NATIONAL ICT AUSTRALIA

L4/Darwin: Evolving UNIX

Charles Gray
Research Engineer, National ICT Australia
charles.gray@nicta.com.au

Australian Government
Department of Communications, Information Technology and the Arts
Australian Research Council

NICTA Members
THE AUSTRALIAN NATIONAL UNIVERSITY
UNSW
NSW First for Business
Department of State and Regional Development
BusinessACT ACT GOVERNMENT
Victoria The Place To Be
THE UNIVERSITY OF MELBOURNE
The University of Sydney
Queensland Government
Griffith UNIVERSITY
QUT Queensland University of Technology
THE UNIVERSITY OF QUEENSLAND AUSTRALIA
NICTA Partners

Outline

1. Project Overview
2. BSD on the Mach microkernel
3. Porting Darwin to the L4 microkernel
4. Project Status

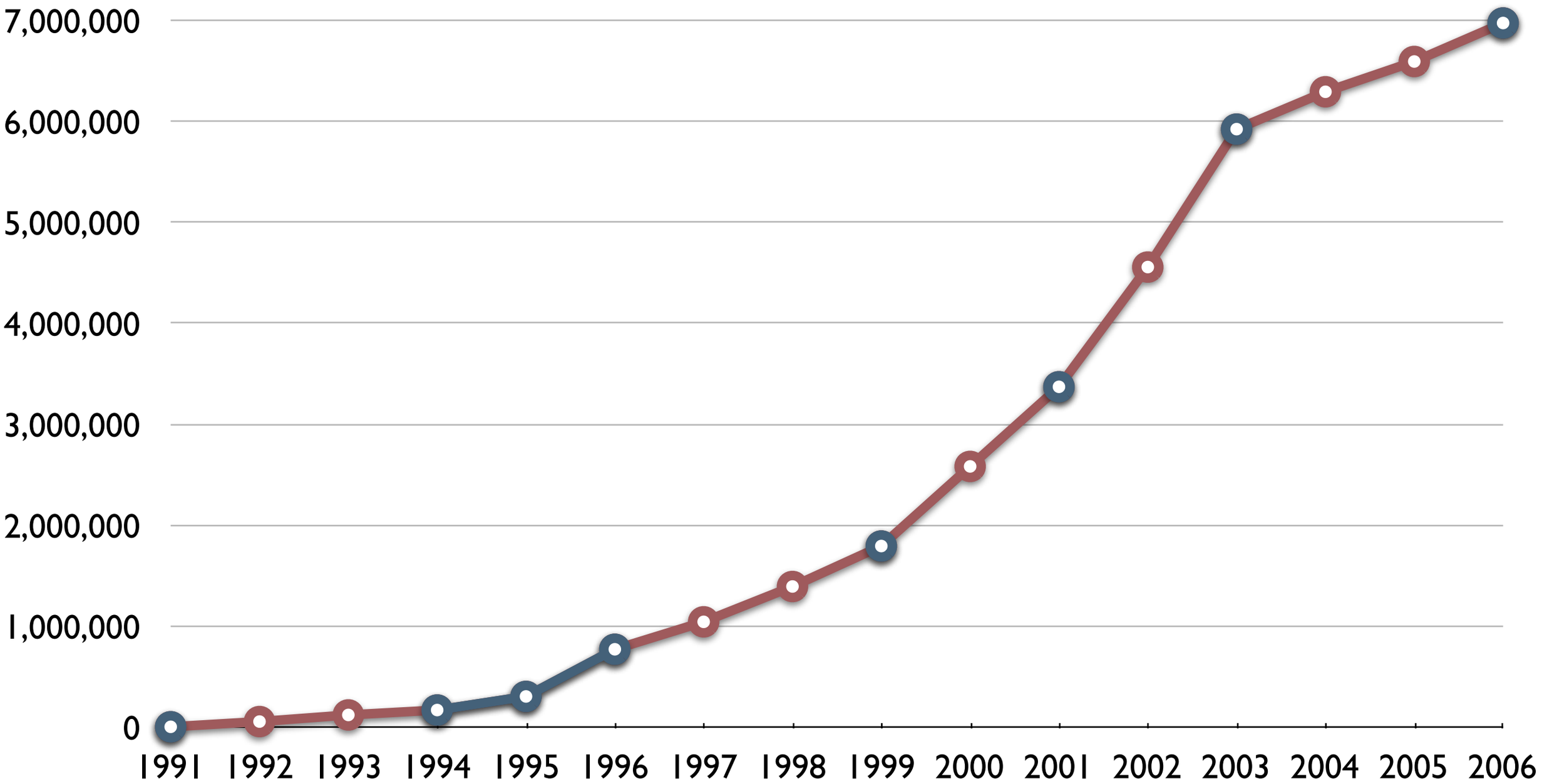
Darbat Overview

- Para-virtualised Darwin kernel
- L4 µ-kernel hypervisor
- Isolated kernel and device drivers
- Standard virtualisation benefits

Why?

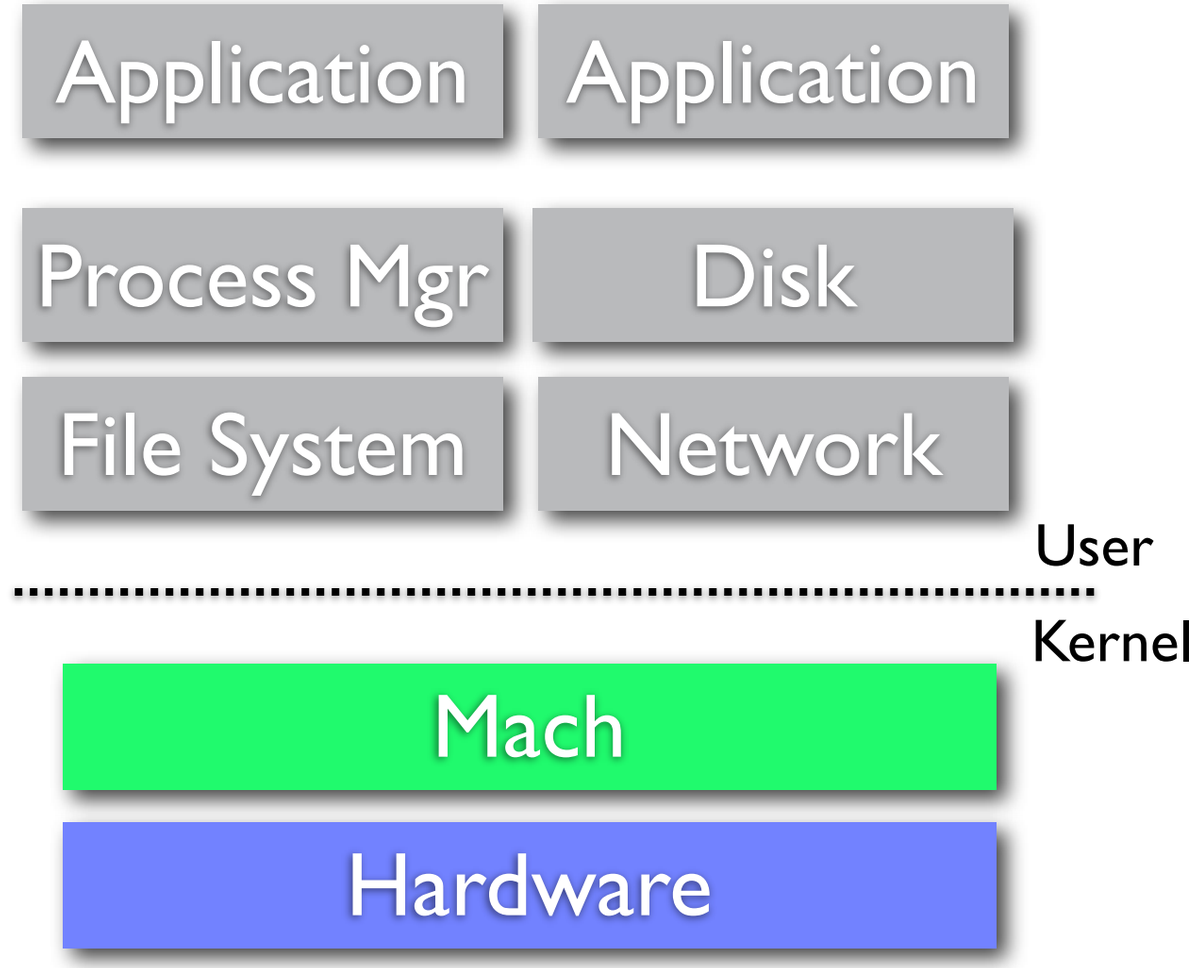
- Virtualisation
- Flexibility
- Robustness
- Code size

Linux Kernel Size (LoC)



Mach

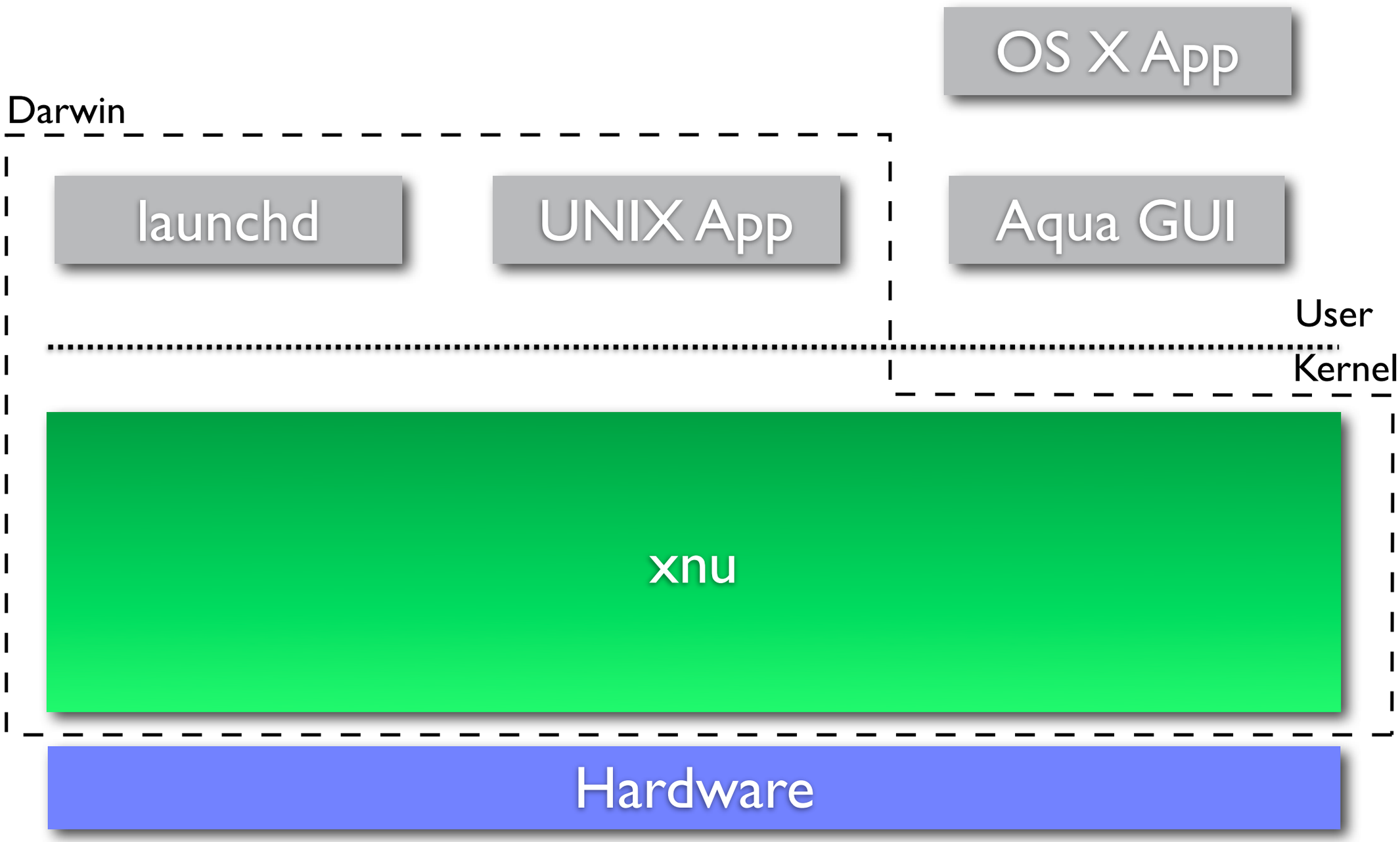
- Once a hot research topic
- Many attempts at BSD on Mach
- Darwin is a monolithic Mach kernel
- Gave microkernels a bad name



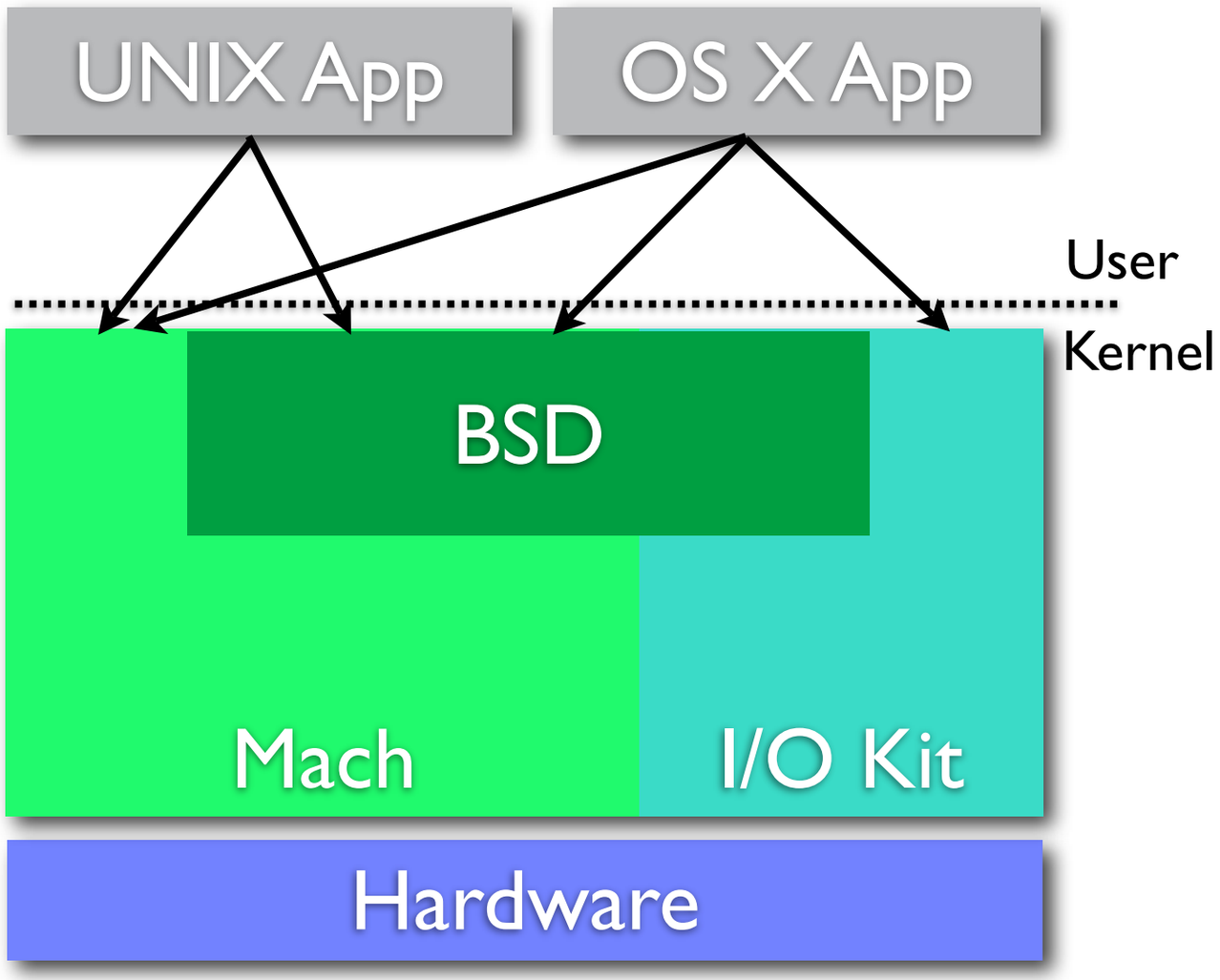
What's in a kernel?

- XML parser
- Decompression algorithms
- Linker
- Network routing
- Device drivers
- Strings

Mac OS X Architecture

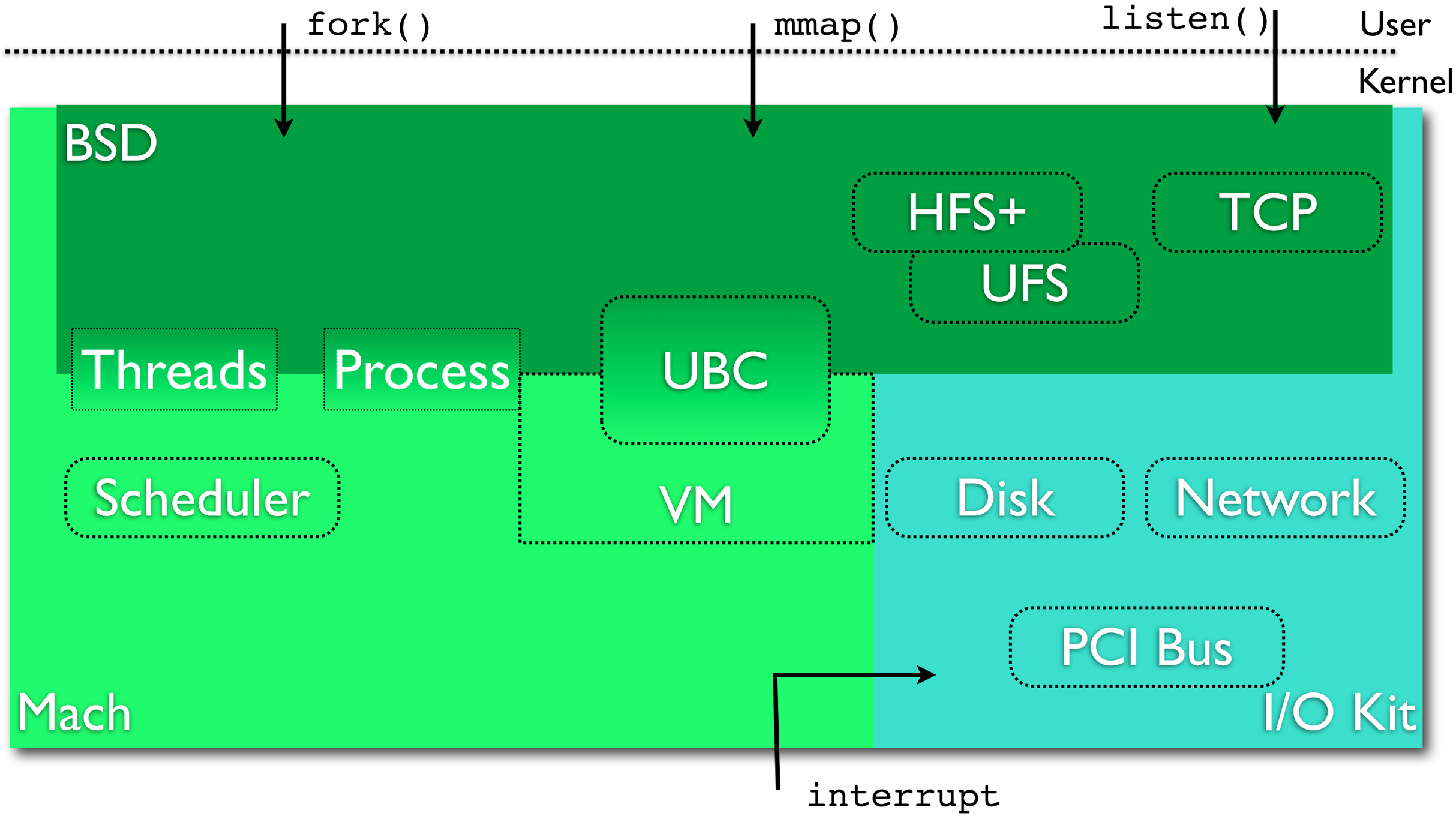


Darwin

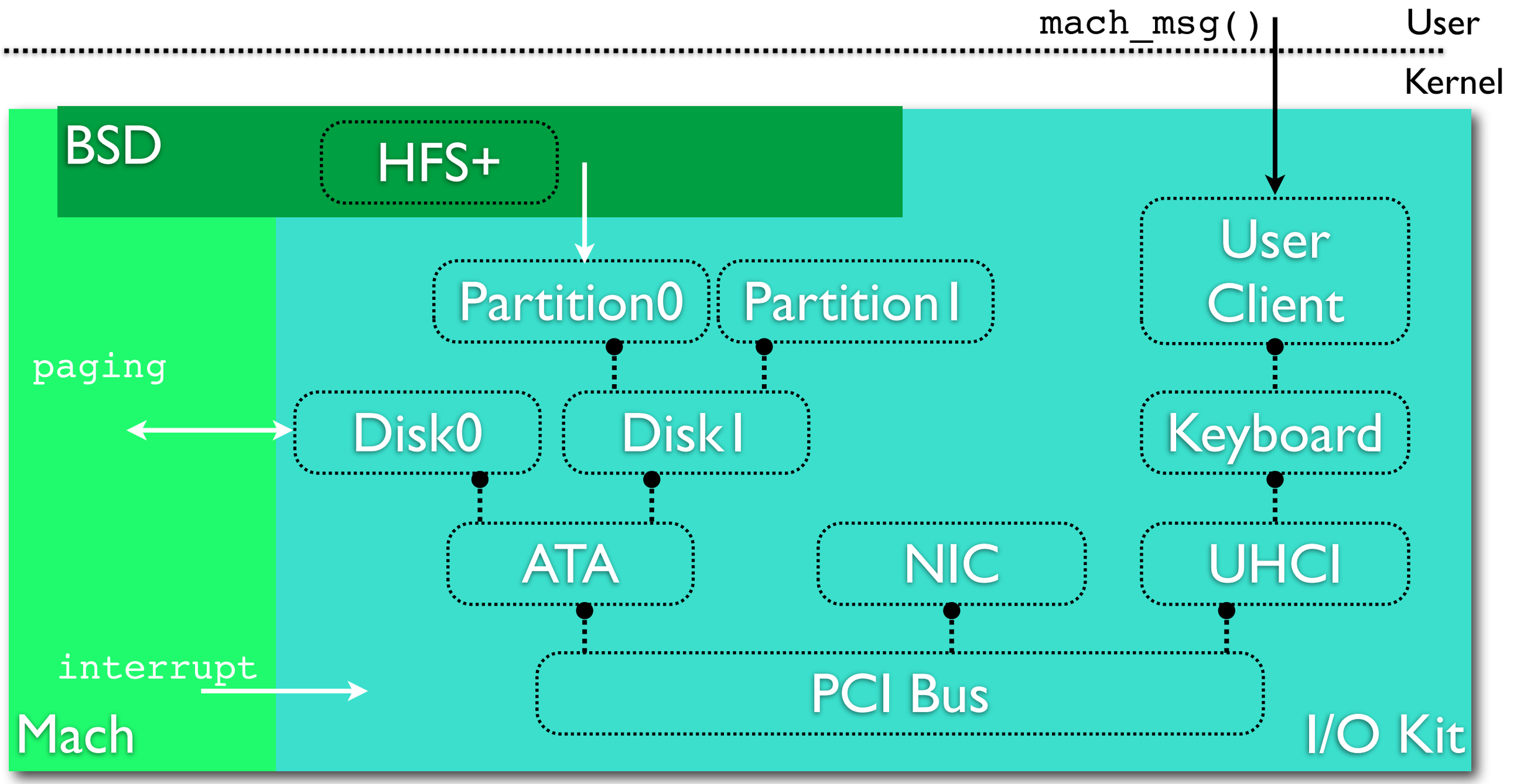


- Unique kernel design
  - Mach 3.0 µ-kernel
  - 4.4 BSD derivative
  - I/O Kit device drivers
- Mac OS X applications
- Full FreeBSD 5 UNIX environment

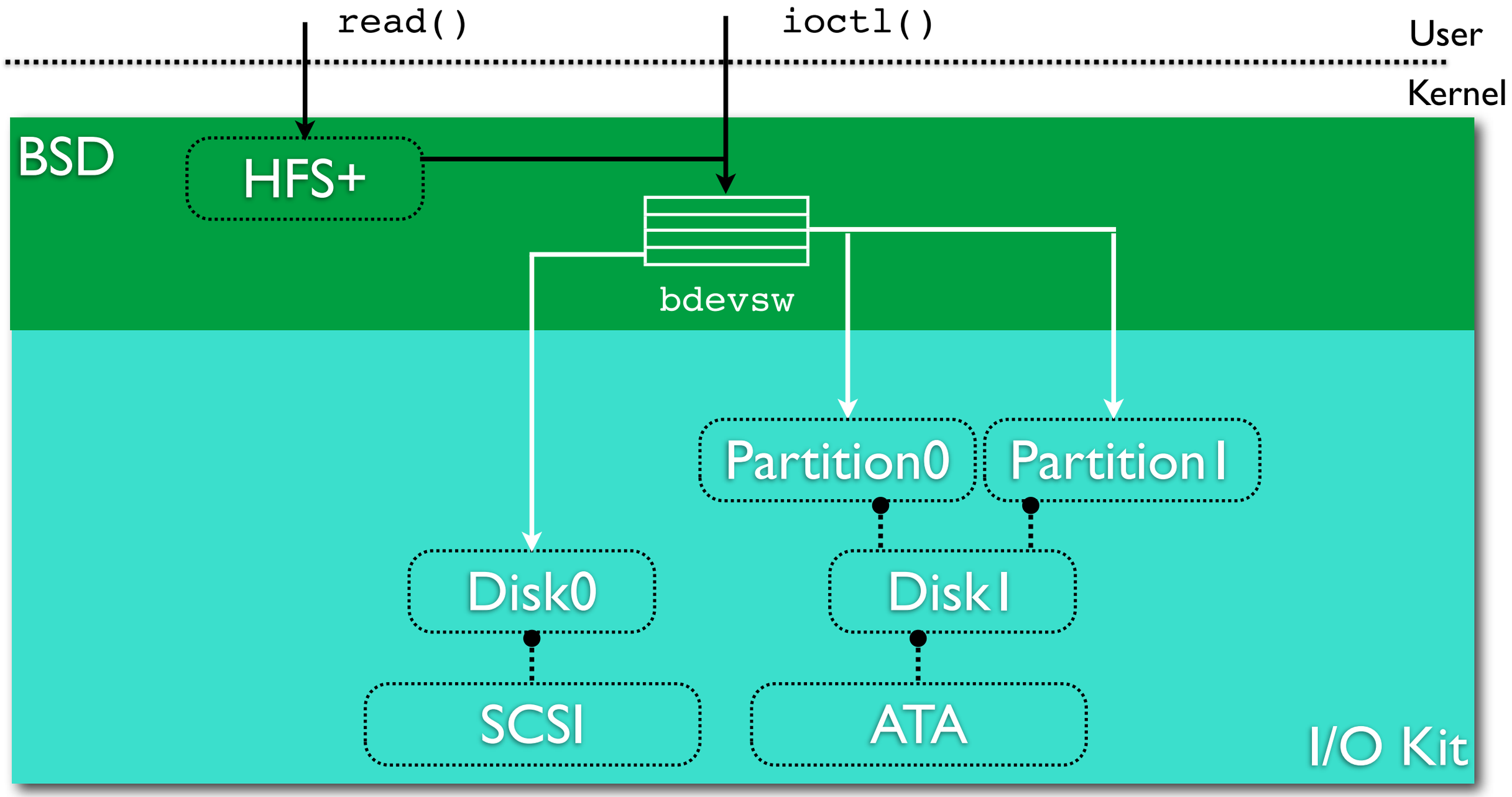
BSD Layer


fork()
mmap()
listen()
User
Kernel
BSD
HFS+
TCP
UFS
Threads
Process
UBC
Scheduler
VM
Disk
Network
PCI Bus
Mach
I/O Kit
interrupt

The I/O Kit


mach_msg()
User
Kernel
BSD
HFS+
Partition0
Partition1
User Client
paging
Disk0
Disk1
Keyboard
ATA
NIC
UHCI
interrupt
PCI Bus
Mach
I/O Kit

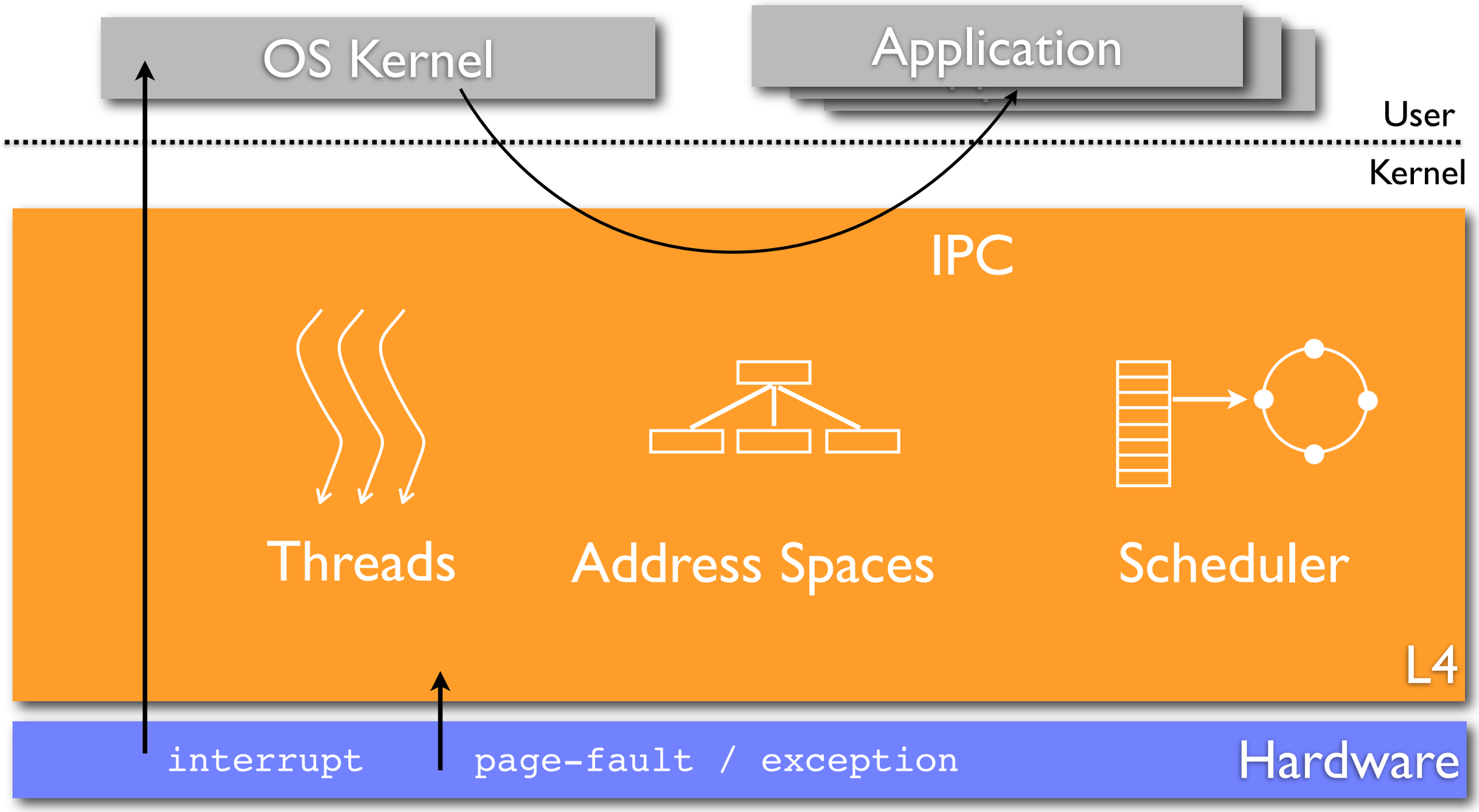
BSD and I/O Kit



L4

- 2nd generation u-kernel
- Learn from mistakes of Mach
- Pay attention to cache, TLB, complexity
- Make microkernels work
- 10-20x performance improvement over Mach

L4 µ-kernel



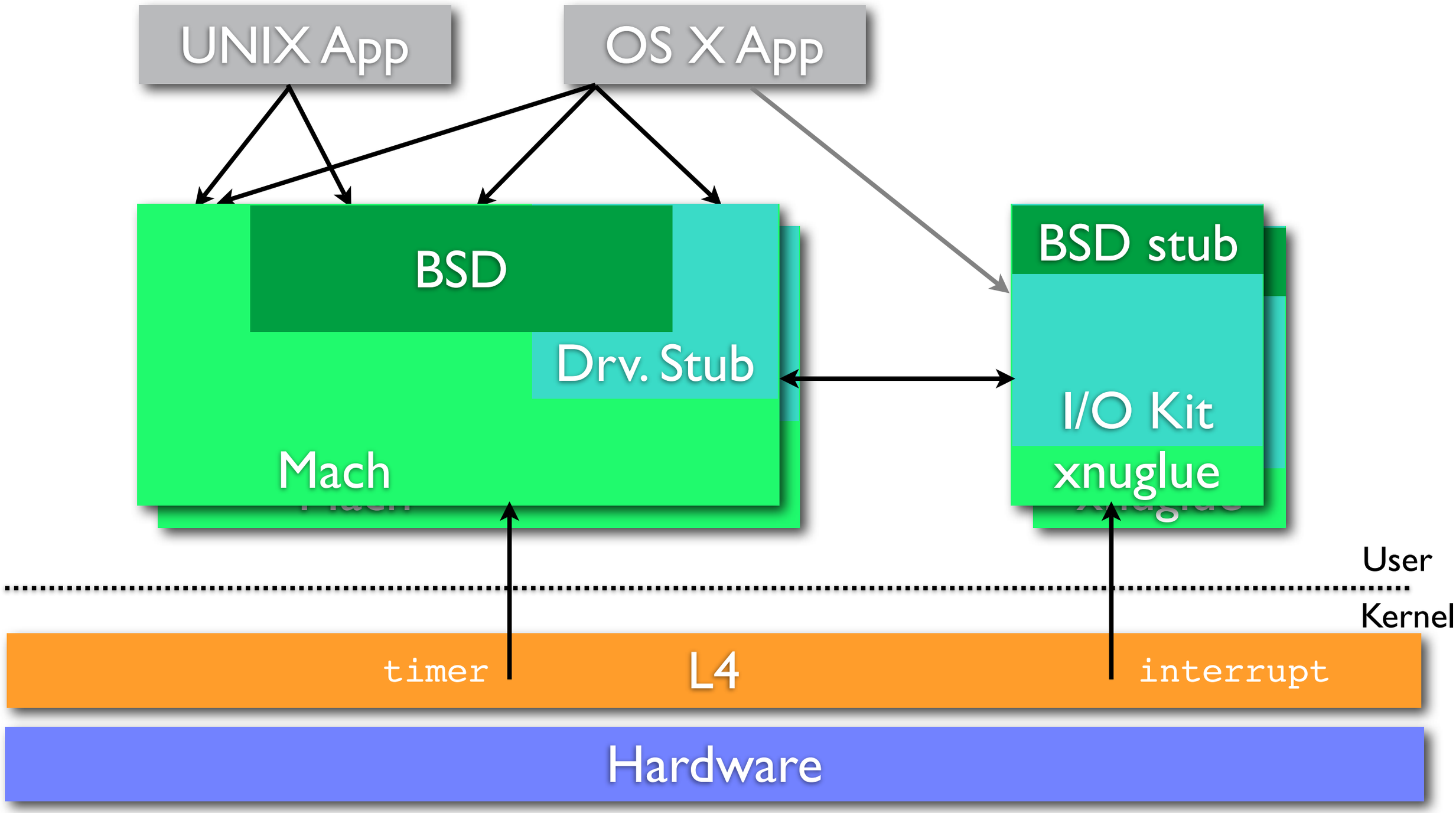
★not to scale

Darbat

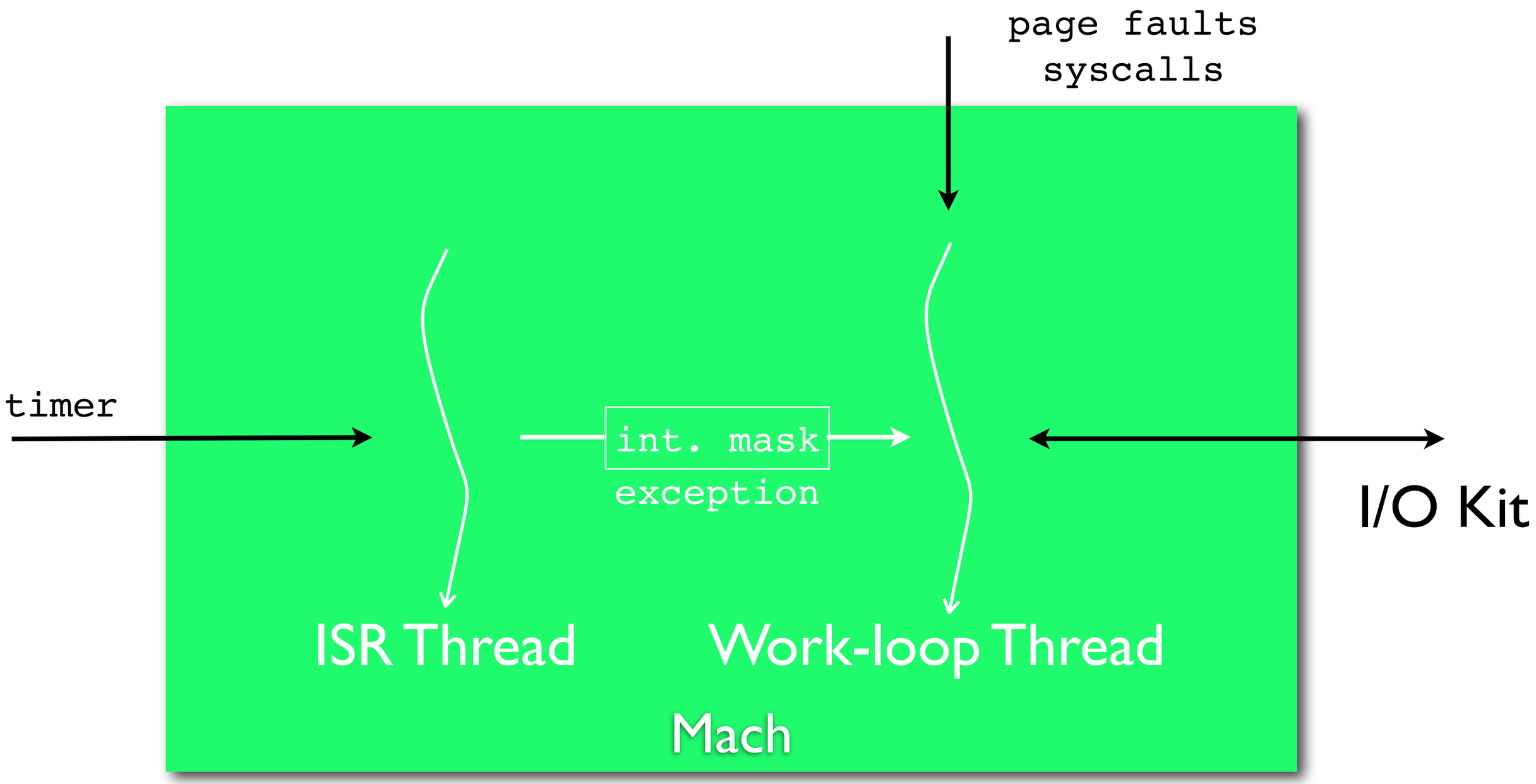
- Decompose Darwin kernel
- Tackle complexity problem
- Add flexibility
- Maintain (or improve) performance
- All with binary compatibility

NATIONAL ICT AUSTRALIA

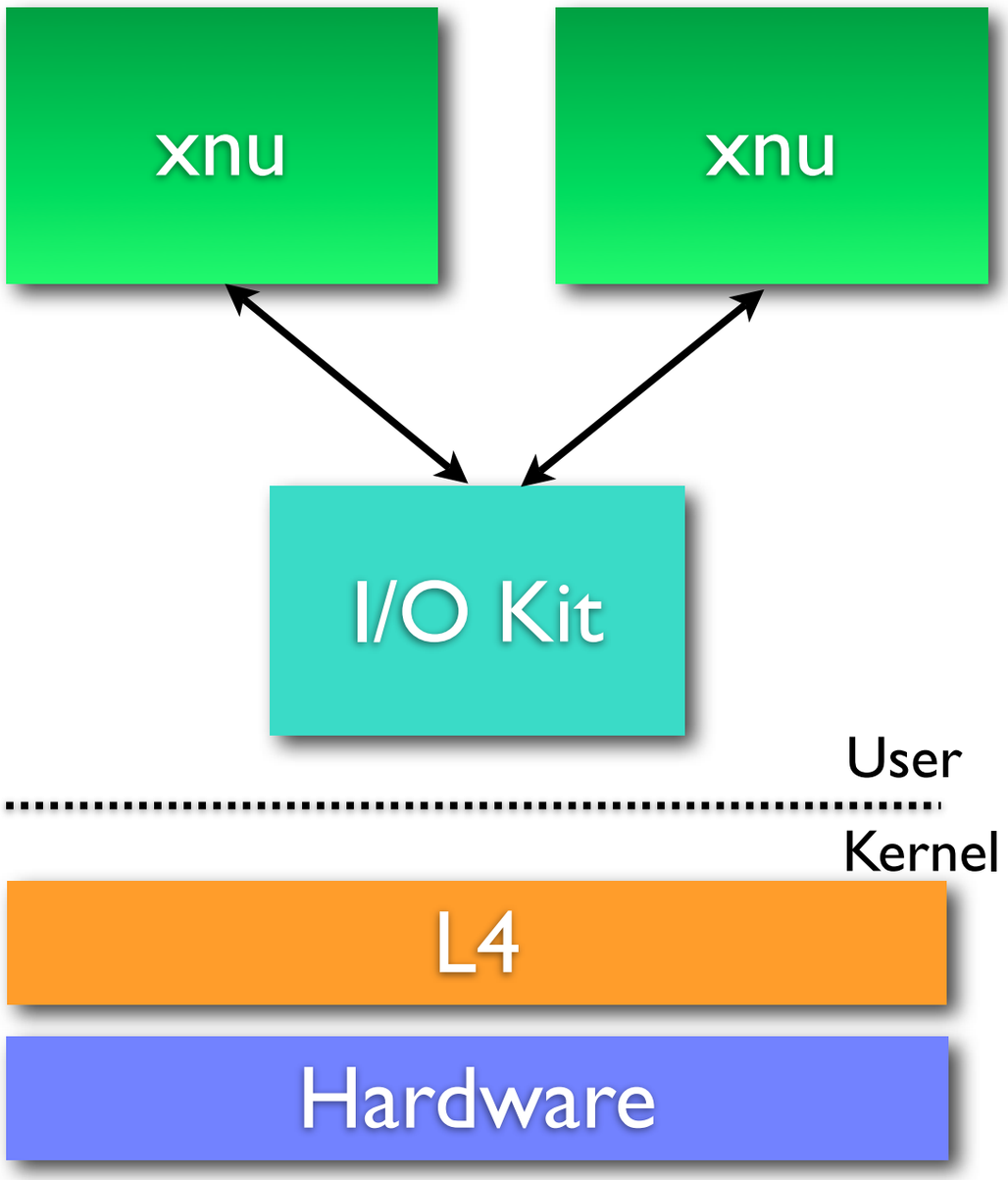
Darbat Model



XNU on L4


page faults
syscalls
timer
int. mask
exception
I/O Kit
ISR Thread
Work-loop Thread
Mach

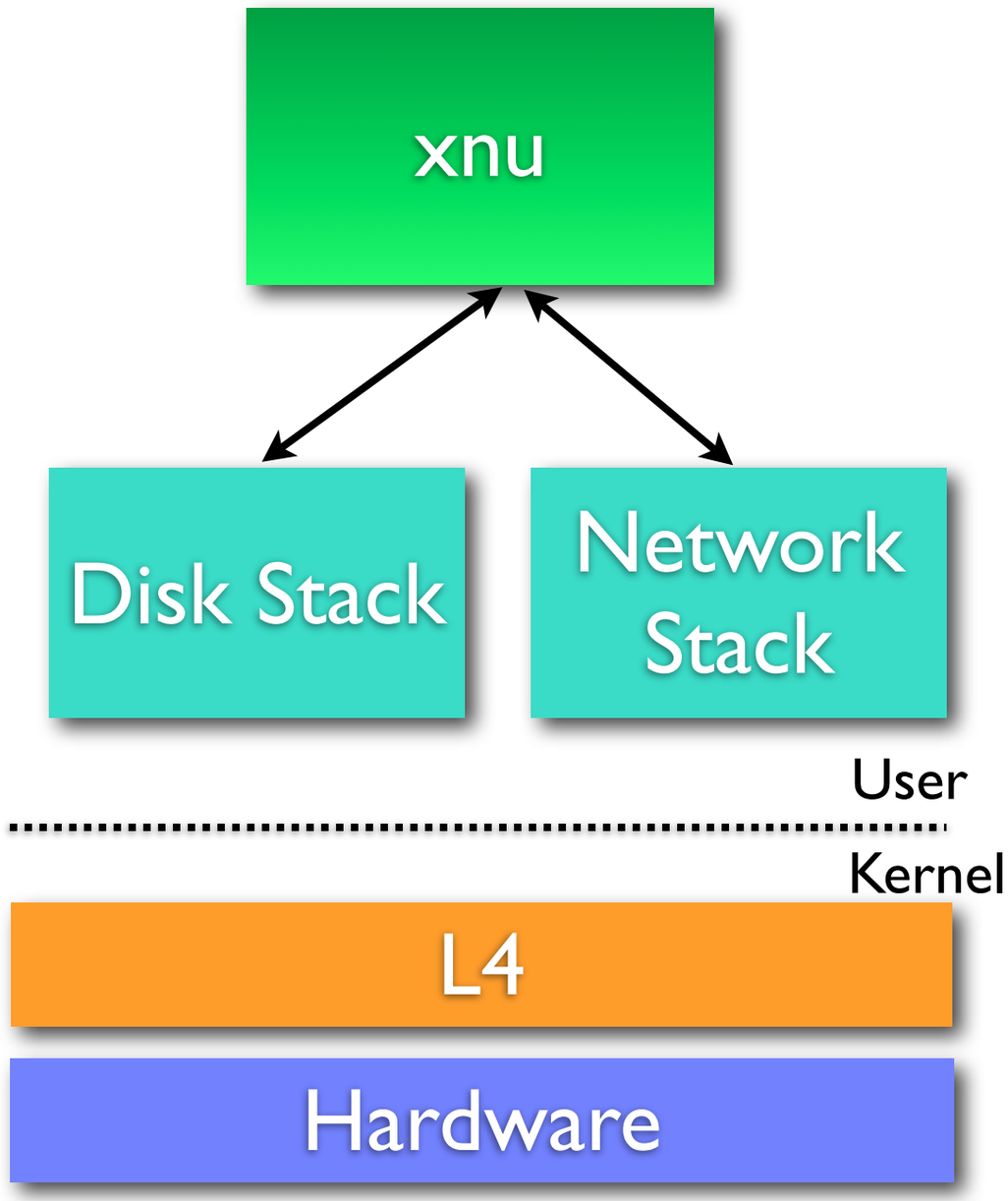
Server Consolidation



- Multiple OS instances
- Kernel support
- Simplified driver model

Crashable Drivers

- Drivers crash
  - Holding locks
  - Hogging resources
- Can isolate drivers
  - and keep performance
- Some devices still critical



NATIONAL ICT AUSTRALIA

VM Aware Scheduling

priority

Application

Application

Application

Application

Application

xnu

xnu

xnu

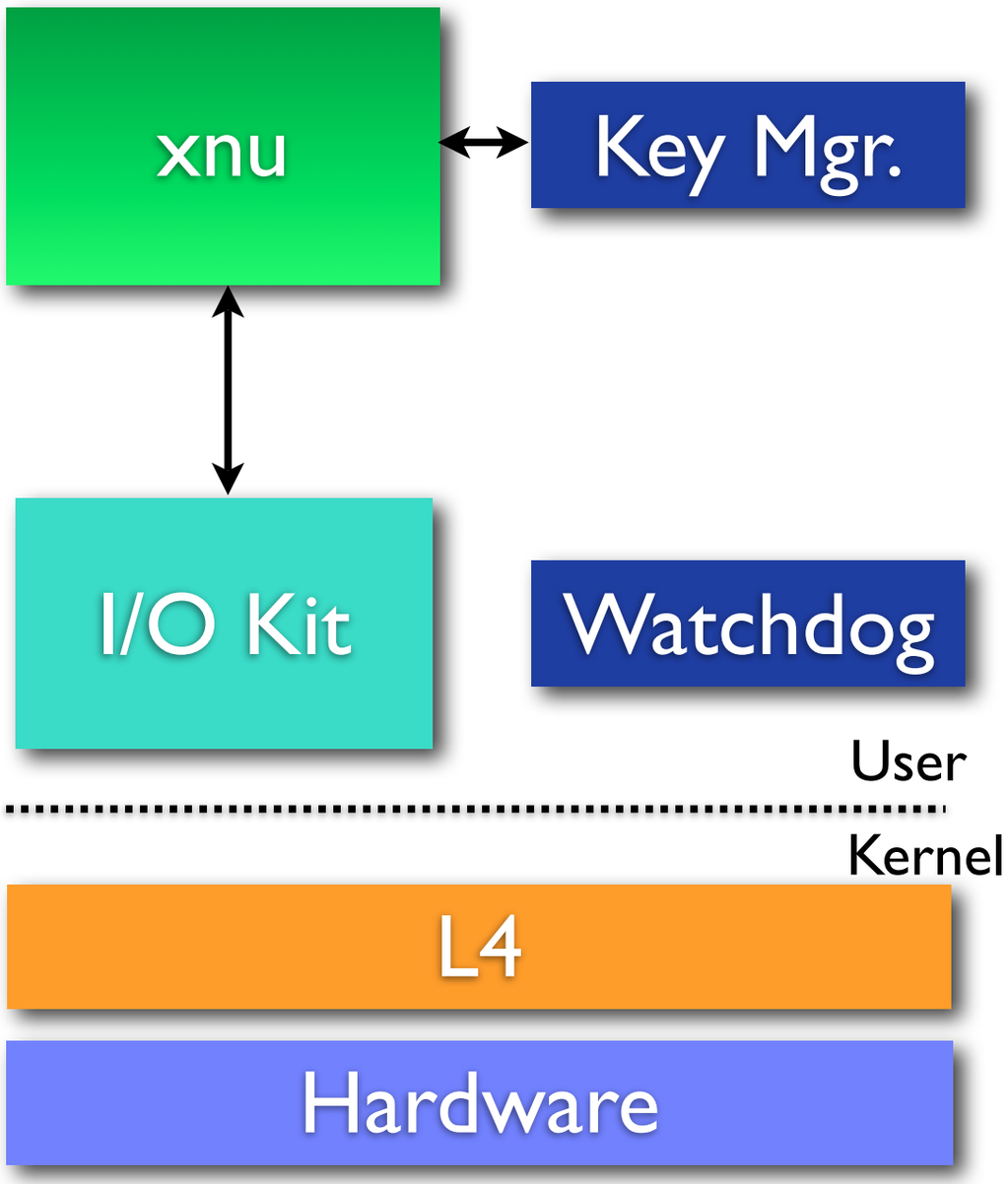
User

Kernel

L4

Hardware

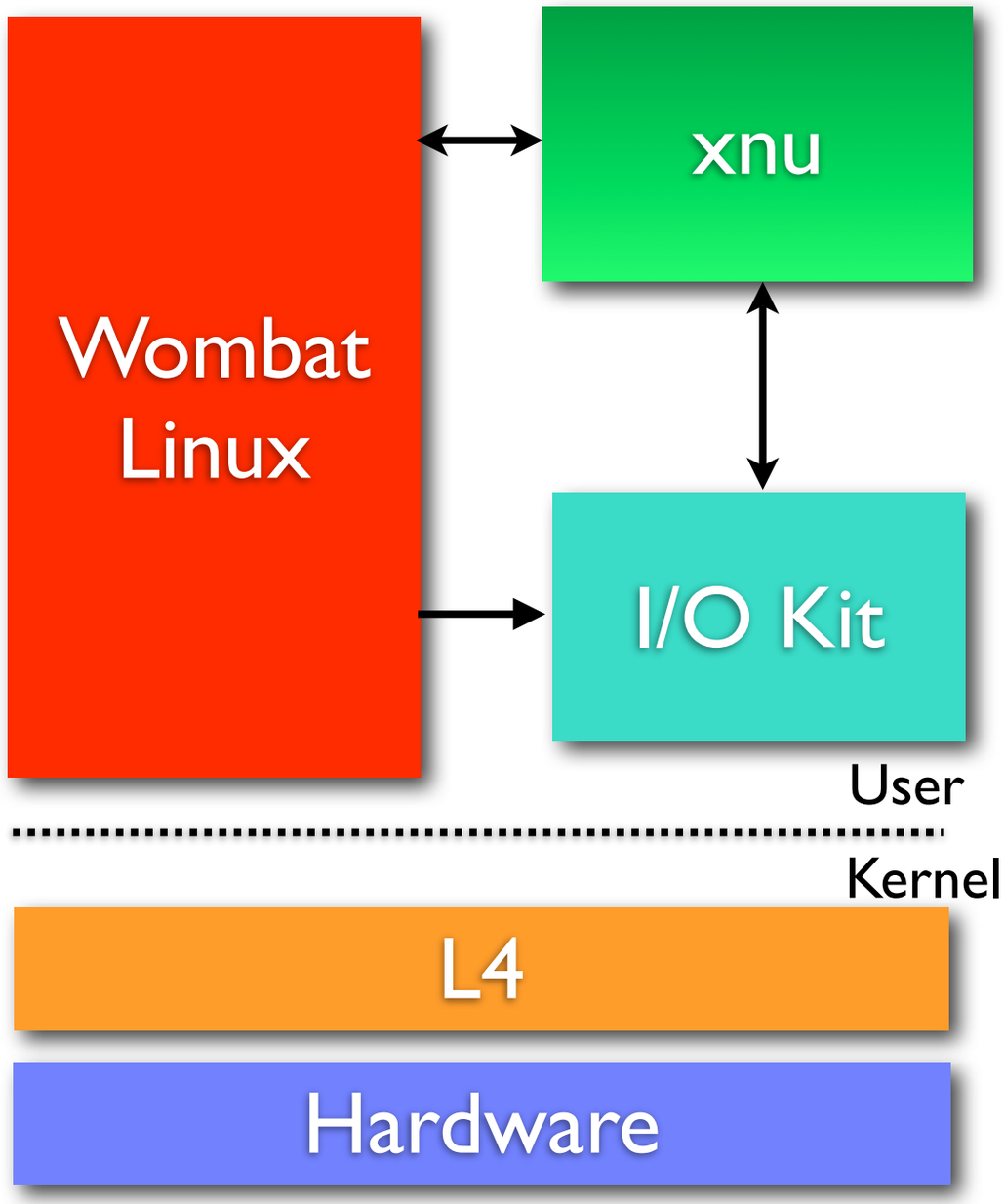
Reliable Servers



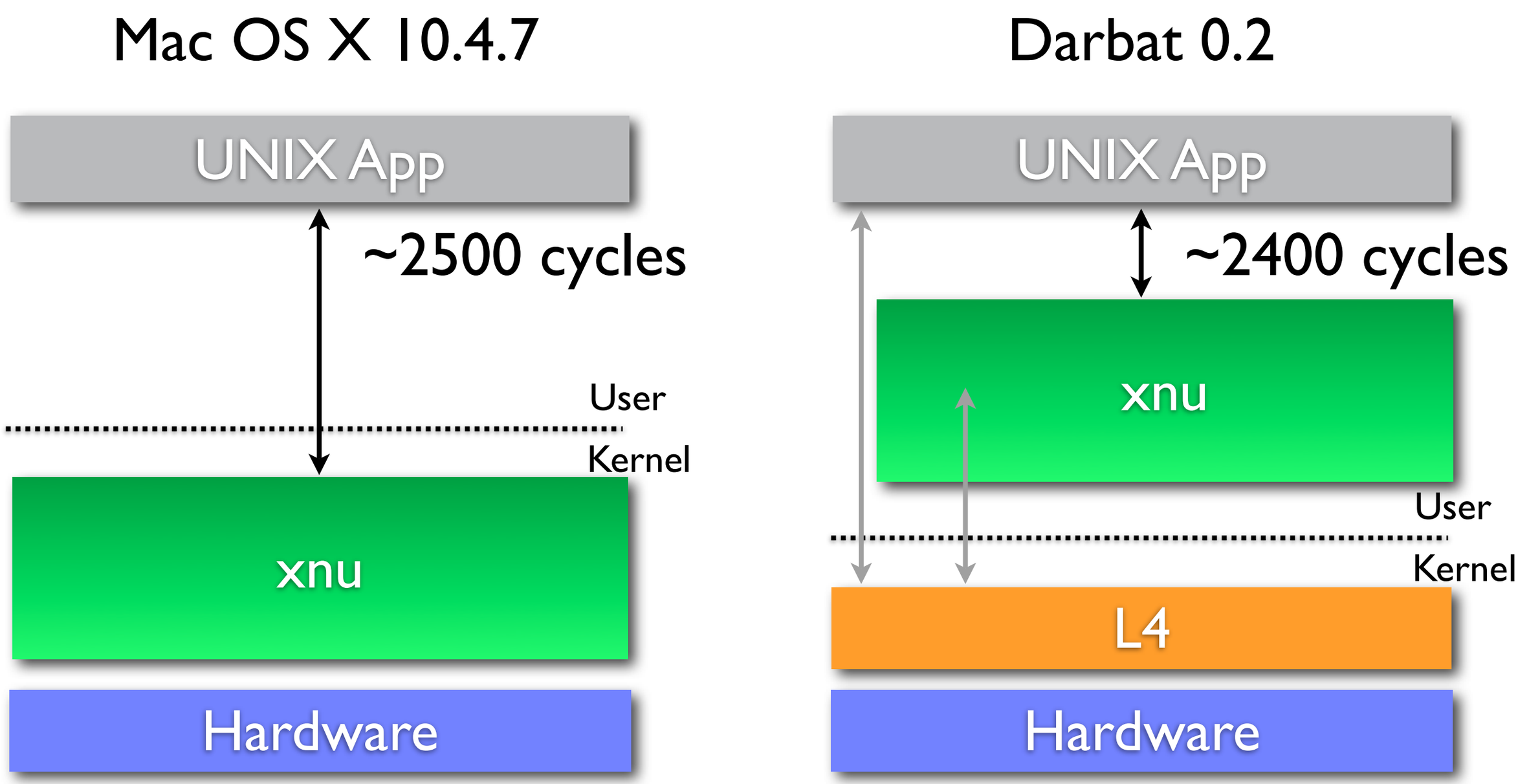
- Microkernel is very resilient
- Protect data from 3rd party modules
- Ensure reliable execution of code
- Cheaper software solutions

Heterogeneous Systems

- L4 is OS (everything?) agnostic
  - Linux
  - *BSD
- Share devices
- Share file systems



System Call Performance

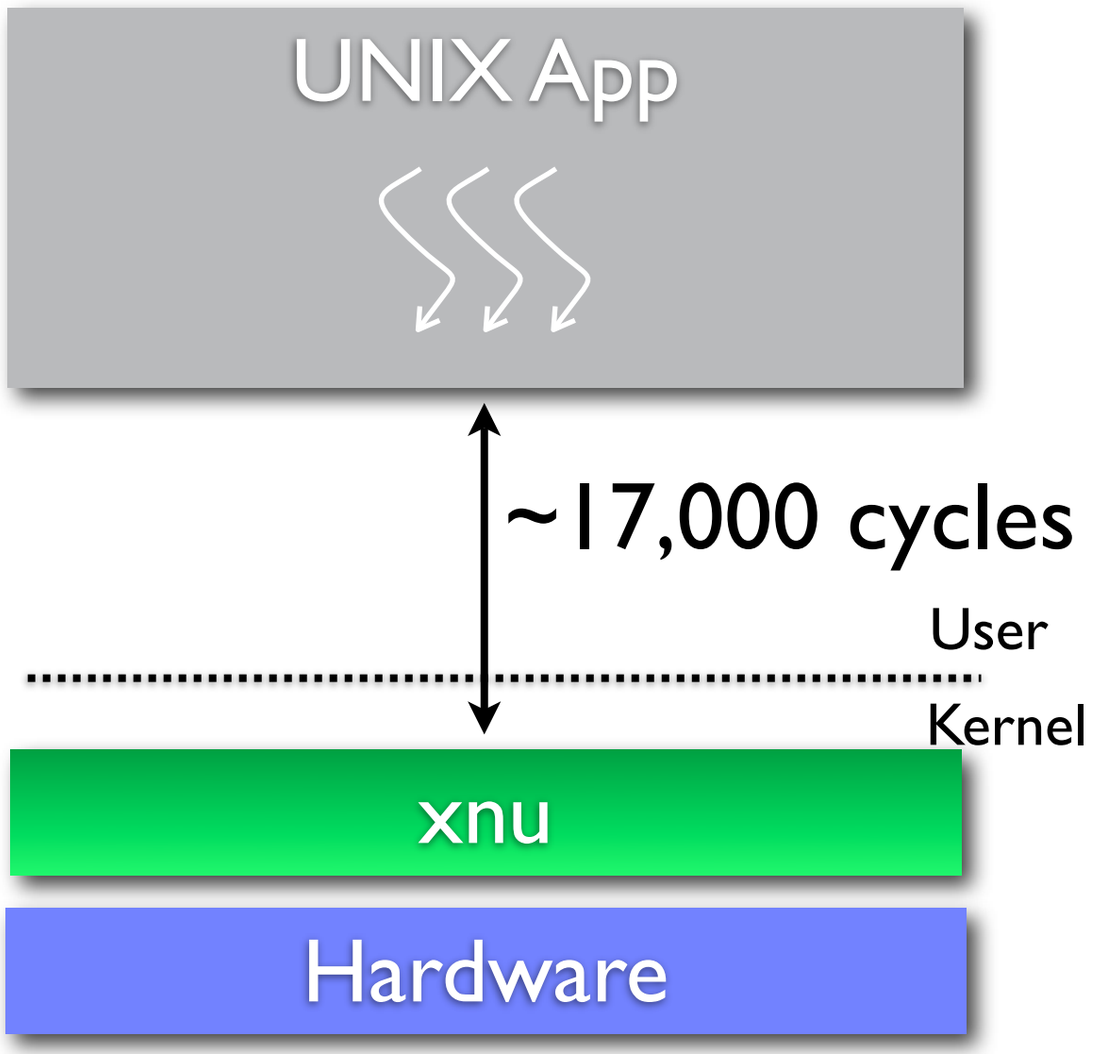


mach_msg null operation

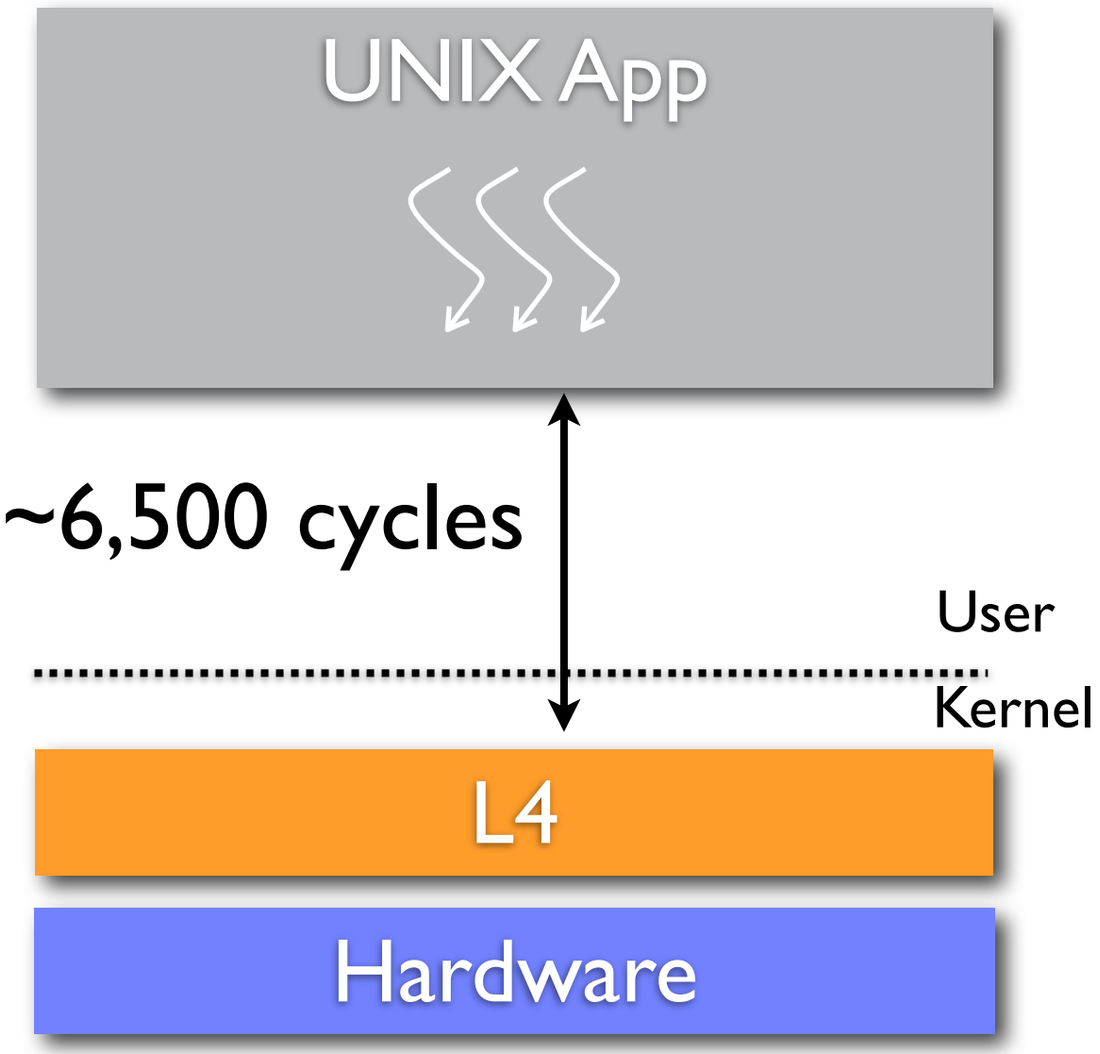
NATIONAL ICT AUSTRALIA

In-task Synchronisation

Mac OS X 10.4.7



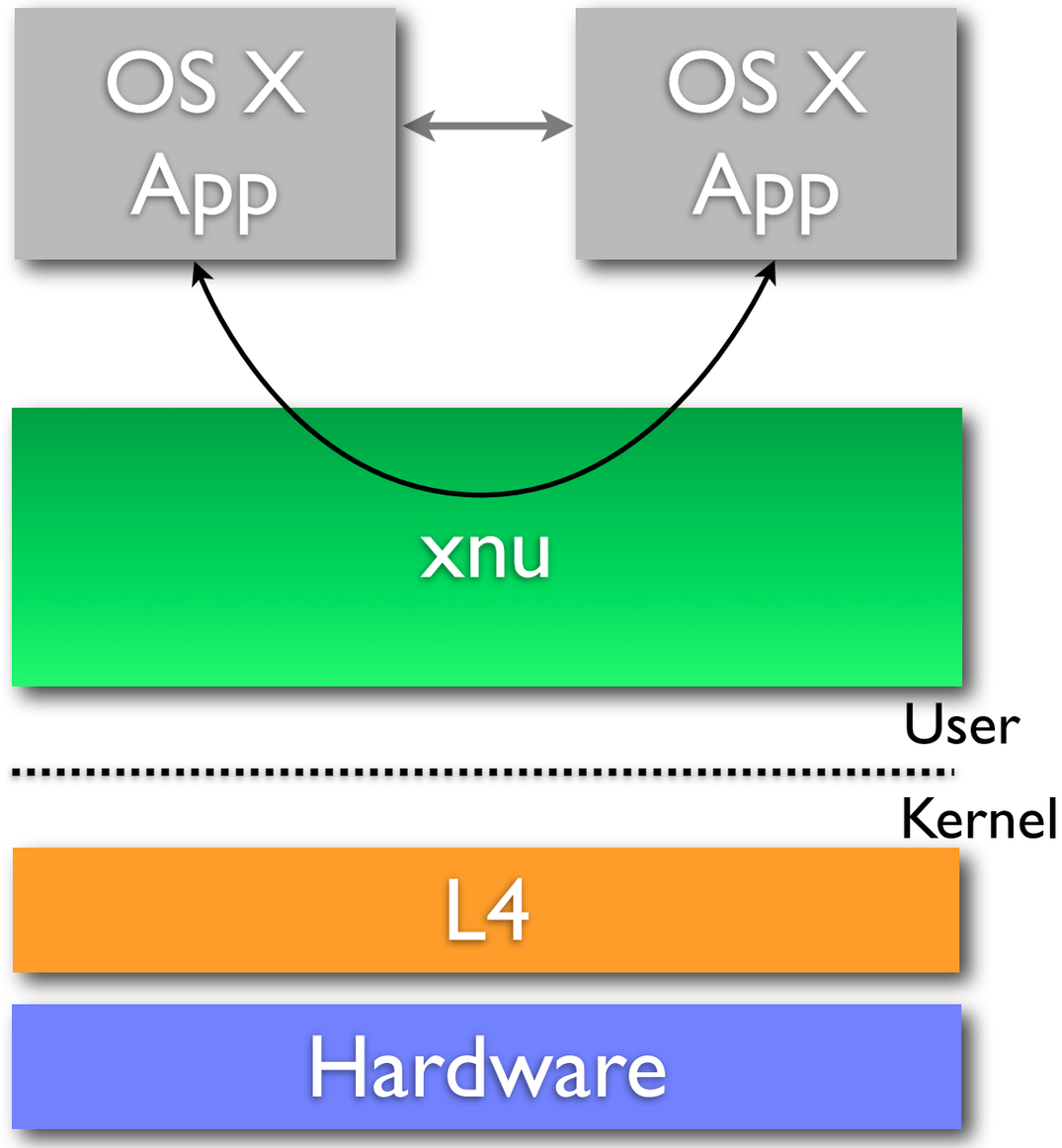
Darbat



pthreads synchronisation

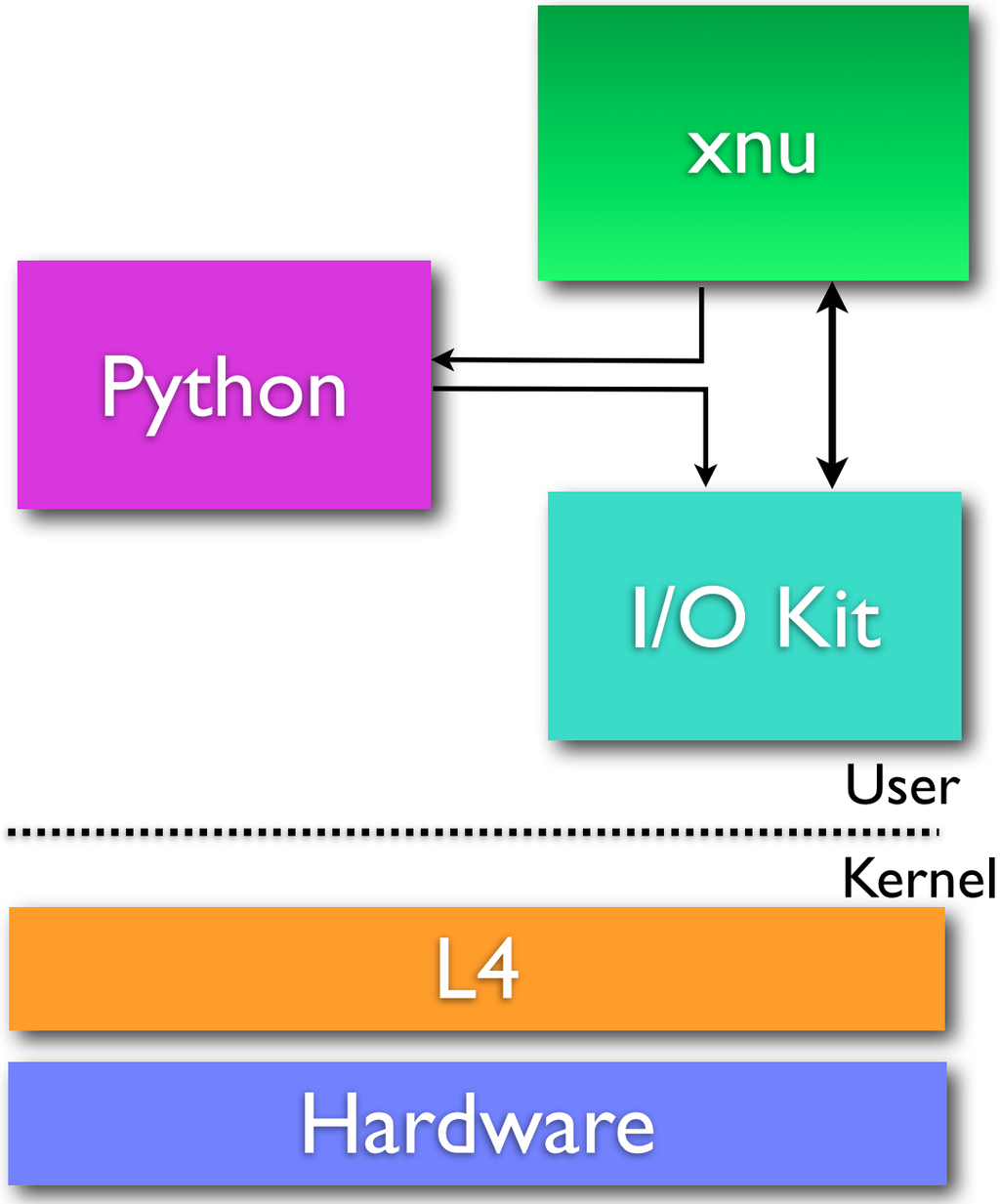
IPC Optimisation

- L4 IPC is a subset of Mach IPC
- Applications rely on Mach semantics
- Optimisation for some messages



Kernel Scripting

- Sometimes you just need a hack
- In-kernel scripts get messy
- Scheduling and locks are a pain
- Don't compromise the whole system



Lessons So Far

- Darwin modularity
- Binary compatibility
- Mac OS X has bugs
- Performance
- Debugging

Future Work?

- Further decomposition
- Improved xnu and I/O Kit bindings
- Usable system!
- Experiment with features
- System benchmarks

Summary

- UNIX kernels are feature-packed
- L4 is one option to address this problem
- Extend UNIX kernel to meet modern usage

Questions?