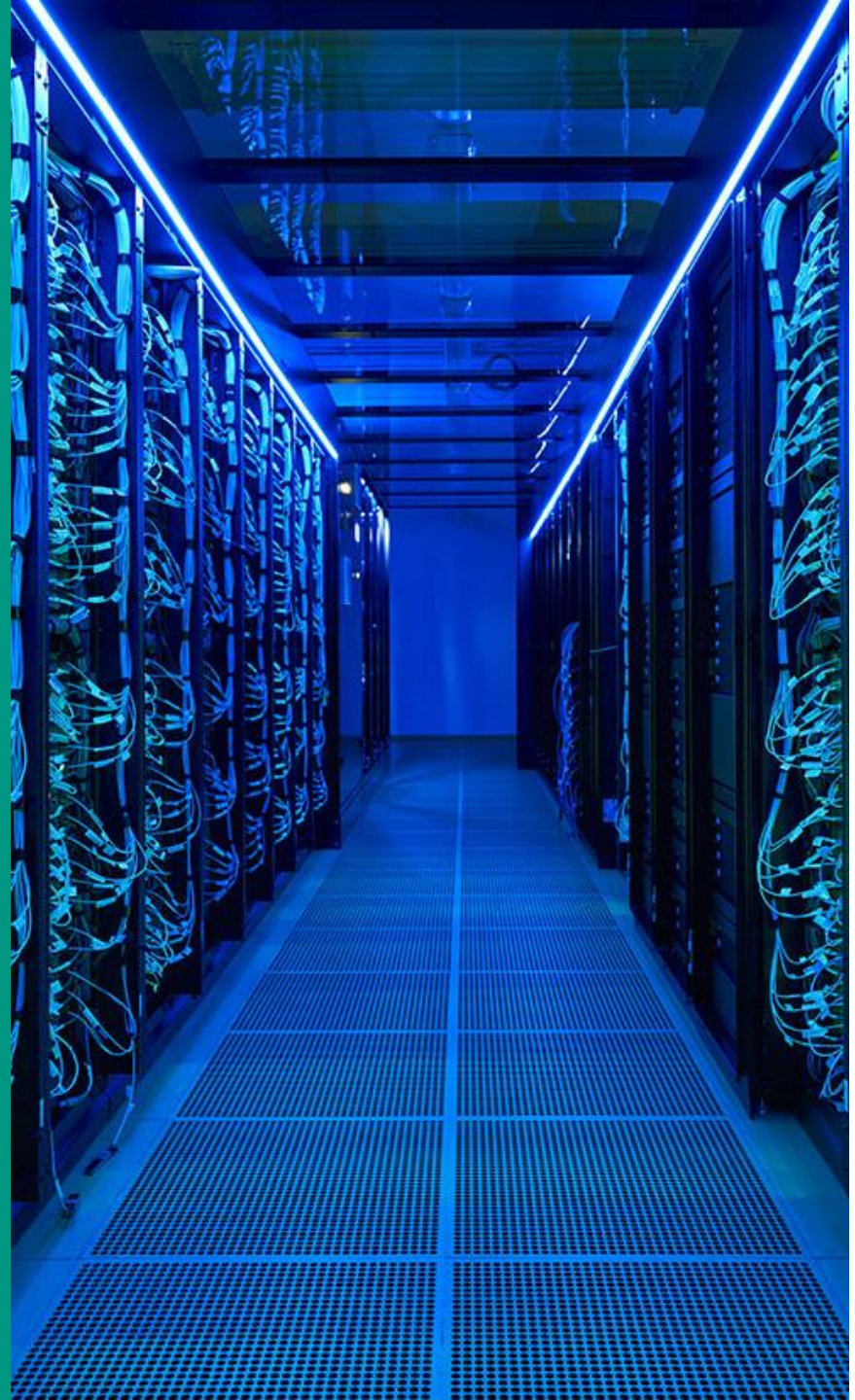
Transparent DAX Mappings

Towards Automatic Kernel Bypass with CXL-Based Hybrid SSDs

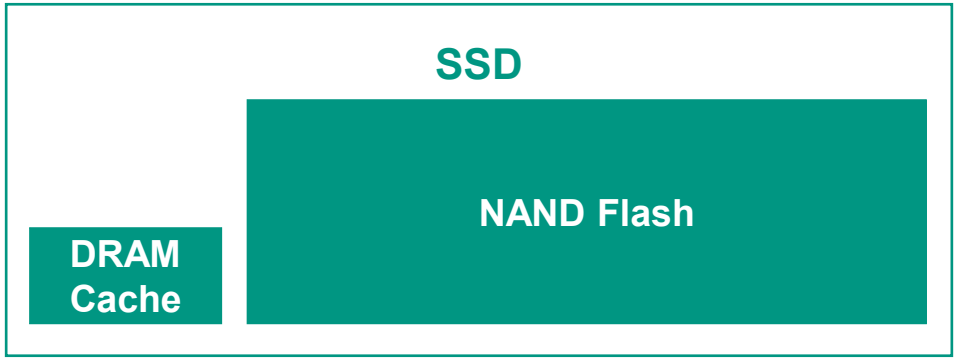
Yussuf Khalil, Daniel Habicht, Pascal Ellinger, Frank Bellosa, Javier González, Adam Manzanares, Vivek Shah



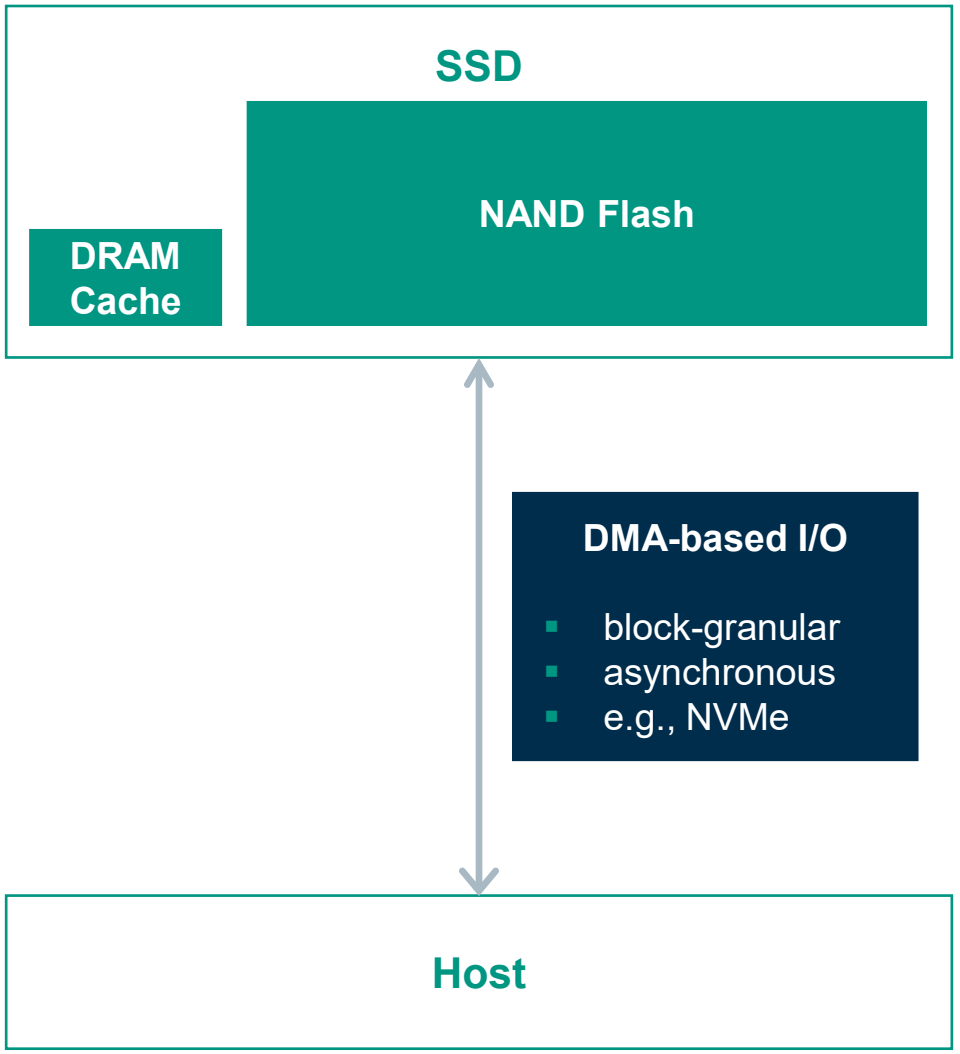
KIT

SAMSUNG

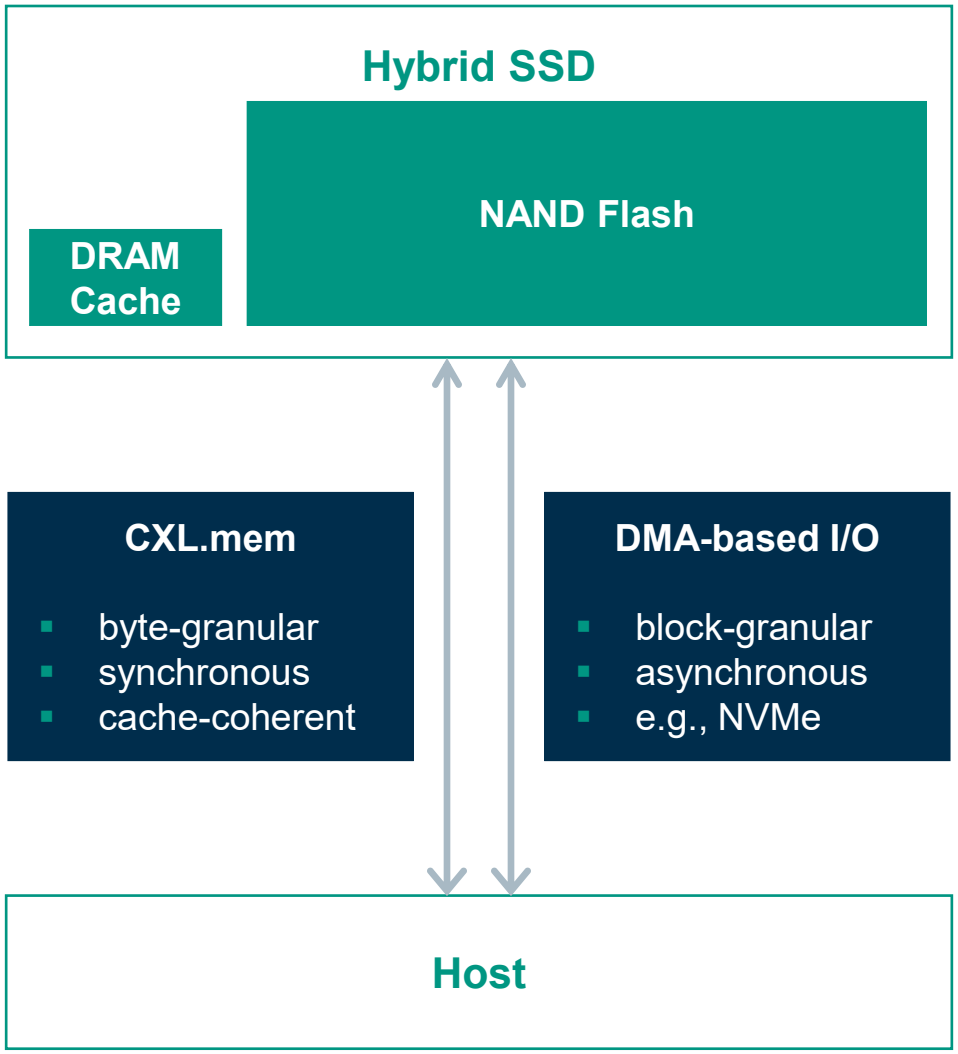
What Is a Hybrid SSD?



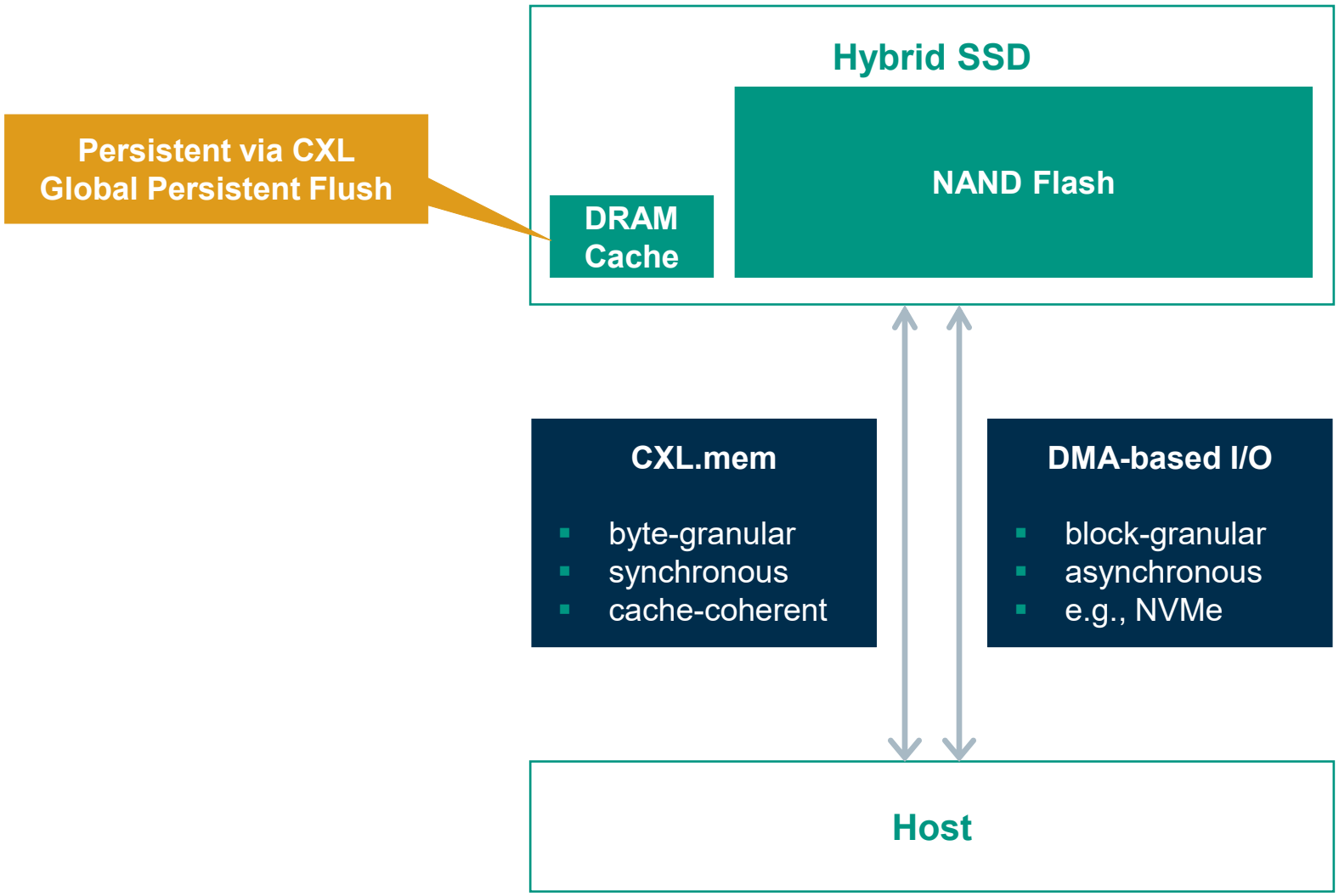
What Is a Hybrid SSD?



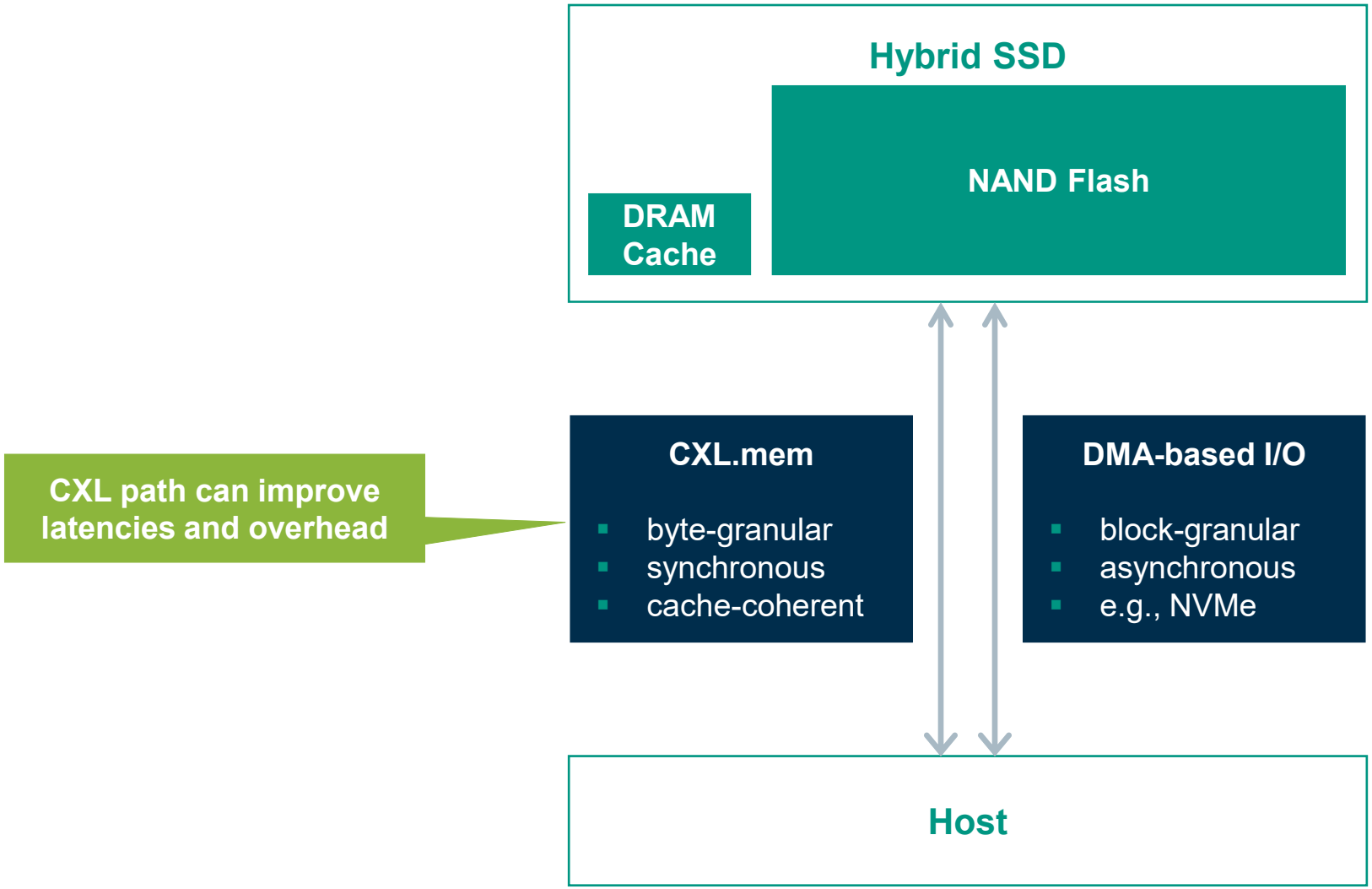
What Is a Hybrid SSD?



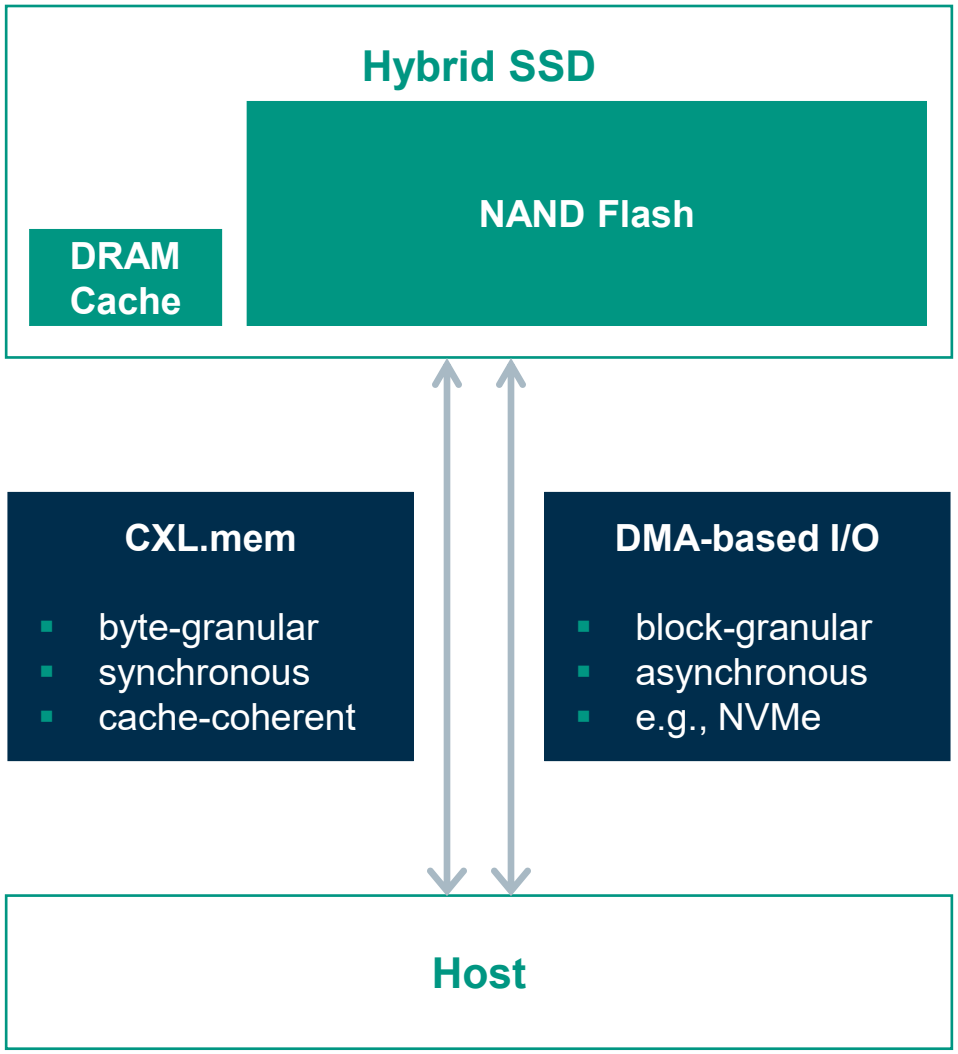
What Is a Hybrid SSD?



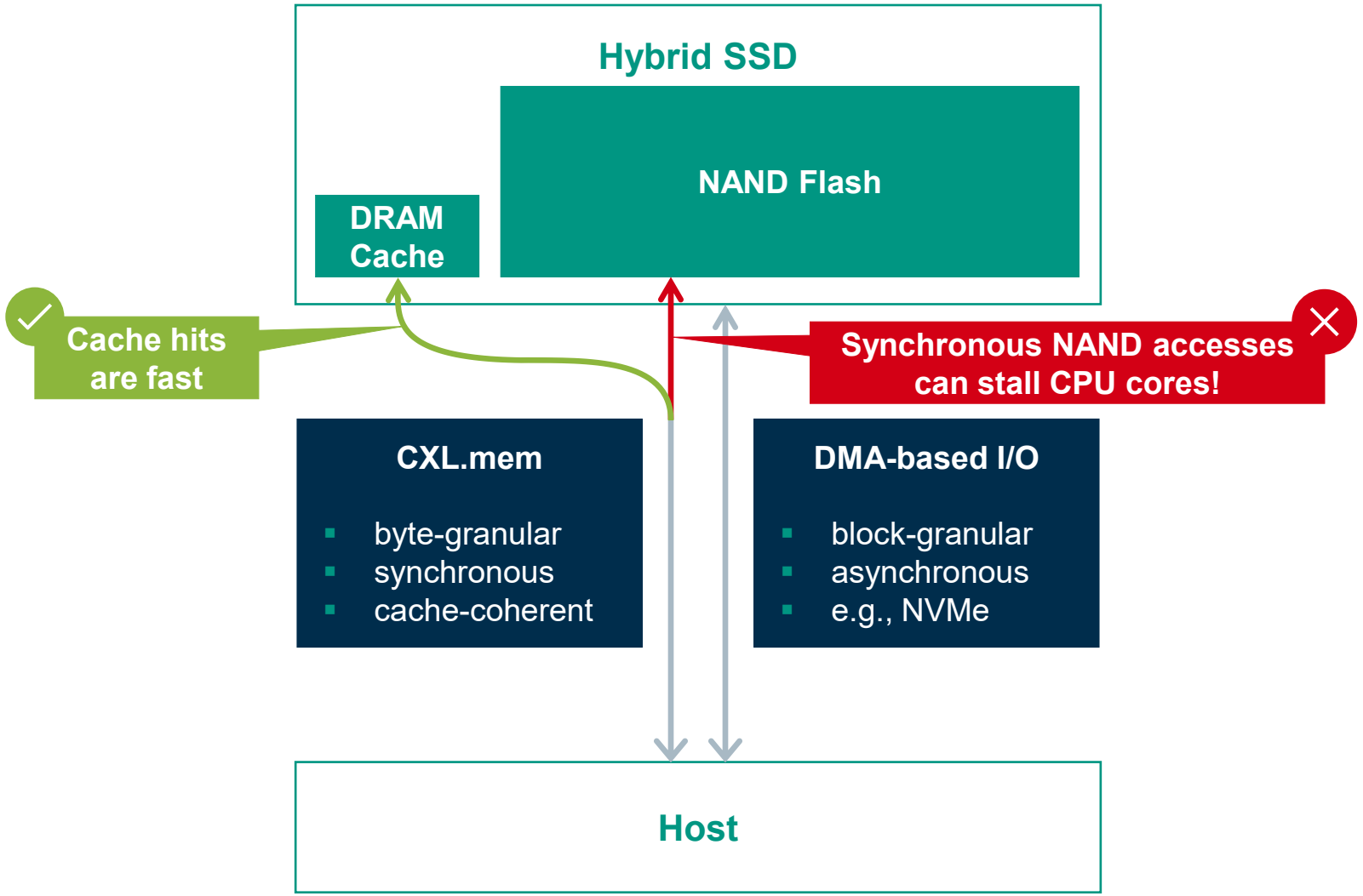
What Is a Hybrid SSD?



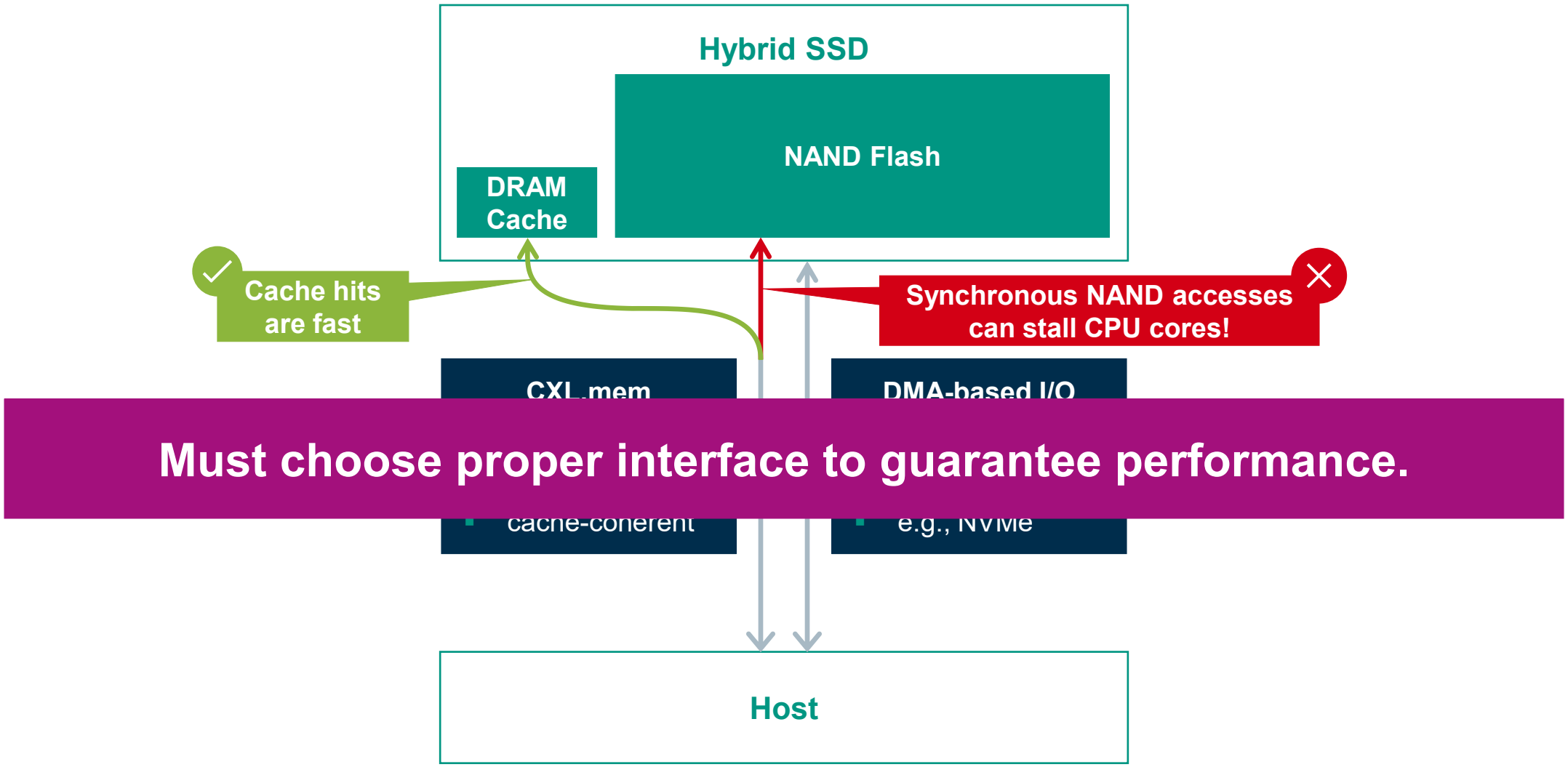
What Is the Problem?



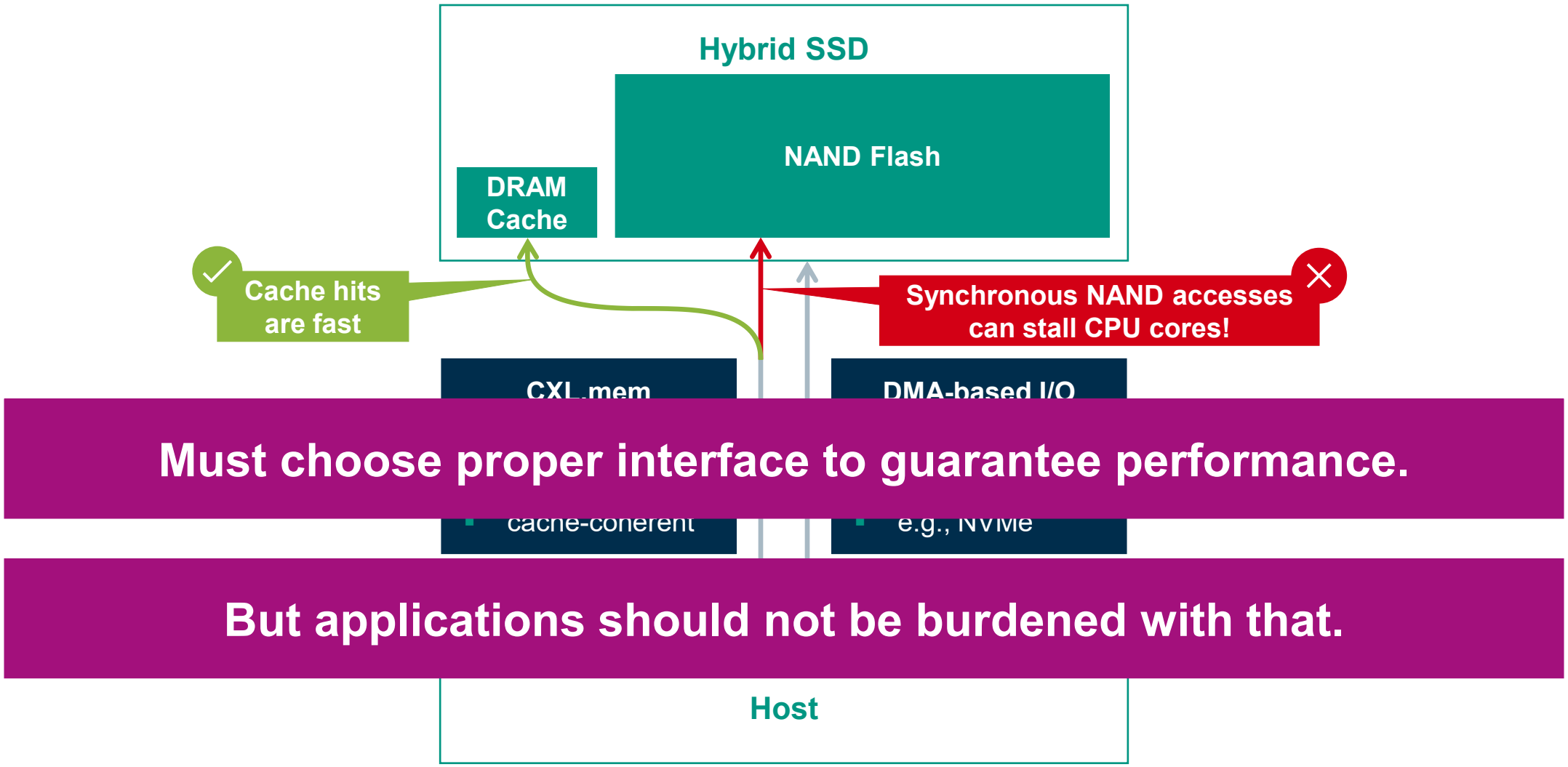
What Is the Problem?



What Is the Problem?



What Is the Problem?



Goals for Our OS Mechanism

Goals for Our OS Mechanism

Performance (or energy) benefit with traditional POSIX read()/write()

Goals for Our OS Mechanism

- Performance (or energy) benefit with traditional POSIX read()/write()
- Performance never worse than with NVMe

Goals for Our OS Mechanism

- Performance (or energy) benefit with traditional POSIX read()/write()
- Performance never worse than with NVMe
- Simple to use, no additional developer effort required

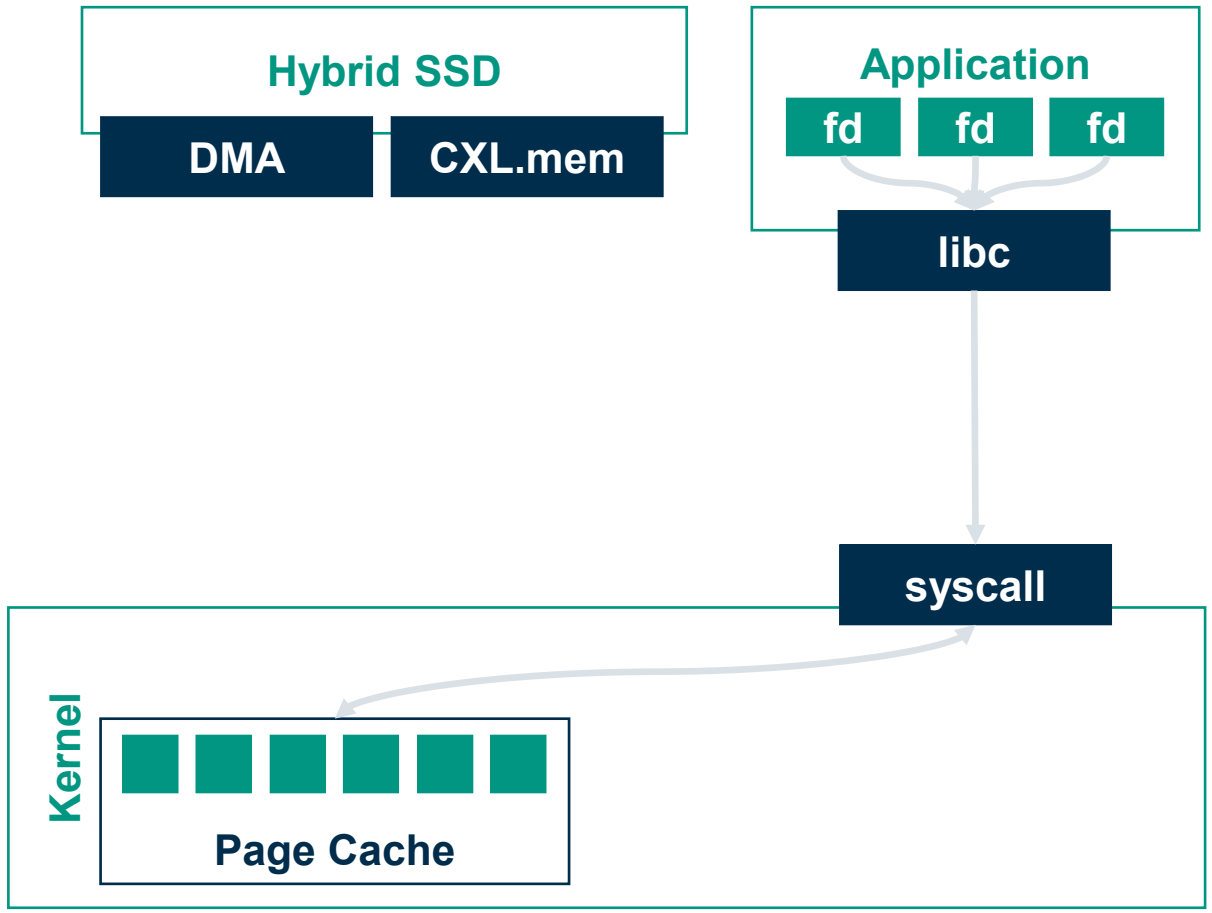
Goals for Our OS Mechanism

- Performance (or energy) benefit with traditional POSIX read()/write()
- Performance never worse than with NVMe
- Simple to use, no additional developer effort required

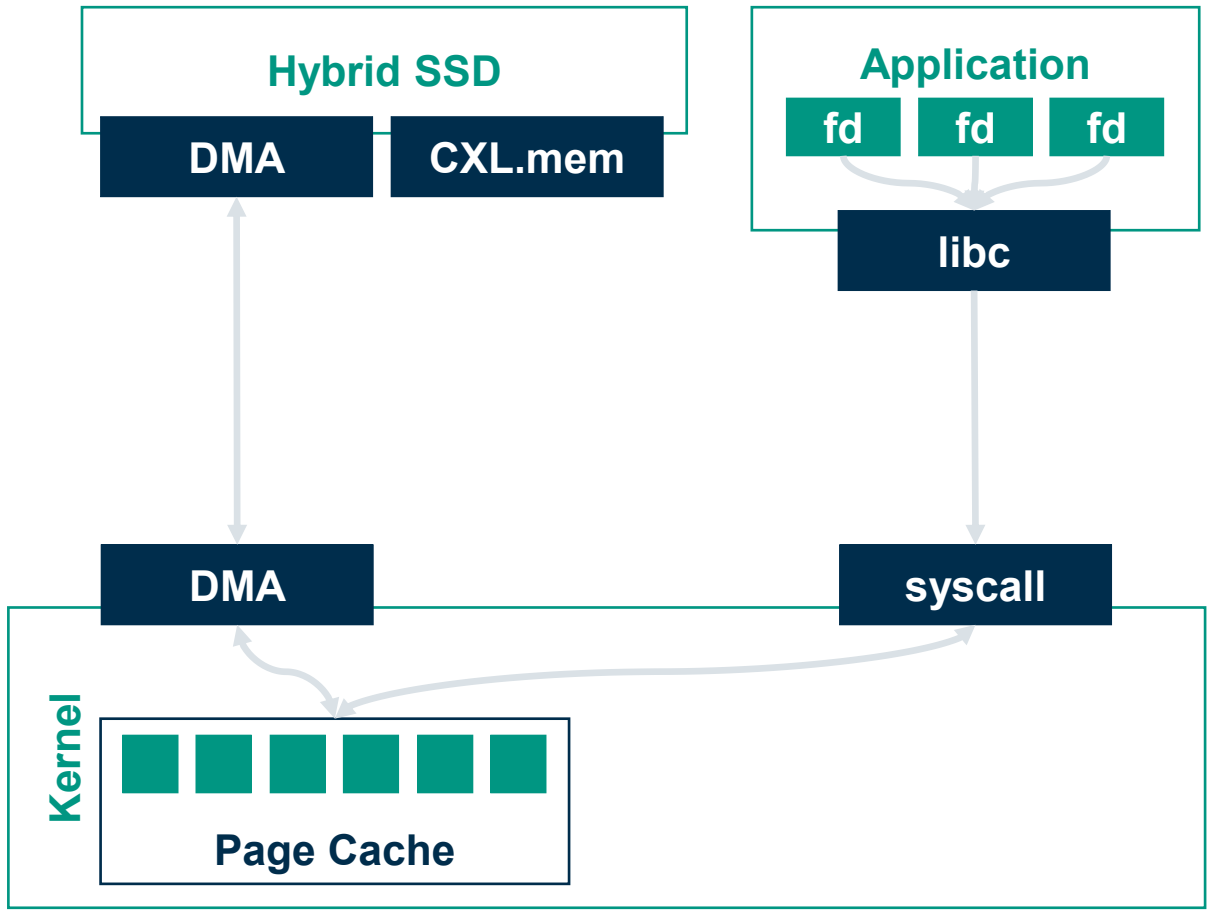
Make hybrid SSDs a drop-in replacement for NVMe SSDs.

Design

Design

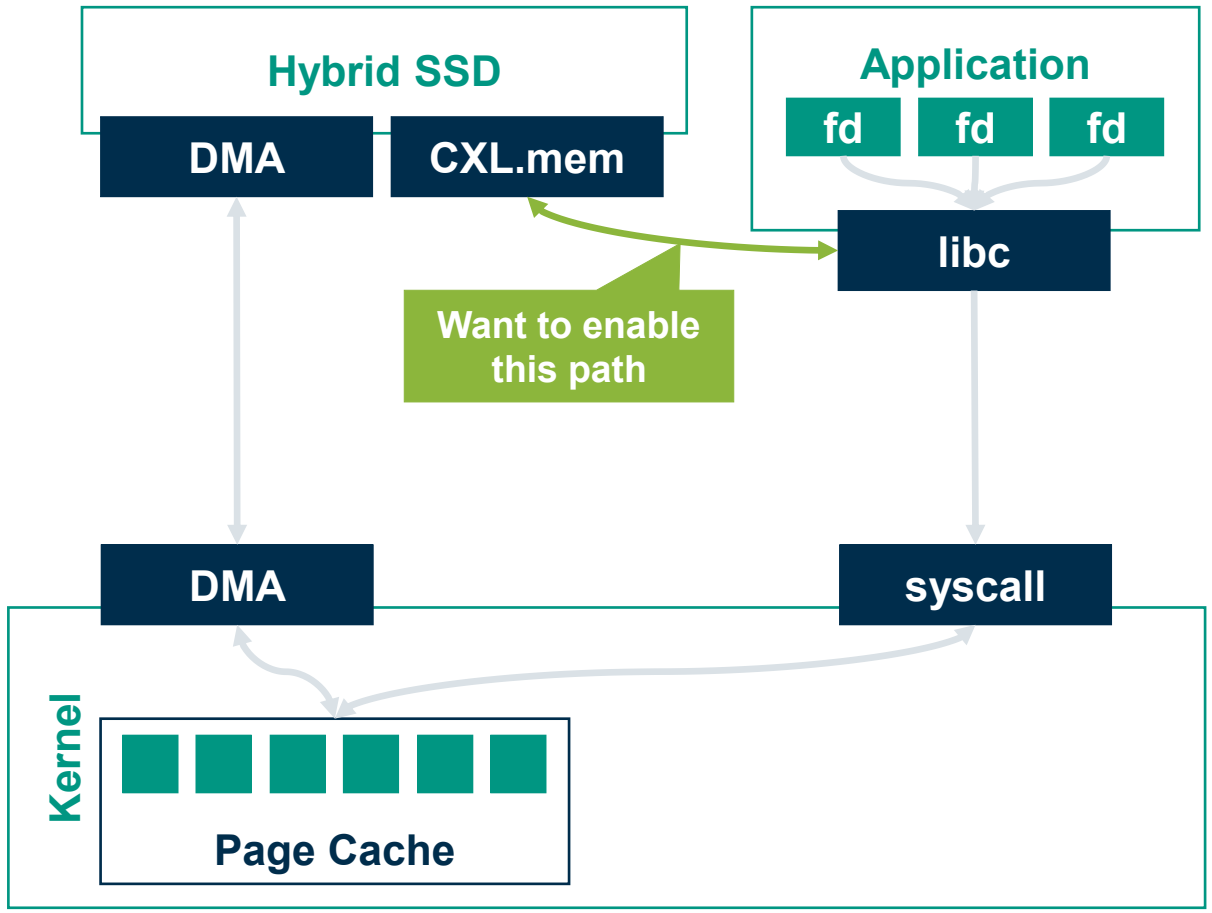

Hybrid SSD
DMA
CXL.mem
Application
fd
fd
fd
libc
syscall
Kernel
Page Cache

Design

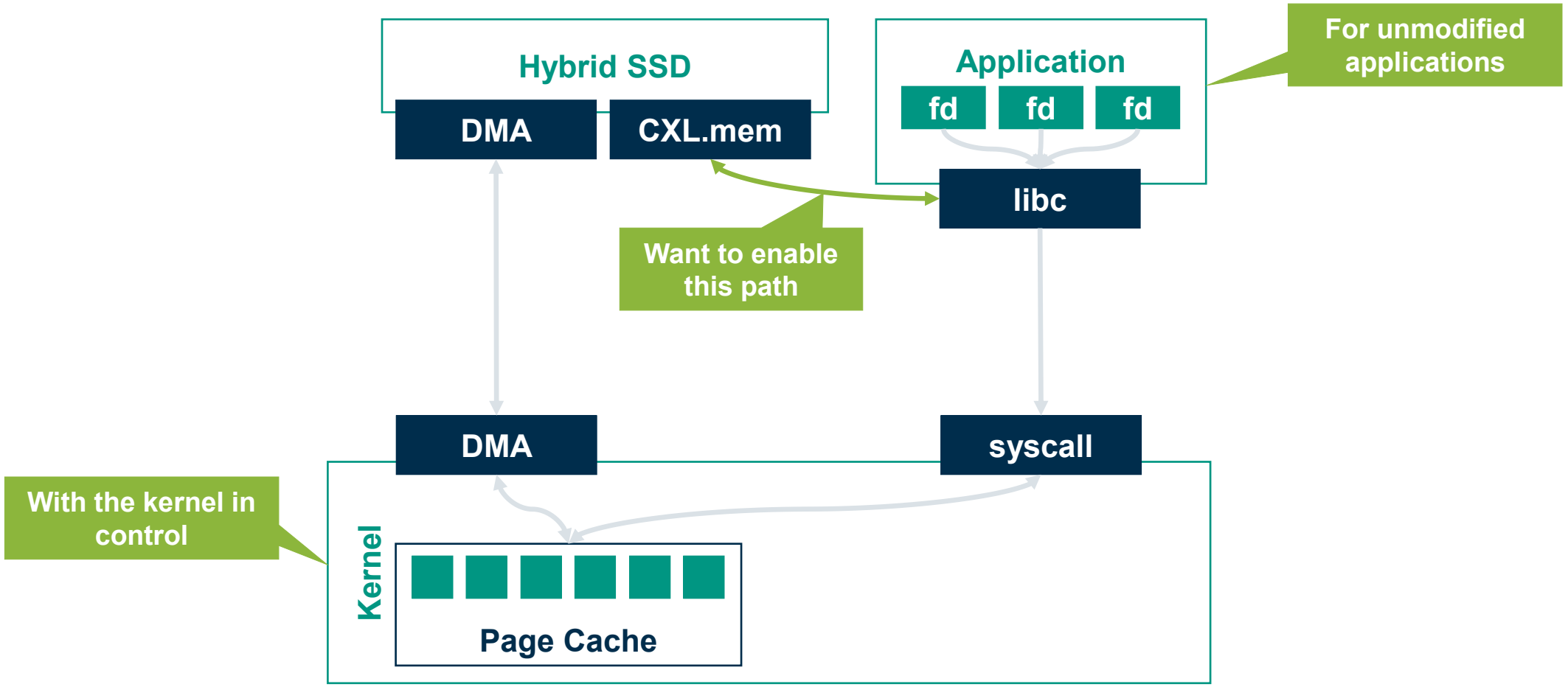


Traditional read()/write() path

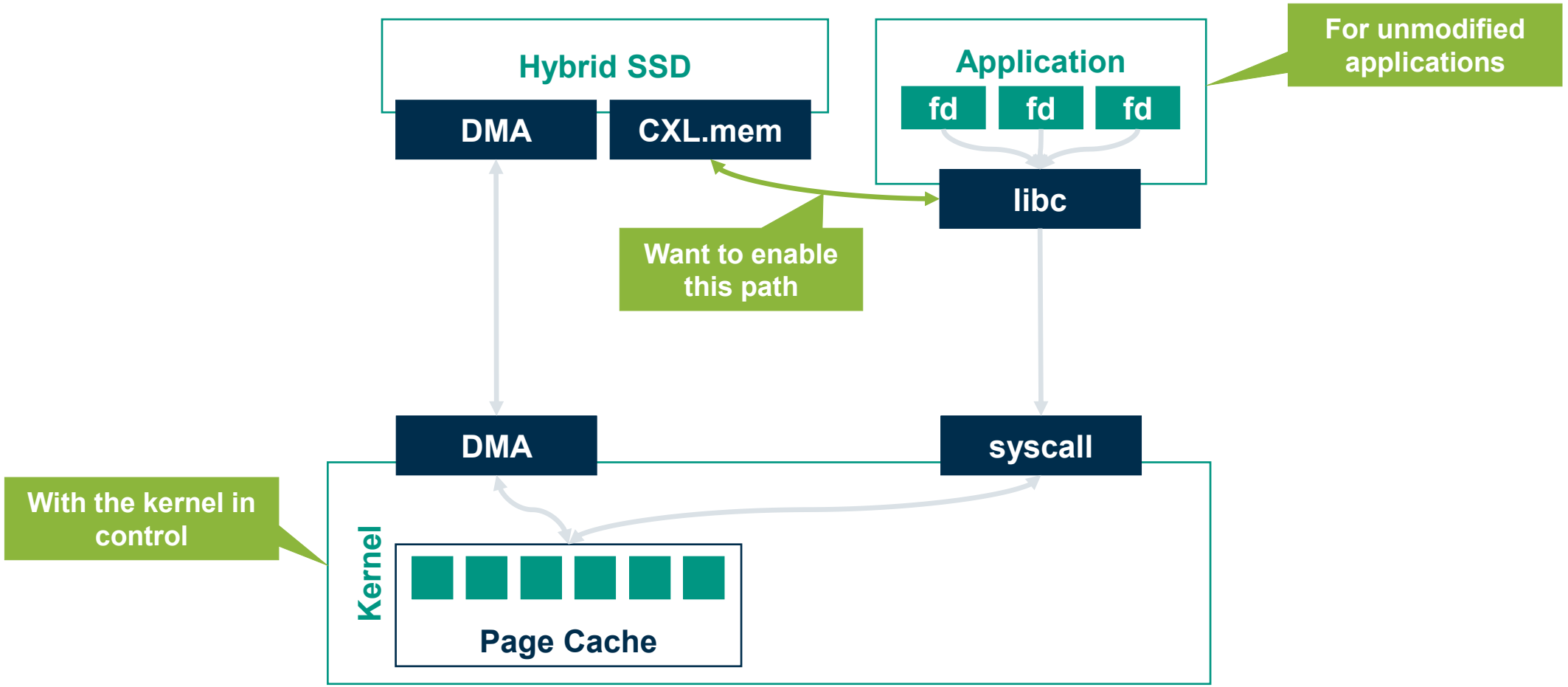
Design


Hybrid SSD
DMA
CXL.mem
Application
fd
fd
fd
libc
Want to enable this path
DMA
syscall
Kernel
Page Cache

Design


Hybrid SSD
DMA
CXL.mem
Application
fd
fd
fd
For unmodified applications
libc
Want to enable this path
DMA
syscall
With the kernel in control
Kernel
Page Cache

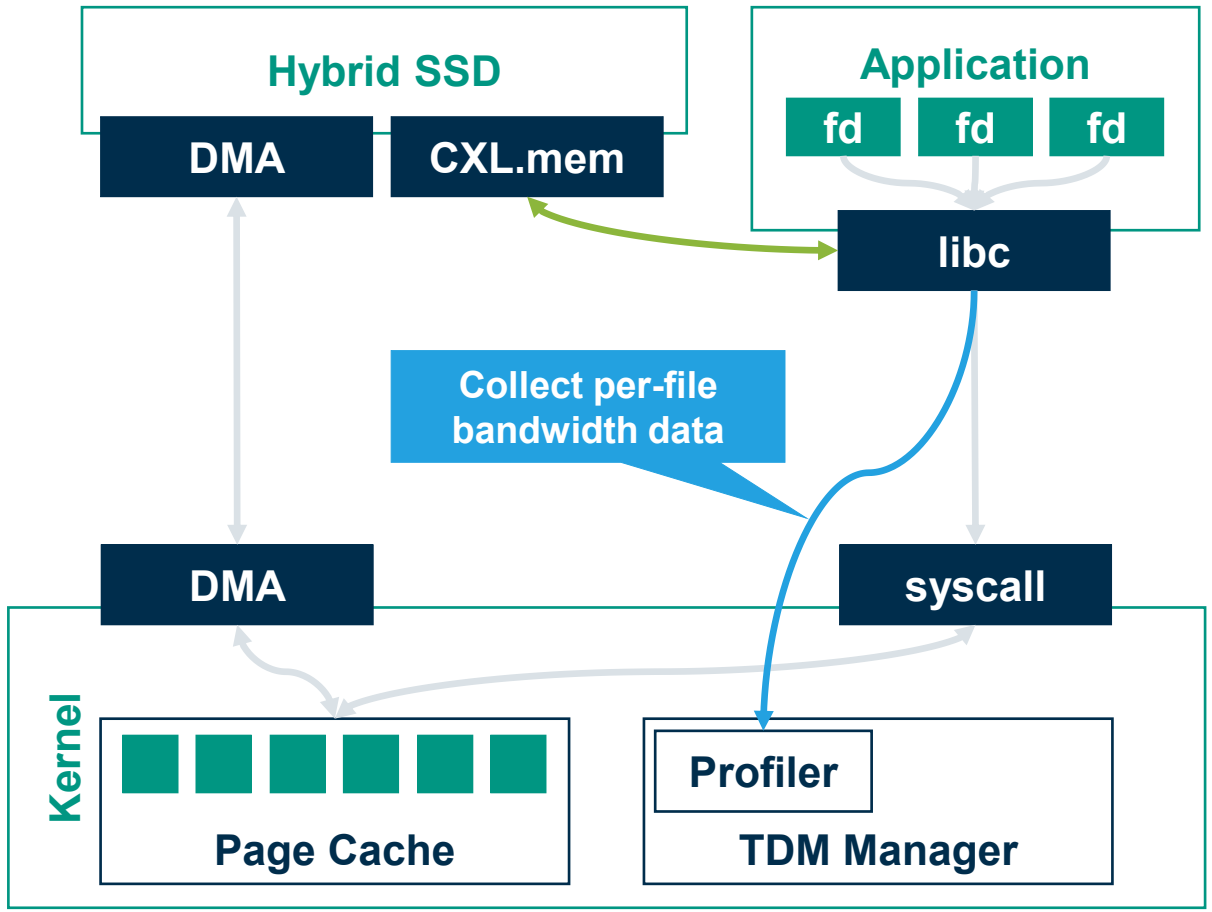
Design



Idea: High-bandwidth files should use DAX.

Design

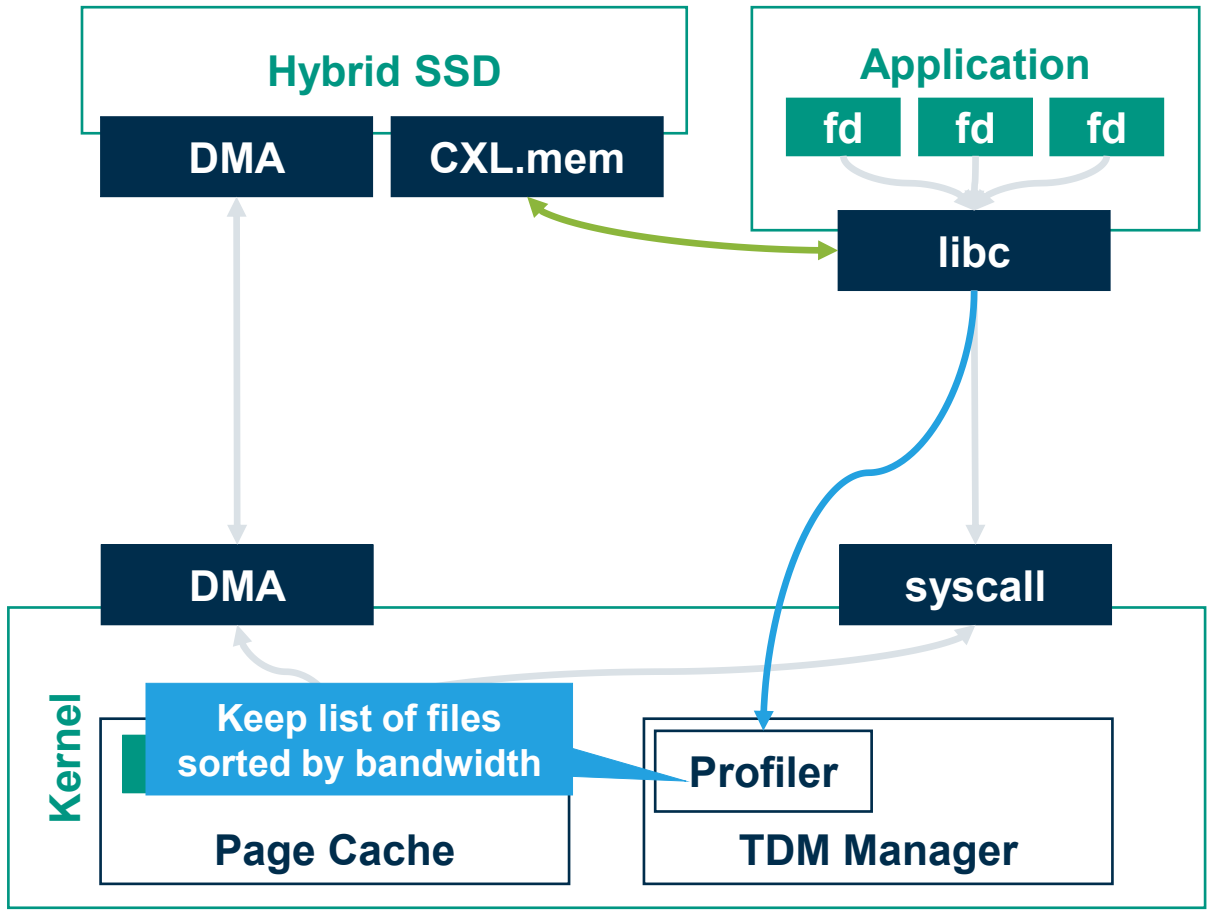
TDM = Transparent DAX Mapping



Idea: High-bandwidth files should use DAX.

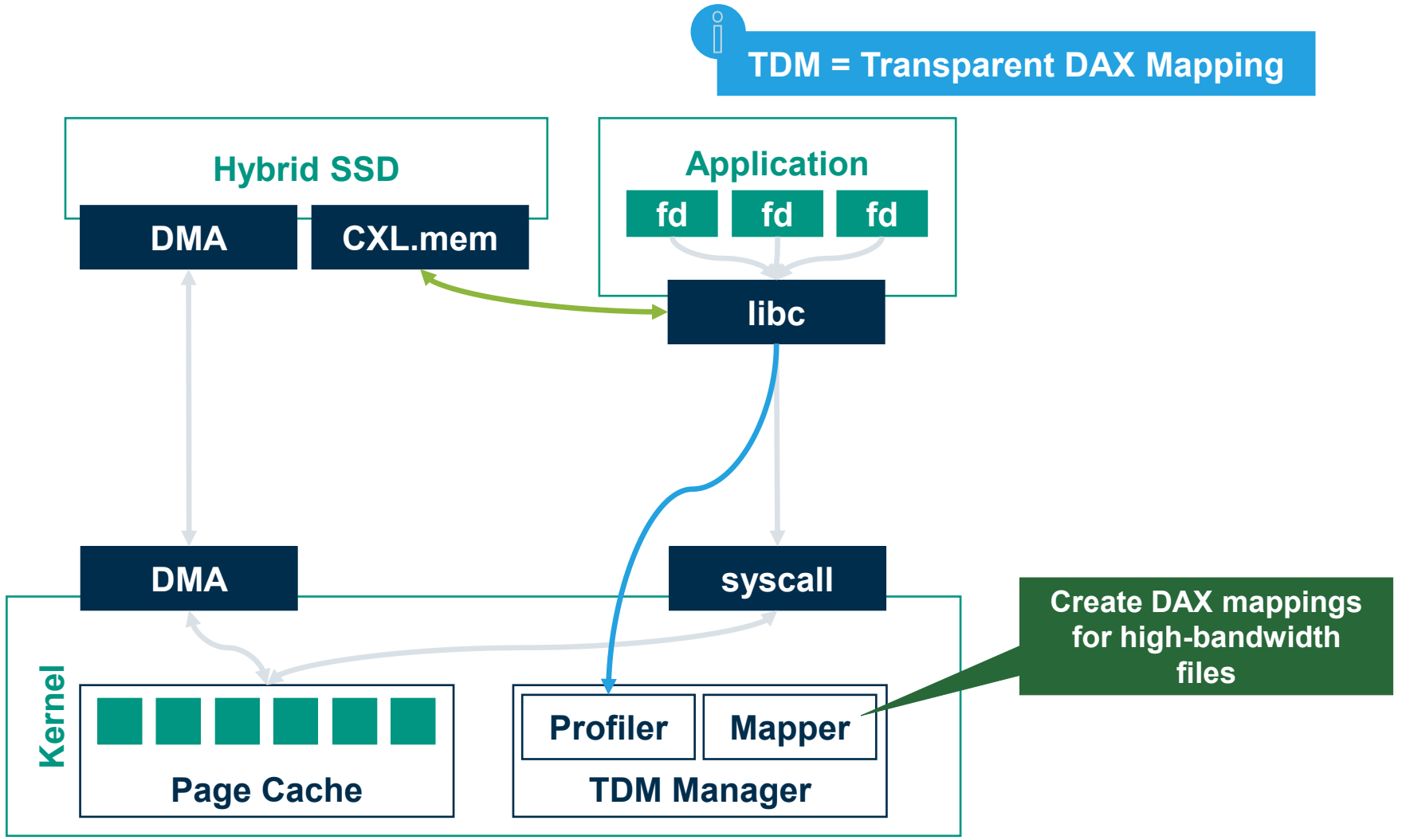
Design

TDM = Transparent DAX Mapping



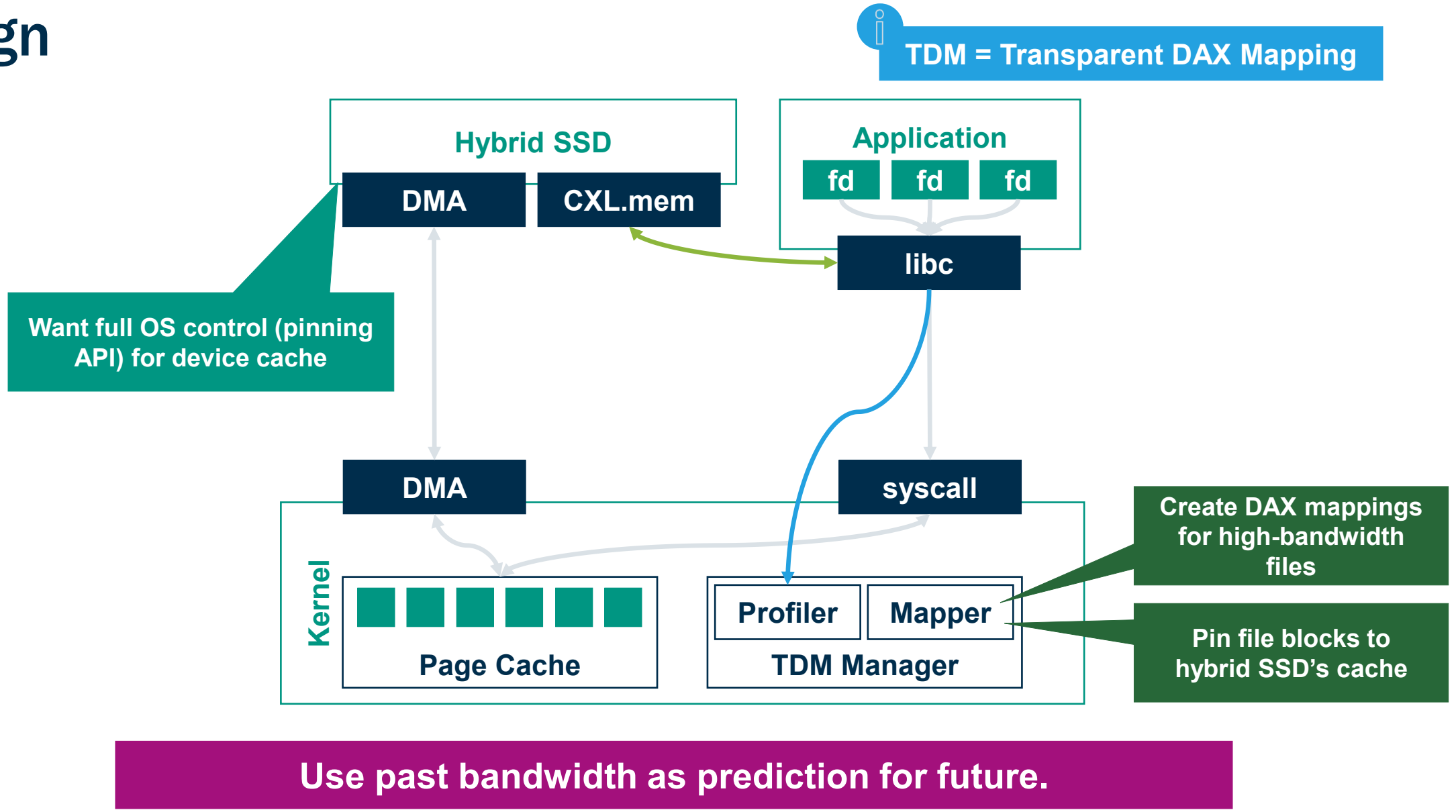
Idea: High-bandwidth files should use DAX.

Design

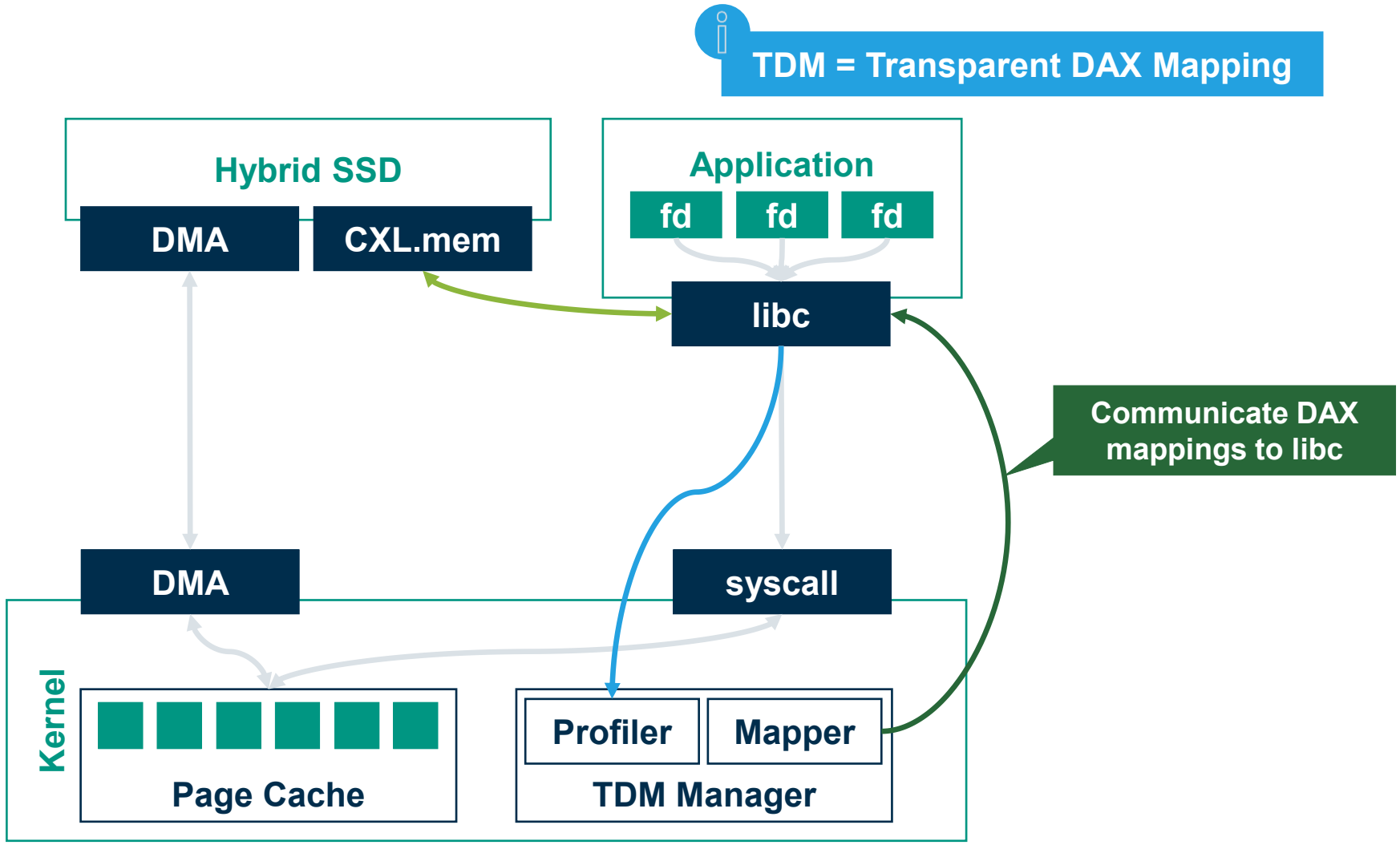


Use past bandwidth as prediction for future.

Design

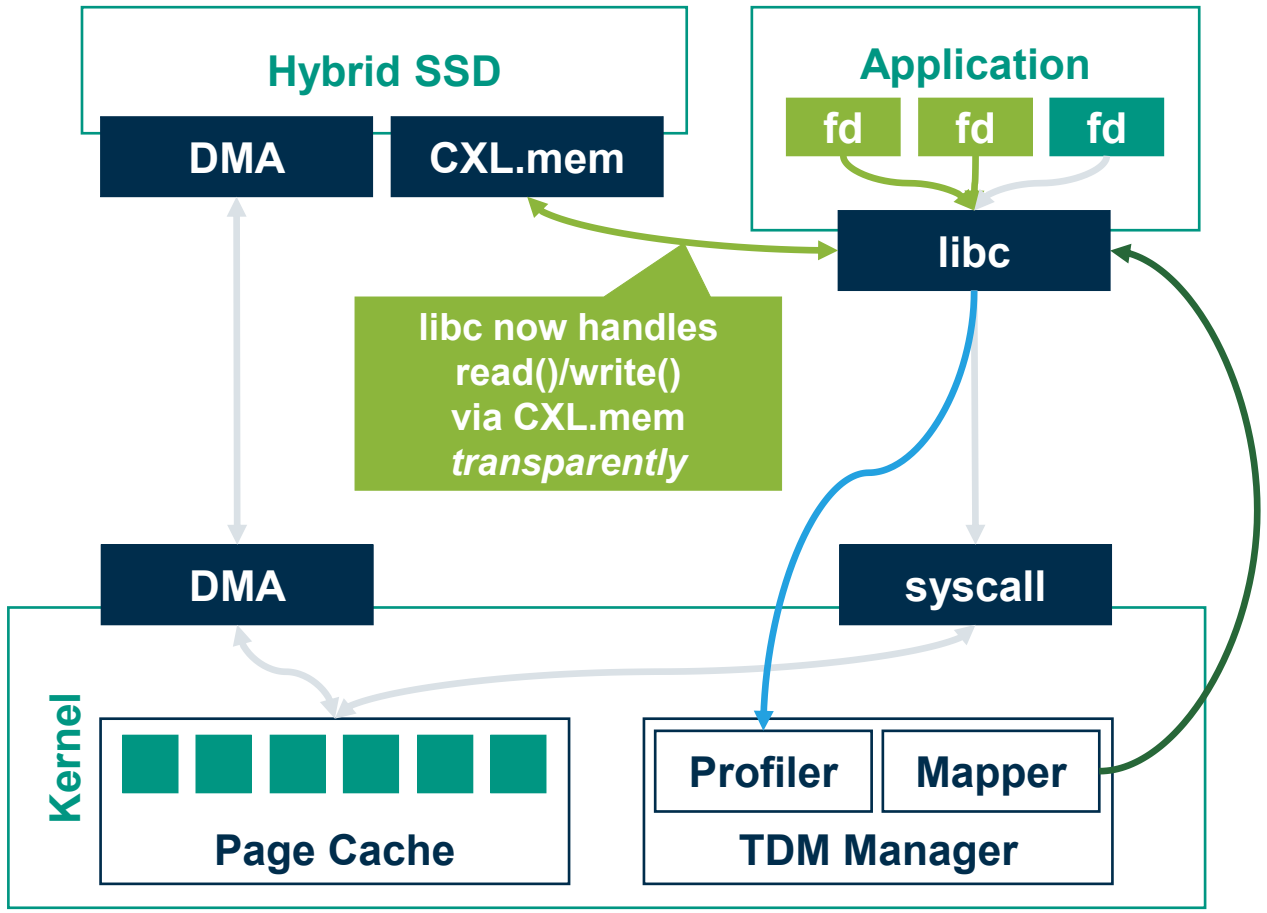

TDM = Transparent DAX Mapping
Hybrid SSD
DMA
CXL.mem
Application
fd
fd
fd
libc
Want full OS control (pinning API) for device cache
DMA
syscall
Kernel
Page Cache
Profiler
Mapper
TDM Manager
Create DAX mappings for high-bandwidth files
Pin file blocks to hybrid SSD's cache
Use past bandwidth as prediction for future.

Design

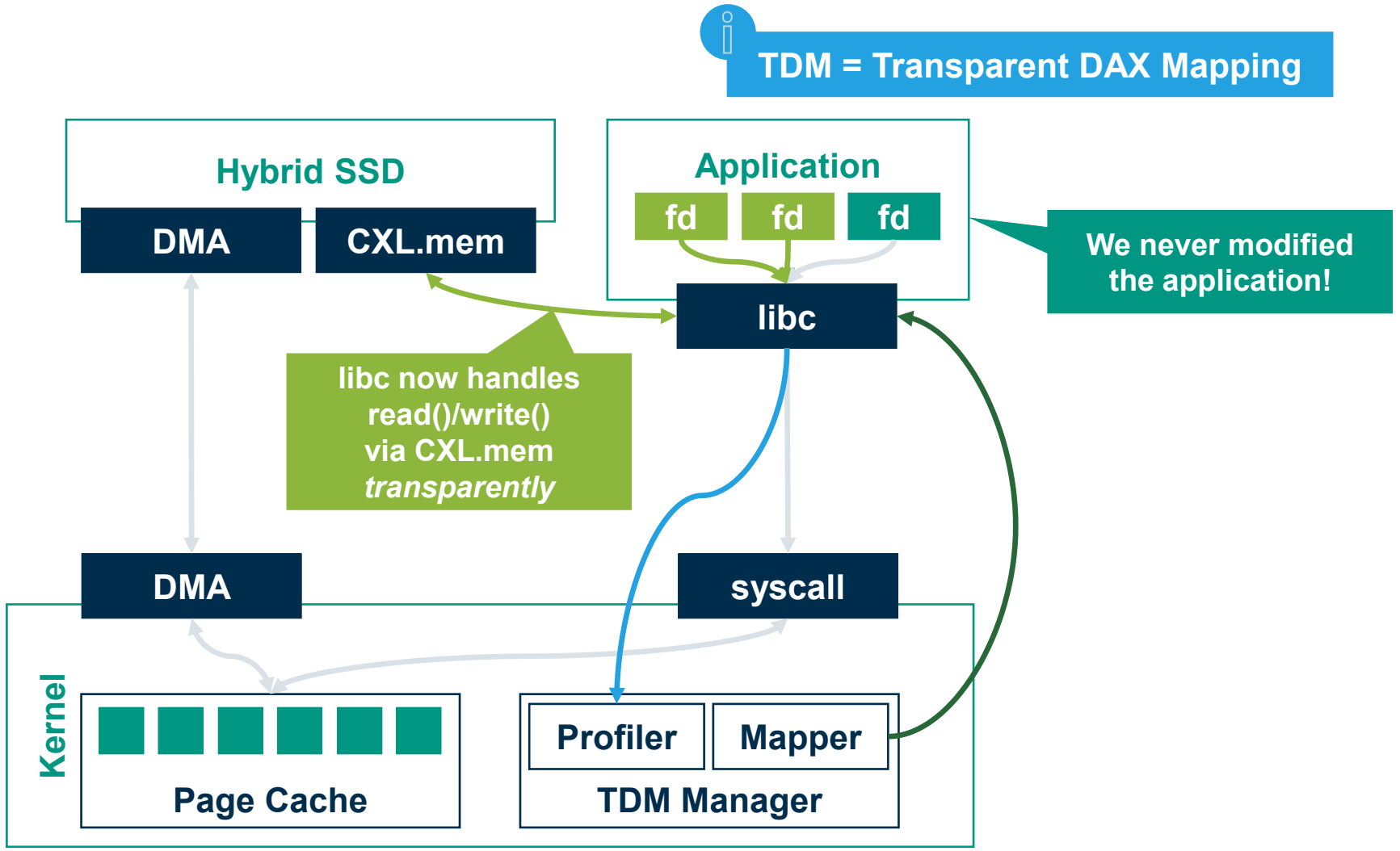

TDM = Transparent DAX Mapping
Hybrid SSD
DMA
CXL.mem
Application
fd
fd
fd
libc
Communicate DAX mappings to libc
DMA
syscall
Kernel
Page Cache
Profiler
Mapper
TDM Manager

Design

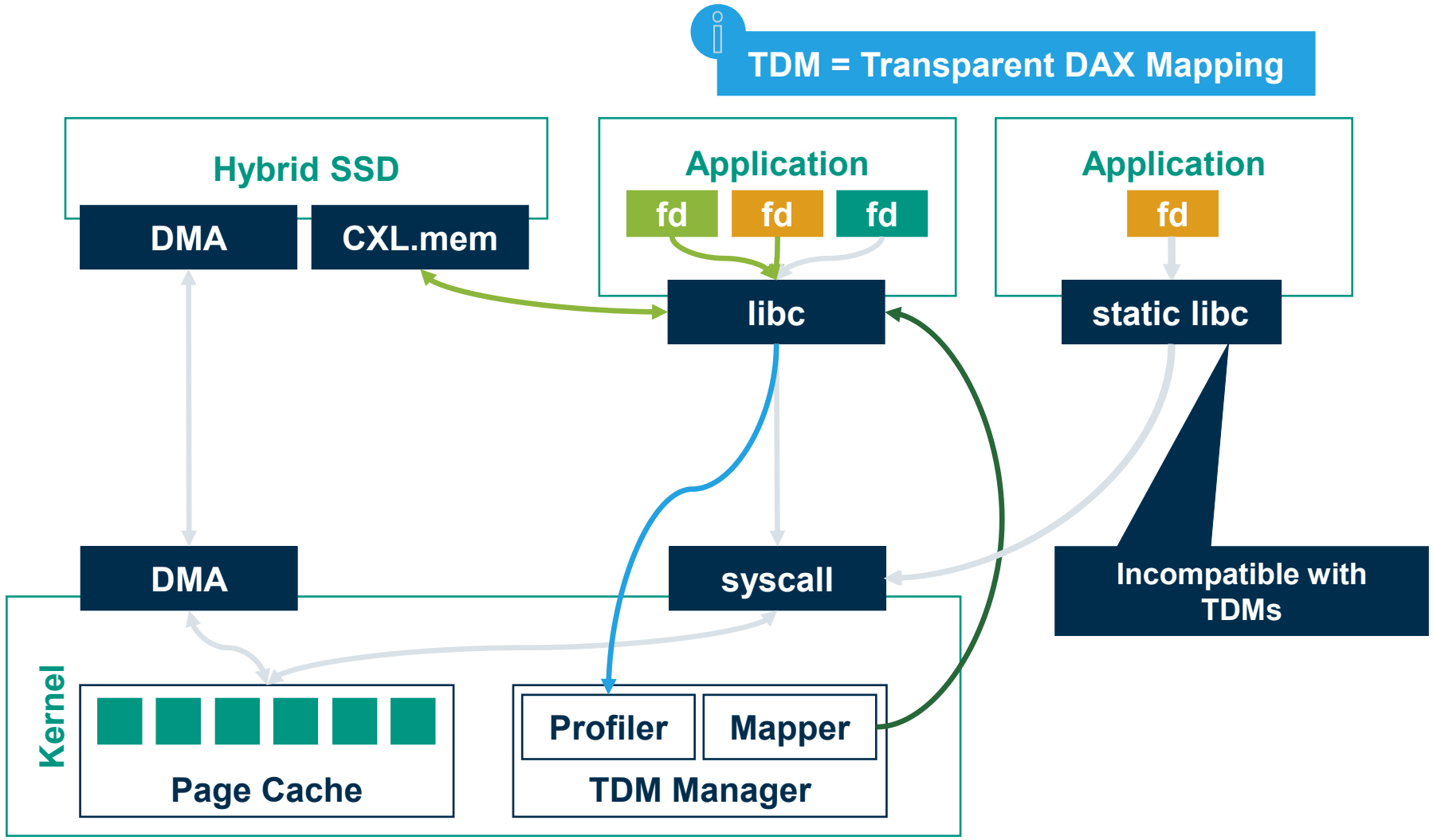
TDM = Transparent DAX Mapping



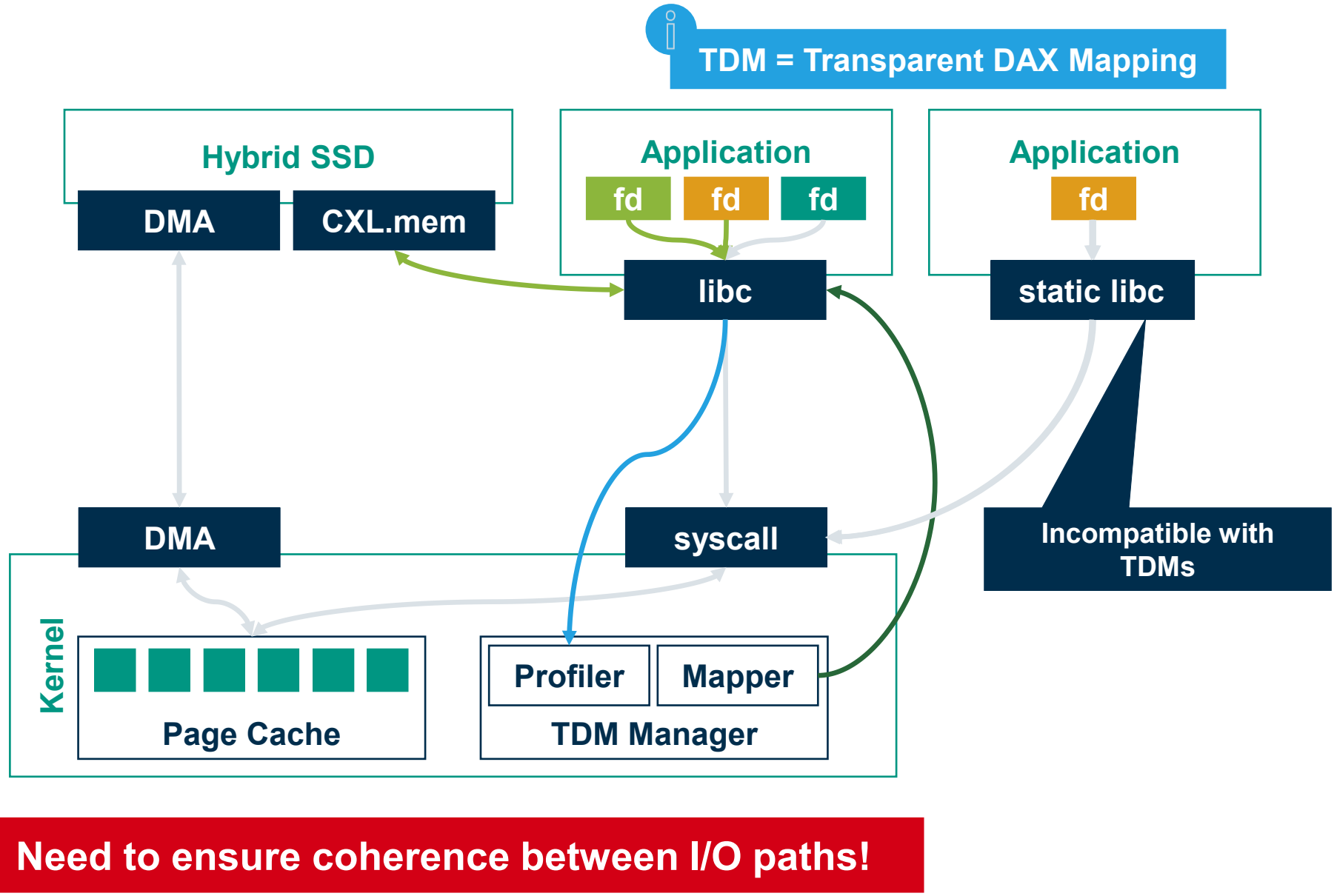
Design


TDM = Transparent DAX Mapping
Hybrid SSD
DMA
CXL.mem
Application
fd
fd
fd
We never modified the application!
libc
libc now handles read()/write() via CXL.mem *transparently*
DMA
syscall
Kernel
Page Cache
Profiler
Mapper
TDM Manager

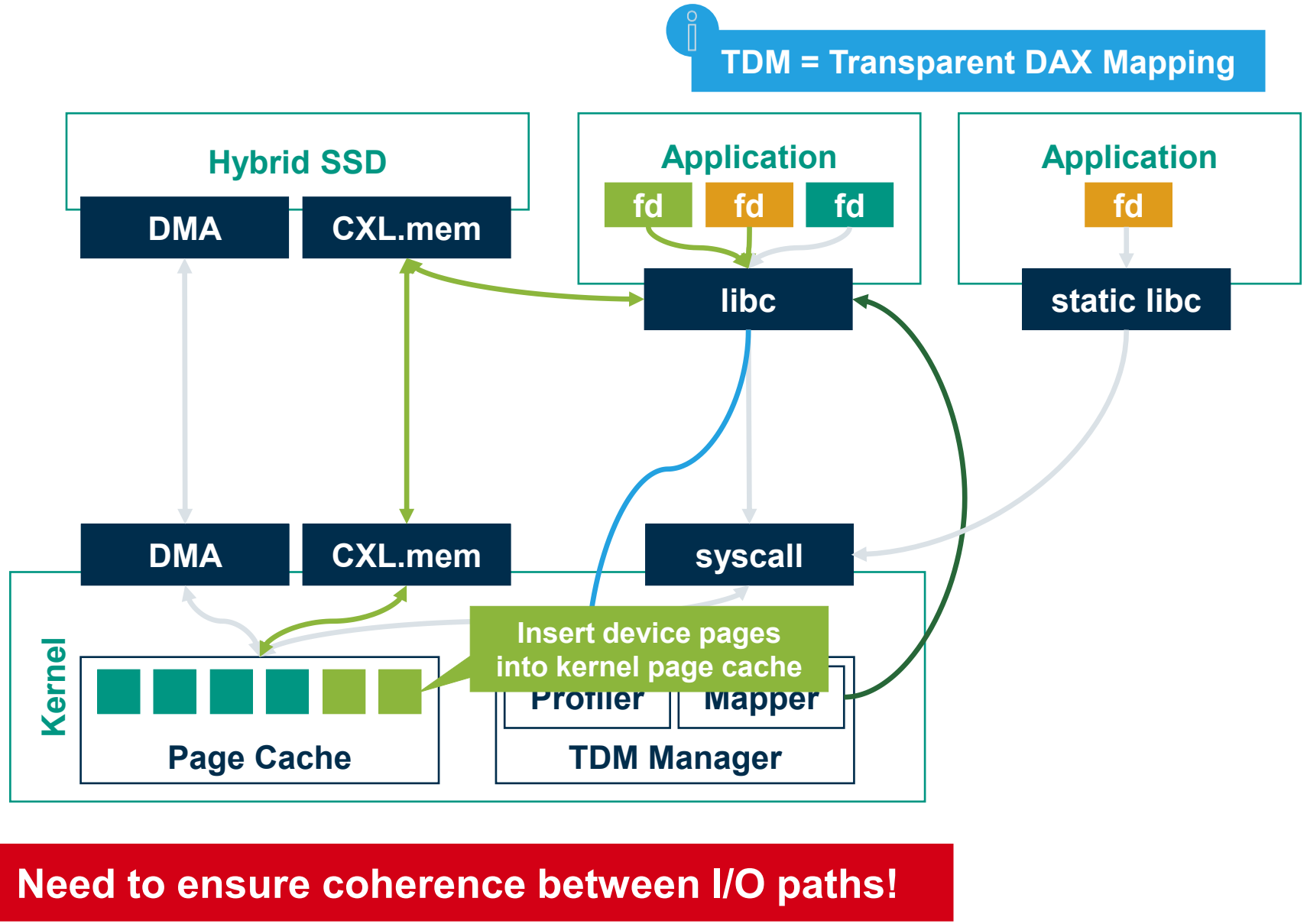
Design


TDM = Transparent DAX Mapping
Hybrid SSD
DMA
CXL.mem
Application
fd
fd
fd
Application
fd
libc
static libc
DMA
syscall
Incompatible with TDMs
Kernel
Page Cache
Profiler
Mapper
TDM Manager

Design


TDM = Transparent DAX Mapping
Hybrid SSD
DMA
CXL.mem
Application
fd
fd
fd
Application
fd
libc
static libc
DMA
syscall
Incompatible with TDMs
Kernel
Page Cache
Profiler
Mapper
TDM Manager
Need to ensure coherence between I/O paths!

Design


TDM = Transparent DAX Mapping
Hybrid SSD
DMA
CXL.mem
Application
fd
fd
fd
Application
fd
libc
static libc
DMA
CXL.mem
syscall
Kernel
Insert device pages into kernel page cache
Page Cache
Profiler
Mapper
TDM Manager
Need to ensure coherence between I/O paths!

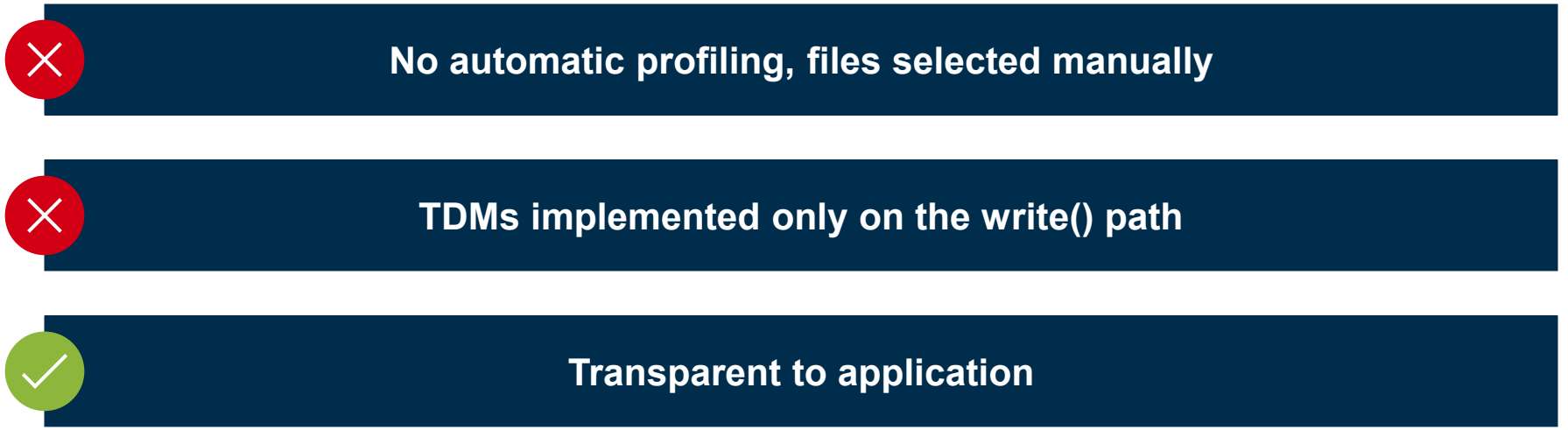
Preliminary Implementation

Preliminary Implementation

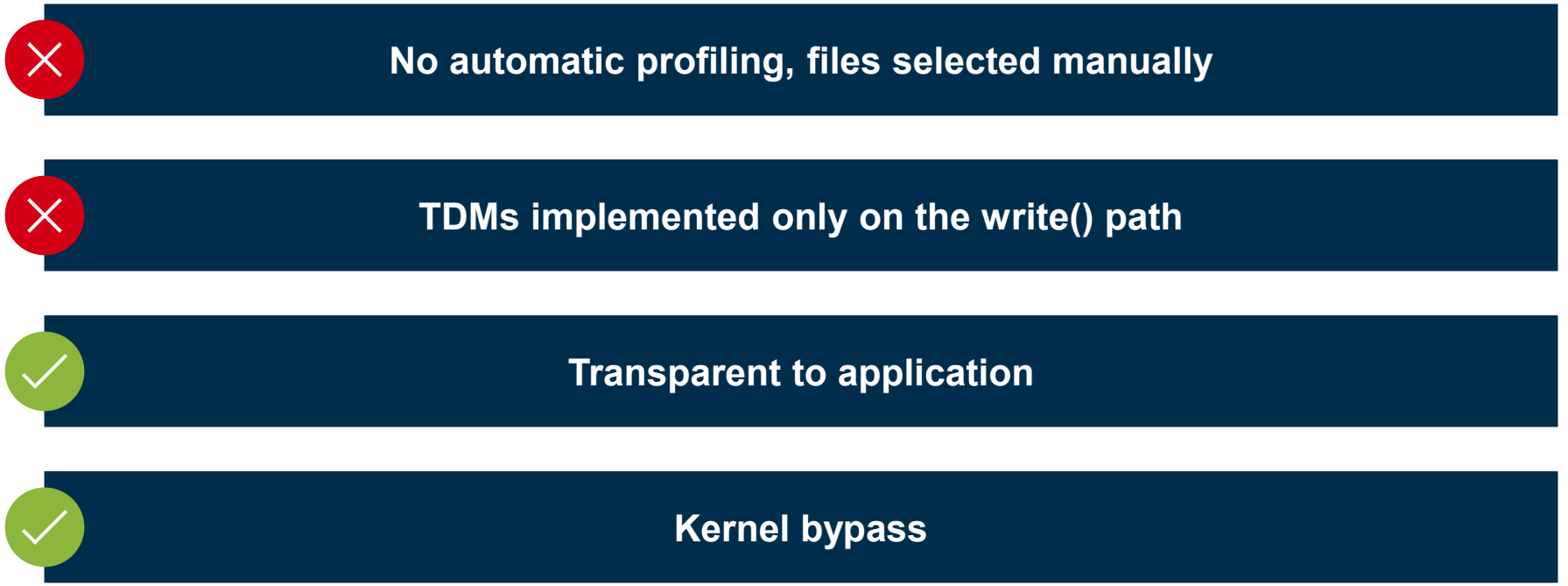
Preliminary Implementation

- No automatic profiling, files selected manually
- TDMs implemented only on the write() path

Preliminary Implementation



Preliminary Implementation



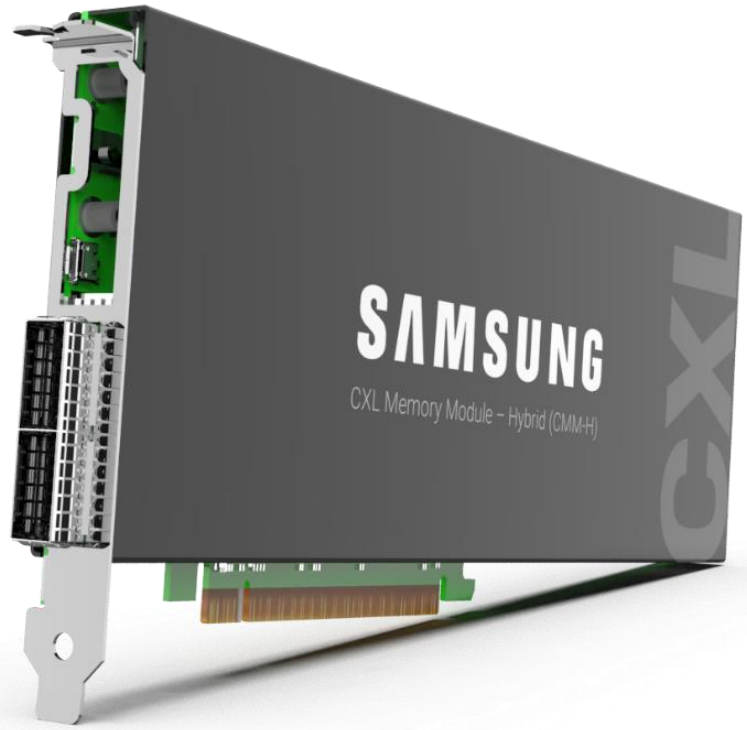
Preliminary Implementation

- ✗ No automatic profiling, files selected manually
- ✗ TDMs implemented only on the write() path
- ✓ Transparent to application
- ✓ Kernel bypass

This is early-stage work.

Evaluation: Samsung CMM-H Prototype

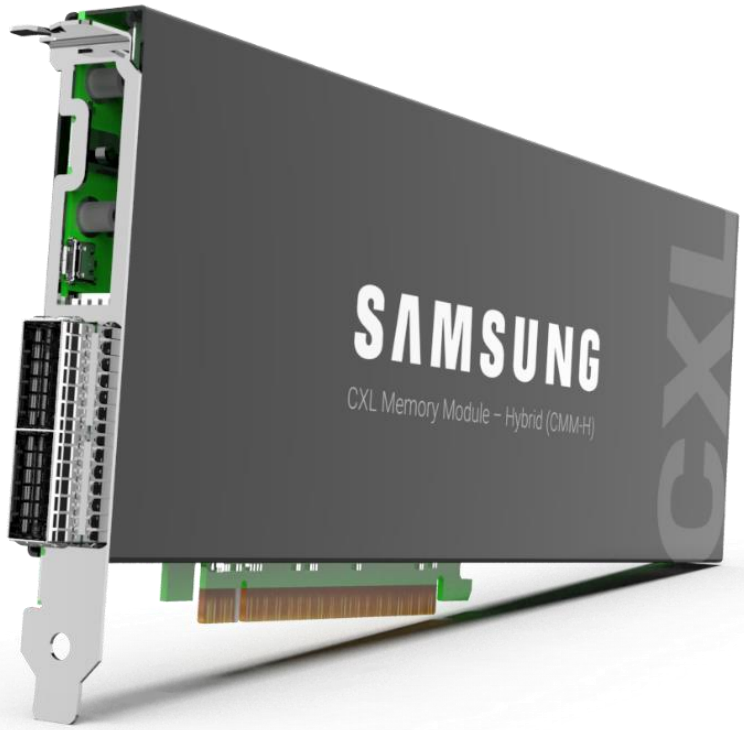
- FPGA-based CXL 2.0 x8 frontend
  - Entire storage capacity accessible via CXL.mem
- Internal Samsung PM9A3 960 GB NVMe SSD
  - Via internal PCIe 4.0 x4 bus
- 48 GiB DRAM cache



Evaluation: Samsung CMM-H Prototype

- FPGA-based CXL 2.0 x8 frontend
  - Entire storage capacity accessible via CXL.mem
- Internal Samsung PM9A3 960 GB NVMe SSD
  - Via internal PCIe 4.0 x4 bus
- 48 GiB DRAM cache

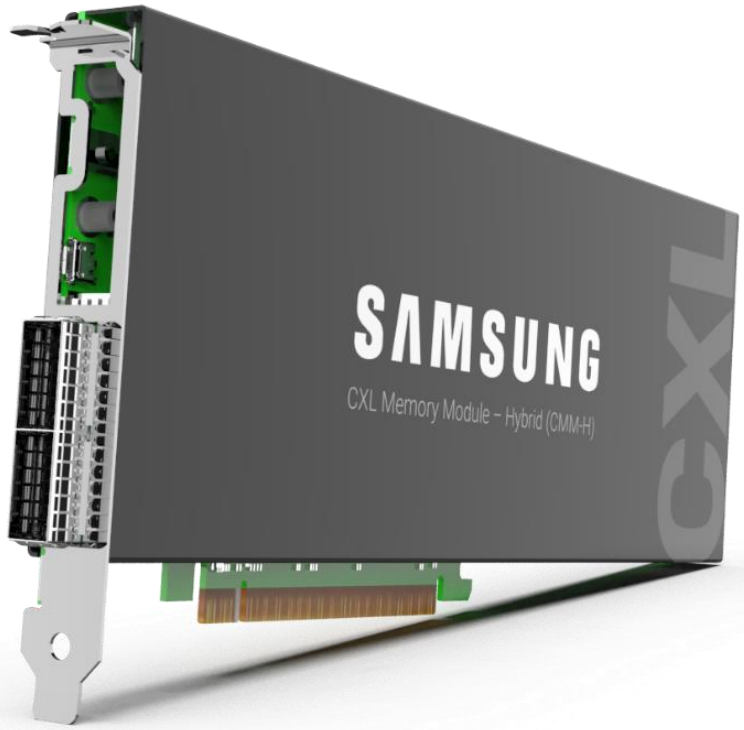
- Internal LRU cache manager
  - Cache management via **prefetch()/evict()** API
  - No explicit cache pinning currently possible



Evaluation: Samsung CMM-H Prototype

- FPGA-based CXL 2.0 x8 frontend
  - Entire storage capacity accessible via CXL.mem
- Internal Samsung PM9A3 960 GB NVMe SSD
  - Via internal PCIe 4.0 x4 bus
- 48 GiB DRAM cache

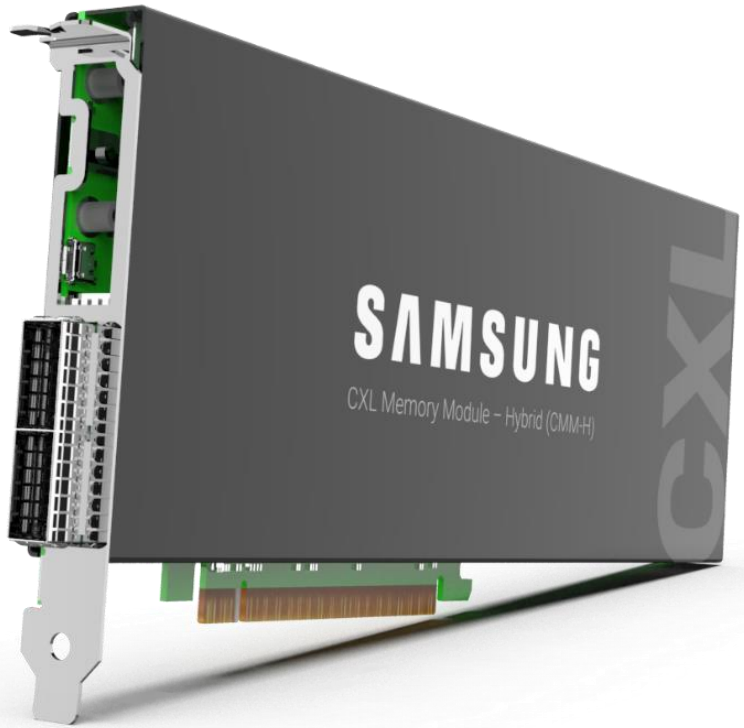
- Internal LRU cache manager
  - Cache management via **prefetch()/evict()** API
  - No explicit cache pinning currently possible
- DMA interface not yet available
  - Kernel uses CXL as well



Evaluation: Samsung CMM-H Prototype

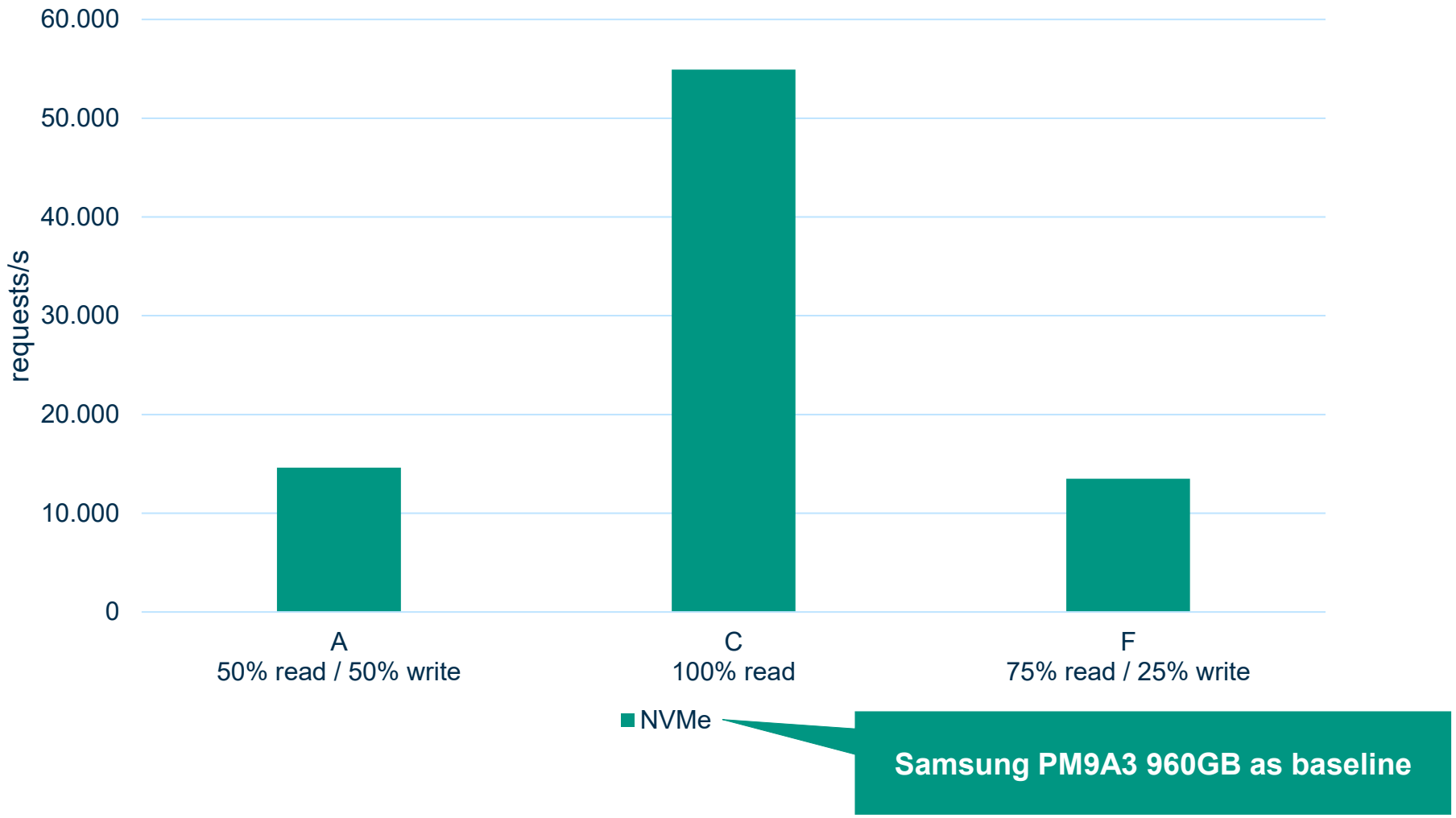
- FPGA-based CXL 2.0 x8 frontend
  - Entire storage capacity accessible via CXL.mem
- Internal Samsung PM9A3 960 GB NVMe SSD
  - Via internal PCIe 4.0 x4 bus
- 48 GiB DRAM cache

- Internal LRU cache manager
  - Cache management via **prefetch()/evict()** API
  - No explicit cache pinning currently possible
- DMA interface not yet available
  - Kernel uses CXL as well

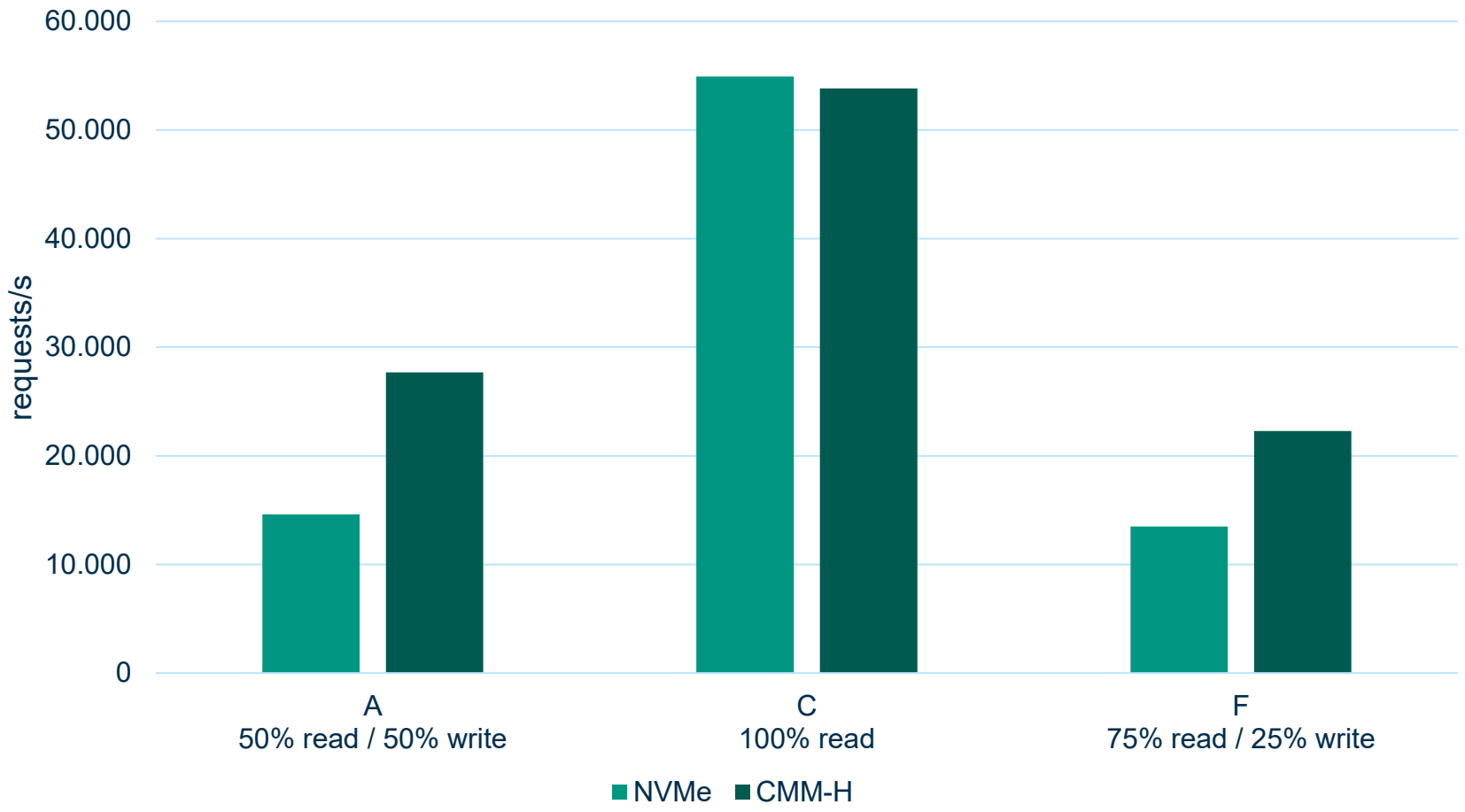


Current implementation focuses on kernel bypass functionality.

Evaluation: YCSB on Valkey

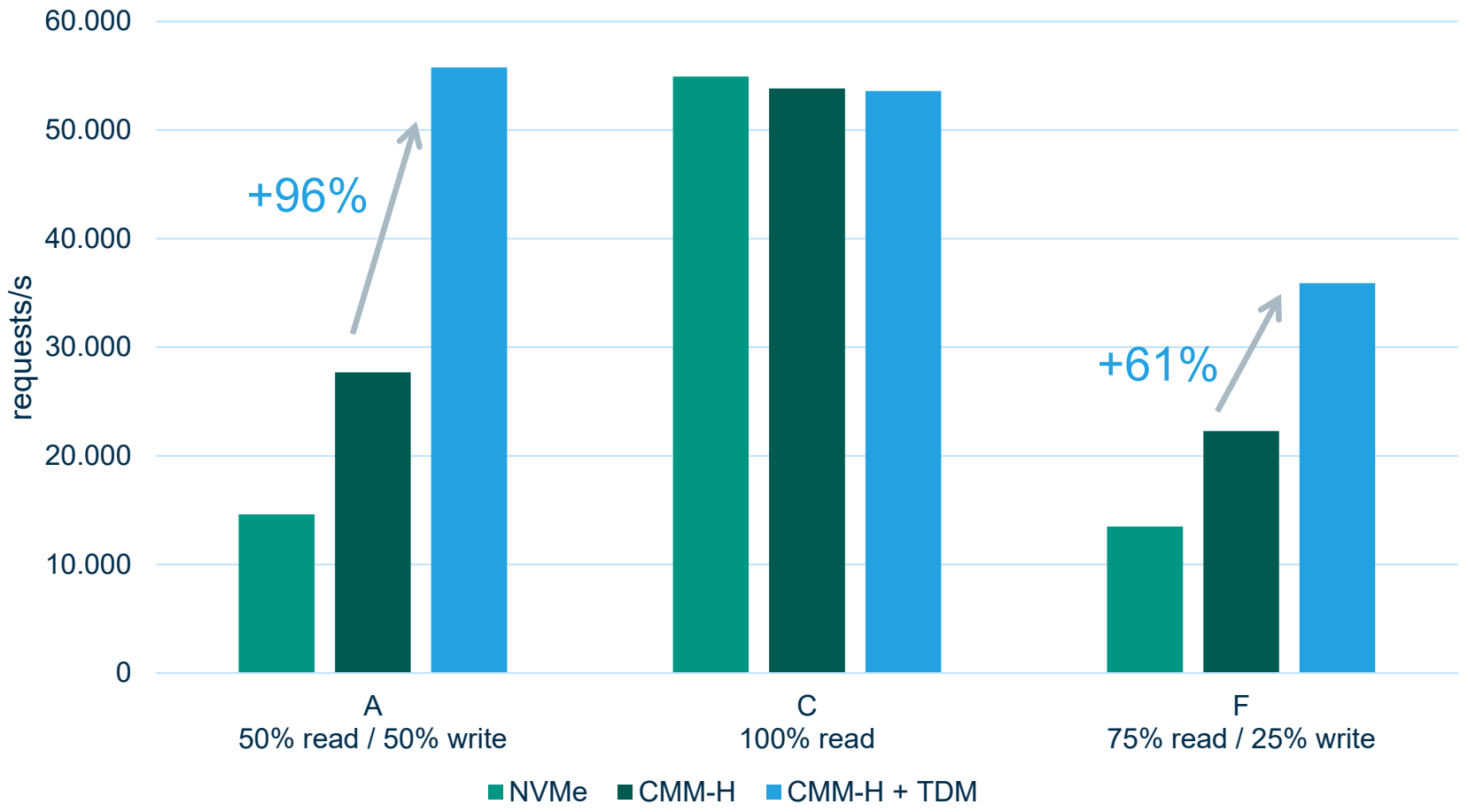


Evaluation: YCSB on Valkey



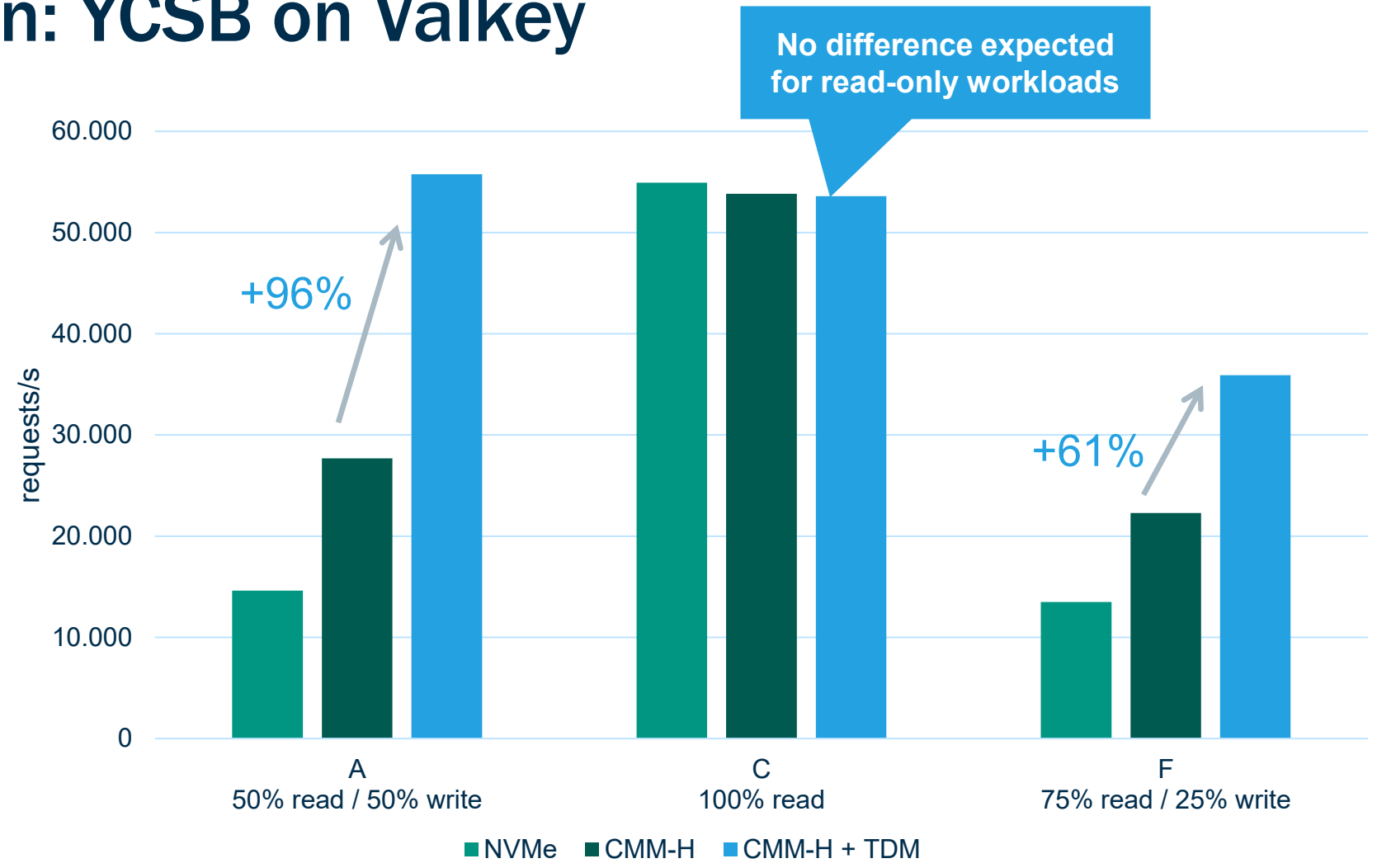
Without TDMs, CMM-H is already faster thanks to reduced latencies.

Evaluation: YCSB on Valkey



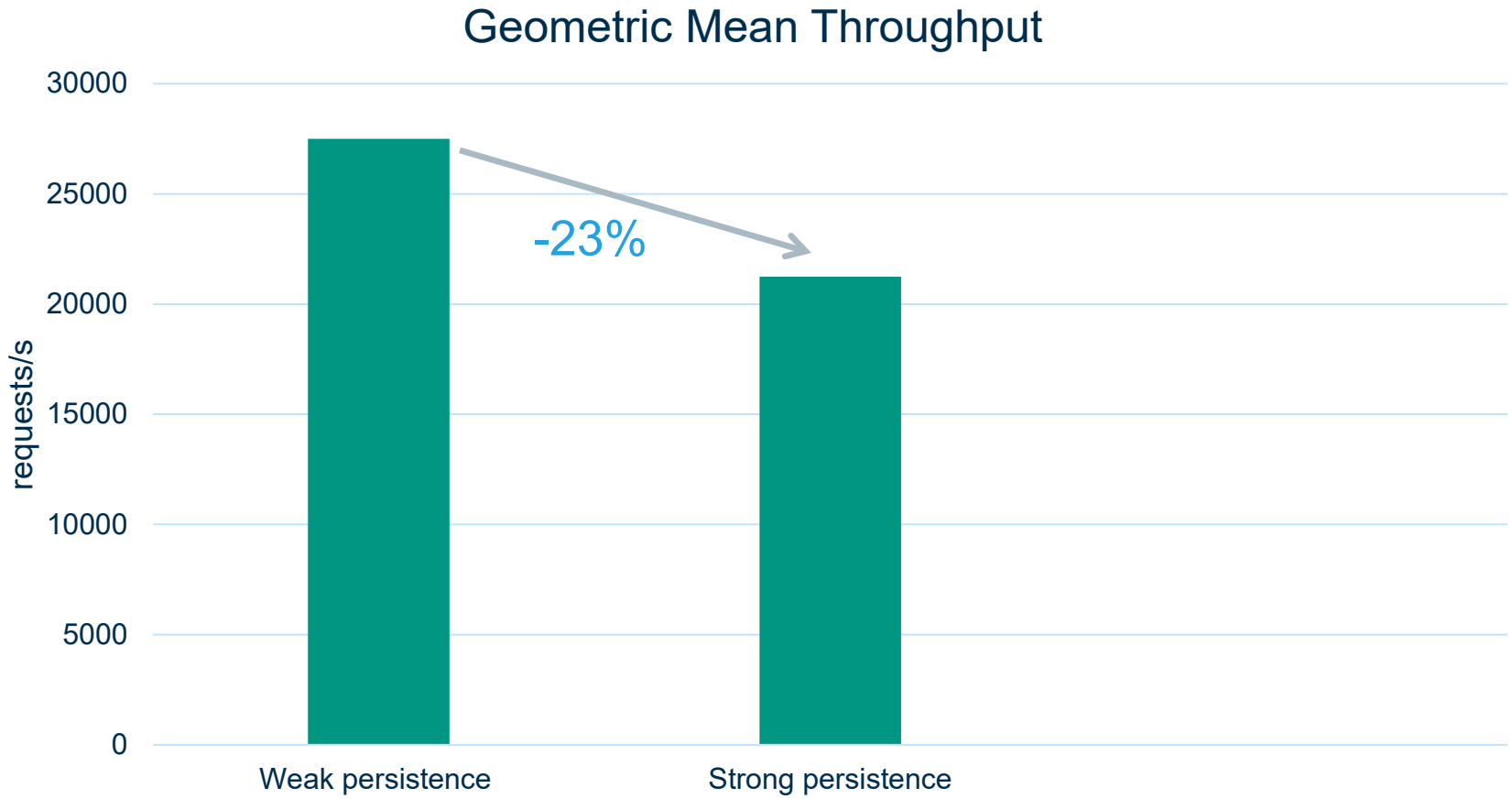
TDMs further increase throughput by up to 96%.

Evaluation: YCSB on Valkey



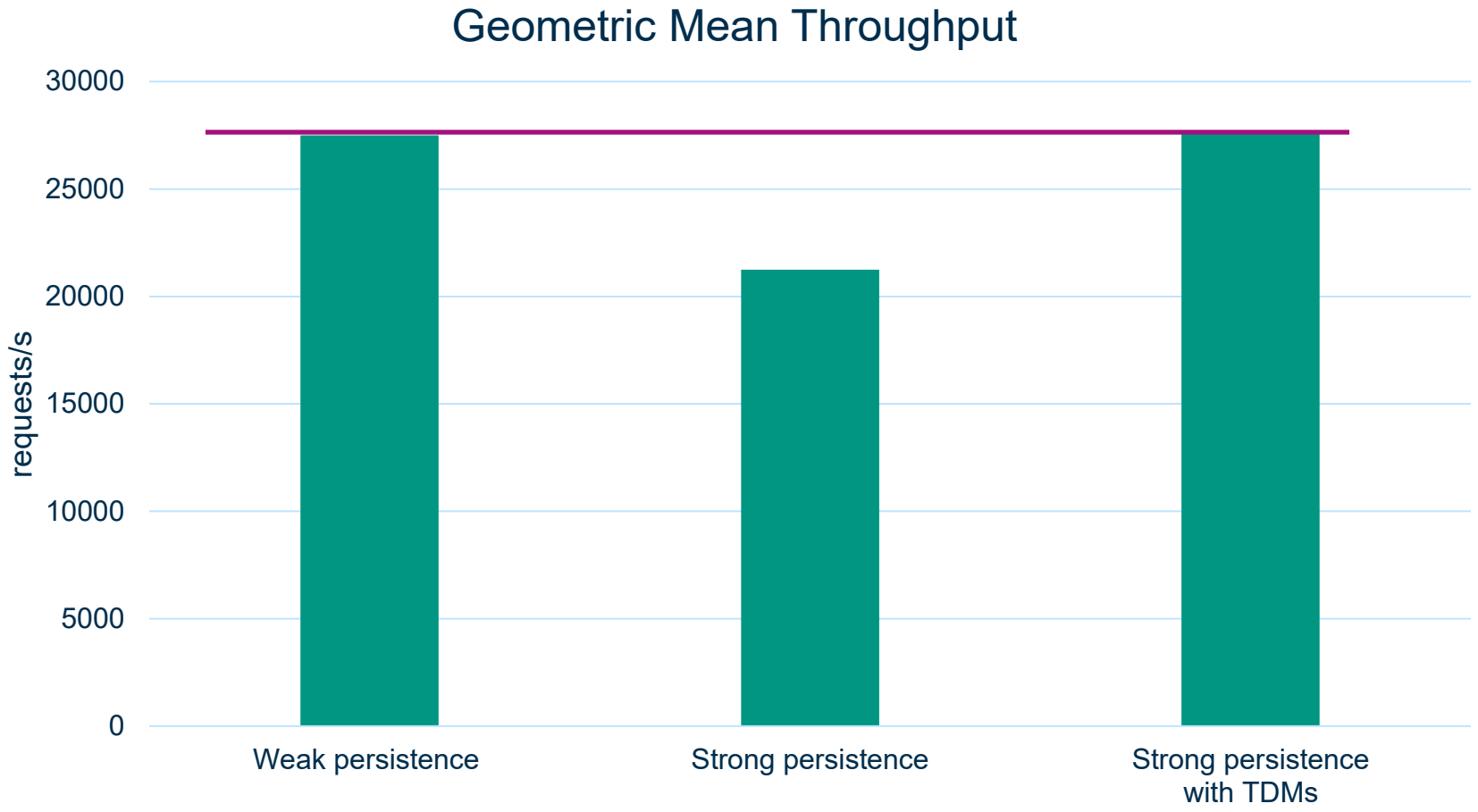
TDMs further increase throughput by up to 96%.

Evaluation: YCSB on Valkey



Strong persistence = fsync() after every write

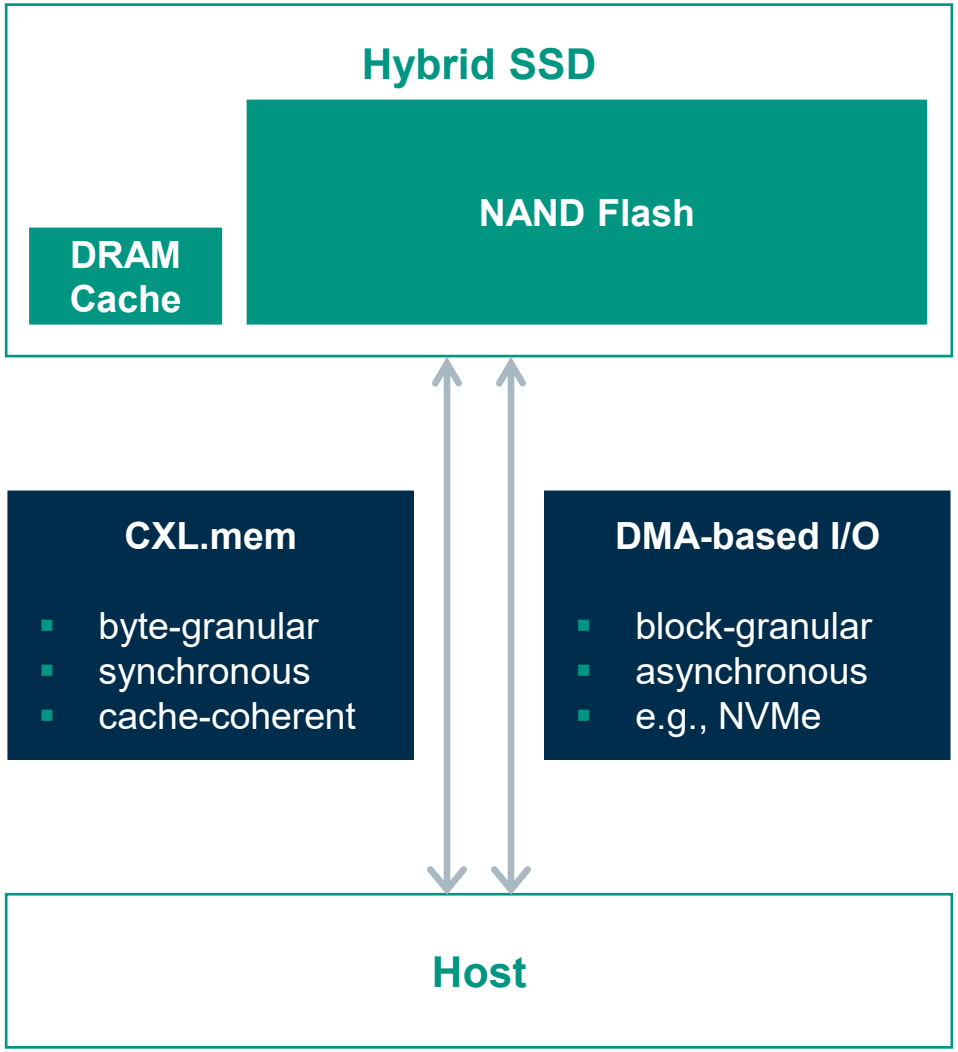
Evaluation: YCSB on Valkey



TDMs make persistence *cheap*.

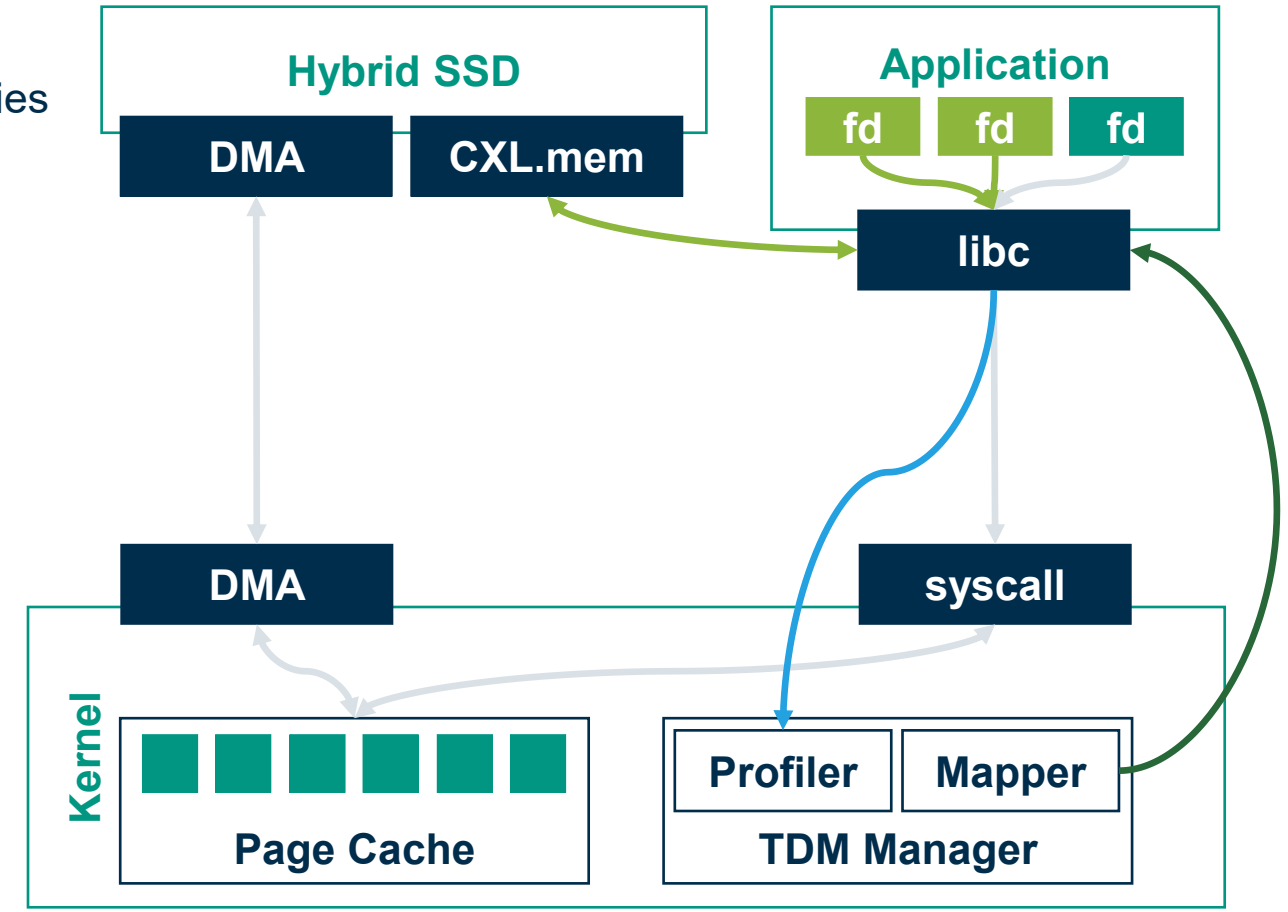
Conclusion

- Hybrid SSDs warrant novel resource management strategies



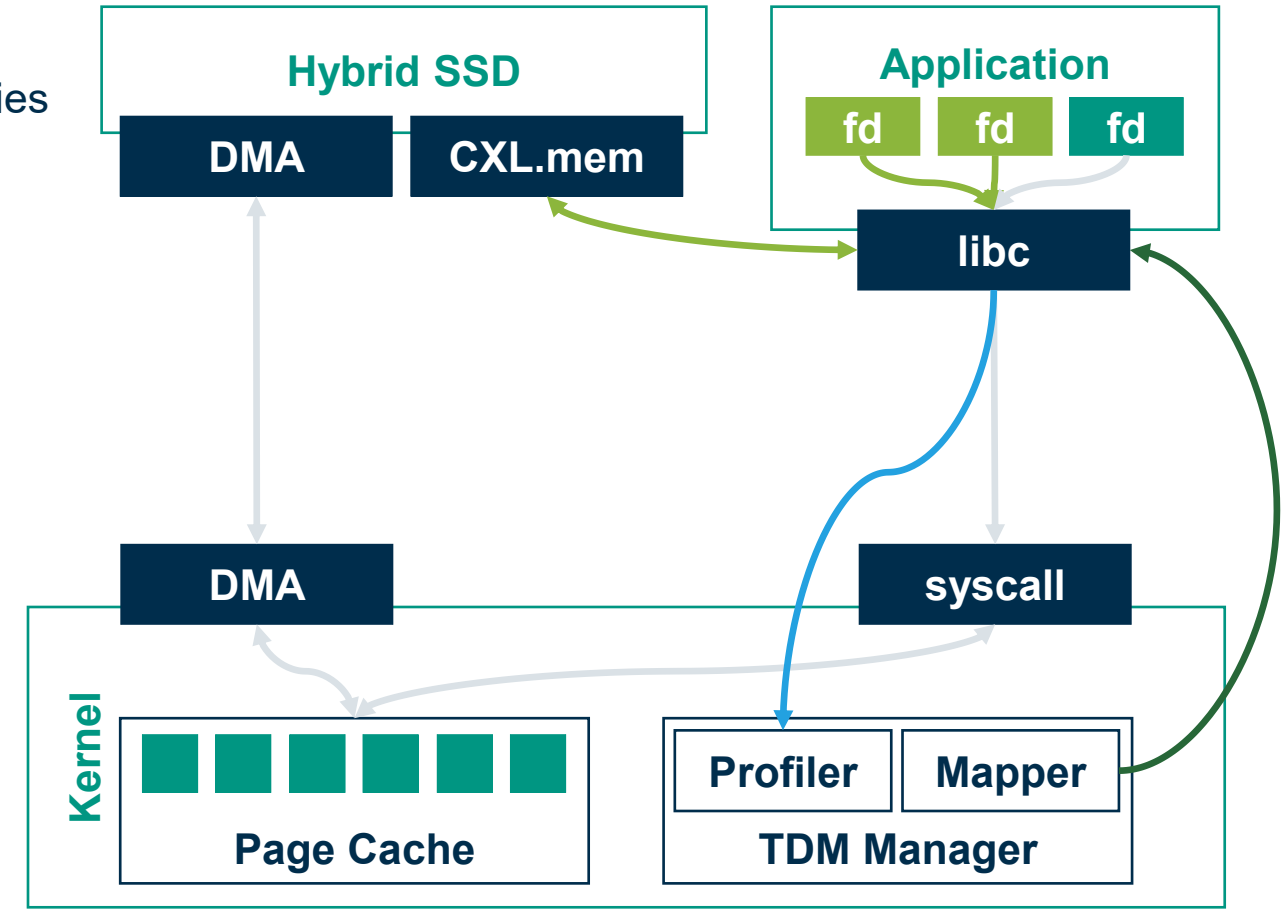
Conclusion

- Hybrid SSDs warrant novel resource management strategies
- Transparent DAX Mappings (TDMs)
  - …put the OS in control of the device's cache
  - …avoid synchronous NAND accesses
  - …enable direct access for high-bandwidth files
  - …employ lightweight and automatic profiling
  - …do not require application modifications



Conclusion

- Hybrid SSDs warrant novel resource management strategies
- Transparent DAX Mappings (TDMs)
  - …put the OS in control of the device's cache
  - …avoid synchronous NAND accesses
  - …enable direct access for high-bandwidth files
  - …employ lightweight and automatic profiling
  - …do not require application modifications
- Preliminary implementation shows potential of approach
  - Up to **96%** increased throughput in Valkey
  - Strong persistence becomes **cheap**



Conclusion

- Hybrid SSDs warrant novel resource management strategies
- Transparent DAX Mappings (TDMs)
  - …put the OS in control of the device's cache
  - …avoid synchronous NAND accesses
  - …enable direct access for high-bandwidth files
  - …employ lightweight and automatic profiling
  - …do not require application modifications
- Preliminary implementation shows potential of approach
  - Up to **96%** increased throughput in Valkey
  - Strong persistence becomes **cheap**
- Our work makes the case for a pinning API in hybrid SSDs