The Systems

Systems History

OS/370, OS/2
PDP-11 System V

BSD UNIX
Minix: Intel
CMU Mach 3: OSF/1, xnu
VAX-VMS/Windows NT
Sun Solaris
Linux

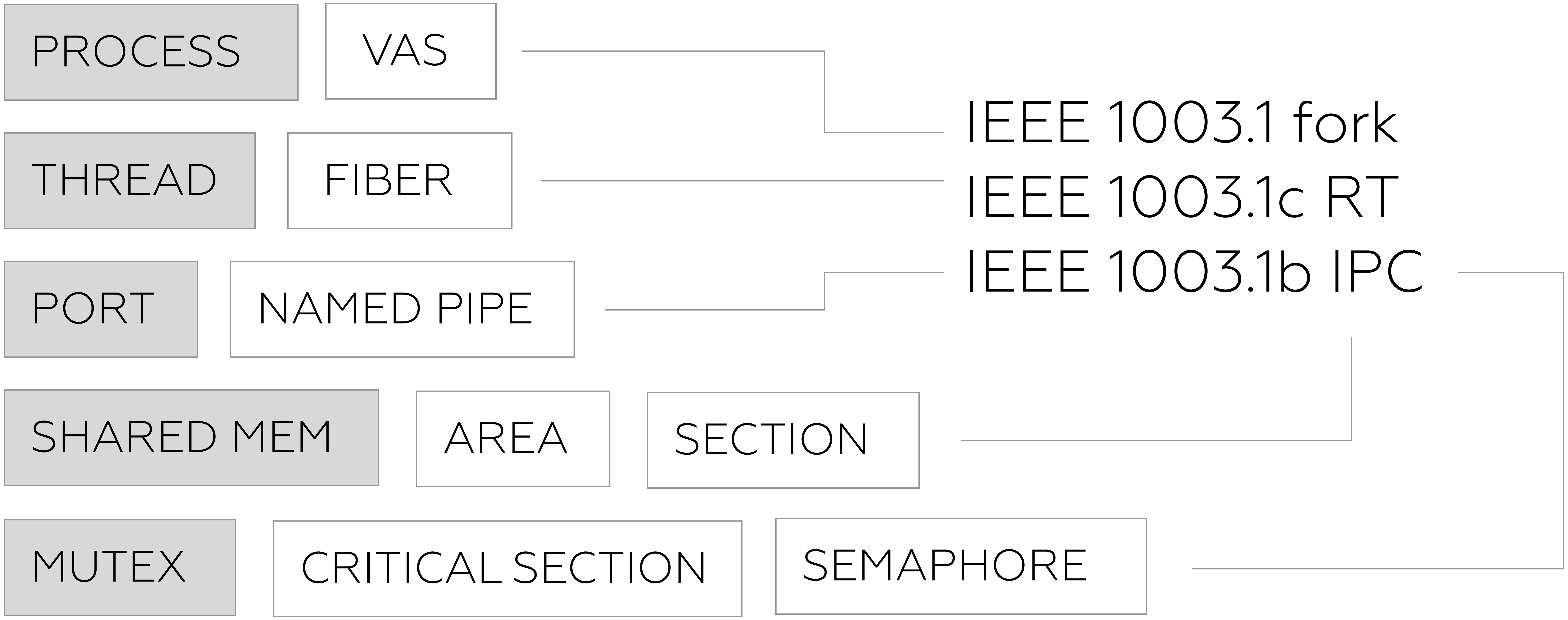
Amiga
RISC OS
Atari TOS

BeOS
WxWorks
TRON
QNX
Mac OS
L4

Squeak / Smalltalk
Plan9
Lisp Machines

Mirage / OCaml
HaLVM / Haskell
LING / Erlang

POSIX / Unix / System V



(+1) HAL IBM, (+1) VMS WNT

TIMERS	CPU	INTERRUPTS	CPU
ROOT	OBJECTS	ACL	RWX
SCHED	ALGORITHMS		
DRIVERS	VFS	BUS	DEV

Runtime File Formats

Mach-O	DEC OSF/1, DEC Tru64 UNIX, CMU Mach 3, xnu/Darwin
Portable Executable (PE/COFF)	Windows NT, BeOS, OS/2
Executable and Linkable Format (ELF)	Linux, QNX, Solaris
WebAssembly	

Object Code

headers
.code
.data
.reloc
.idata

- objdump
- link
- cc
- as

File Systems

Streams

Zetabytes Volumes

ACL

Transactional Log

RAID Stripes

Sparse Files

Query Language

Workstation Filesystems: BeFS, NTFS, zfs, ext4
Network Filesystems: NFS, IPFS, glusterfs
Specialized Filesystems: exFAT, ISO-9660

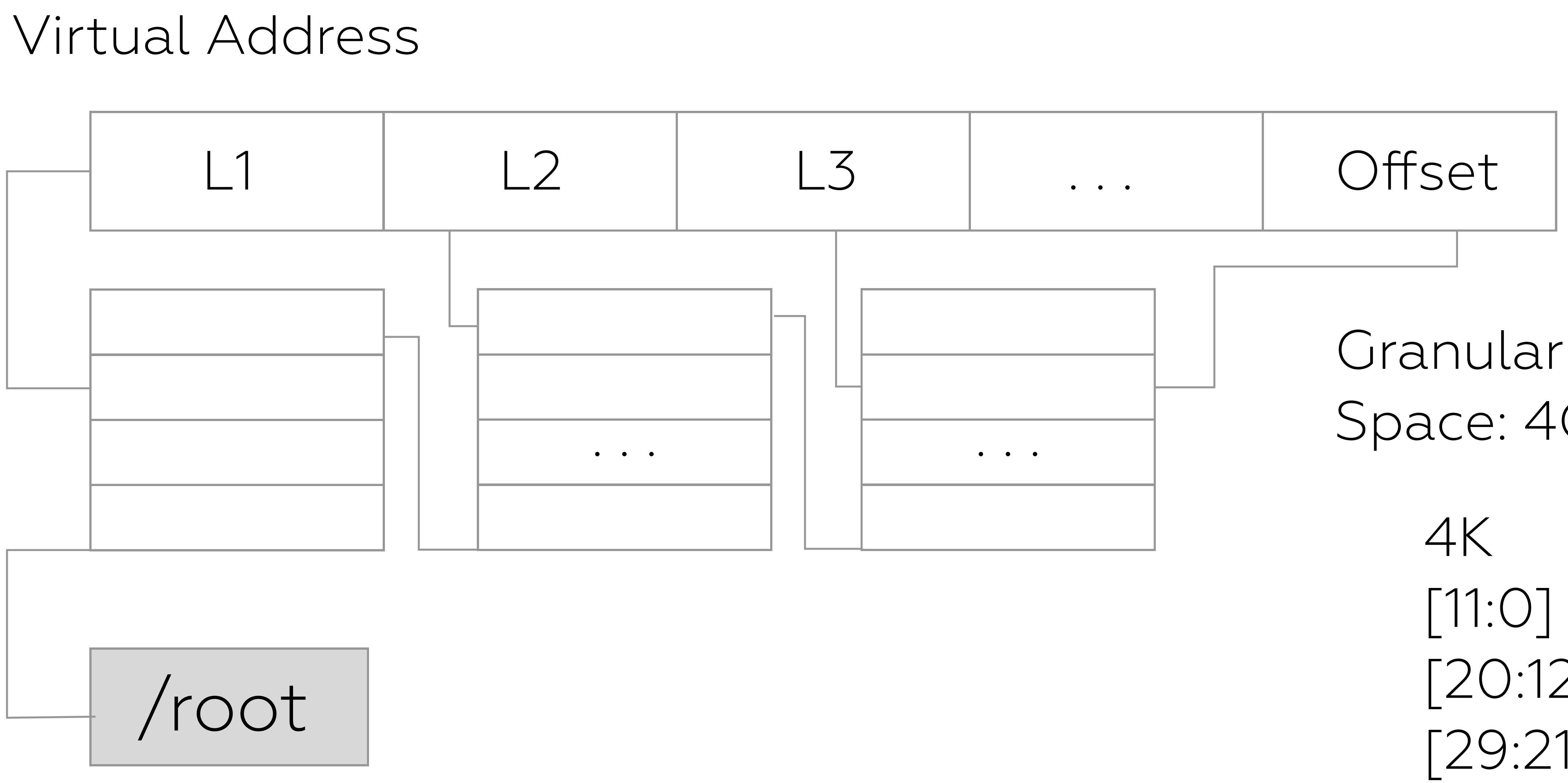
Runtime Libraries

MSVCRT	Windows
CLIB	Linux
MUSL	BSD
Foundation	Mac

- Heap Management
- Input/Output
- Streams/Buffers
- UTF-8/Encodings
- etc.

Memory Management

ARM



Granularity: 4K 16K 64K
Space: 4GB 16GB 1TB 4TB 16TB 256TB 4PB

4K	64K
[11:0] offset	[15:0] in-page offset
[20:12] L3 index	[28:16] L3 index
[29:21] L2 index	[41:29] L2 index
[38:30] L1 index	[47:42] L1 index
[47:39] L0 index	[63] TTBR0
[63] TTBR0	

Memory Management

Intel

Granularity: 4K 2M 4M 1GB
Space: 4GB, 1TB, 4PB, 16EB (~18*10^18)

4K
[31:22] 1024-entry
[21:12] 1024-entry
[11:0] offset

4M
[31:22] 1024 entry
[21:0] offset

LM 1G
[63:48] sign
[47:39] 512-entry
[38:30] 512-entry
[29:0] offset

LM 2M
[63:48] sign
[47:39] 512-entry
[38:30] 512-entry
[29:21] 512-entry
[20:0] offset

LM 4K
[63:48] sign
[47:39] 512-entry
[38:30] 512-entry
[29:21] 512-entry
[20:12] 512-entry
[11:0] offset

PAE 4K
[31:30] 4-entry
[29:21] 512-entry
[20:12] 512-entry
[11:0] offset

PAE 2M
[31:30] 4-entry
[29:21] 512-entry
[20:0] offset

Boot Process

MBR | GPT

BIOS | uBoot | UEFI | OpenFirmware

grub | BCD

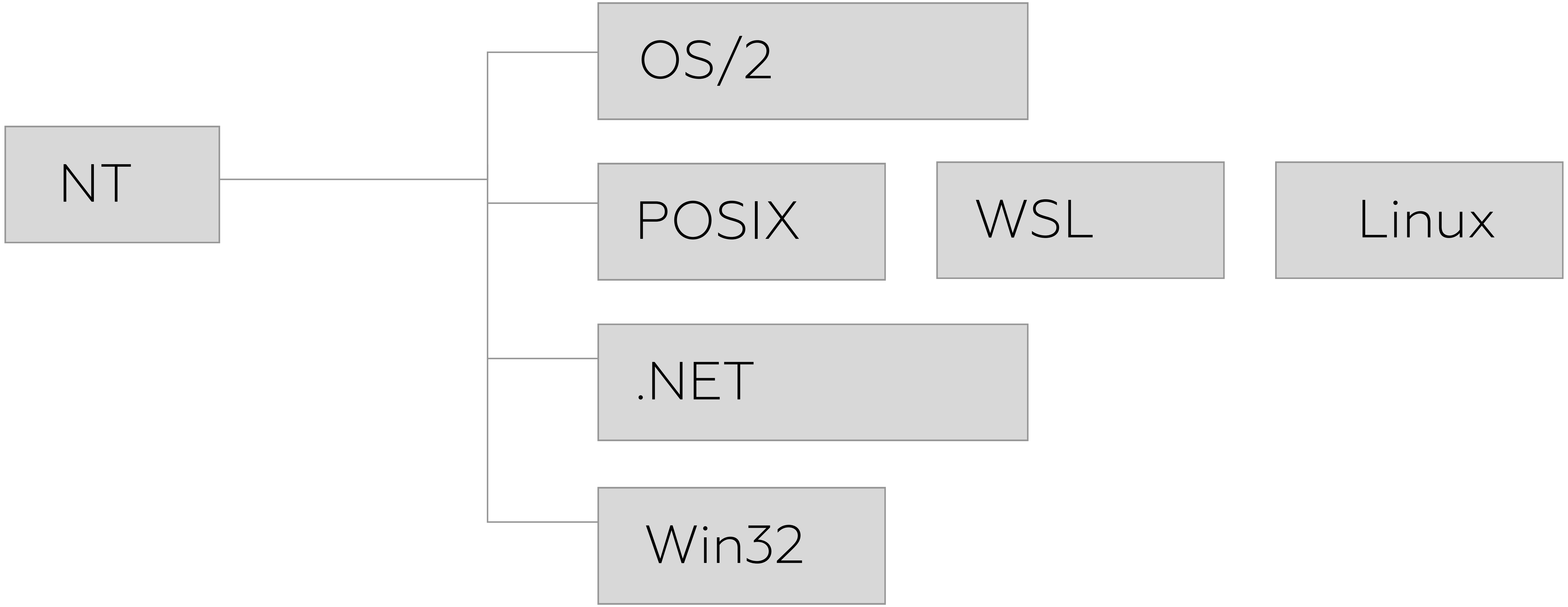
MBR, BIOS: DOS Ages, Partition Tables, Hardware Abstraction Layer
GPT: Modern Partition Table
uBoot: Evaluation Boards, MIPS, ARM32 Devices
OpenFirmware: SGI, Sun, HP, Apple PowerPC
UEFI: Modern HAL

Hardware Programming

DMA copy	OpenCL, CUDA programming
Async I/O	Linux/aio, NtIoCompletionPort
Interrup Packets	Device Drivers
IOCTL	User

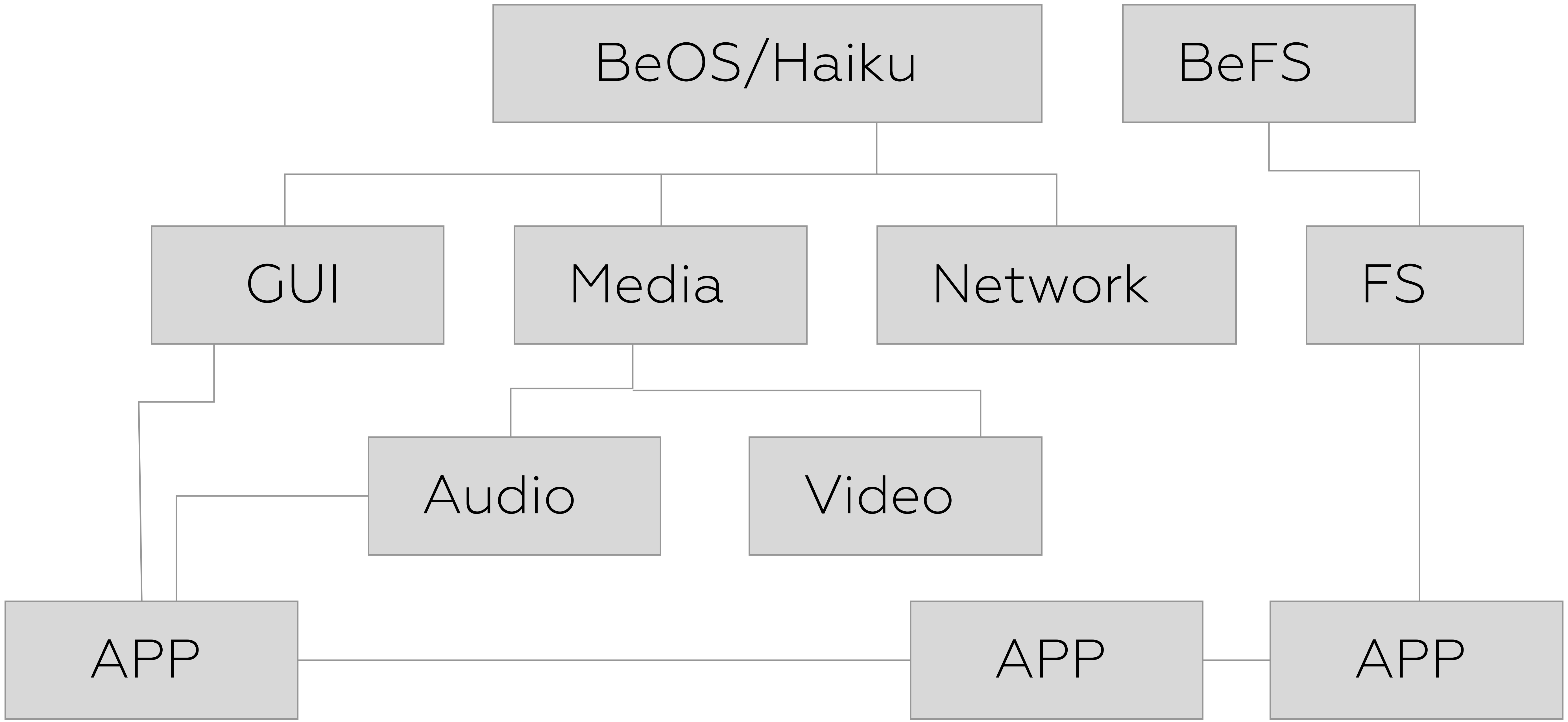
Microkernels Era

Windows NT

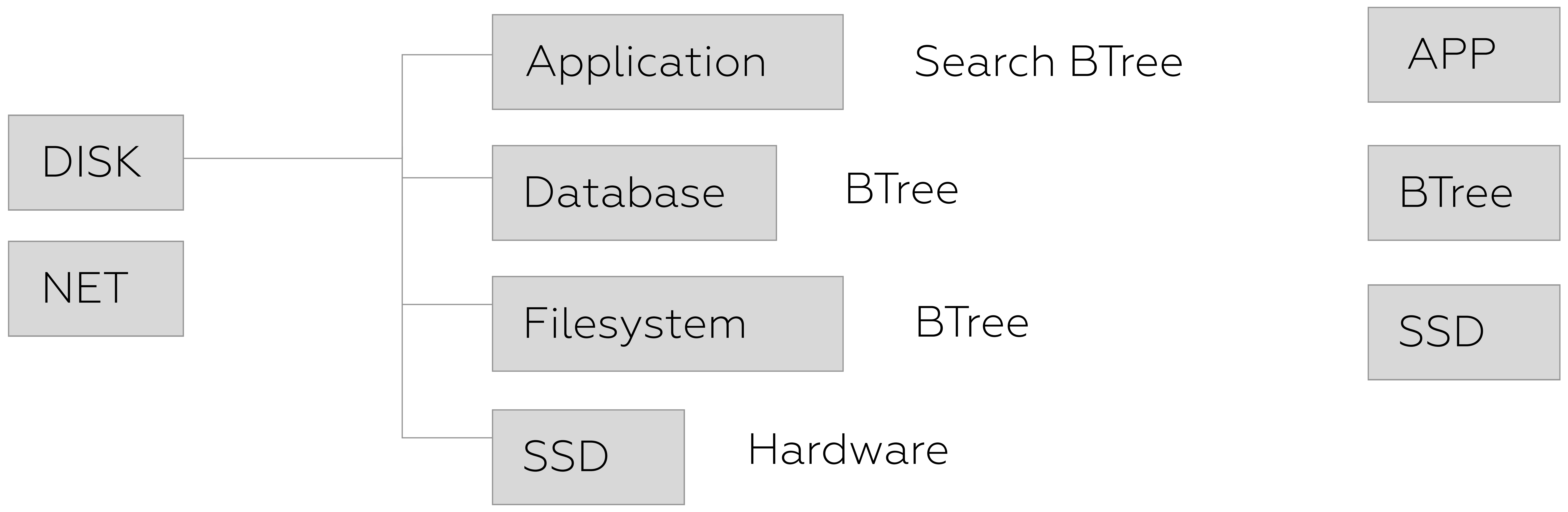


Book. Helen Custer. Inside Windows NT

BeOS



The Problem of Legacy Systems



Improvements

CPU — Avoid Context Switching

MEM — Zero Copy, Garbage Collection Free, CAS, CMPXCHG8B

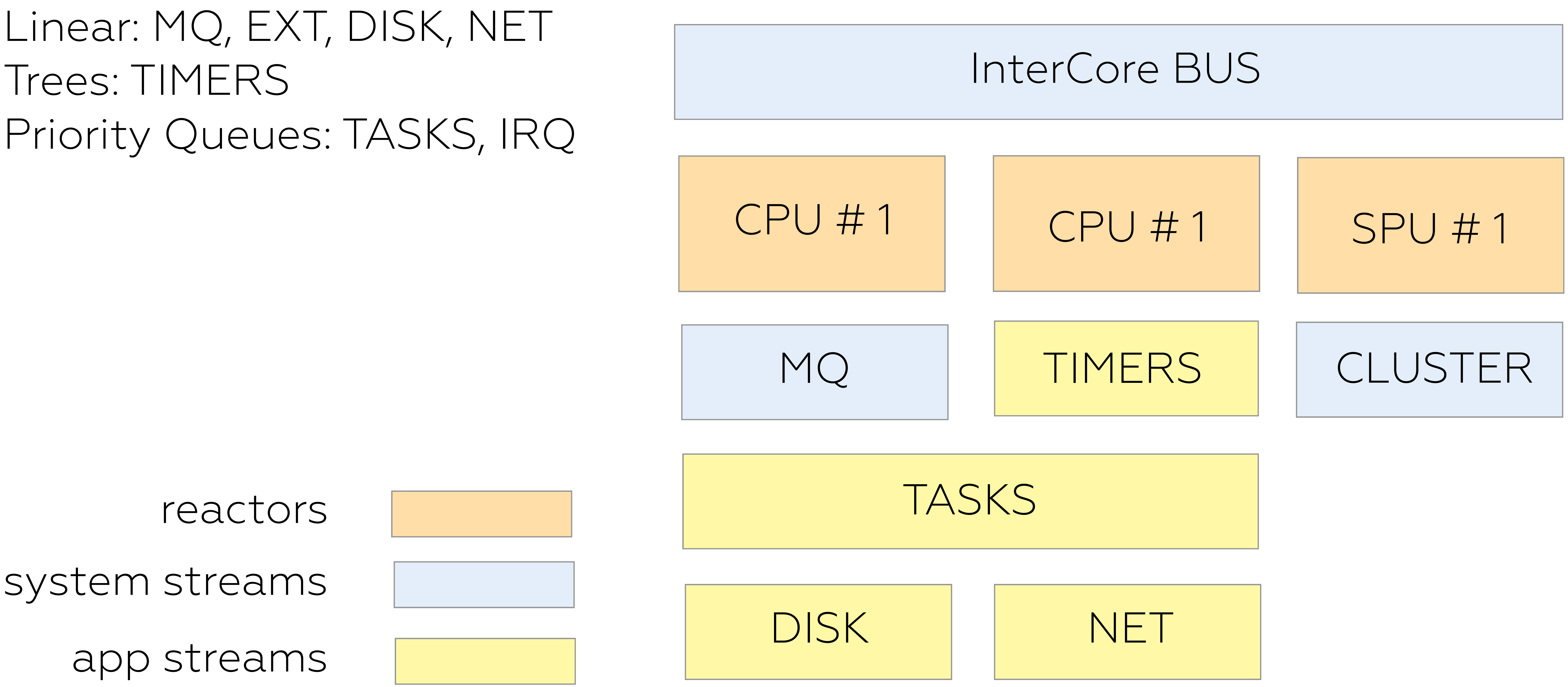
Built-in Language / Yield FSM — LLVM / JIT — Fast Script VM

OS — No Hashtables, Skip Lists instead
No Semaphores and locking primitives
No OS!

Unikernels

Talk Structure

Scheduler: Round-Robin, Priority Queues, Tree Flavours
Scheduler Actors: Features, Timers, Async I/O
Streams Backends: Zero-copy, Message Passing
Linear Backends: Async I/O Disk Streams, Network Streams
Indexed Backends: Timers, Actors
Backpressured Message Bus/Buffers: Arc/Vec prealloc
Class: Low Latency, Real Time


Linear: MQ, EXT, DISK, NET
Trees: TIMERS
Priority Queues: TASKS, IRQ
InterCore BUS
CPU # 1
CPU # 1
SPU # 1
MQ
TIMERS
CLUSTER
TASKS
DISK
NET
reactors
system streams
app streams

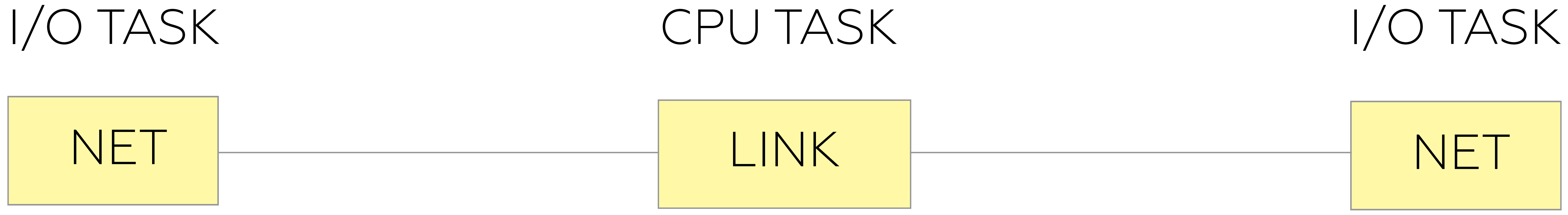
MQ

Queue Types

SPSC/LINK	4-10ns Lowest Latency Possible
MPSC/SUB	10-40ns Reducer or Subscribe Polling
SPMC/PUB	10-40ns Publisher Multicursor

FAST DELIVERY CASE

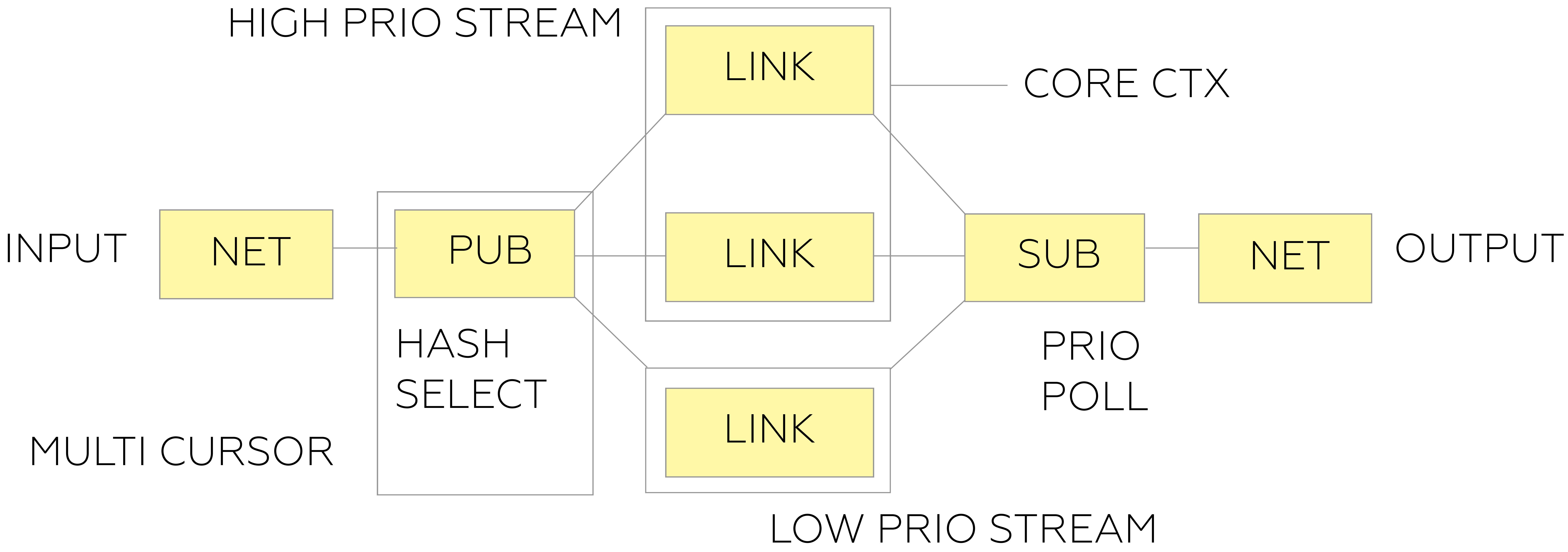
Single Threaded Task Configuration
to be compared as reference



You can use inplace message modifying and reduce copies to unpack and pack.

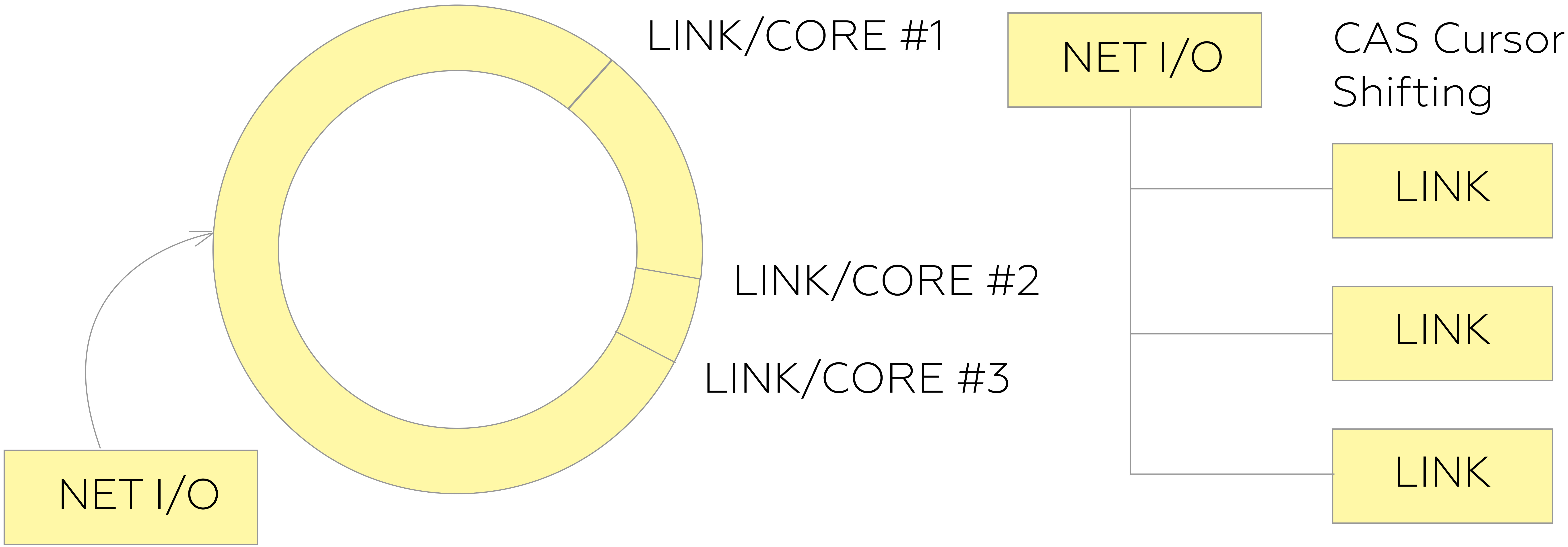
LOAD BALANCING CASE

Load Balancing
of Priority Streams per Core Buckets



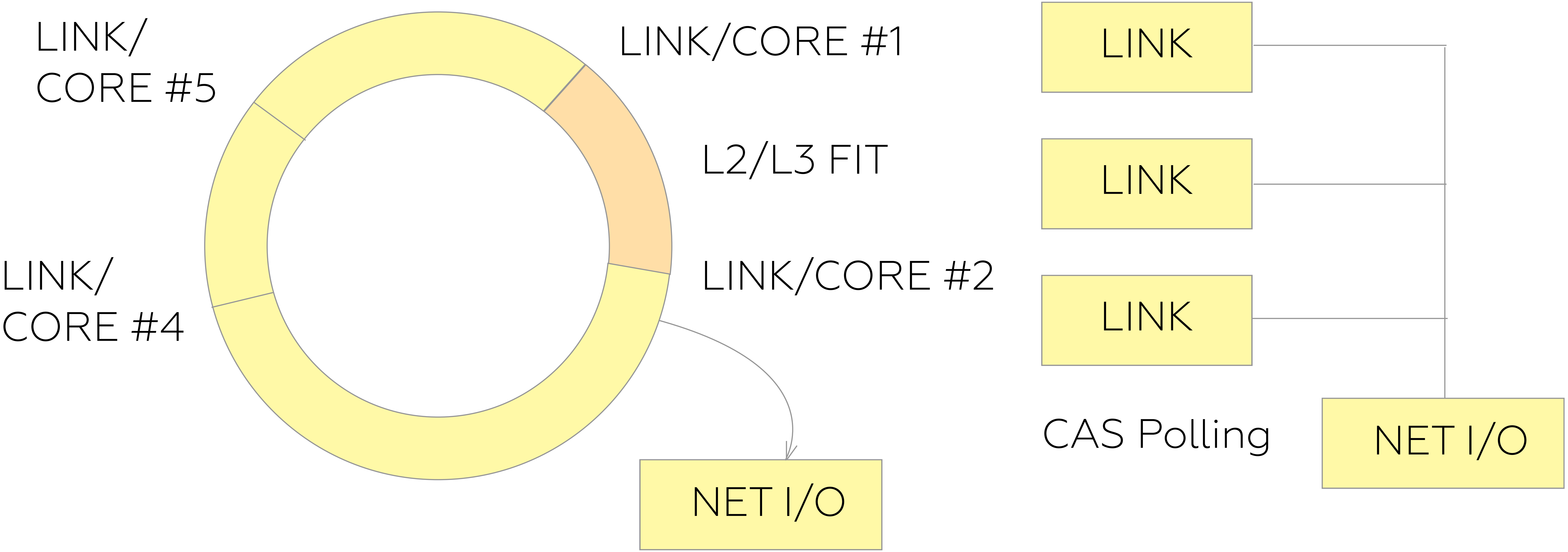
PUBLISHER CASE

PUB Implementation for Zero-Copy
Multiple Consumer Publishing (SPMC)



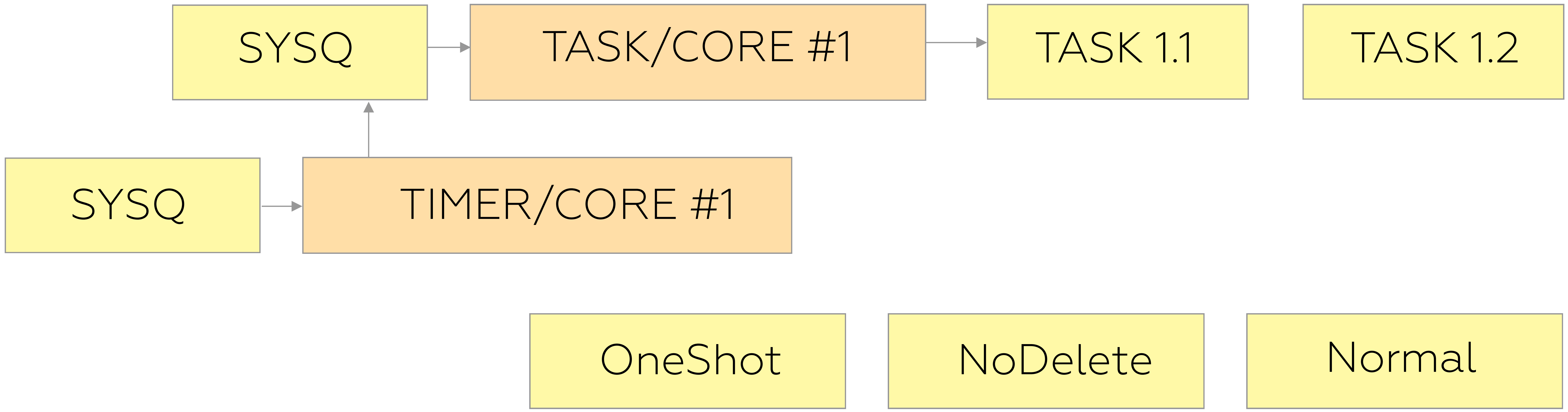
SUBSCRIBER CASE

Multicursor Implementation of SUB (MPSC)
for InterCore Queue Migrations and Cache Locality

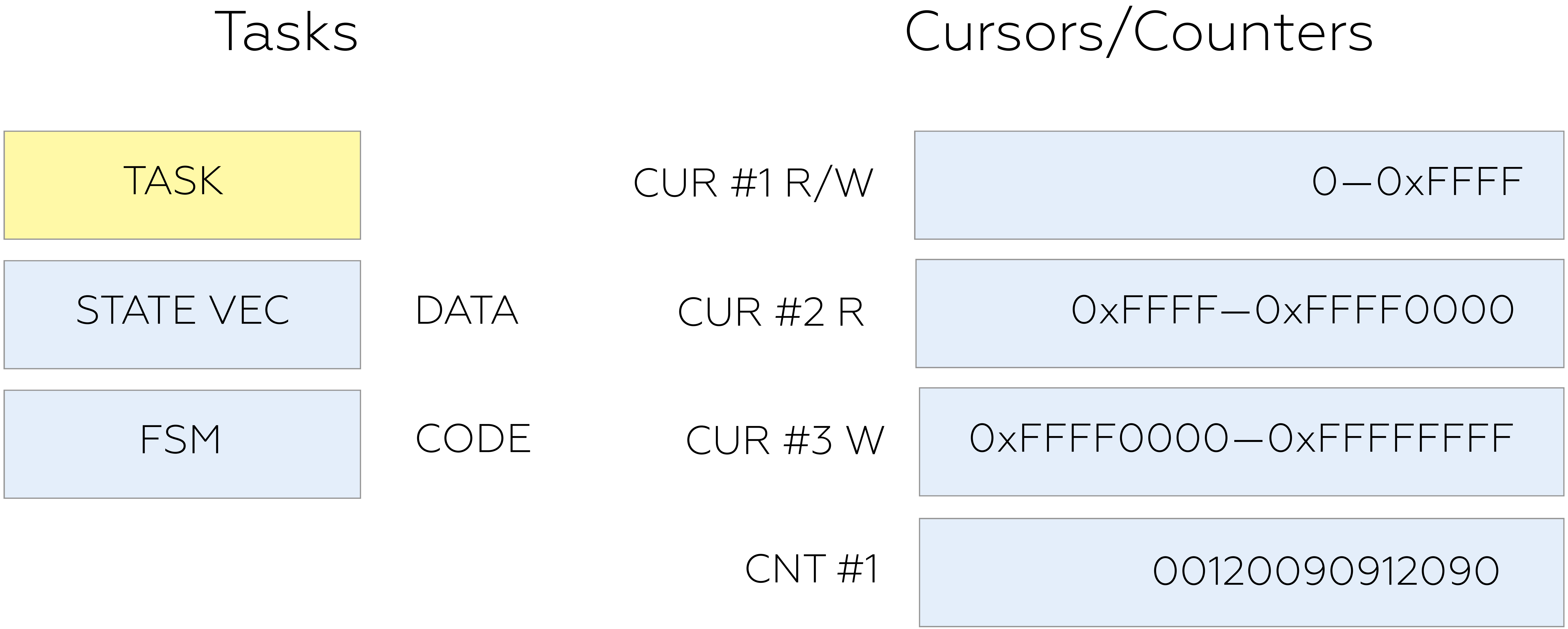


TIMERS

Scheduler Reactors can communicate throught InterCore transport for Timers.



NoDelete Timers use Linear Firing Round Robin of 4 swaps otherwise LogN.



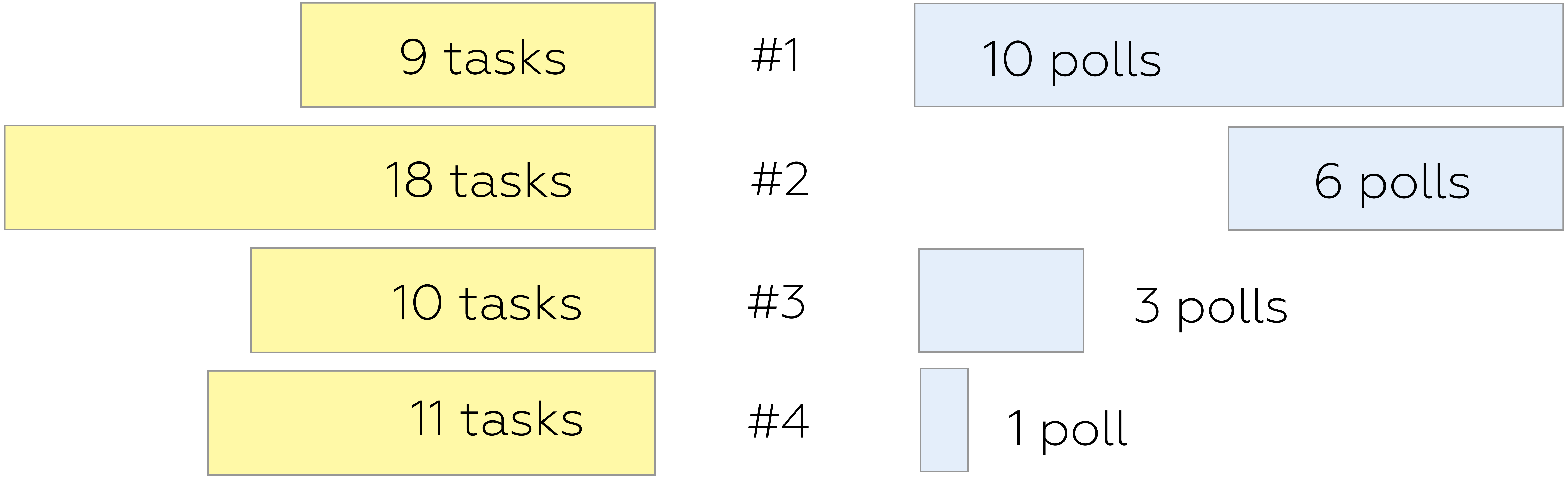
Tasks
Cursors/Counters
TASK
STATE VEC
DATA
FSM
CODE
CUR #1 R/W
0—0xFFFF
CUR #2 R
0xFFFF—0xFFFF0000
CUR #3 W
0xFFFF0000—0xFFFFFFFF
CNT #1
00120090912090

Capacity: 239 Time: 20

Workload: 48 Total: 400

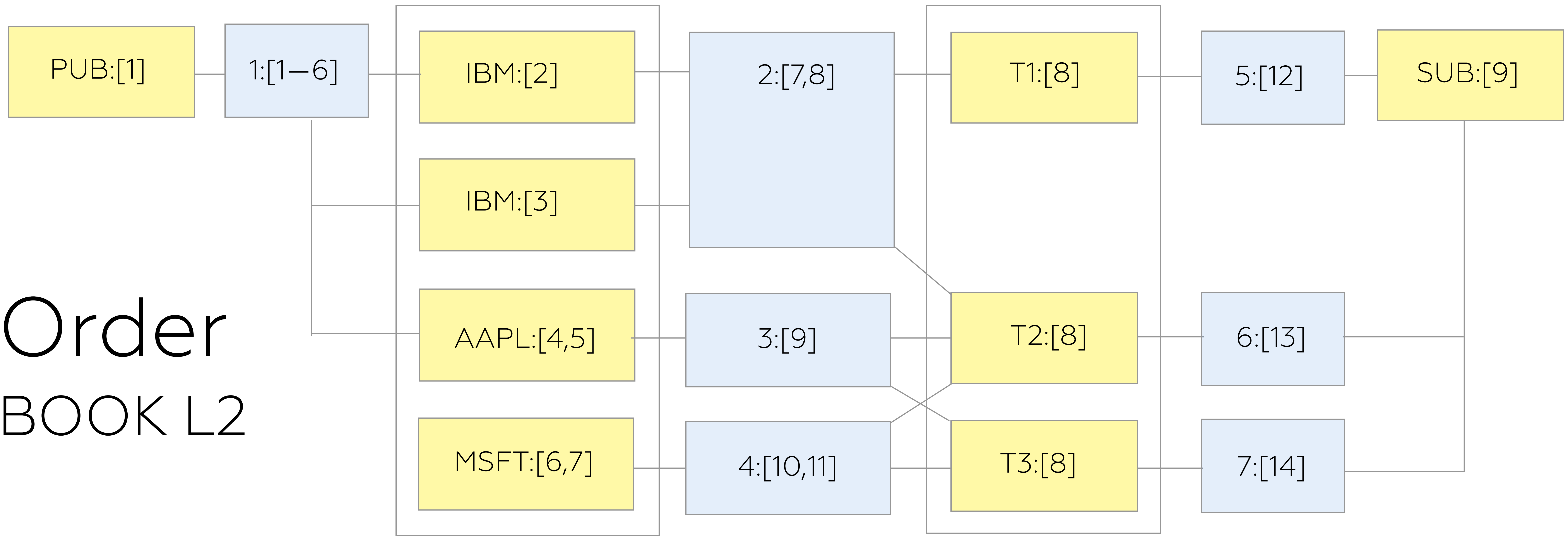
Avg Task Consumtion
Accumulated in the Task Stream

$$\Sigma \text{ Tasks} * \text{Polls} * \text{AvgTime} = \text{Capacity}$$



prios: [10,6,3,1]

12x CPU Cores: In: [1] Order Books: [2,3,4,5,6,7] Traders: [8] Out: [9]



Order
BOOK L2

8x32K MEM Regions: Input Queue: [1] Reducing Queues: [2,3,4,5,6,7]

ITERATORS

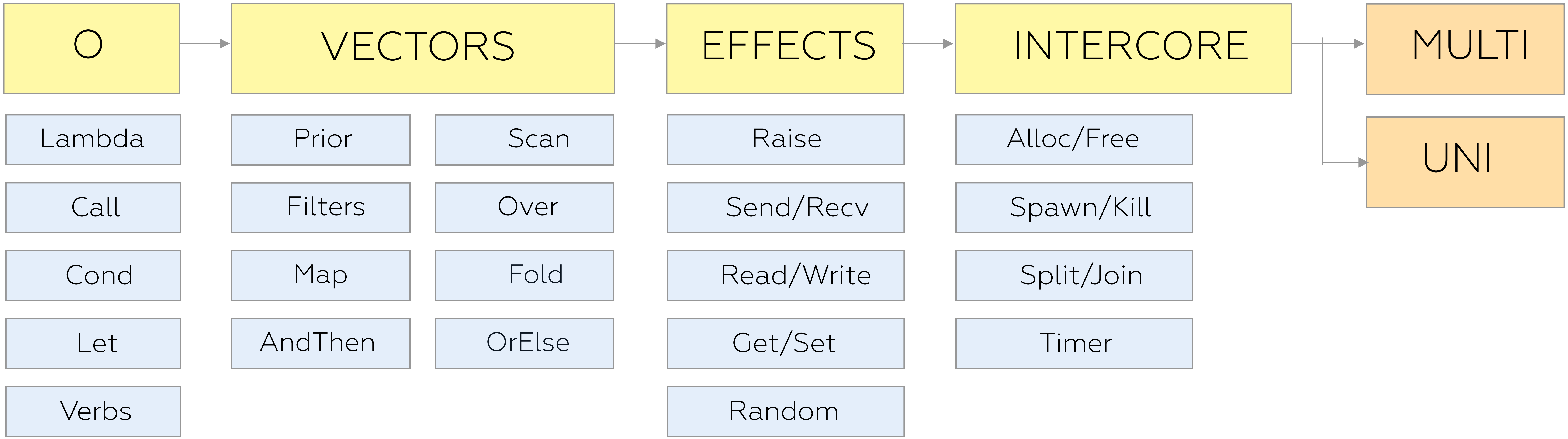
```
+/{x*y}[(1;3;4;5;6);(2;6;2;1;3)]

vec1.iter().zip(vec2.iter()).map(|(i, j)|i * j).sum()
```

```
$ objdump ./target/release/o -d | grep vpmul
   2251c:c5 d5 f4 fb          vpmuludq %ymm3,%ymm5,%ymm7
   22525:c4 41 55 f4 c0       vpmuludq %ymm8,%ymm5,%ymm8
   2253a:c5 d5 f4 db          vpmuludq %ymm3,%ymm5,%ymm3
   22547:c5 cd f4 ec          vpmuludq %ymm4,%ymm6,%ymm5
   22550:c5 cd f4 ff          vpmuludq %ymm7,%ymm6,%ymm7
   22562:c5 cd f4 e4          vpmuludq %ymm4,%ymm6,%ymm4
   22595:c5 d5 f4 fb          vpmuludq %ymm3,%ymm5,%ymm7
```

INTERPRETER

Unified Combinators of Language and Streams
Interpretation for Unicore and Multicore



The motivation is to keep LLVM vectorizer continuous happy