OS Extensibility: SPIN and Exokernels

Robert Grimm
New York University

The Three Questions

- What is the problem?
- What is new or different?
- What are the contributions and limitations?

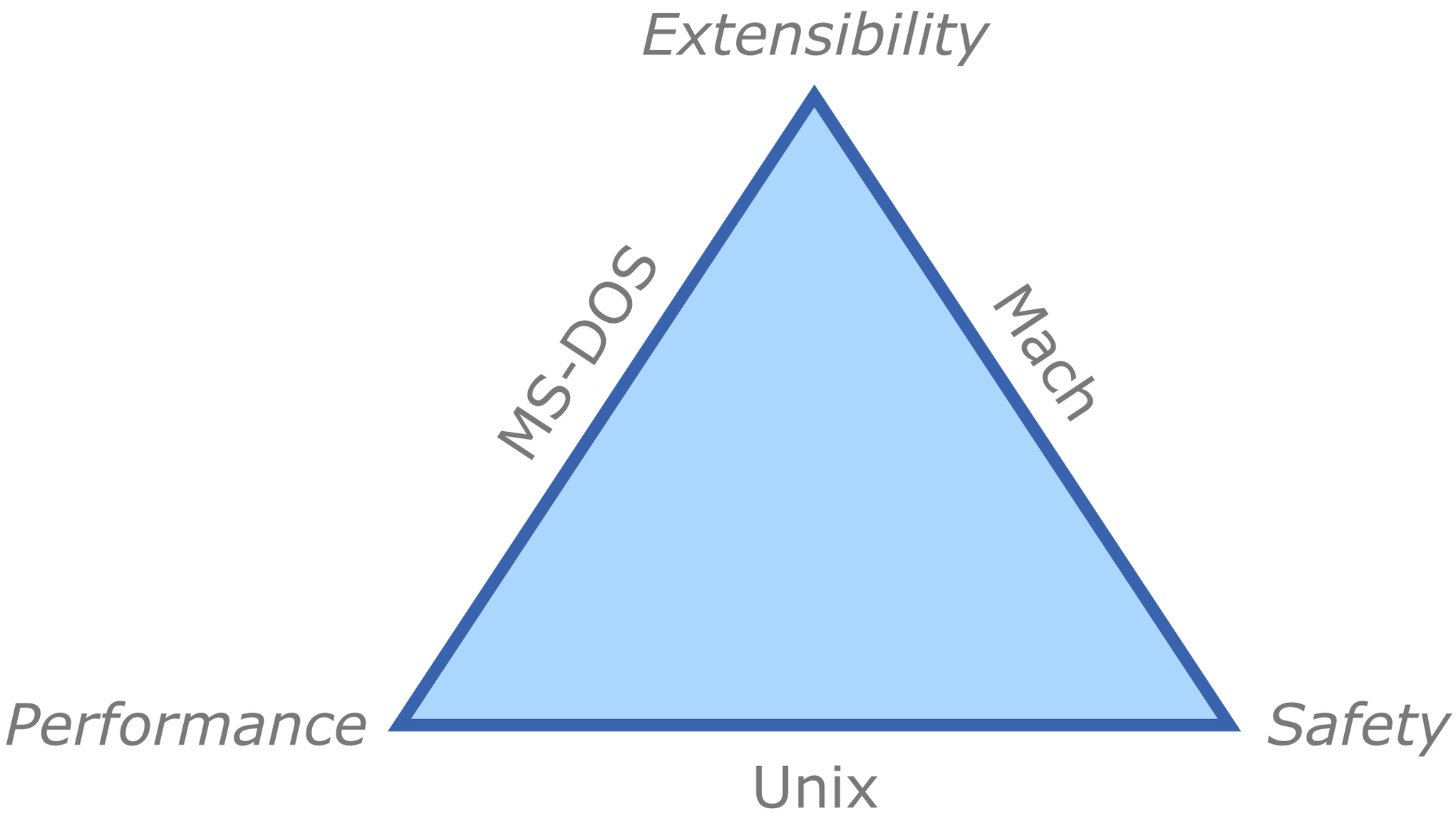
OS Abstraction Barrier

- Fixed high-level abstractions
  - Hurt application performance
  - Hide information
  - Limit functionality
- Examples
  - Buffer cache management
  - Persistent storage

Goals

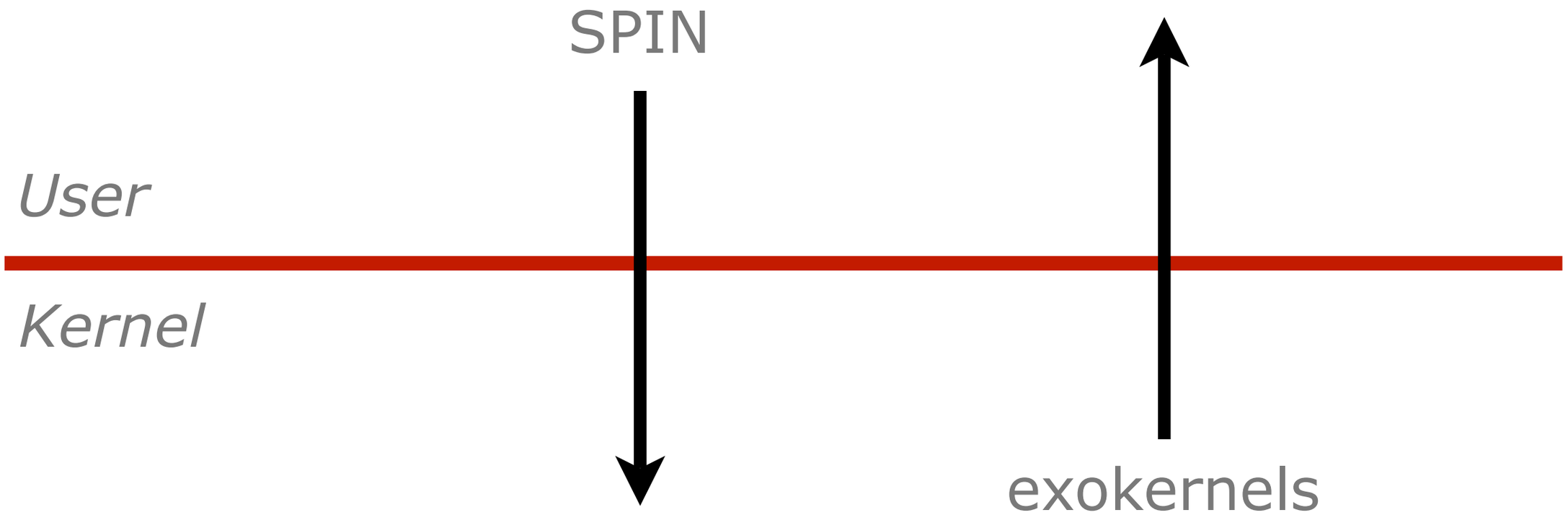
- Extensibility
  - Applications introduce specialized policies/services
- Safety
  - Kernel, applications, services are protected from each other
- Performance
  - Extensibility and safety have low cost

Why Is This Hard?



* Key challenge: Can we get all three in a single system?

The Two Approaches



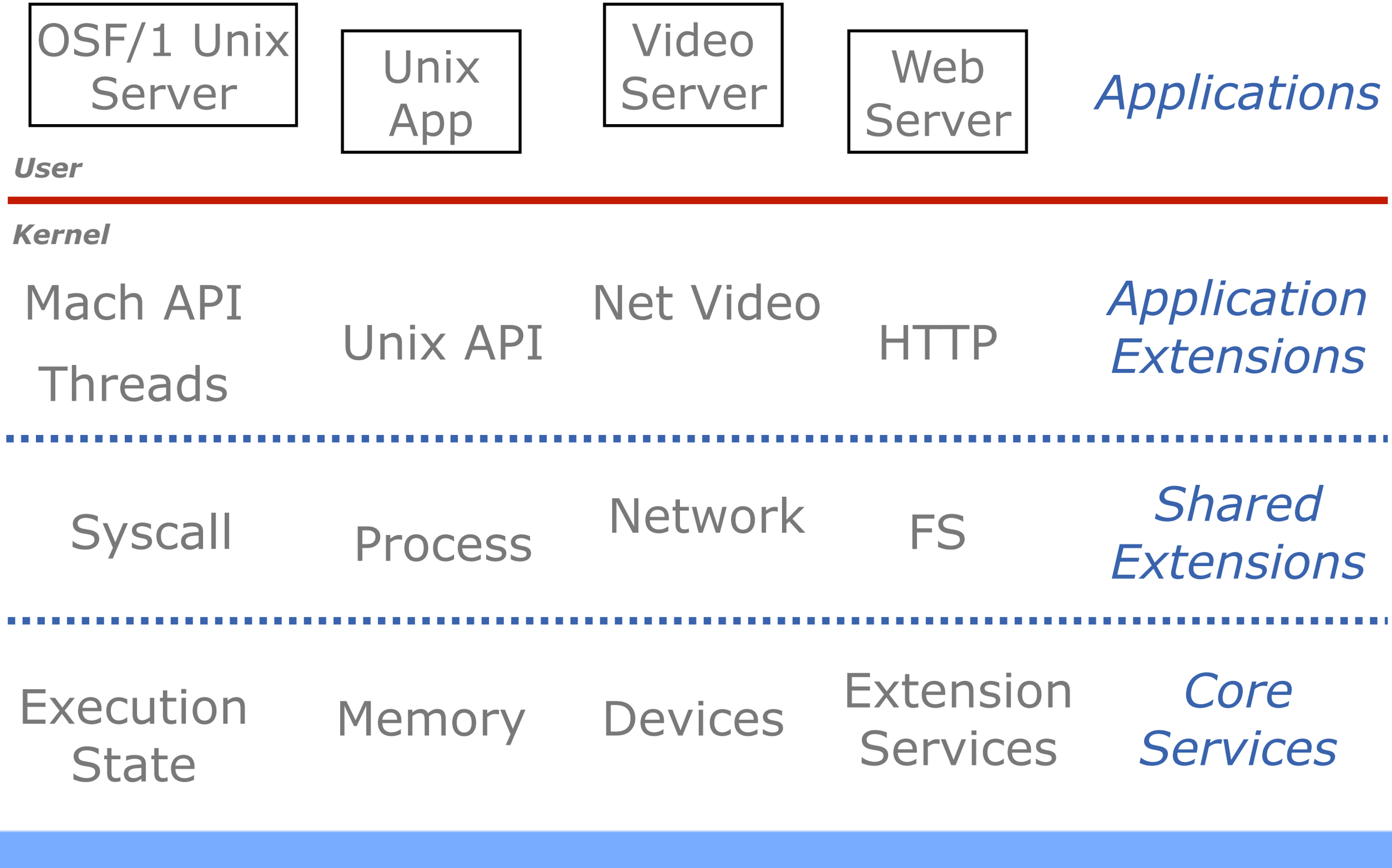
- SPIN: move application-specific functionality in kernel
- exokernels: make kernel barrier as low as possible

SPIN

SPIN Approach

- Put extension code into the kernel
  - Cheap communication
- Use language protection features
  - Static safety
- Dynamically impose on any service
  - Fine-grained extensibility

The Big Picture



Modula-3

- Type-safe programming language
- Interfaces
- Garbage collection
- Other features
  - Objects, generic interfaces, threads, exceptions

- Most of kernel written in Modula-3
  - Drivers "borrowed" from DEC OSF/1
- Extensions must be written in Modula-3
- User-space applications written in any language

Safety

- Capabilities
  - Simply a pointer
    - Can we pass capabilities to user-land?
  - Protection domains
    - Language-level
    - Limit visibility of names
    - Enforced at dynamic link time

Extensibility

- Extension model
  - Events
    - Indicate the occurrence of some condition
  - Event handlers
    - May execute synchronously, asynchronously, in bounded time
  - Guards
    - Restrict invocation of event handlers based on arguments
- Mechanism
  - Event dispatcher
  - Common case: an (indirect) procedure call
    - Module implementing the interface

Core Services

- Memory management
  - Physical addresses
  - Virtual addresses
  - Translations
- Thread management
  - Signals to scheduler
    - Block, unblock
  - Signals to thread manager
    - Checkpoint, resume

Performance

- It works

Exokernels

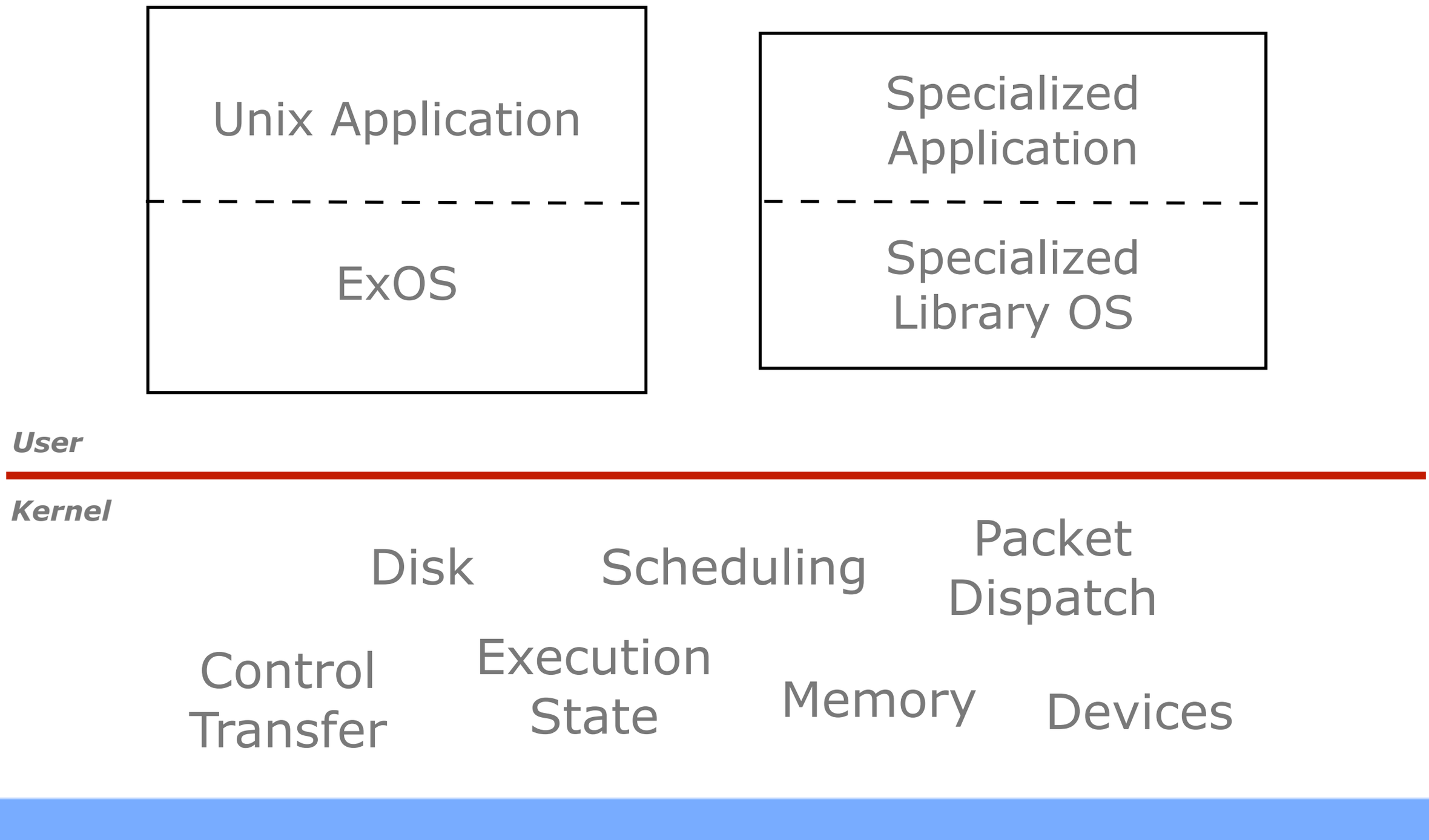
Exokernels Approach

- Make the application do it!

Exokernels Approach (again)

- Separate protection and management
- Expose allocation
- Expose (physical) names
- Expose revocation
- Expose information

The Big Picture



At the Core

- Processor time slices
- Process environments
  - Hardware exceptions (Aegis, Xok)
  - Timer interrupts (Aegis, Xok)
  - Protected entries for building IPC (Aegis, Xok)
  - Virtual memory addressing
    - Aegis: guaranteed mappings, apps notified of TLB misses
    - Xok: hardware page tables, apps specify mappings
  - Hierarchical capabilities (Xok only)
- Book keeping

Aegis	MIPS DECstations
Xok	x86 PCs

How to Protect Shared Abstractions?

- Software regions
  - Provide access to memory *only* through system calls
  - Are typically more fine-grained than pages
- Hierarchically-named capabilities
  - Easily restrict access
- Wake-up predicates
  - Ensure that processes get their time in the limelight
- Robust critical sections
  - Provides isolation with low overhead and without requiring cooperation

The Disk

- Problem
  - How to store meta-data?
    - Ownership of disk blocks
- Failed approaches
  - Simple capabilities
    - Where to put them? In the block? A separate area?
  - Self-descriptive meta-data
    - How expressive is the description language?
    - How much space is used for descriptions?
  - Template-based descriptions
    - Again, how expressive is the description language?

The Disk (cont.)

- Untrusted deterministic functions
  - Programmatic templates that specify pointed-to blocks
- Shared data
  - System-wide buffer cache registry
    - Entries can be pinned
- Ordered disk writes
  - Ensure consistency after crash
    - Never reuse on-disk resource before nullifying pointers
    - Never create pointers before data has been initialized
    - Never reset old pointers before new one has been set

Performance

- It works
- It scales

Smackdown

SPIN Issues

- Trusted compiler
- Resource control

Exokernels Issues

- Extension model
- Downloaded code
  - Wake-up predicates (to identify interesting events)
  - Dynamic packet filters (to identify packets)
  - Application-specific handlers (to process packets)
  - Untrusted deterministic functions (to identify meta-data)
- Complexity of disk management

What Do You Think?