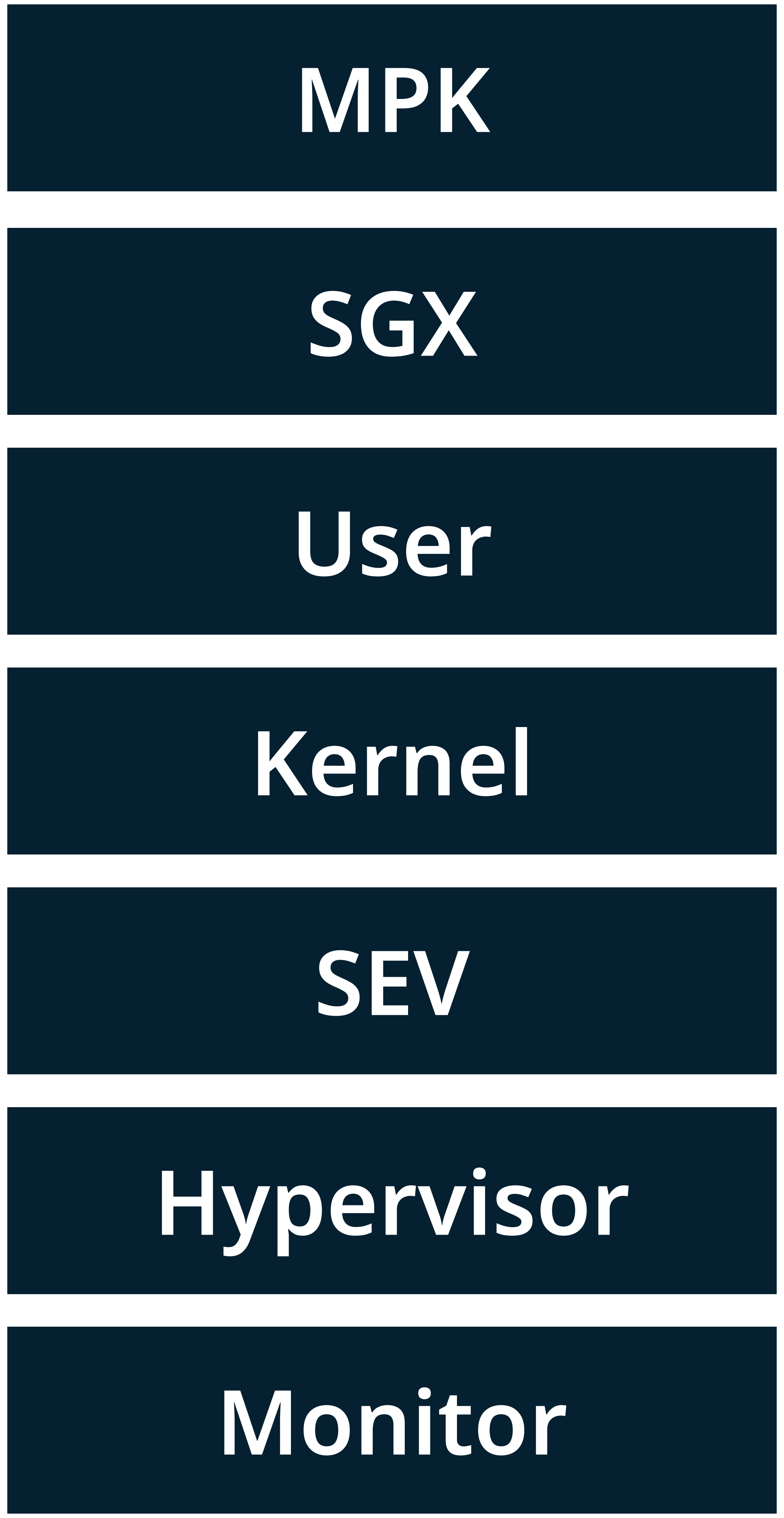
Software-Defined CPU Modes

Michael Roitzsch, Till Miemietz, Christian von Elm, Nils Asmussen

So Many CPU Modes



MPK

SGX

User

Kernel

SEV

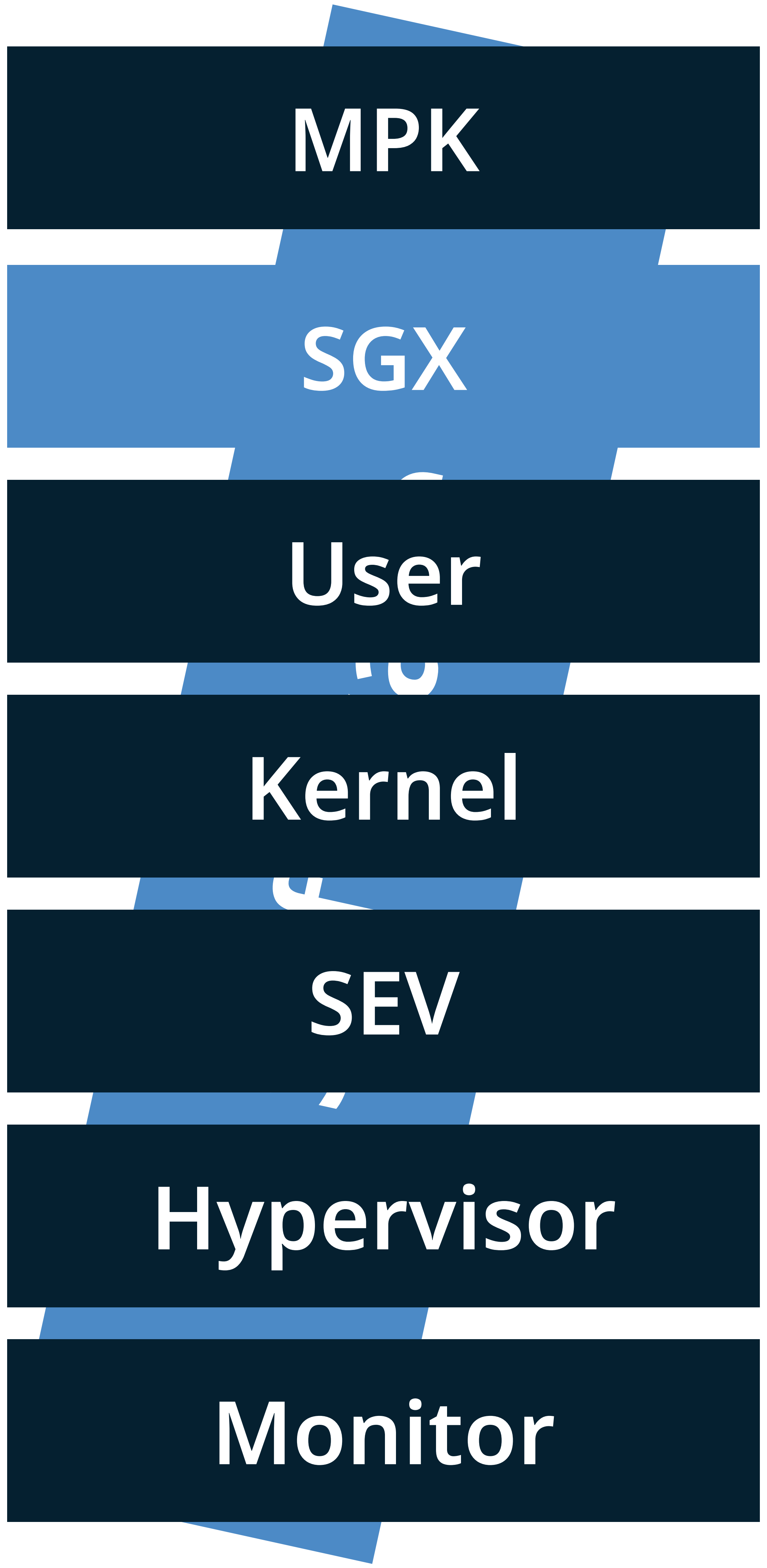
Hypervisor

Monitor

So Many CPU Modes

What if …

CPU modes were programmable?



MPK

SGX

User

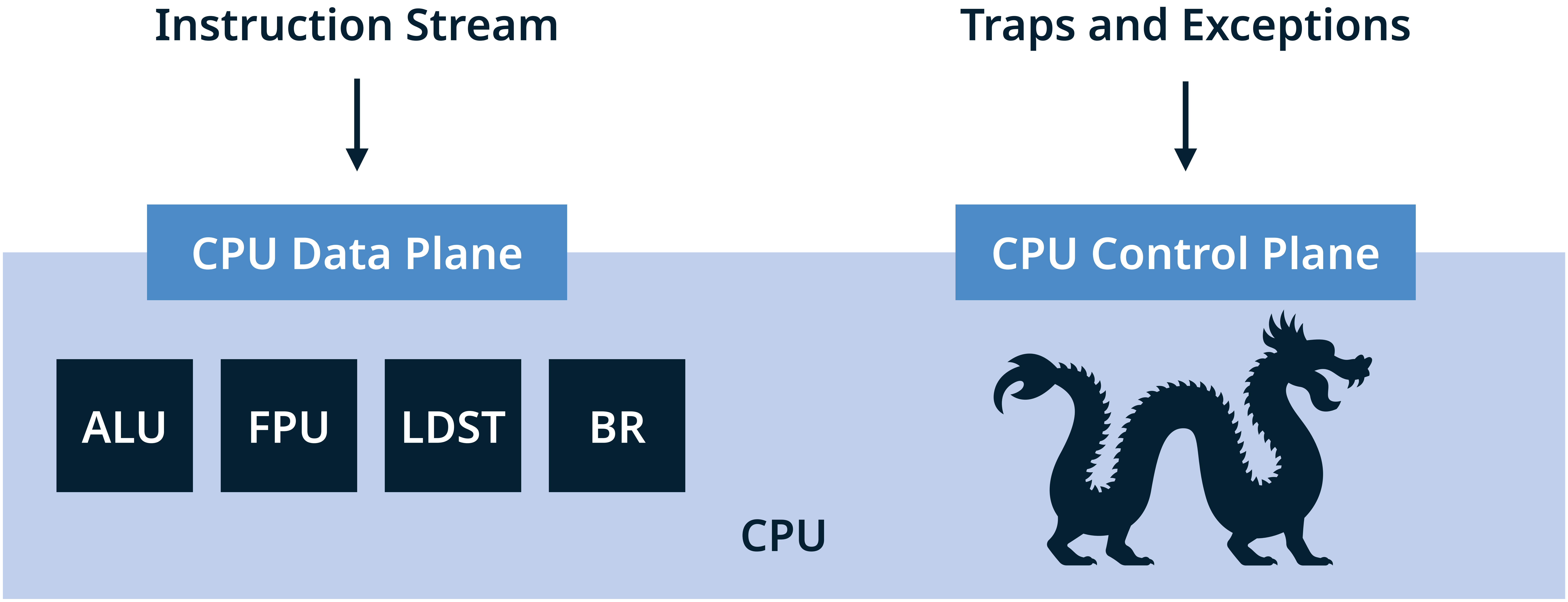
Kernel

SEV

Hypervisor

Monitor

CPU Programmability



Instruction Stream

Traps and Exceptions

CPU Data Plane

CPU Control Plane

ALU

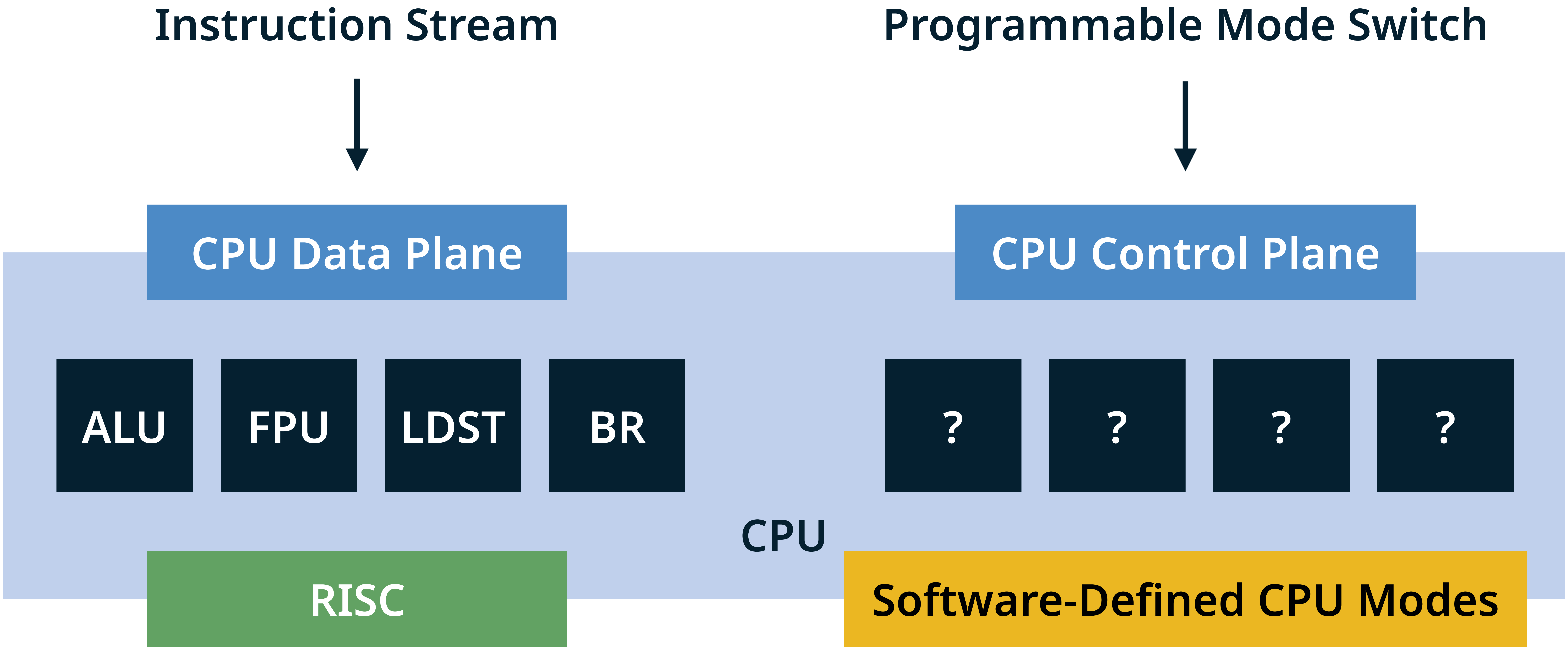
FPU

LDST

BR

CPU

CPU Programmability



Instruction Stream

Programmable Mode Switch

CPU Data Plane

CPU Control Plane

ALU

FPU

LDST

BR

?

?

?

?

CPU

RISC

Software-Defined CPU Modes

Why?

- **Type II hosted hypervisors:** nest guest kernel/user in host user mode
- **Dune:** paging control and dirty bits access can help GC performance
- **SPDK/DPDK:** lightweight isolation when sharing devices
- **Custom in-app sandboxes:** run JIT code with write-xor-execute

Steps of a Mode Transition

mode switch triggered by instruction like sysenter or side effect

reconfigure trigger behavior of instructions

reconfigure memory access permissions

transfer selected state of the exited mode to the entered mode

Mode Configuration Table

parent-call

current

child-call

ID	Parent ID	Entry Point	Trap Behavior
0	—	0x8000	00000000
1	0	0xAA00	00100010
2	1	0x1230	01100010
3	2	0xFF40	01100011

kernel

parent-return

user

sandbox

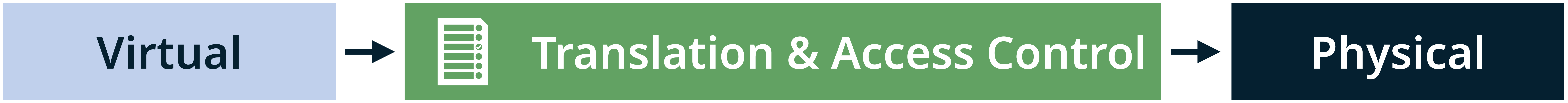
child-return

Mode Configuration Table

ID	Parent ID	Entry Point	Trap Behavior	MemPerm Root
0	—	0x8000	00000000	0xD000
1	0	0xAA00	00100010	0xEB00
2	1	0x1230	01100010	0x3210
3	2	0xFF40	01100011	0x4040

Mode Configuration Table

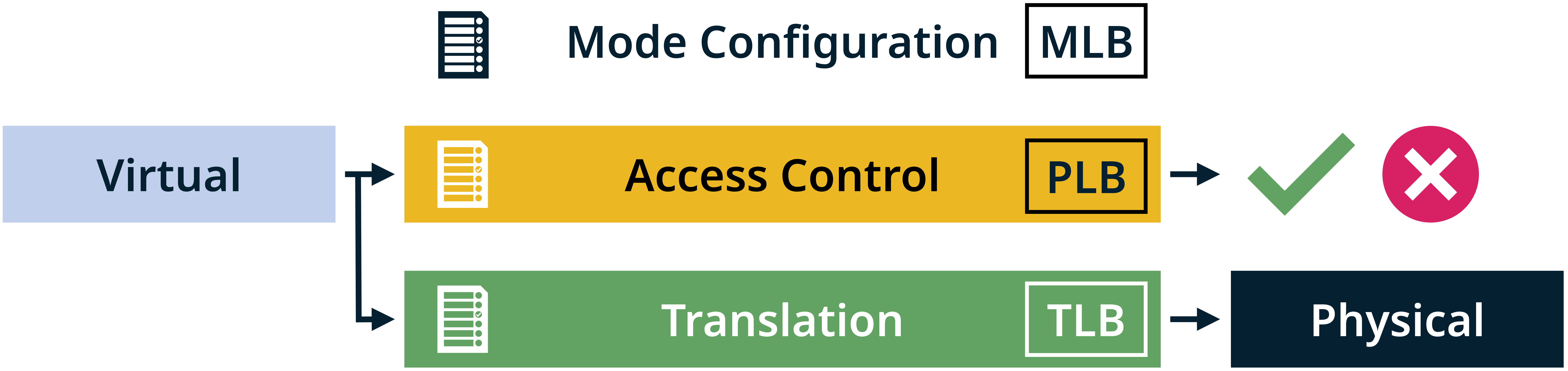
ID	Parent ID	Entry Point	Trap Behavior	MemPerm Root
0	—	0x8000	00000000	0xD000
1	0	0xAA00	00100010	0xEB00
2	1	0x1230	01100010	0x3210
3	2	0xFF40	01100011	0x4040



Virtual → Translation & Access Control → Physical

Mode Configuration Table

ID	Parent ID	Entry Point	Trap Behavior	MemPerm Root
0	—	0x8000	00000000	0xD000
1	0	0xAA00	00100010	0xEB00
2	1	0x1230	01100010	0x3210
3	2	0xFF40	01100011	0x4040



Mode Transition Programmability

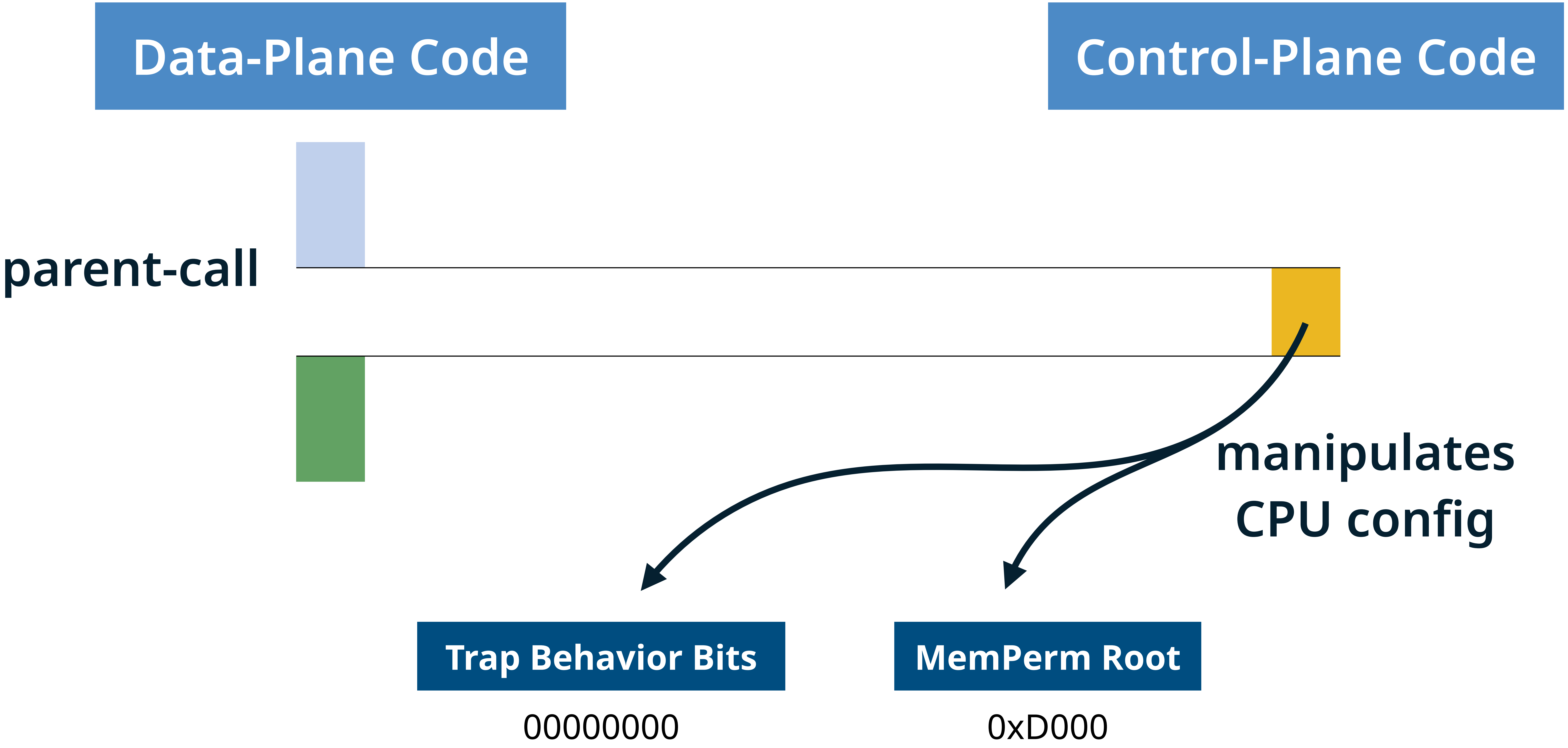
mode switch triggered by instruction like sysenter or side effect

reconfigure trigger behavior of instructions

reconfigure memory access permissions

transfer selected state of the exited mode to the entered mode

Control-Plane Code



Data-Plane Code

Control-Plane Code

parent-call

manipulates CPU config

Trap Behavior Bits

00000000

MemPerm Root

0xD000

Where to run the Control-Plane Code?

Idea 1: Control-plane processor

Idea 2: Mode switch mode

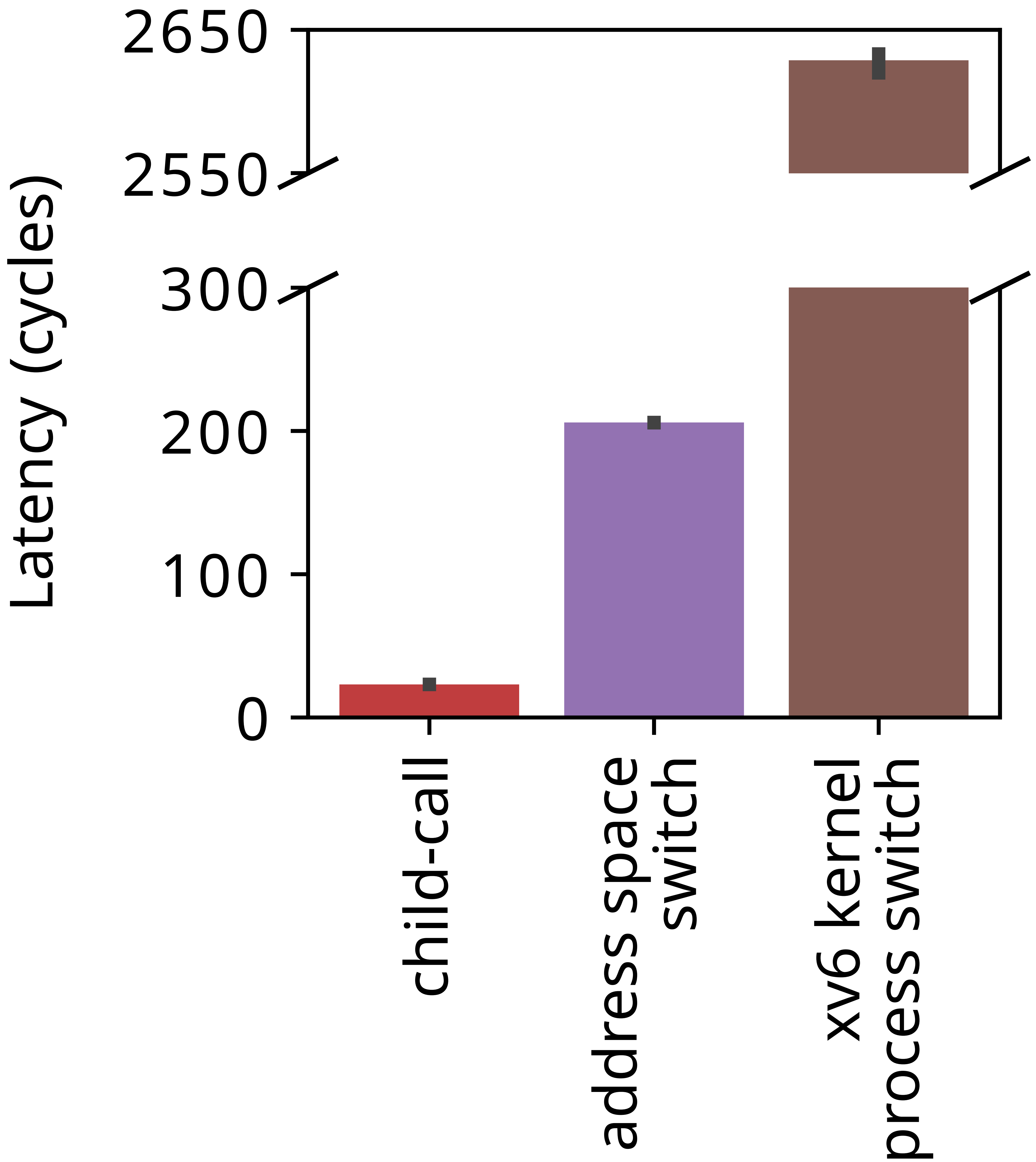
… it’s the last mode you’ll ever need

Summary and Discussion

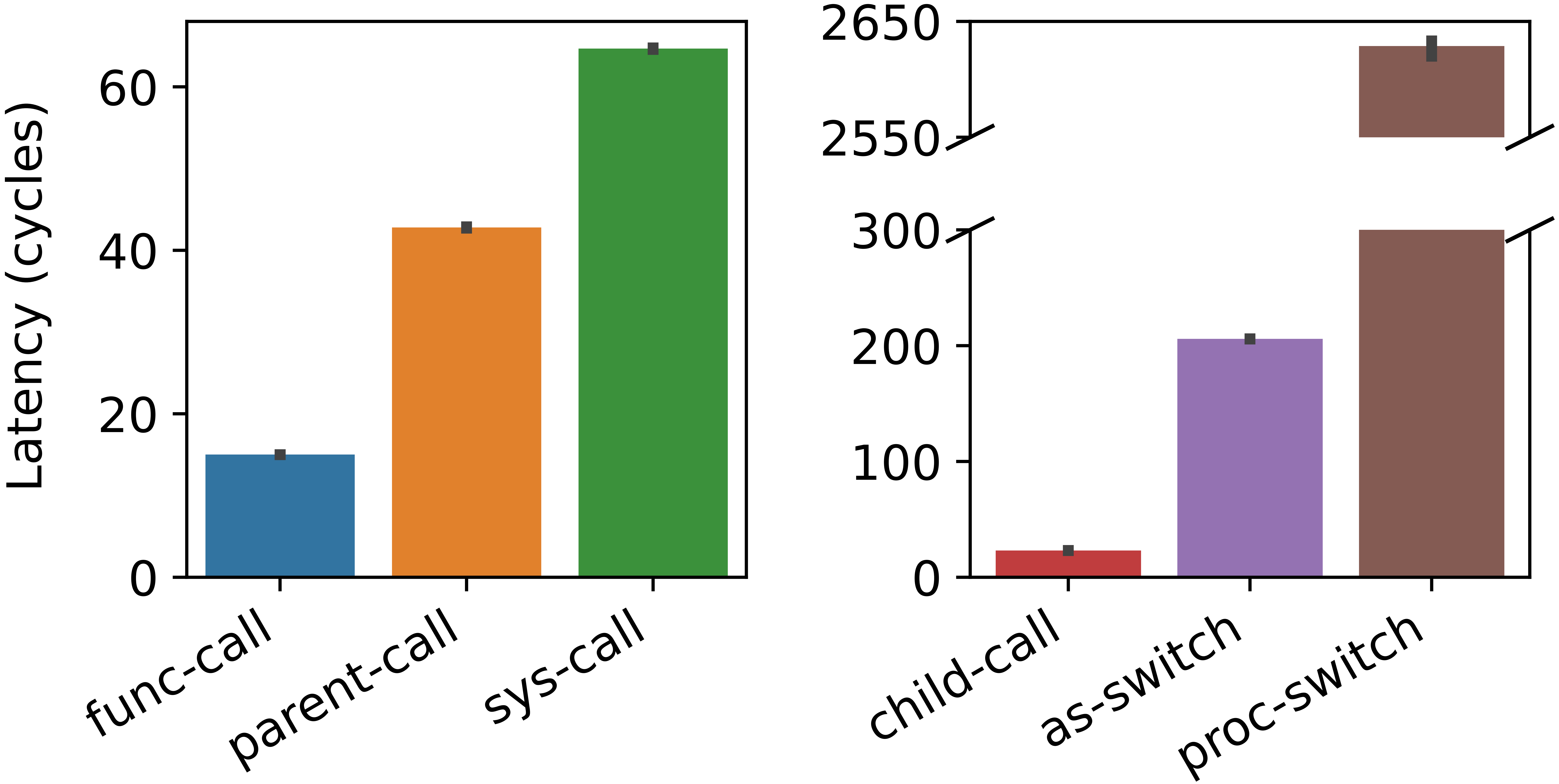
software-defined CPU modes:
useful and feasible

Open Questions

- memory protection: data structure, relationship to ISA-capabilities?
- security and complexity: more openness, API complexity?
- flexibility and performance: additional energy cost of MLB/PLB?
- compiler integration: modes as part of programming models



Evaluation



Latency (cycles)
0
20
40
60
func-call
parent-call
sys-call
2650
2550
300
200
100
0
child-call
as-switch
proc-switch