SDC

STORAGE DEVELOPER CONFERENCE

SNIA ■ SANTA CLARA, 2015

Planning for the Next Decade of NVM Programming

Andy Rudoff

Principal Engineer

Intel Corporation

SDC
STORAGE DEVELOPER CONFERENCE
SNIA ■ SANTA CLARA, 2015

Planning for the Next Decade of NVM Programming

Andy Rudoff
Principal Engineer
Intel Corporation
Member: SNIA NVM Programming TWG

The Story So Far:

In the beginning the Universe was created.

This has made a lot of people very angry and been widely regarded as a bad move.

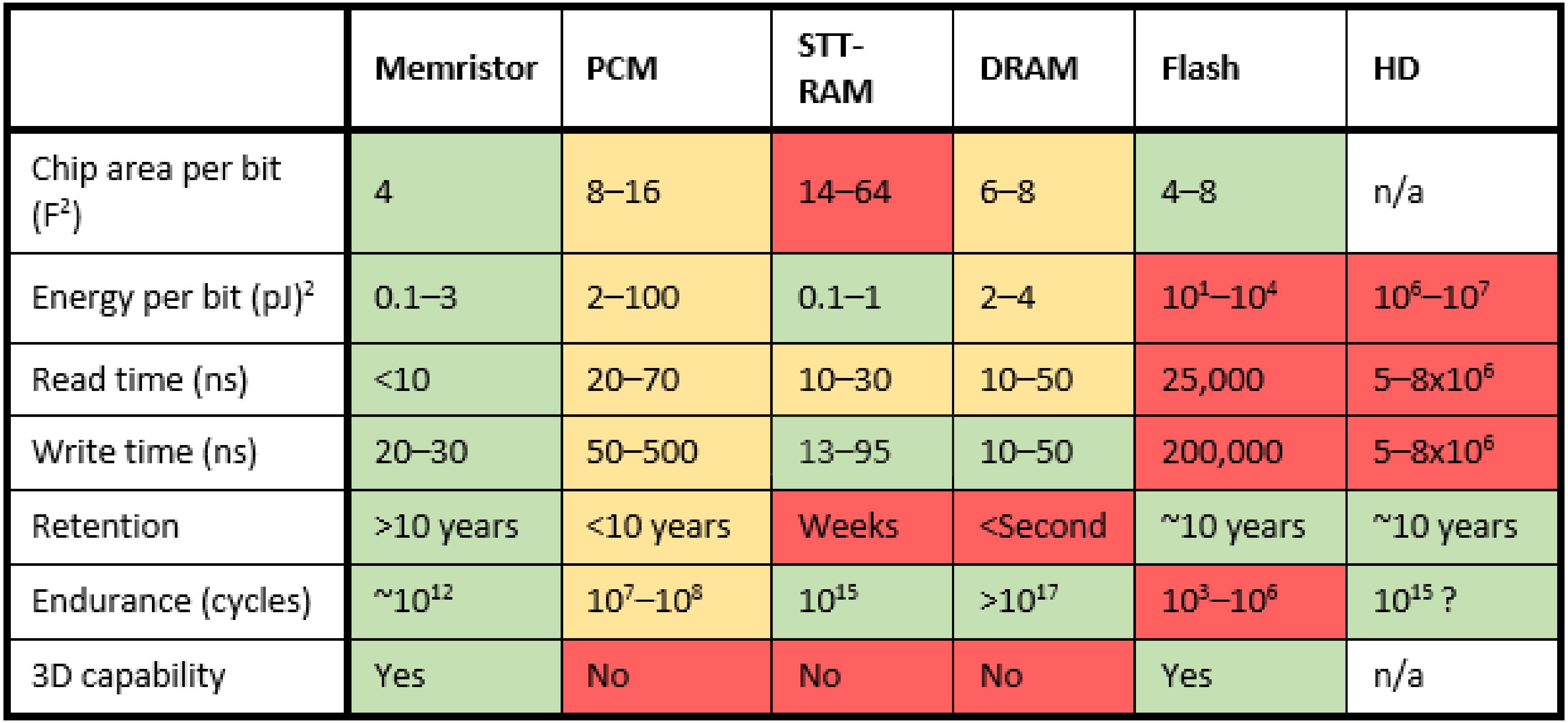
The Story So Far:

In the beginning the Universe was created.

This has made a lot of people very angry and been widely regarded as a bad move.

- Douglas Adams, *The Restaurant at the End of the Universe*

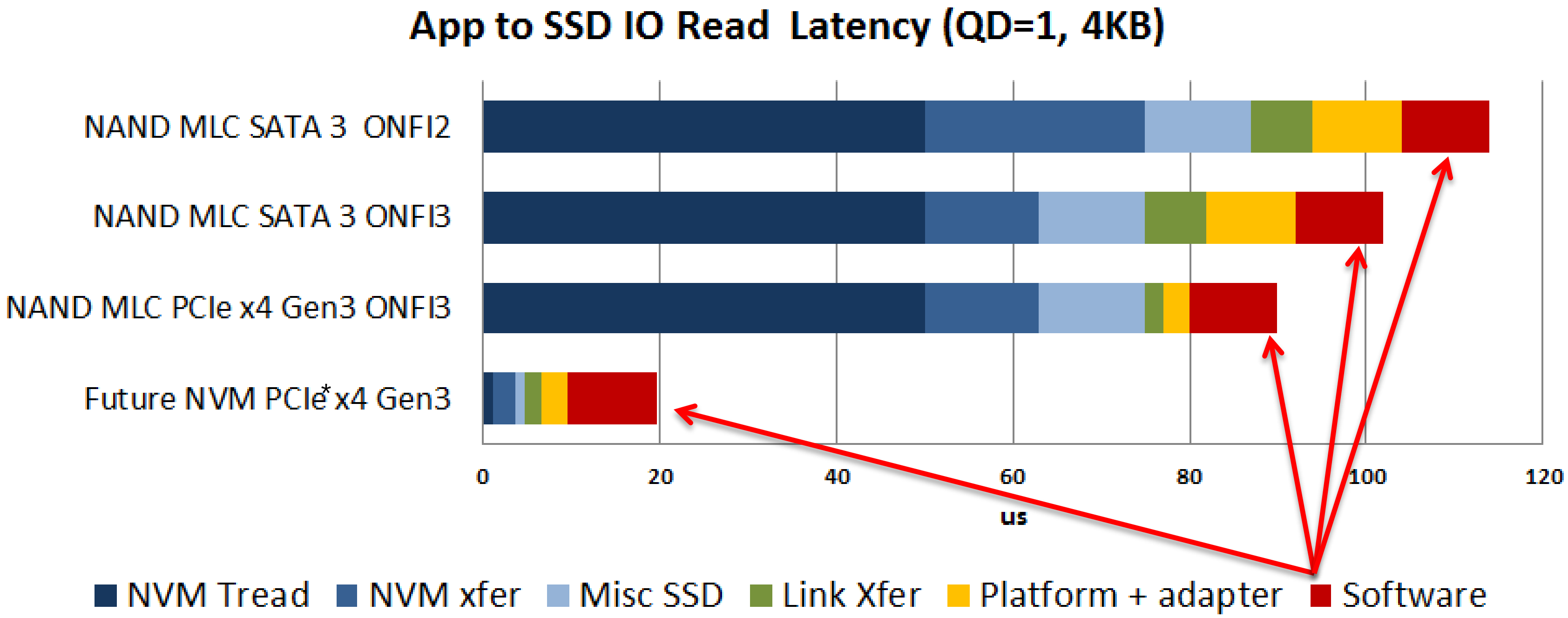
Table 1. Comparison of data storage technologies. (Data drawn from public sources and HP internal research).



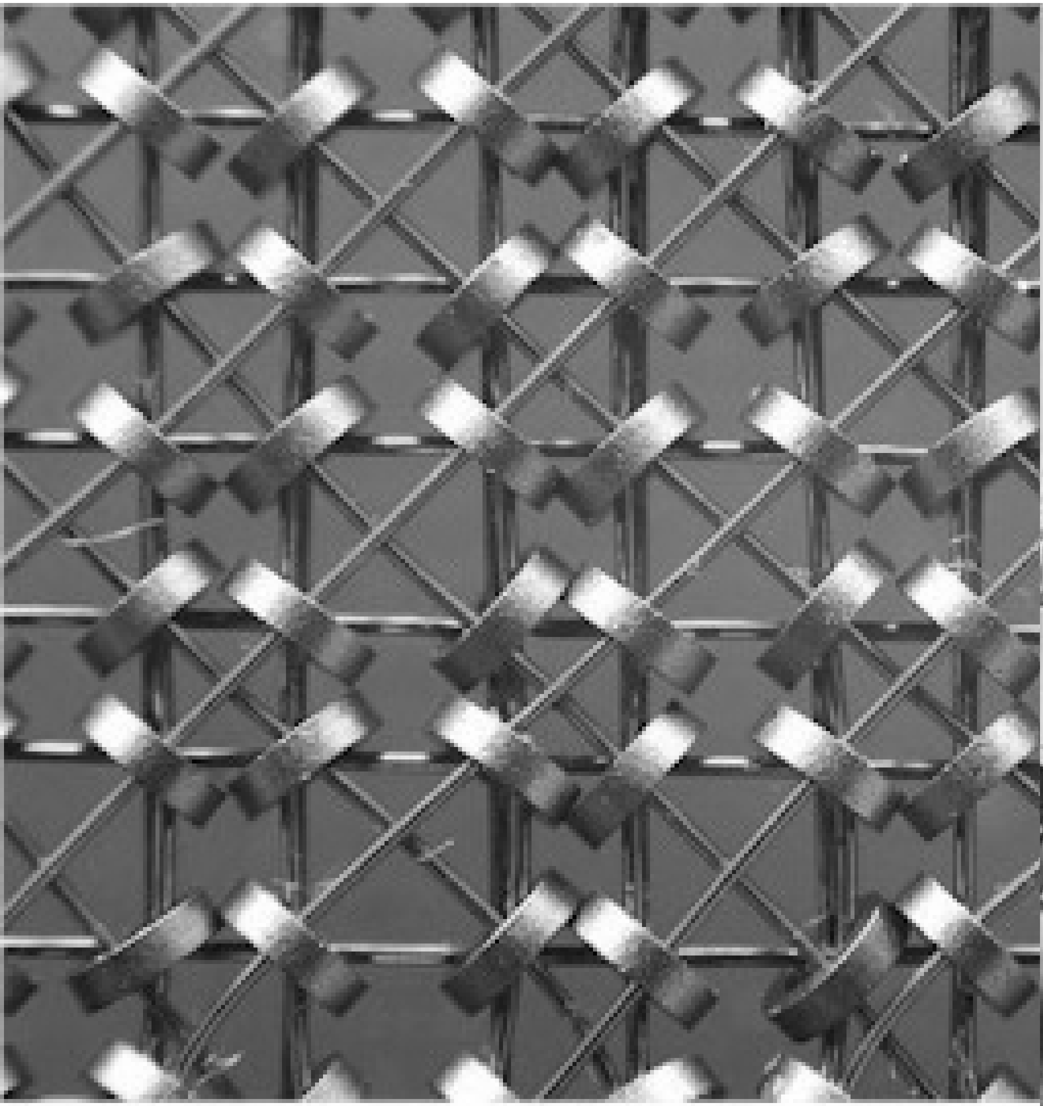
	Memristor	PCM	STT-RAM	DRAM	Flash	HD
Chip area per bit (F^2)	4	8–16	14–64	6–8	4–8	n/a
Energy per bit (pJ)[2]	0.1–3	2–100	0.1–1	2–4	10^1–10^4	10^6–10^7
Read time (ns)	<10	20–70	10–30	10–50	25,000	5–8x10^6
Write time (ns)	20–30	50–500	13–95	10–50	200,000	5–8x10^6
Retention	>10 years	<10 years	Weeks	<Second	~10 years	~10 years
Endurance (cycles)	~10^{12}	10^7–10^8	10^{15}	>10^{17}	10^3–10^6	10^{15} ?
3D capability	Yes	No	No	No	Yes	n/a

Source:
http://www8.hp.com/hpnext/posts/beyond-dram-and-flash-part-2-new-memory-technology-data-deluge

Moving the Focus to SW Latency



Memory or Storage?



Persistence:

Storage

- Block I/O only
- Sync or Async
  - DMA master

- High Capacity?
- Drive-serviceability?

- NAND, NVMe, PCIe

pmem

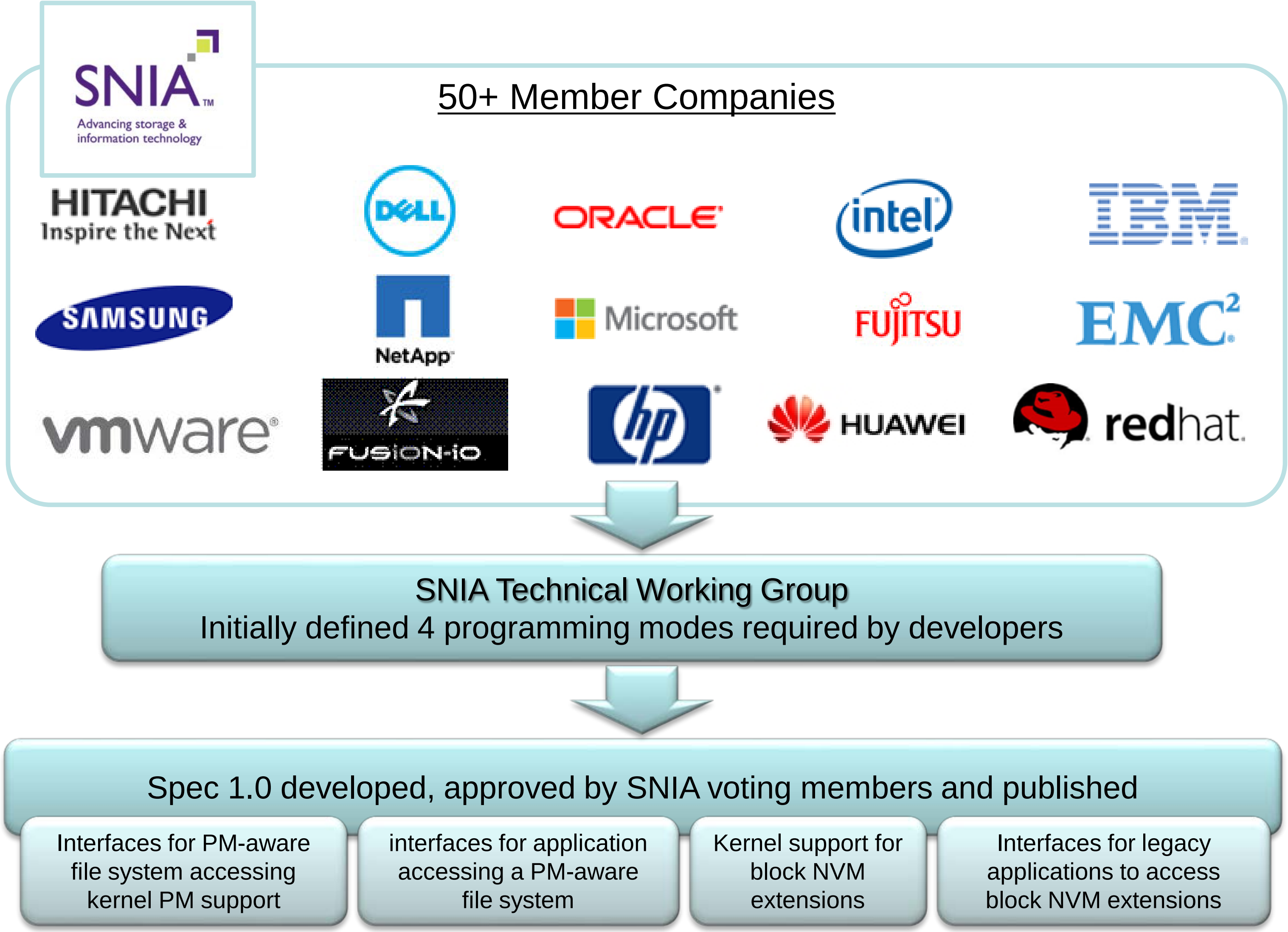
- Byte addressable
- Sync (probably)
  - DMA slave

- Growing Capacity?
- NVDIMM

- Not NAND, NVMe
  - PCIe?

pmem: The New Tier

- Byte-addressable memory, but persistent
- Must be *reasonable* to stall a CPU waiting for a load to finish
  - So, not NAND NVM based
- Can do small I/O
  - DIMMs are 64B cache line accessible
- Can DMA to it
  - Receive data from network directly to persistence!



50+ Member Companies

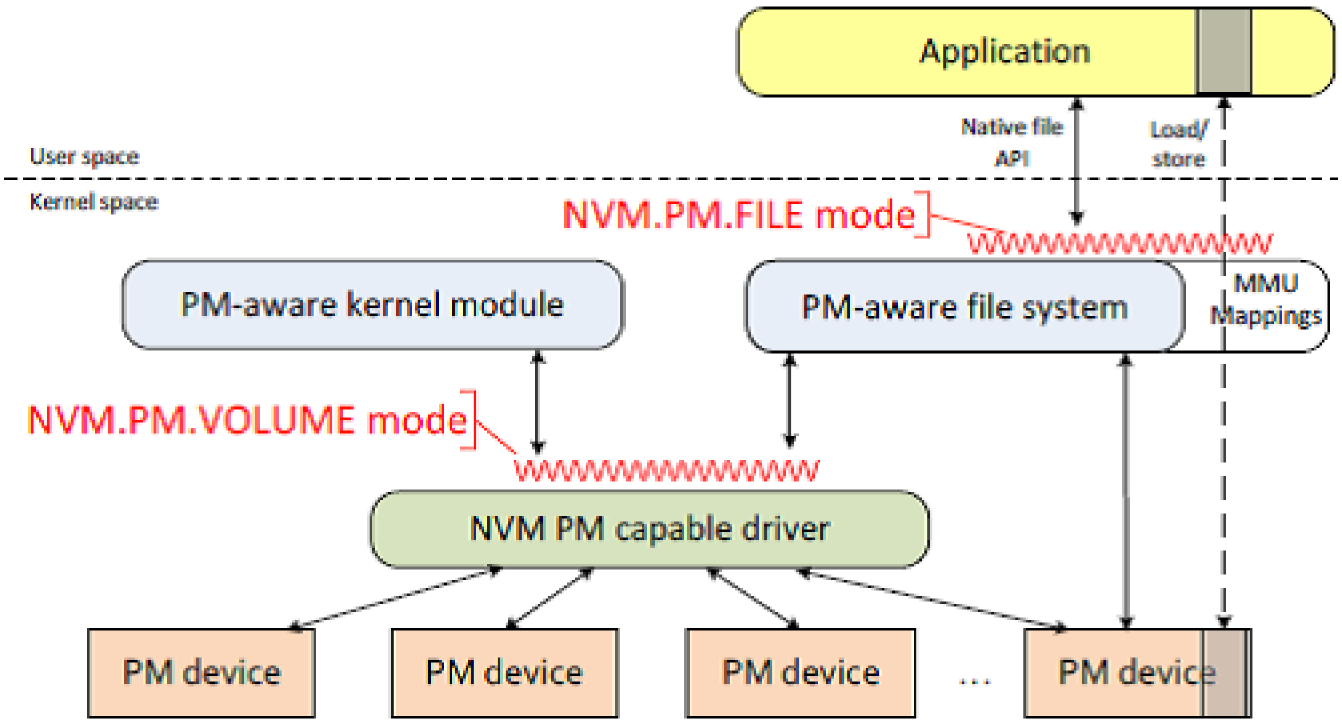
SNIA Technical Working Group
Initially defined 4 programming modes required by developers

Spec 1.0 developed, approved by SNIA voting members and published

- Interfaces for PM-aware file system accessing kernel PM support
- interfaces for application accessing a PM-aware file system
- Kernel support for block NVM extensions
- Interfaces for legacy applications to access block NVM extensions

http://snia.org/sites/default/files/NVMProgrammingModel_v1.pdf

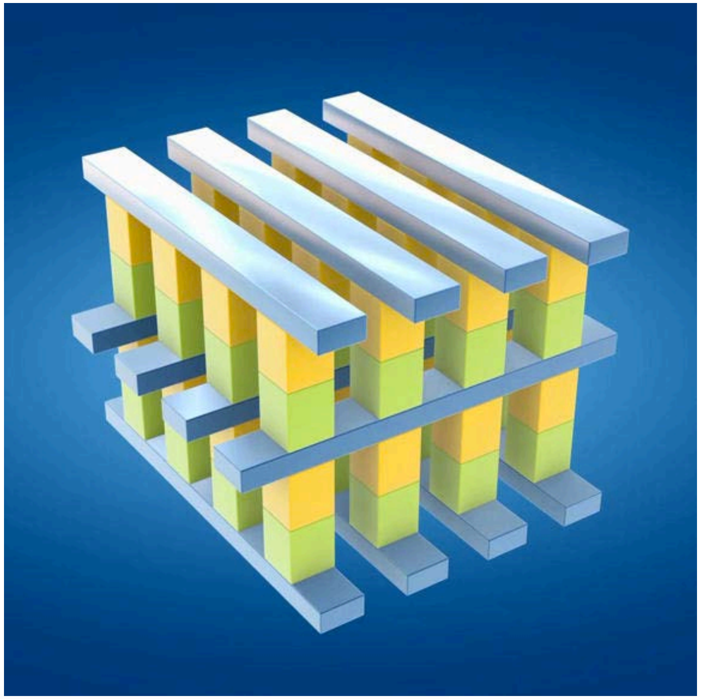
Defining the NVM Programming Model

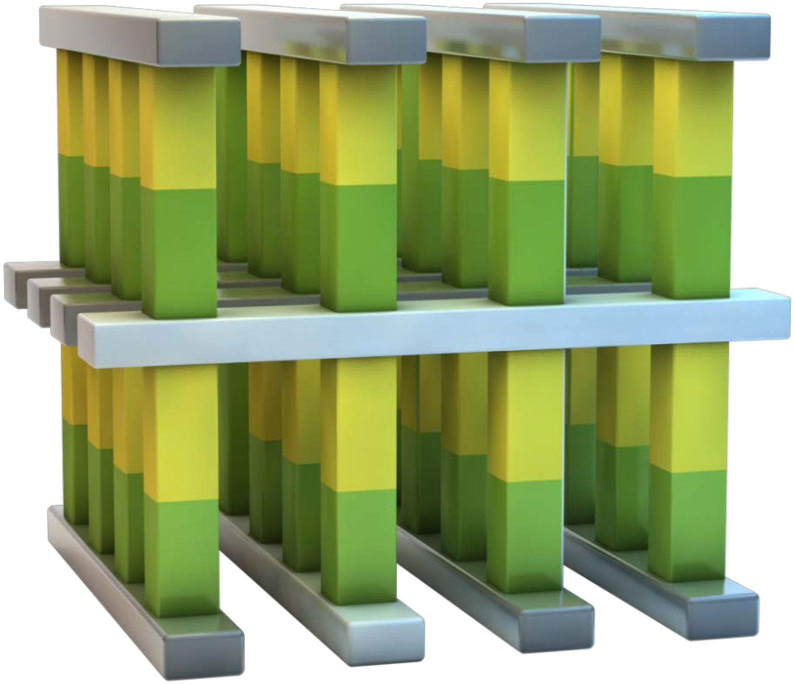


Recent Announcements

- Intel® 3D XPoint™ Technology

- The Intel DIMM





1000X
faster

THAN NAND

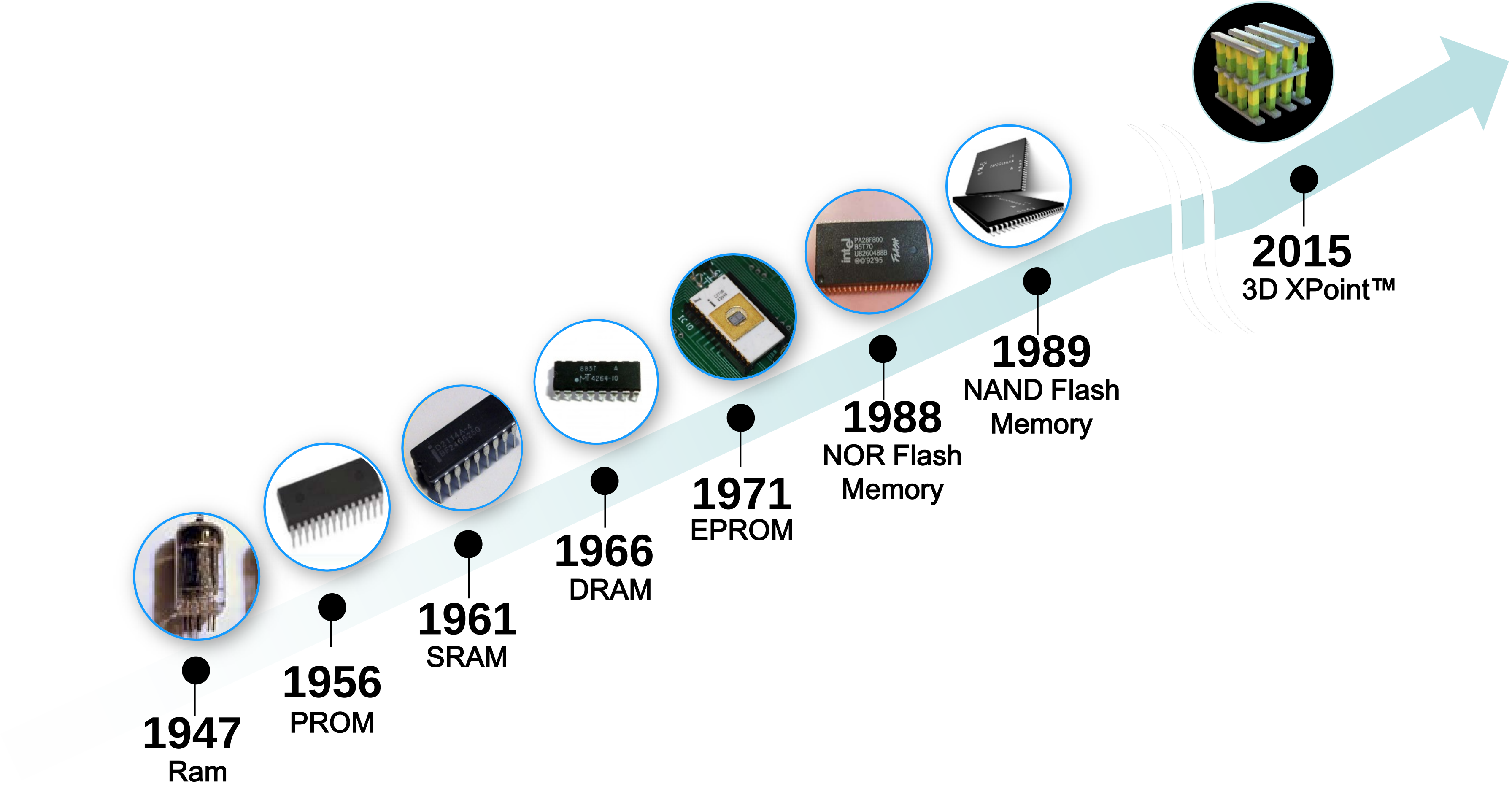
1000X
endurance

OF NAND

10X
denser

THAN CONVENTIONAL MEMORY

The Memory Timeline



1947
Ram

1956
PROM

1961
SRAM

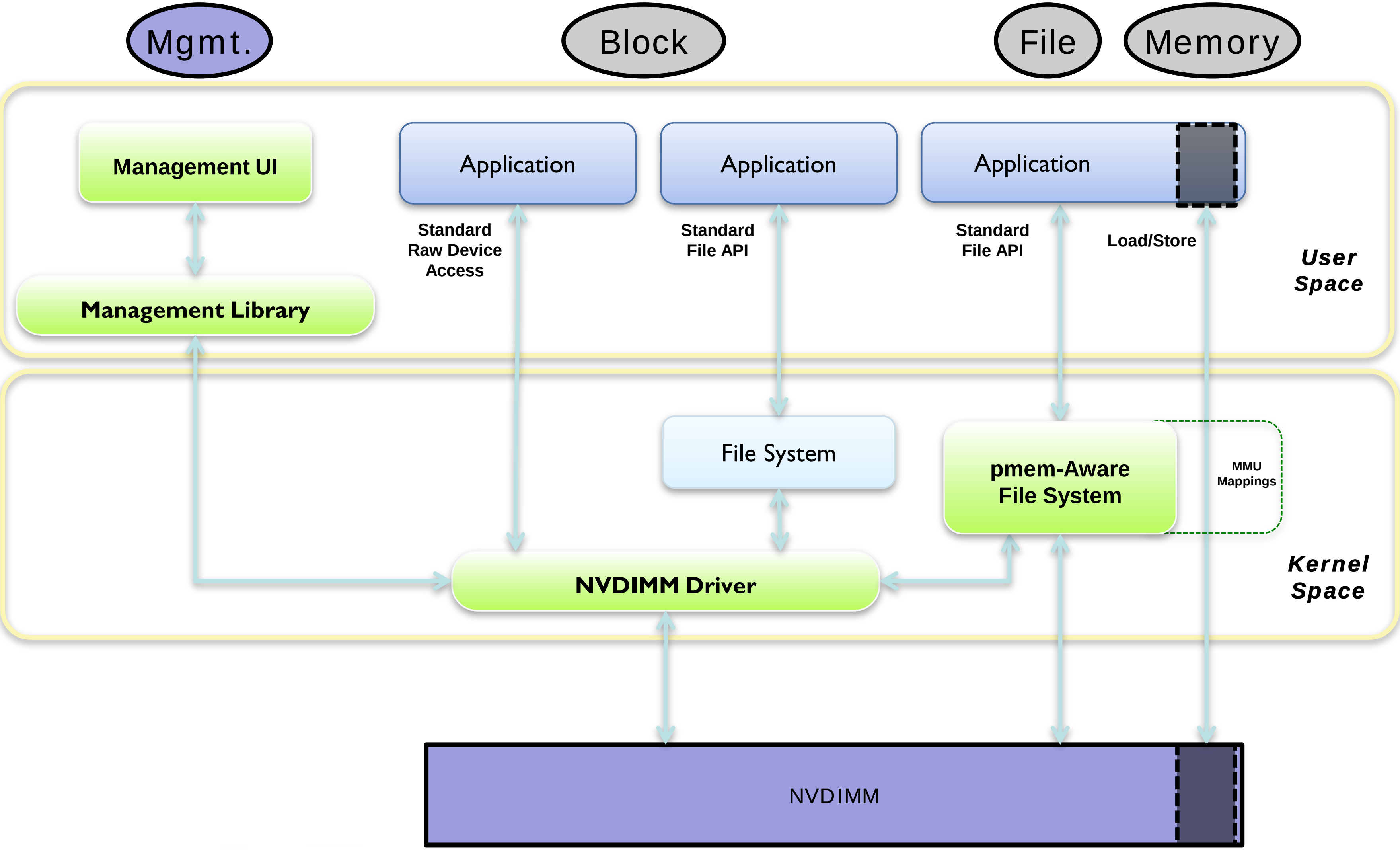
1966
DRAM

1971
EPROM

1988
NOR Flash
Memory

1989
NAND Flash
Memory

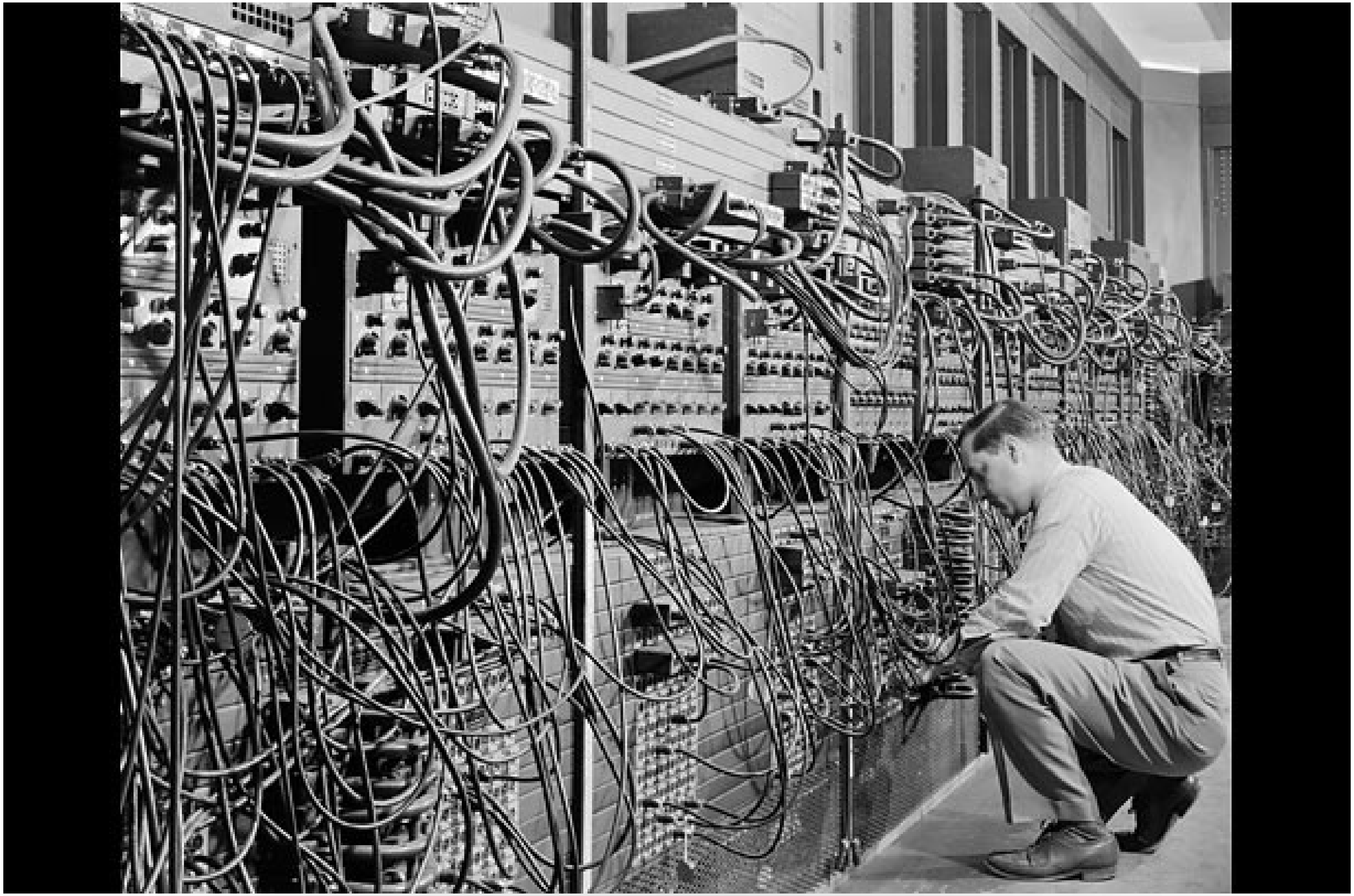
2015
3D XPoint™


Mgmt.
Block
File
Memory
Management UI
Application
Application
Application
Management Library
Standard
Raw Device
Access
Standard
File API
Standard
File API
Load/Store
User
Space
File System
pmem-Aware
File System
MMU
Mappings
NVDIMM Driver
Kernel
Space
NVDIMM

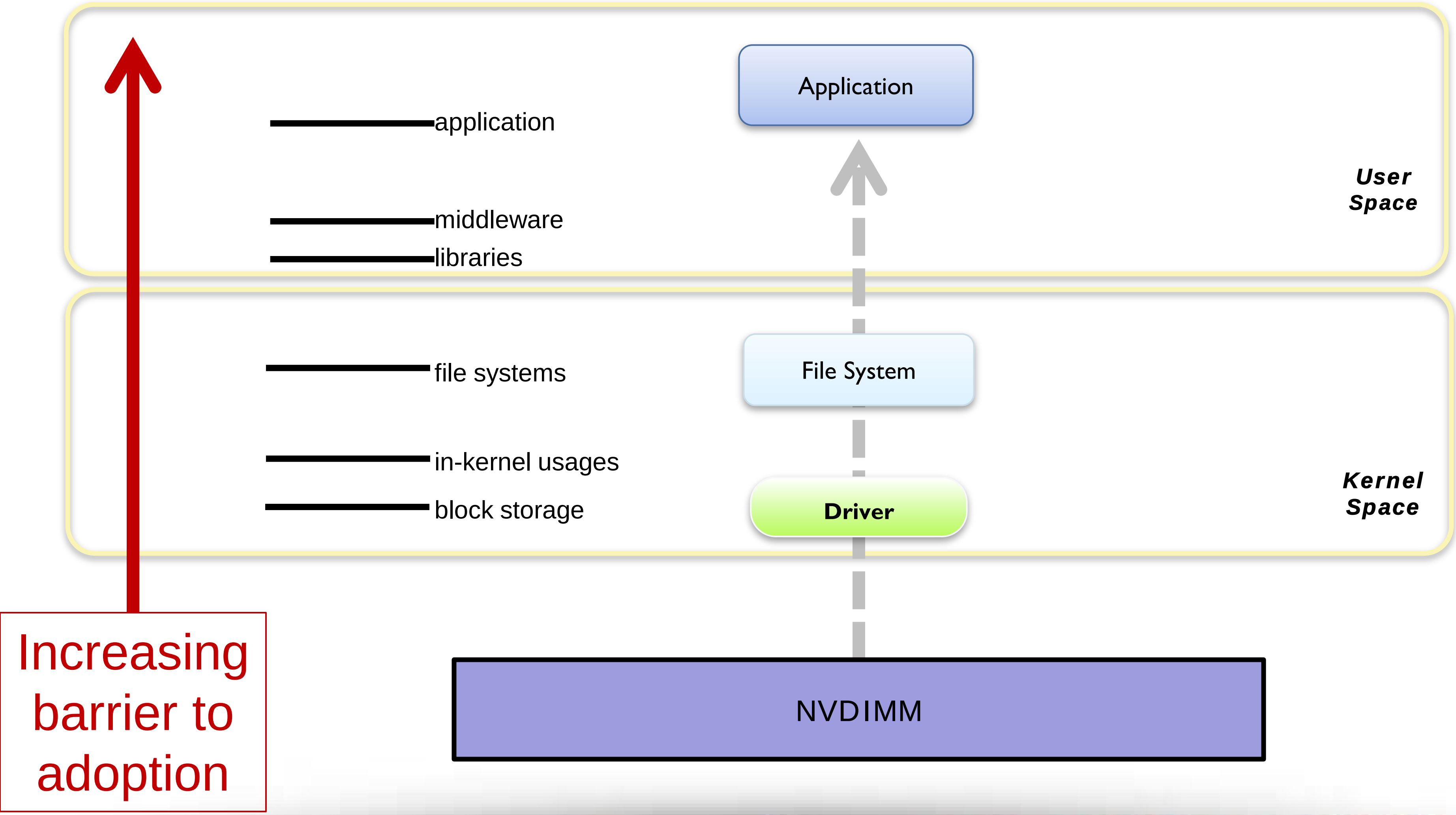
(Announced) State of SW Ecosystem

- Detecting pmem
  - BIOS creates ACPI-style information for OS
  - Defined in ACPI 6.0
- Linux support upstream
  - Exposing pmem as block storage
    - Generic NVDIMM driver for Linux released
  - Exposing pmem for direct access
    - Linux DAX upstream
  - Naming pmem areas
    - Linux ext4+DAX support upstream
  - KVM Changes upstream
- Support in other operating systems emerging
  - Neal’s talk Yesterday
    - *Storage Class Memory Support in the Windows Operating System*
  - Heavy OSV Involvement in TWG

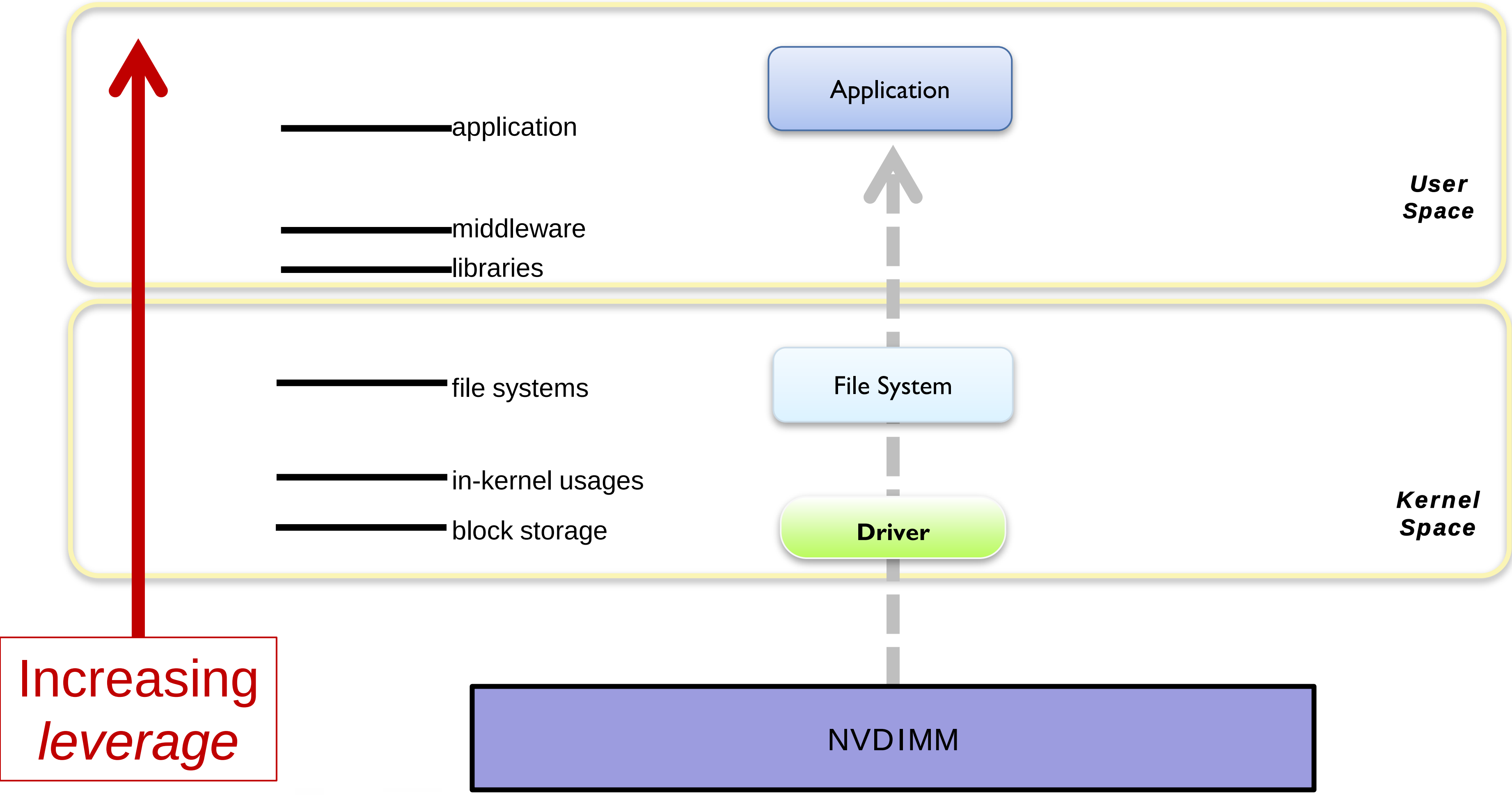
The Next Decade…



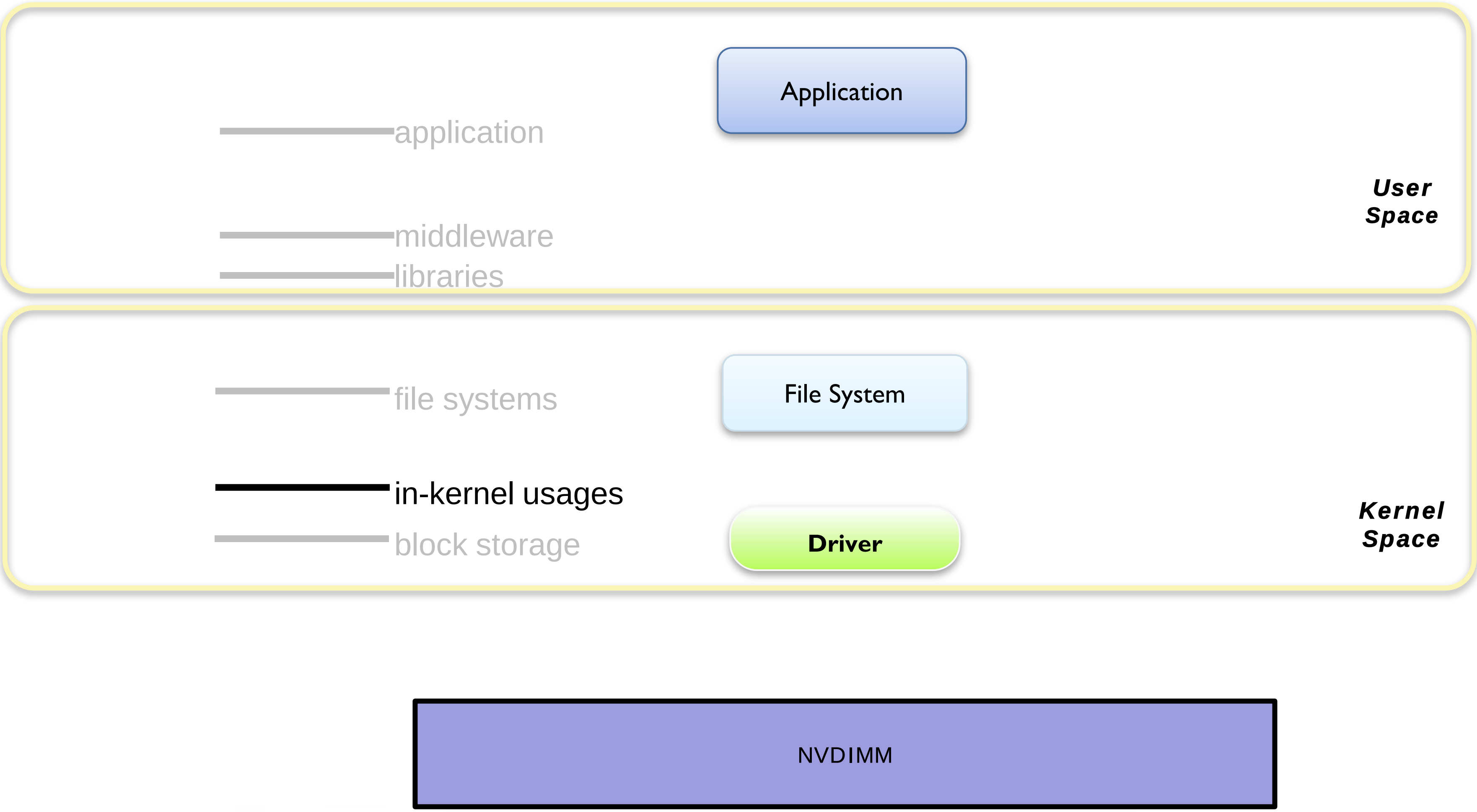
Transparency Levels



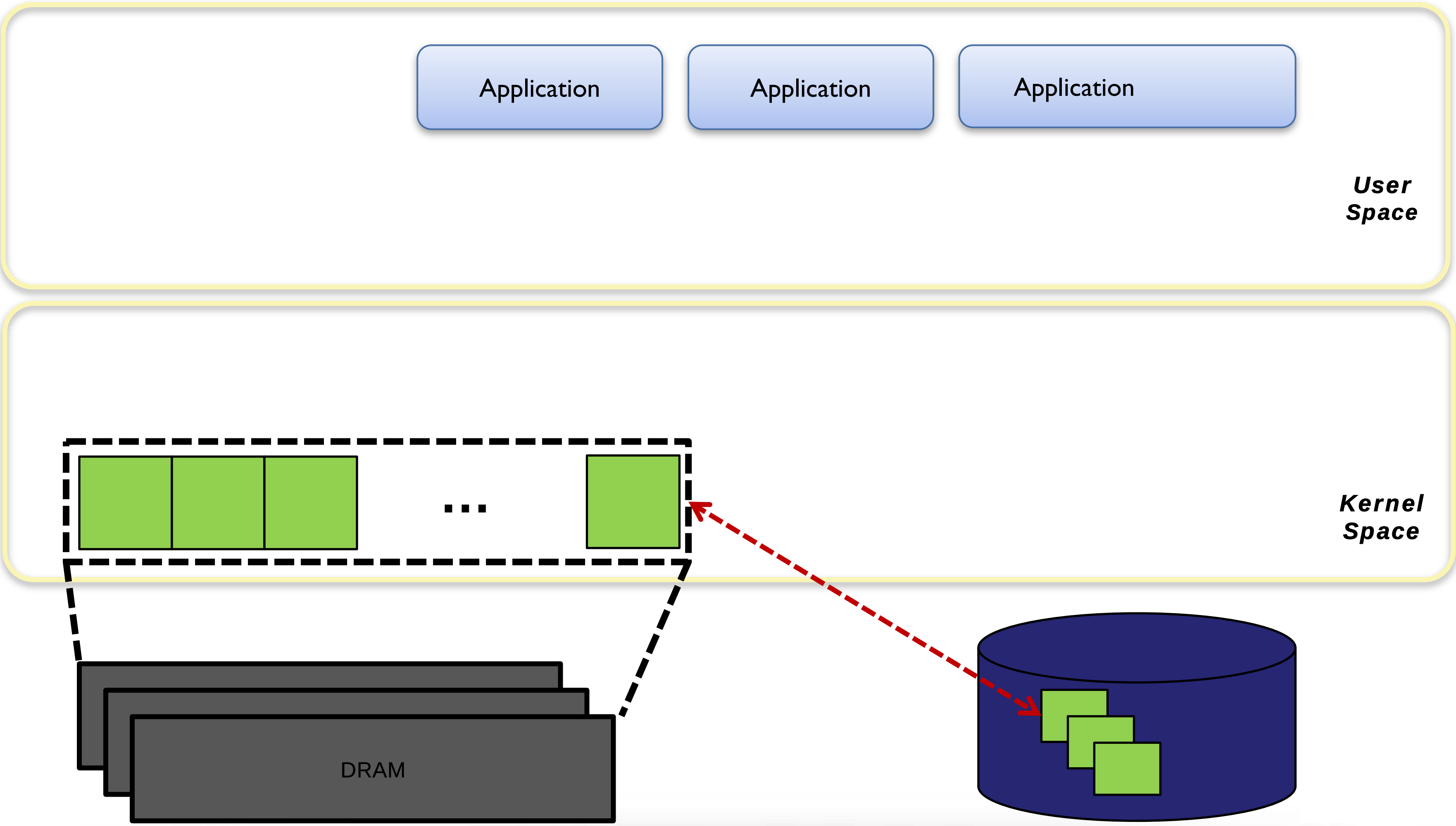
Transparency Levels



One Transparent Example: *pmem Paging*



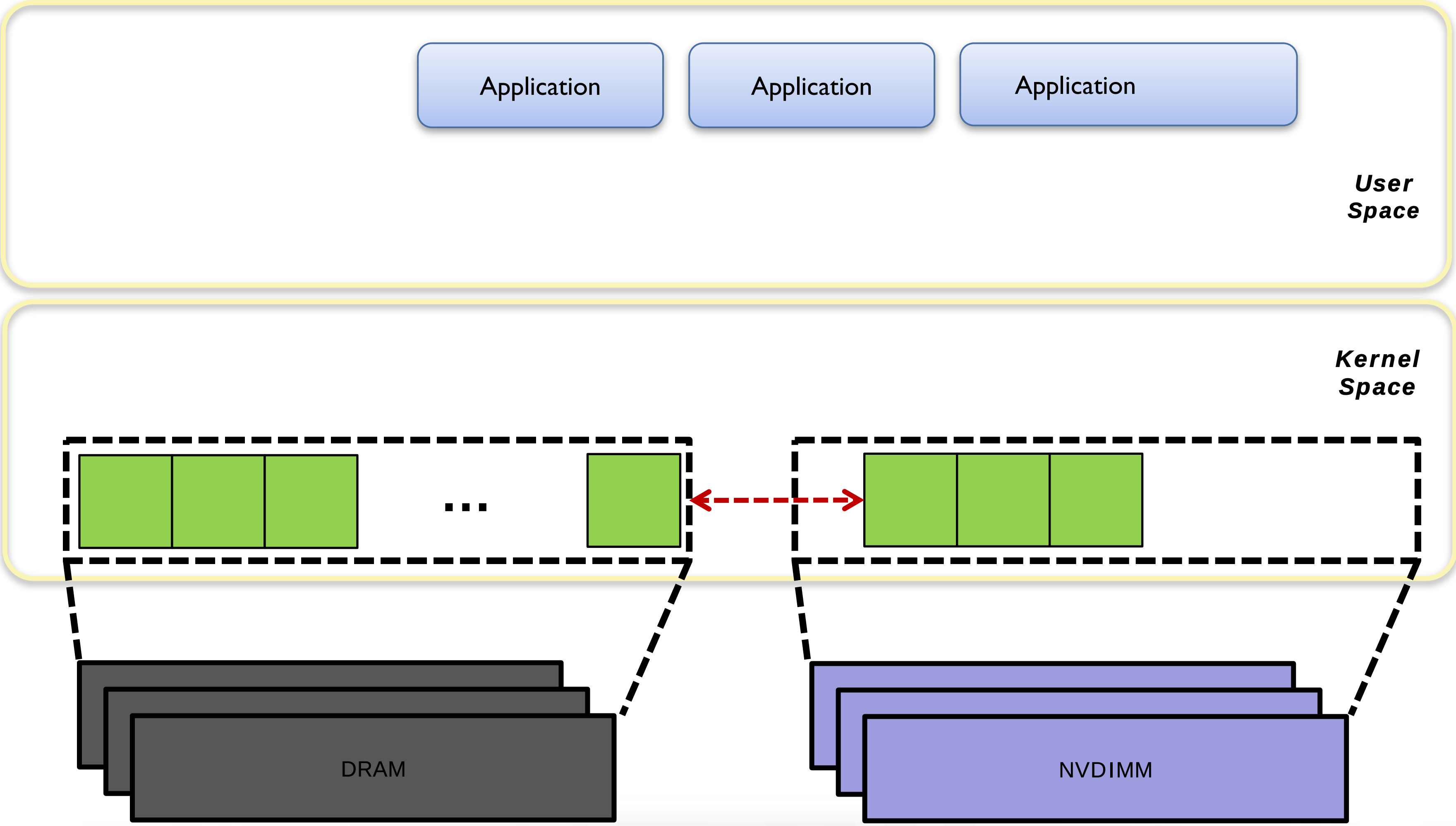
Paging from the OS Page Cache



Attributes of Paging
(and why everyone avoids it)

- Major page faults
  - Block I/O (page I/O) on demand
  - Context switch – there and back again
  - Latency of block stack
- Available memory looks much larger
  - But penalty of fault is significant
- Page in must pick a victim
  - Based on simplistic R/M metric
  - Can surprise an application
- Many enterprise apps opt-out
  - Managing page cache themselves
  - Using intimate date knowledge for paging decisions
- Interesting example: Java GC

Paging to pmem



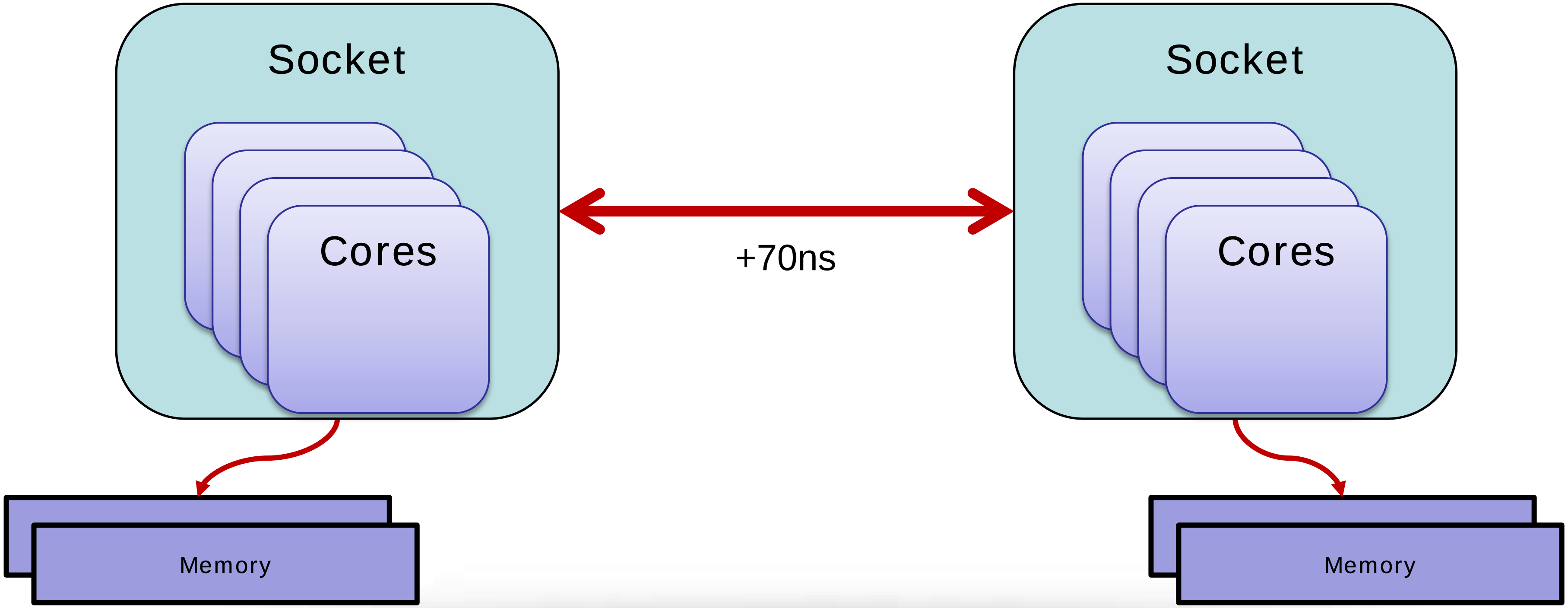
When Will pmem Paging be Cost Effective?

- When pmem costs less than (or close to) DRAM
- When pmem performance approaches DRAM
- When pmem capacity becomes significant

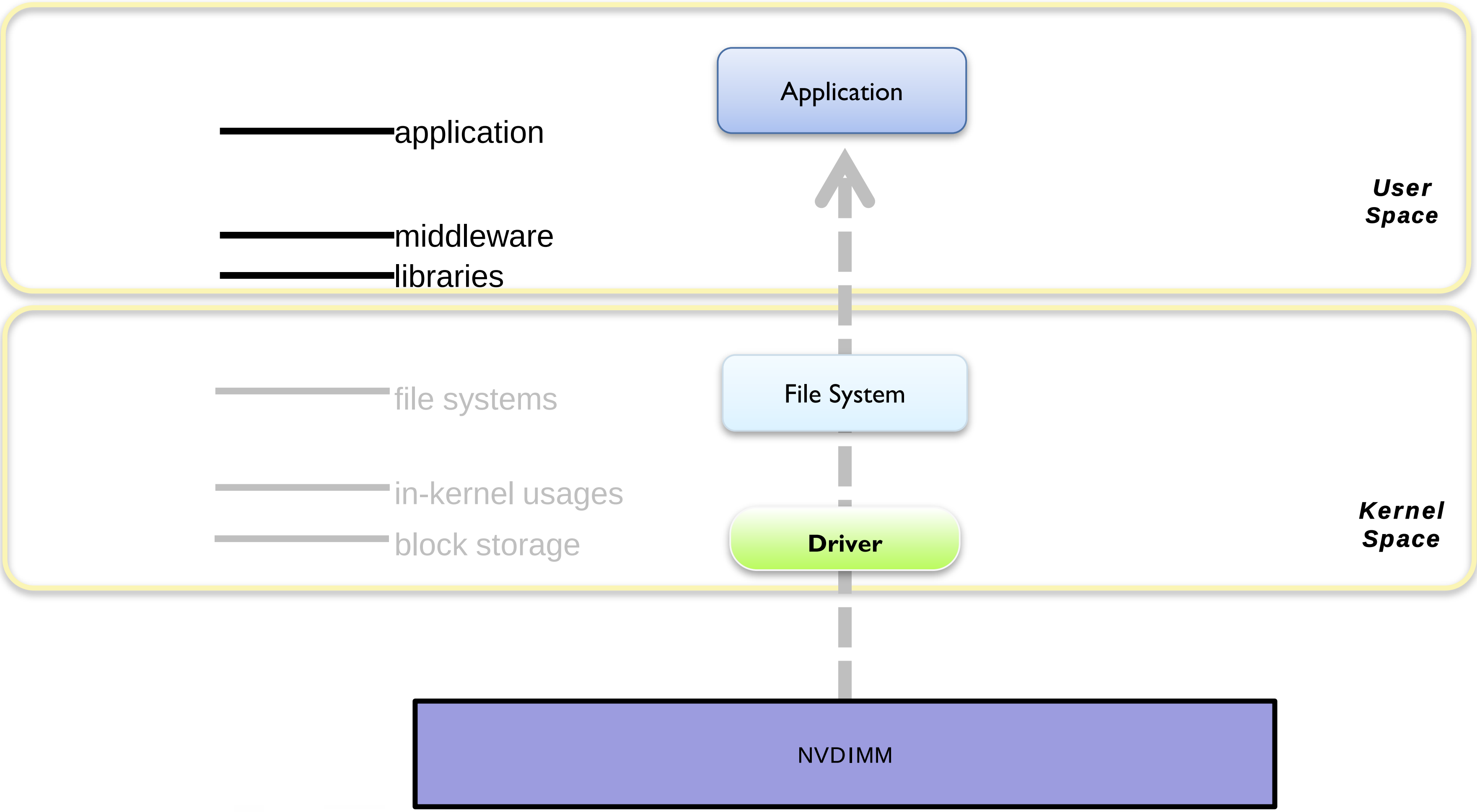
- When pmem is as reliable as memory
  - Probably needs to exceed memory reliability due to the fact it is persistent

Not just for pmem…

- High-bandwidth memory
- NUMA localities
- Different NVM technologies



Extending into User Space



application

middleware

libraries

file systems

in-kernel usages

block storage

Application

File System

Driver

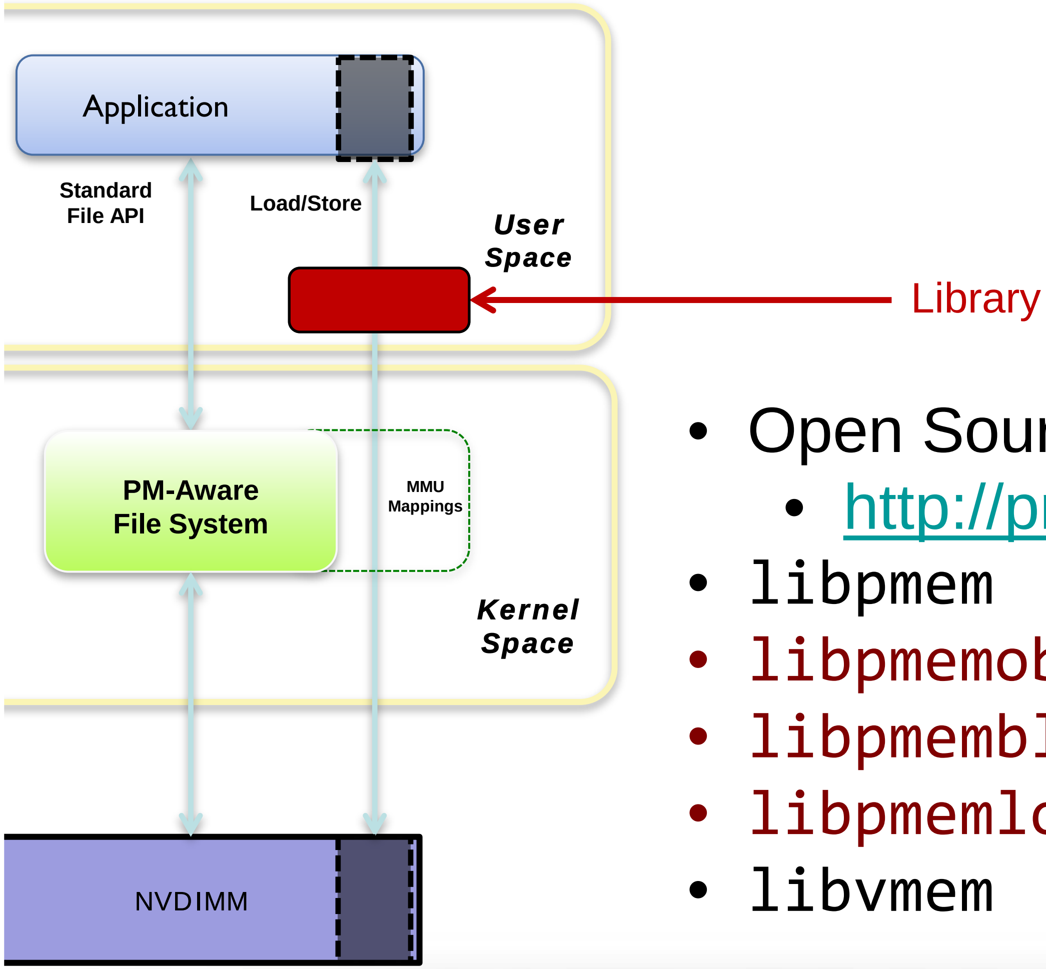
NVDIMM

User Space

Kernel Space

NVM Library: pmem.io

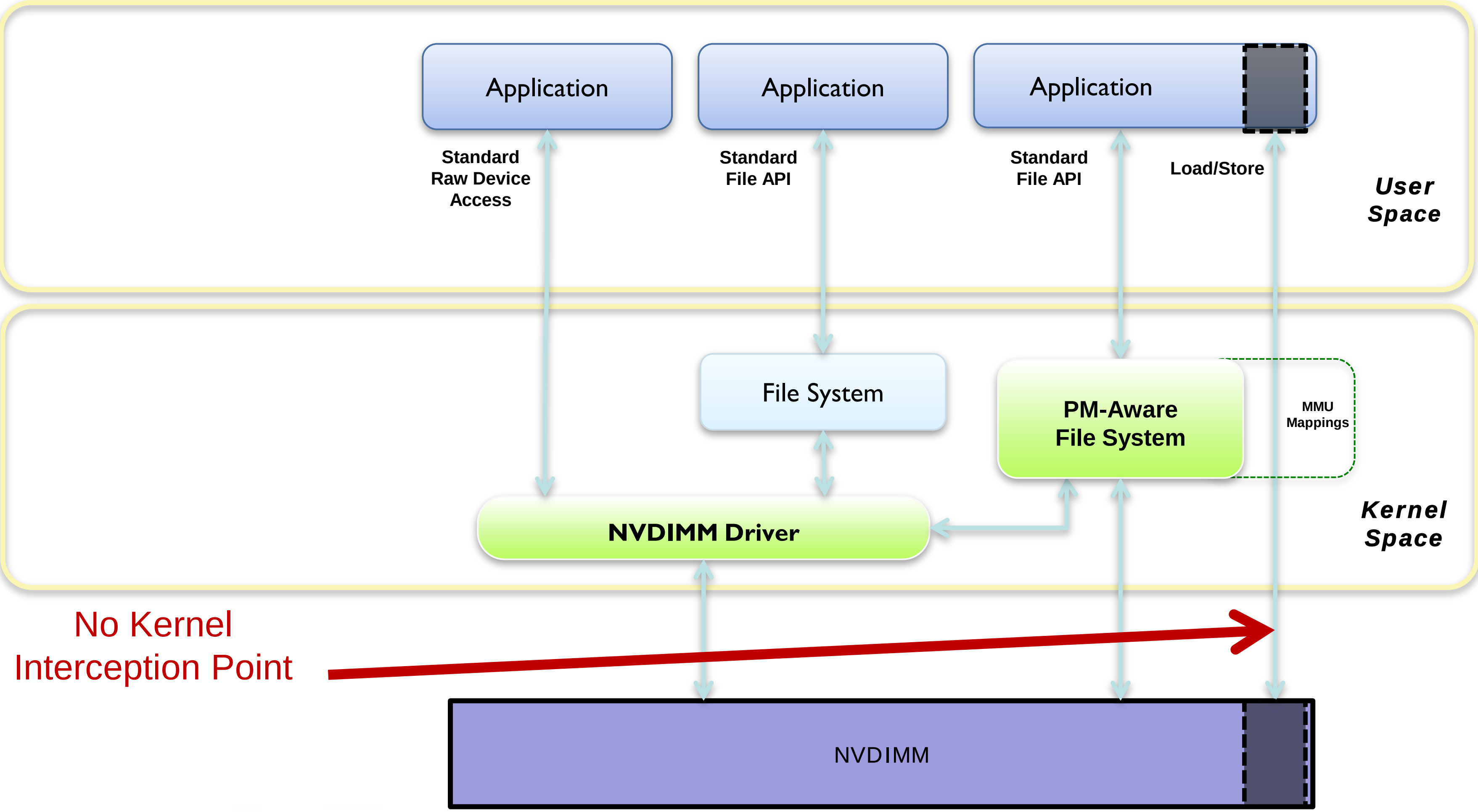
64-bit Linux Alpha Release



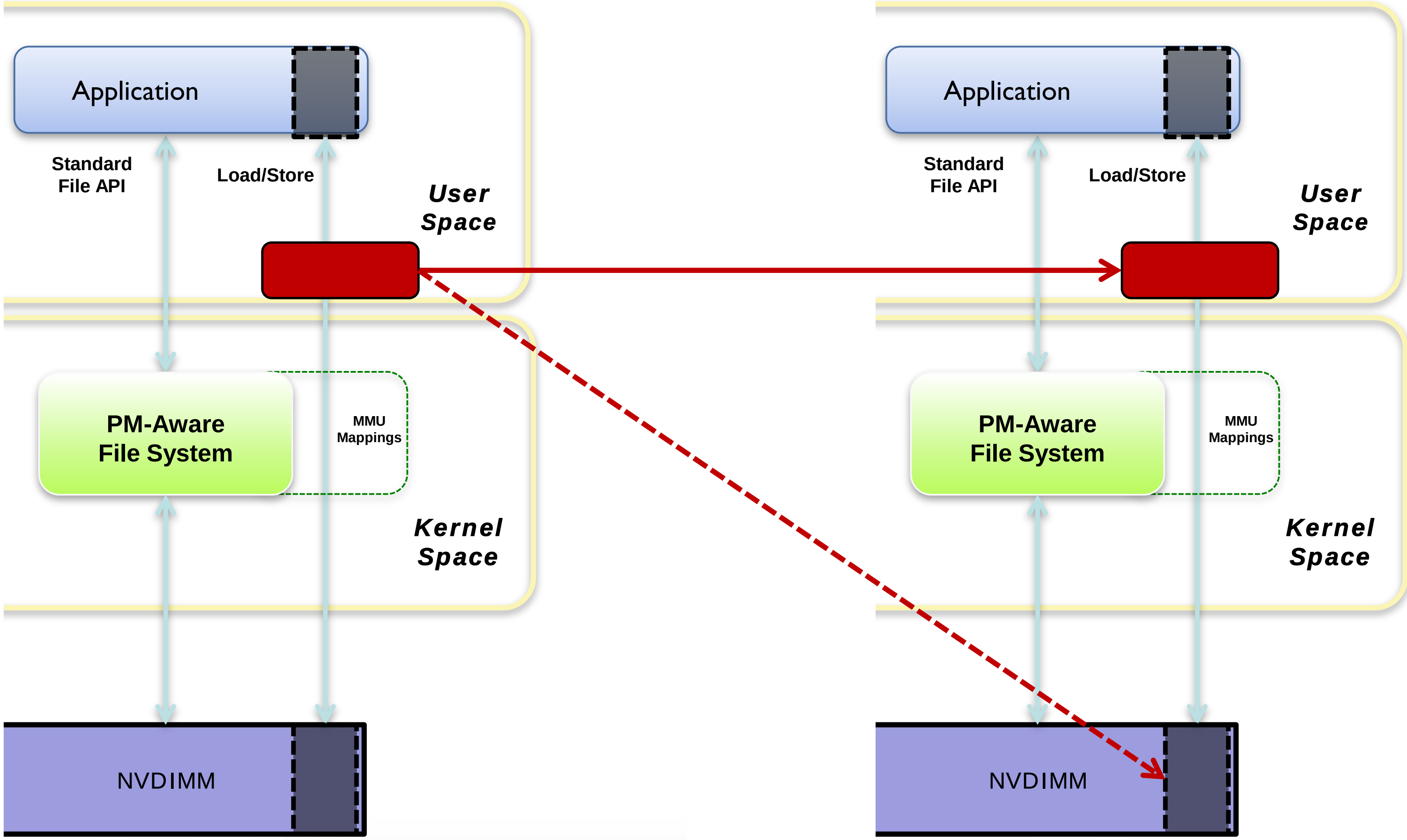
- Open Source
  - http://pmem.io
- `libpmem`
- `libpmemobj`
- `libpmemblk`
- `libpmemlog`
- `libvmem`

Transactional

Replication Challenge of pmem



RDMA to pmem

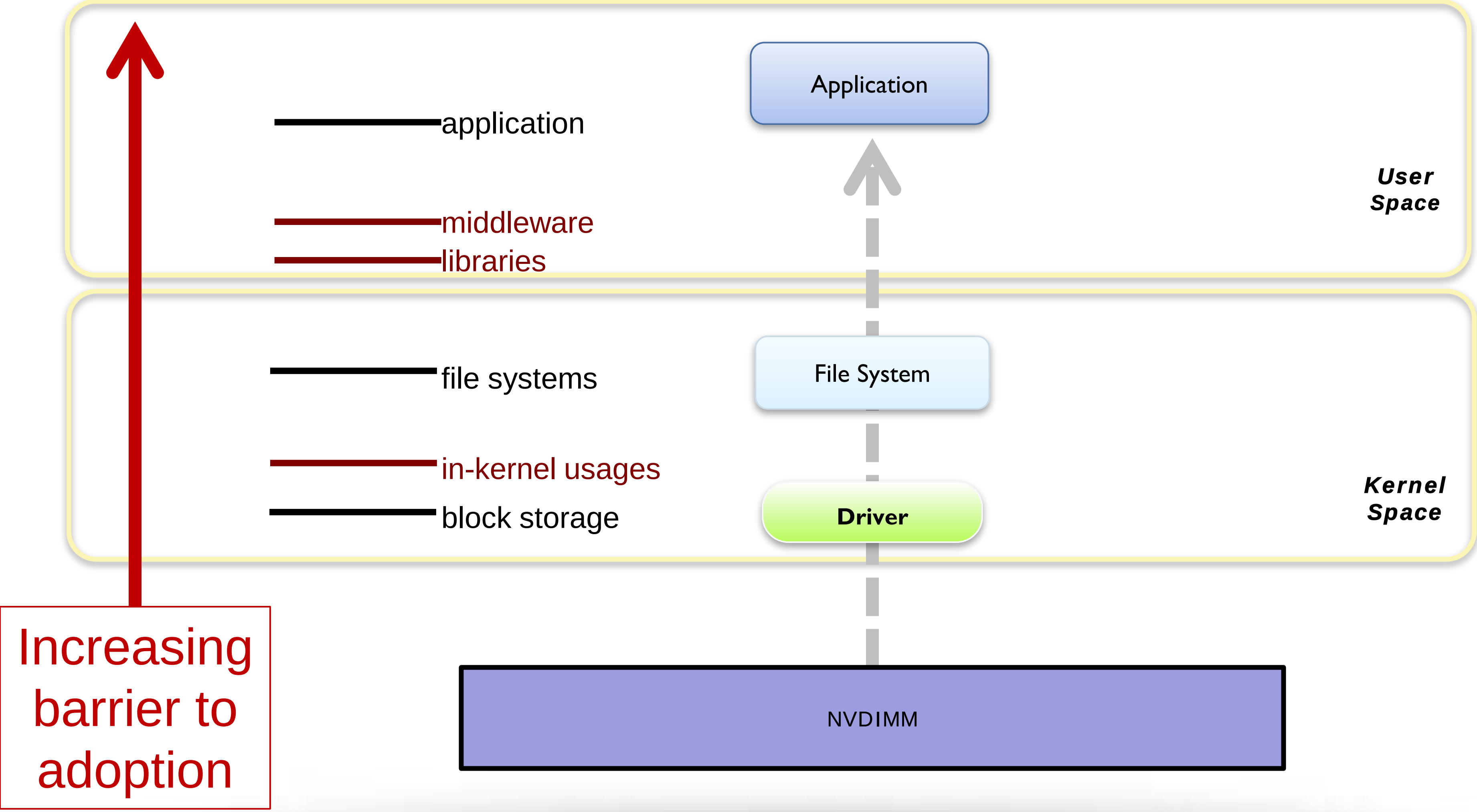

Application
Standard File API
Load/Store
User Space
PM-Aware File System
MMU Mappings
Kernel Space
NVDIMM
Application
Standard File API
Load/Store
User Space
PM-Aware File System
MMU Mappings
Kernel Space
NVDIMM

Non-Transparent pmem Use Cases

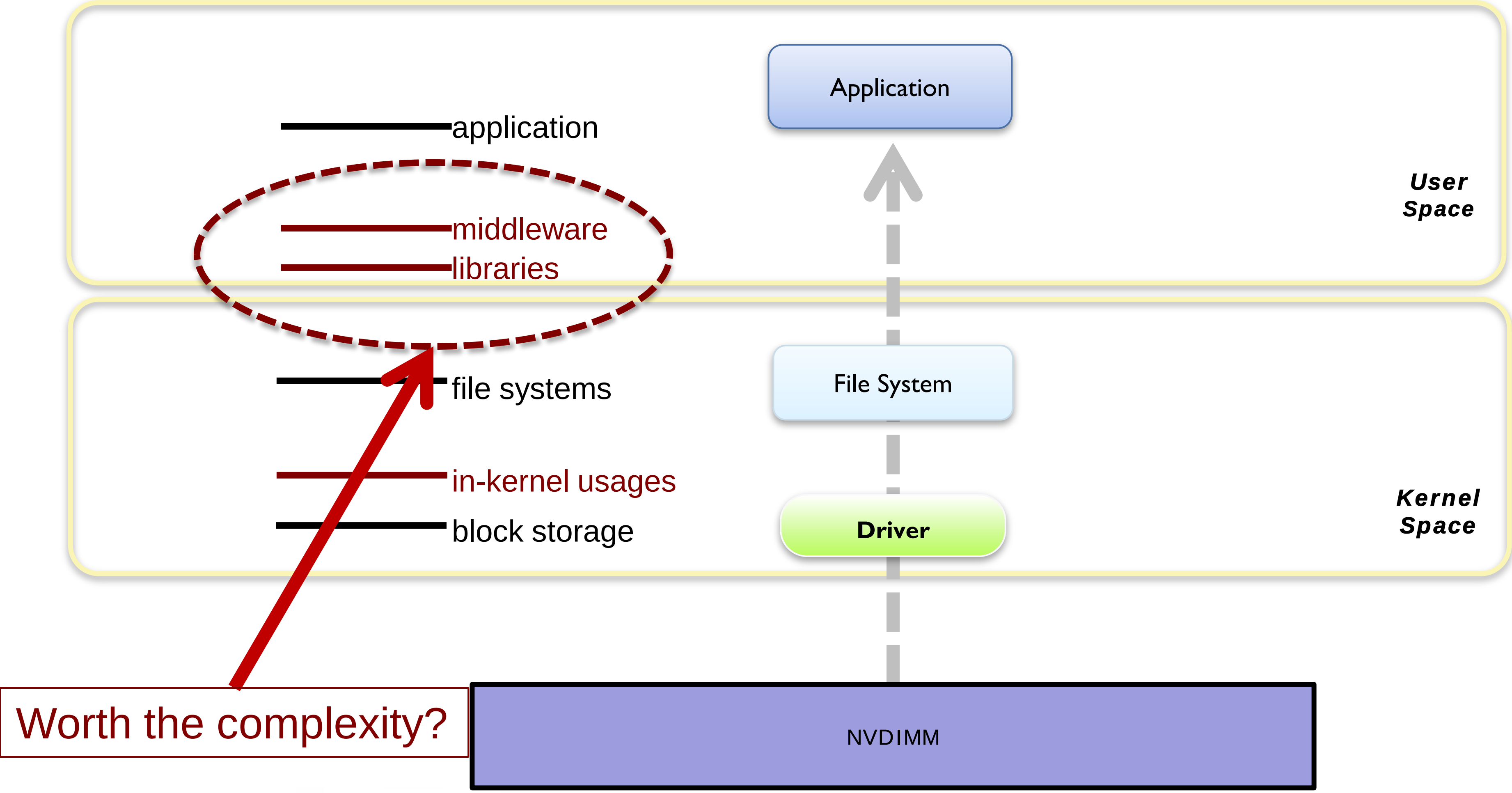
- Volatile caching
  - Due to capacity, relative simplicity

- In-memory database
- Storage appliance write cache
  - Also for large structures like dedup tables
  - Leverage RDMA capability
- Large, byte-addressable data structures
  - Example: HBASE hash table
- HPC
  - Example: checkpoint
  - Example: distributed versioned object store

Prediction: The Sweet Spots

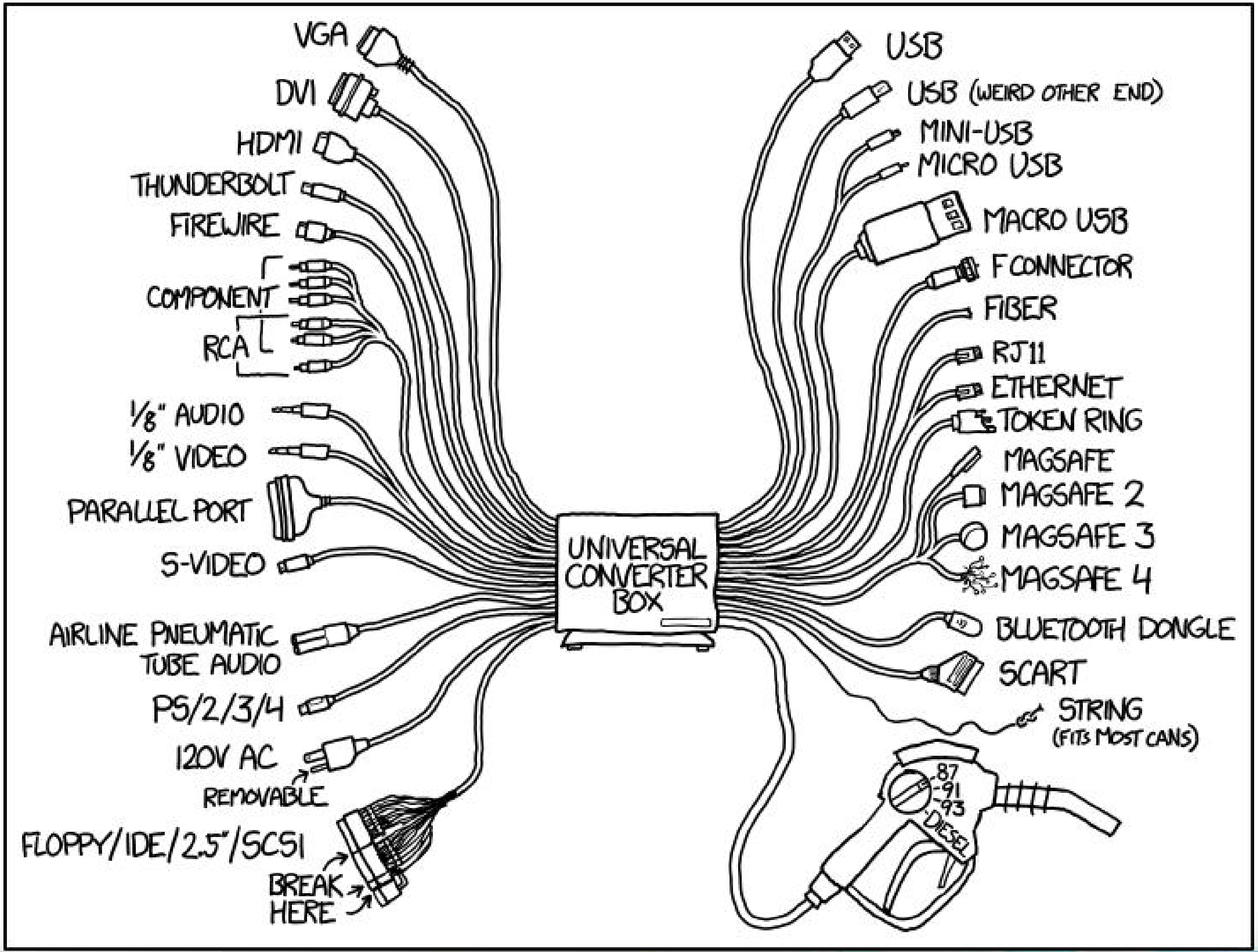


Prediction: The Big Challenge



Worth the complexity?

Summary…


VGA
DVI
HDMI
THUNDERBOLT
FIREWIRE
COMPONENT
RCA
1/8" AUDIO
1/8" VIDEO
PARALLEL PORT
S-VIDEO
AIRLINE PNEUMATIC TUBE AUDIO
PS/2/3/4
120V AC
REMOVABLE
FLOPPY/IDE/2.5"/SCSI
BREAK HERE
UNIVERSAL CONVERTER BOX
USB
USB (WEIRD OTHER END)
MINI-USB
MICRO USB
MACRO USB
F CONNECTOR
FIBER
RJ11
ETHERNET
TOKEN RING
MAGSAFE
MAGSAFE 2
MAGSAFE 3
MAGSAFE 4
BLUETOOTH DONGLE
SCART
STRING
(FITS MOST CANS)
87
-91
-93
DIESEL

Summary

- Building a SW ecosystem for pmem
  - Won’t overcome Enterprise time-to-adoption, but…
  - Linux support upstream
  - Other operating systems progressing
- Cost versus Benefit Challenge
  - Cost of Emerging NVM
  - Cost of application complexity
  - Fall back to transparency at various levels
- What you can do to prepare
  - Learn NVM programming model
  - Map use cases to pmem
  - Contribute to libraries, SW ecosystem

Links to More Information

- SNIA NVM Programming Model
  - http://www.snia.org/forums/sssi/nvmp
- Intel® Architecture Instruction Set Extensions Programming Reference
  - https://software.intel.com/en-us/intel-isa-extensions
- Open Source NVM Library work
  - http://pmem.io
- Linux kernel support & instructions
  - https://github.com/01org/prd
- ACPI 6.0 NFIT definition (used by BIOS to expose NVDIMMs to OS)
  - http://www.uefi.org/sites/default/files/resources/ACPI_6.0.pdf
- Open specs providing NVDIMM implementation examples, layout, BIOS calls:
  - http://pmem.io/documents/
- Google group for pmem programming discussion:
  - http://groups.google.com/group/pmem