkey
Global rkey

Simpler applications

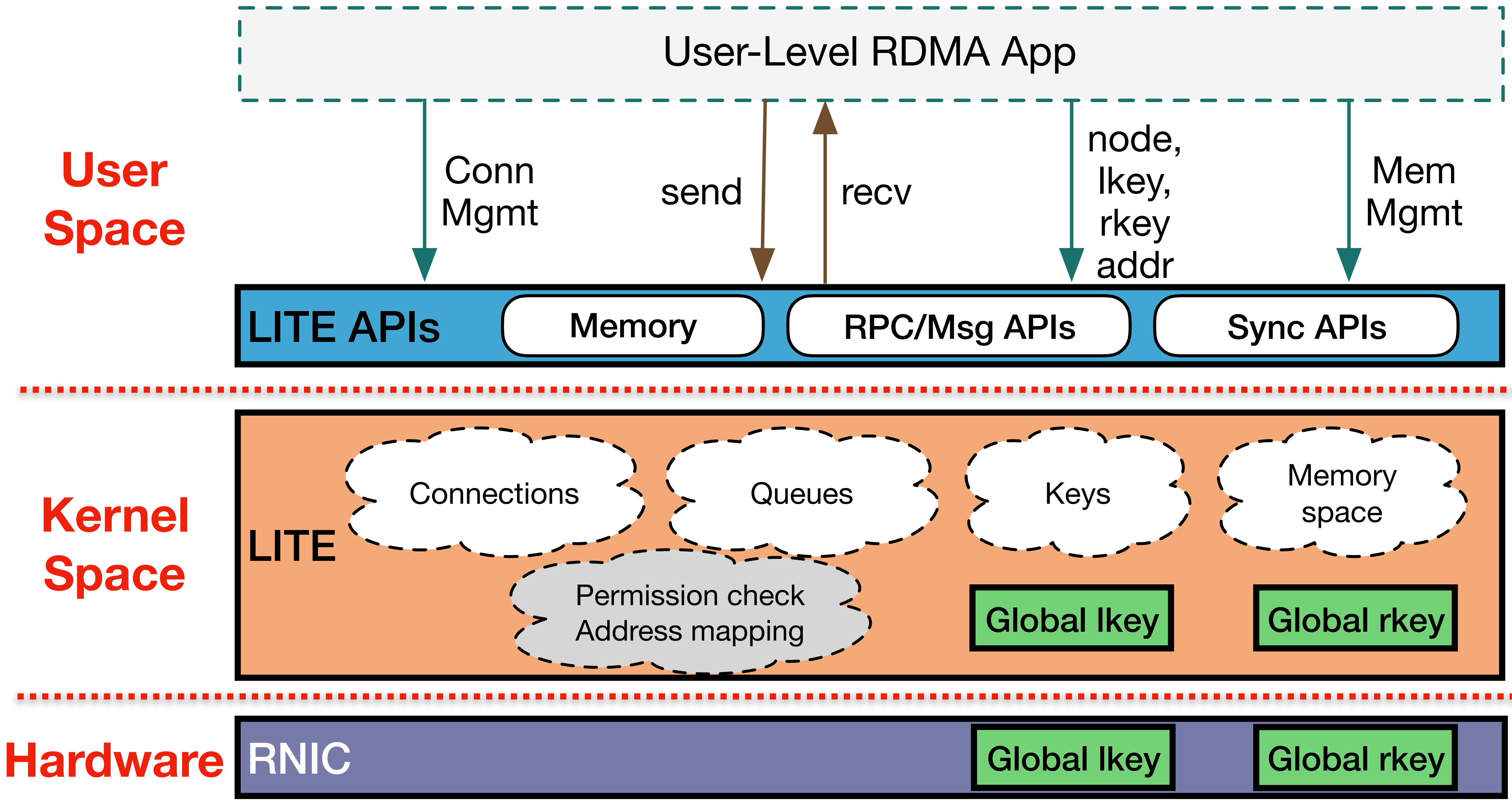


Simpler applications



Cheaper hardware

Simpler applications



Cheaper hardware

Great Performance and scalability

My students, **before** going to their first conference:

Kernel programming is fun; we like doing great engineer work

Projects at WukLab

- ***LITE*** [sosp'17]: **15K** LOC, 80% in kernel
- ***Hotpot*** [SoCC'17]: **19K** LOC, all in kernel
- ***Lego*** OS [ongoing]: **170K** LOC already, all in kernel

My students, **after** going to their first conference:

Why are we writing so many kernel code when other students can get a paper with hundreds lines of user-level code?

Thank you
Questions?

Get LITE at: *[github.com/Wuklab/LITE]*

wuklab.io