



PERSISTENT MEMORY PM SUMMIT

JANUARY 23, 2020 | SANTA CLARA, CA

The Persistent Memory Programming Model

Andy Rudoff, Principal Engineer, Intel



PERSISTENT MEMORY PM SUMMIT

JANUARY 23, 2020 | SANTA CLARA, CA

The Persistent Memory Programming Model

Andy Rudoff, ~~Principal Engineer, Intel~~

Founding Member, SNIA NVM Programming TWG

The Backstory

Eight Years Summarized in Ten Minutes


SNIA Technical Workgroup Involvement

- **2012 (June)**
 - NVM Programming TWG Formed
 - Immediate Participation from key OSVs, IHVs, ISVs
- **2013 (December)**
 - SNIA Publishes the NVM Programming Model version 1.0
- **2015 (March)**
 - SNIA Publishes version 1.1
- **2017 (June)**
 - SNIA Publishes version 1.2

SNIA Starts the Summit

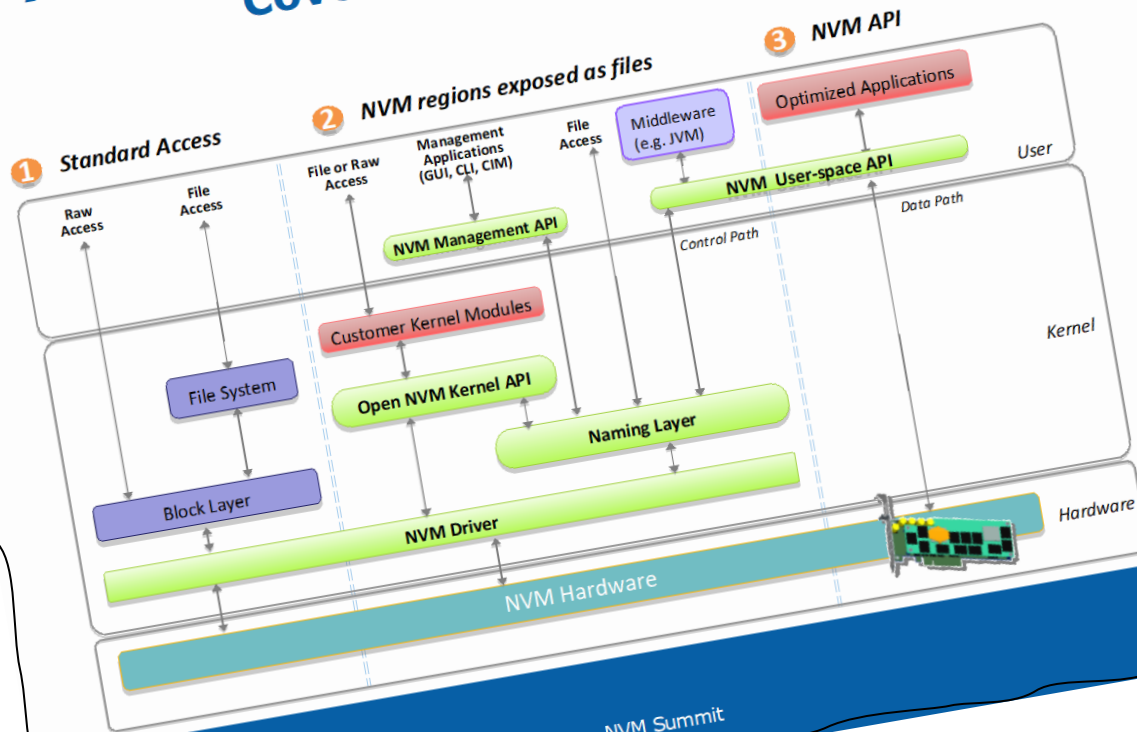
- **2012 (June)**
 - NVM Programming TWG Formed
 - Immediate Participation from key OSVs, IHVs, ISVs
- **2013 (December)**
 - SNIA Publishes the NVM Programming Model version 1.0
- **2015 (March)**
 - SNIA Publishes version 1.1
- **2017 (June)**
 - SNIA Publishes version 1.2

2013 (January)
First PM Summit (“NVM Summit”)



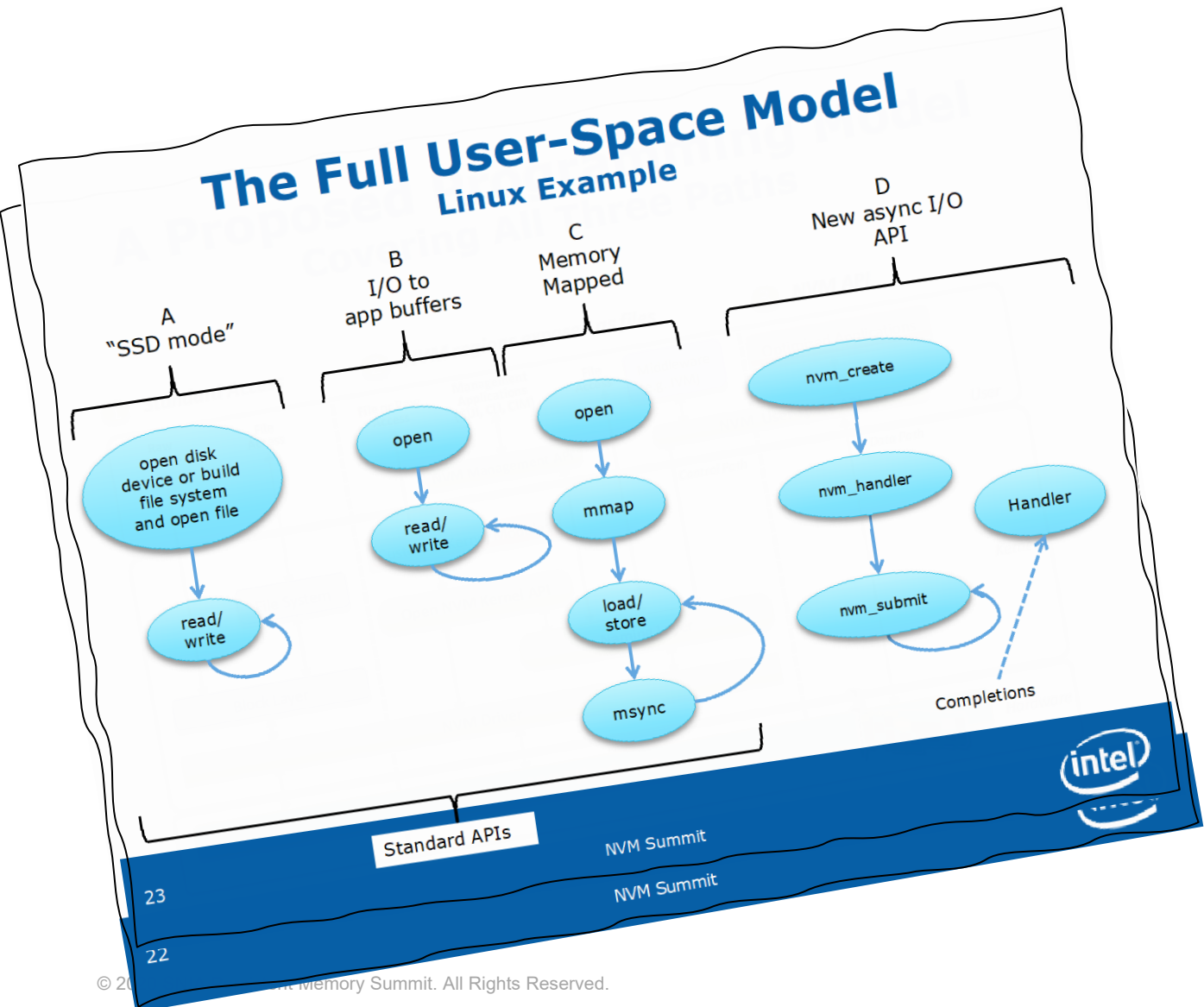
NVM Summit 2013

A Proposed Programming Model Covering All Three Paths

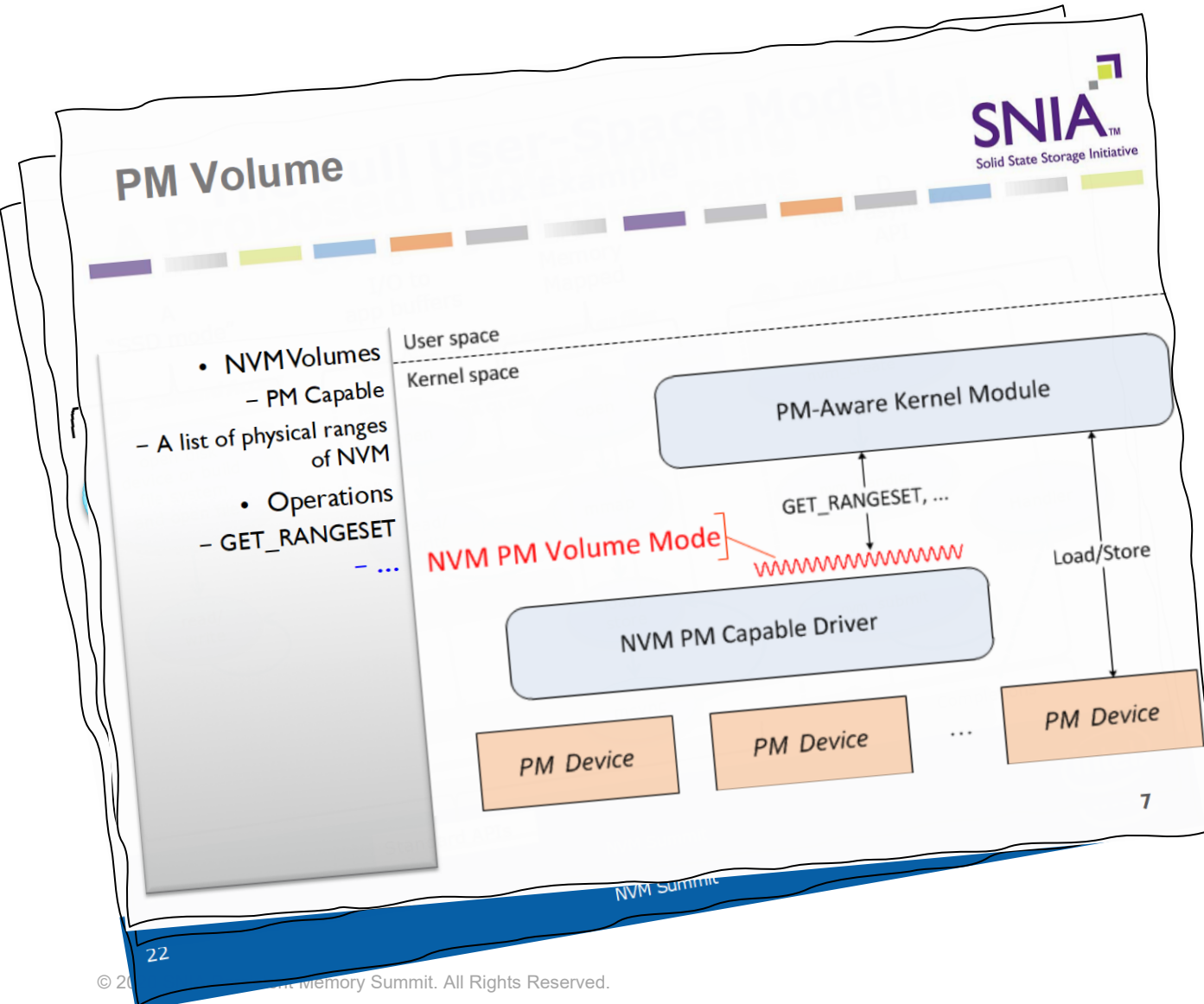


NVM Summit

NVM Summit 2013



NVM Summit 2014

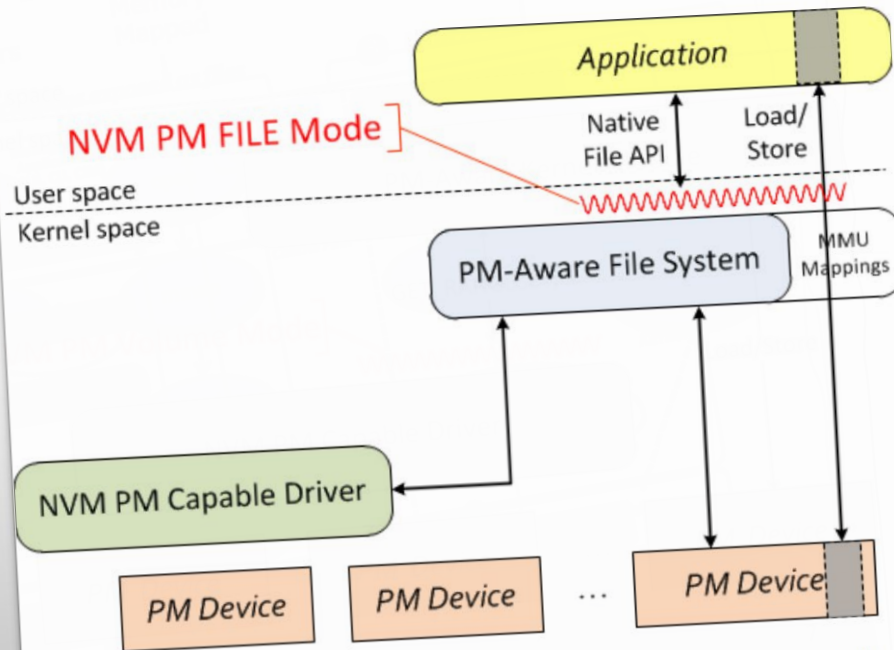


NVM Summit 2014

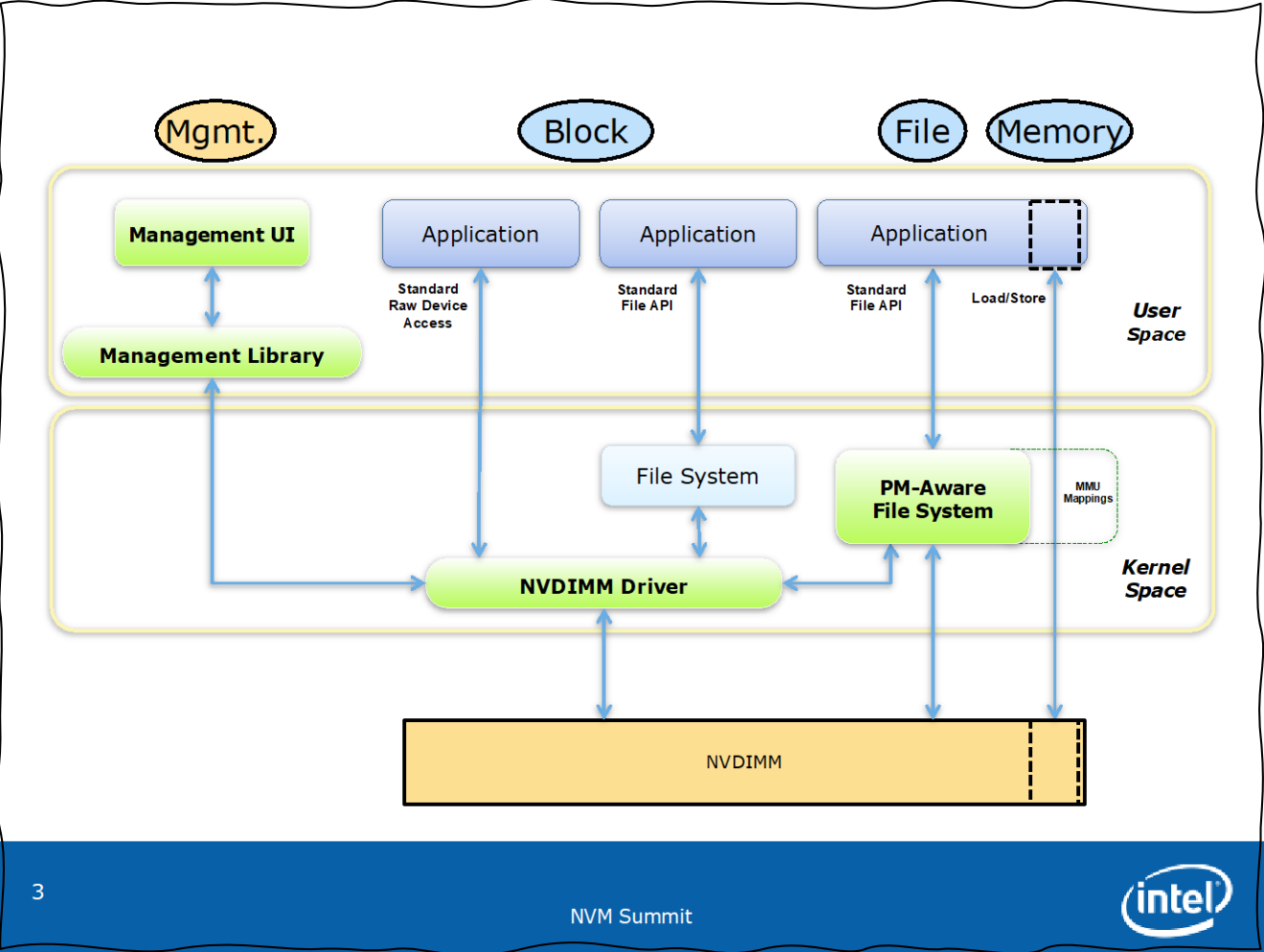
PM File

SNIA™
Solid State Storage Initiative

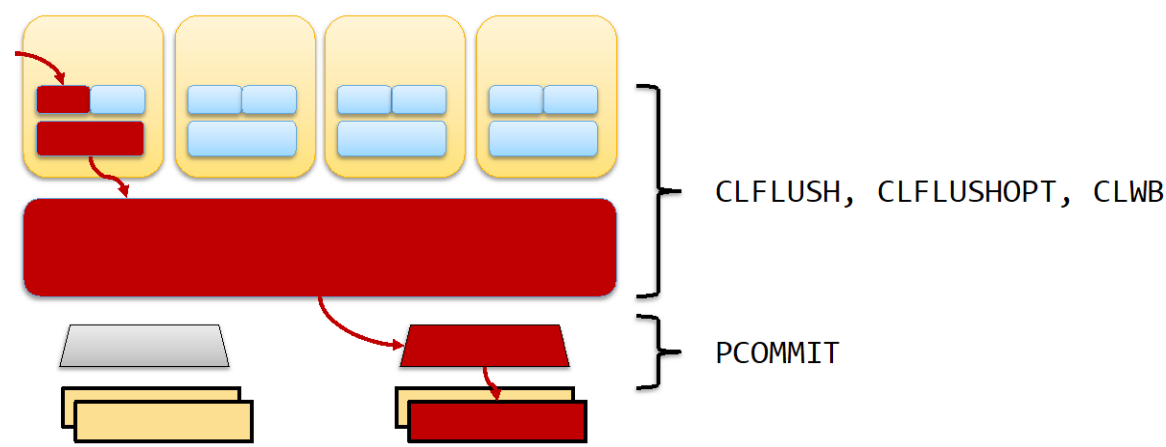
- NVM Files
 - PM Capable
 - Native file APIs and management
- Operations
 - Native open/close read/write
 - NVM.PM.FILE.MAP
 - ...



NVM Summit 2015

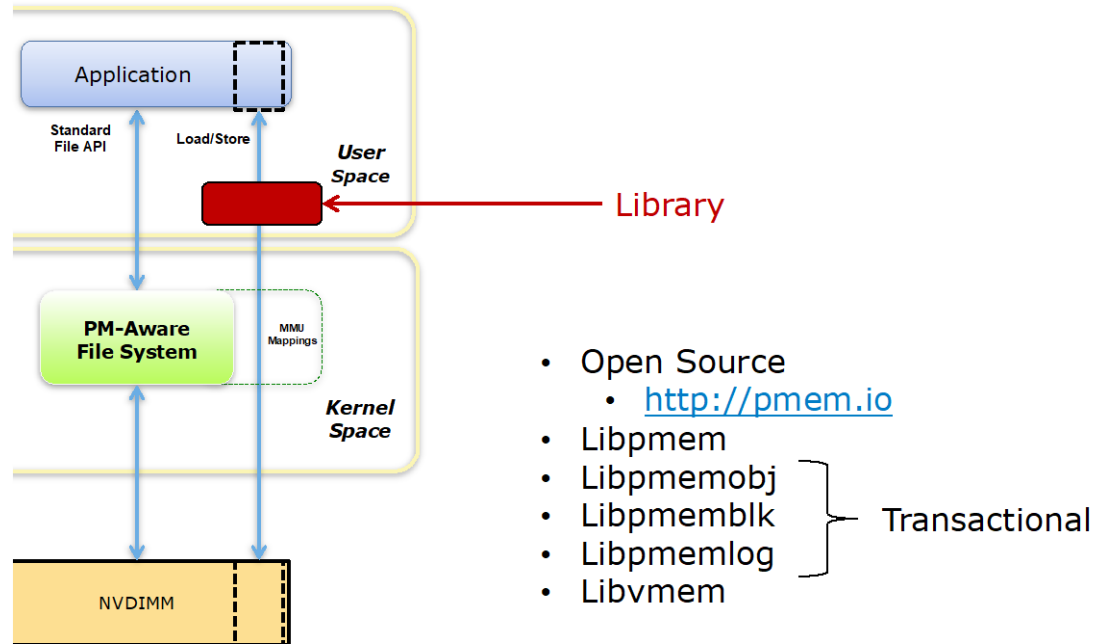


Two Levels of Flushing Writes



NVM Summit 2015

NVM Library: pmem.io 64-bit Linux Initially



12

NVM Summit



8

NVM Summit



3

22

NVM Summit 2016

Application

Standard File API

Load/Store

MMU Mappings

PM-Aware File System

NVDIMM


User Space

Kernel Space

STORAGE INDUSTRY
SUMMIT

Convergence of
Storage and Memory
Developing the Needed
Ecosystem

JANUARY 20, 2016, SAN JOSE, CA



Neal Christiansen
Microsoft
Principal Development Lead
Storage Class Memory in Windows


SNIA
SSSI

SOLID STATE
STORAGE

STORAGE INDUSTRY
SUMMIT

Convergence of
Storage and Memory
Developing the Needed
Ecosystem

JANUARY 20, 2016, SAN JOSE, CA



Jeff Moyer
Red Hat
Principal Software Engineer
Persistent Memory in Linux


SNIA
SSSI

SOLID STATE
STORAGE

STORAGE INDUSTRY
SUMMIT

Convergence of
Storage and Memory
Developing the Needed
Ecosystem

JANUARY 20, 2016, SAN JOSE, CA




Tom Talpey
Microsoft
Storage Architect

Going Remote at Low Latency: a Future
Networked NVM Ecosystem

SNIA
SSSI

SOLID STATE
STORAGE



3

8

12

NVM Summit

22

NVM Summit 2017

Application

Standard File API

Load/Store

User Space

PM-Aware File System

MMU Mappings

Kernel Space

NVDIMM

12

8


3

NVM Summit

STORAGE INDUSTRY
SUMMIT

Convergence of
Storage and Memory
Developing the Needed
Ecosystem

JANUARY 20, 2016, SAN JOSE, CA



Neal Christiansen
Microsoft
Principal Development Lead
Storage Class Memory in Windows

SNIA
SSSI

STORAGE

STORAGE INDUSTRY
SUMMIT

Convergence of
Storage and Memory
Developing the Needed
Ecosystem

JANUARY 18, 2017 | SAN JOSE, CA

SNIA
SSSI

STORAGE


Storage at Memory Speed and
Amazing Future of Virtual Non-Volatile Memory

Jef
Rajesh Venkatasubramanian, Principal Engineer, VMware

STORAGE INDUSTRY
SUMMIT

Convergence of
Storage and Memory
Developing the Needed
Ecosystem

JANUARY 20, 2016, SAN JOSE, CA



Tom Talpey
Microsoft
Storage Architect

SNIA
SSSI

STORAGE

Going Remote at Low Latency: a Future
Networked NVM Ecosystem

intel

© 2017 Persistent Memory Summit. All Rights Reserved.

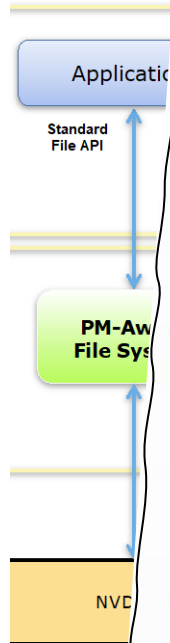
14

PM Summit 2017

NV

Active Work

- More details on flushing to persistence
 - ◆ Includes flushing to remote persistence
- Continue to refine the error model
- Transactions
- APIs?



© 2017 SNIA Persistent Memory Summit. All Rights Reserved.

12

SUMMIT Ecosystem

JANUARY 20, 2016, SAN JOSE, CA



Tom Talpey
Microsoft
Storage Architect

Going Remote at Low Latency: a Future
Networked NVM Ecosystem

SNIA² SOLID STATE
SSSI | STORAGE

PM Summit 2017

Application

Standard File API

PM-Aw File Sys

NVD

12

8

3

NVM Summit

22

Types of Store Barriers


Barrier Type	Current Status
Standard API	Fully specified Fully supported: <ul style="list-style-type: none">Linux (ext4, XFS)Windows (NTFS)
Optimized Flush	Specified, but evolving (ask when safe) <ul style="list-style-type: none">Linux: unsafe except Device DAX<ul style="list-style-type: none">(and new file systems)Windows: safe
Remote Flush	Proposals under discussion (works today with extra round trip)
Deep Flush	Upcoming Specification
Transactions	Built on above via libraries and languages Much more language support to do

© 2017 SNIA Persistent Memory Summit

SUMMIT

Developing Ecosystem

JANUARY 20, 2016, SAN JOSE, CA



Tom Talpey
Microsoft
Storage Architect

Going Remote at Low Latency: a Future Networked NVM Ecosystem

SNIA | SOLID STATE
SSSI | STORAGE

PM Summit 2018

Diagram: PM-Aw File System Architecture

The diagram shows a stack of layers: Application (blue box) at the top, connected to Standard File API (yellow box), which connects to PM-Aw File Sys (green box), which connects to NVD (orange box) at the bottom. Red arrows indicate data flow from the Application layer down to the NVD layer.

TWG Ongoing Work

- **Security**
 - PM Hardware Security Threat Model (balloting)
- **Remote persistent memory (via RDMA)**
 - Ongoing – optimizations for RDMA worked in multiple forums
 - Remote asynchronous flush (under discussion)
- **Higher-level Semantics**
 - As we learn more..

© 2018 SNIA Persistent Memory Summit. All Rights Reserved.

SUMMIT Ecosystem
JANUARY 20, 2016, SAN JOSE, CA

Tom Talpey
Microsoft
Storage Architect

Going Remote at Low Latency: a Future Networked NVM Ecosystem

SNIA | SOLID STATE
SSSI | STORAGE

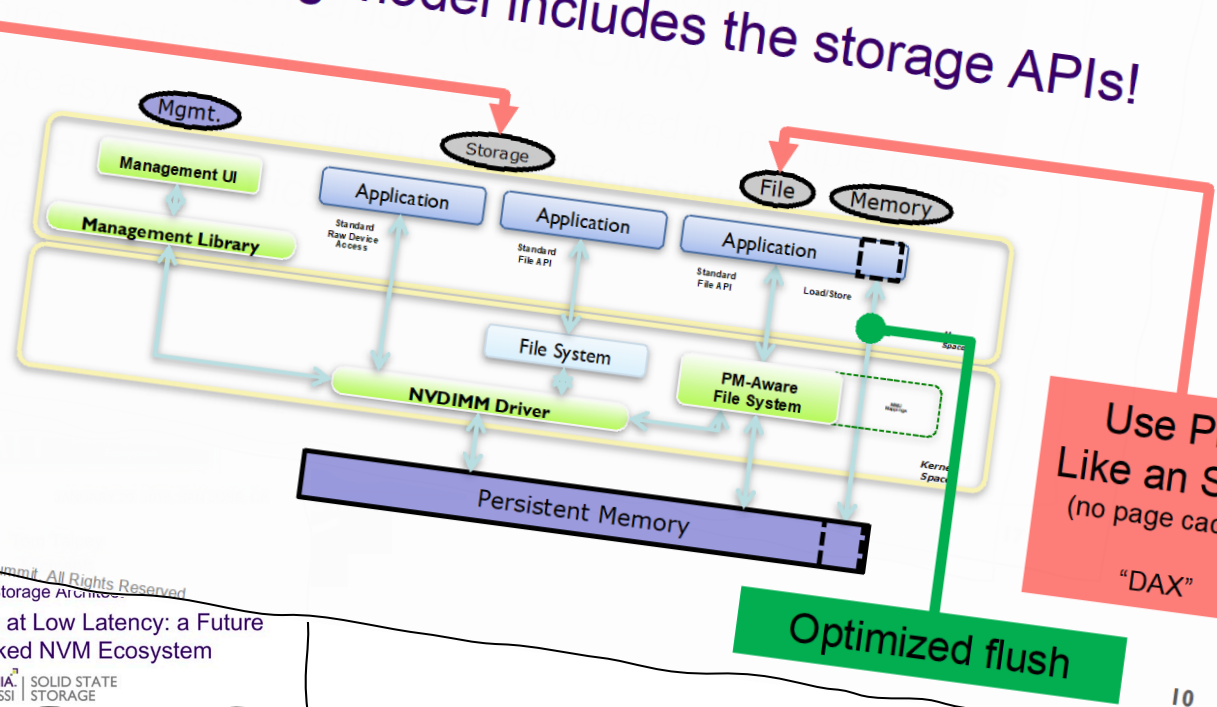
PM Summit 2019

TWG Ongoing Work

Optimized Flush: Flushing from Userspace

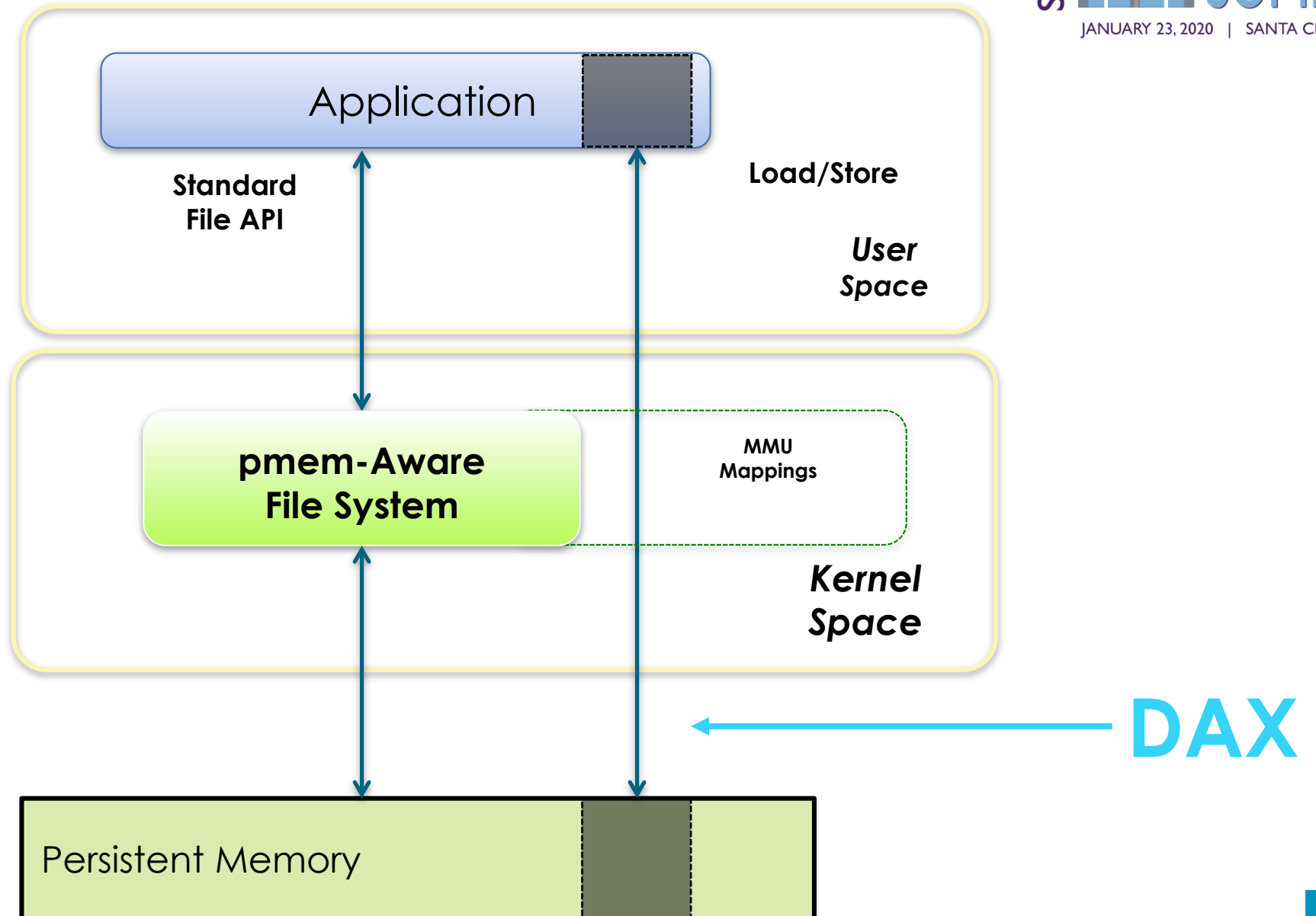
➤ The programming model includes the storage APIs!

Use PM Like an SSD



Going Remote at Low Latency: a Future Networked NVM Ecosystem

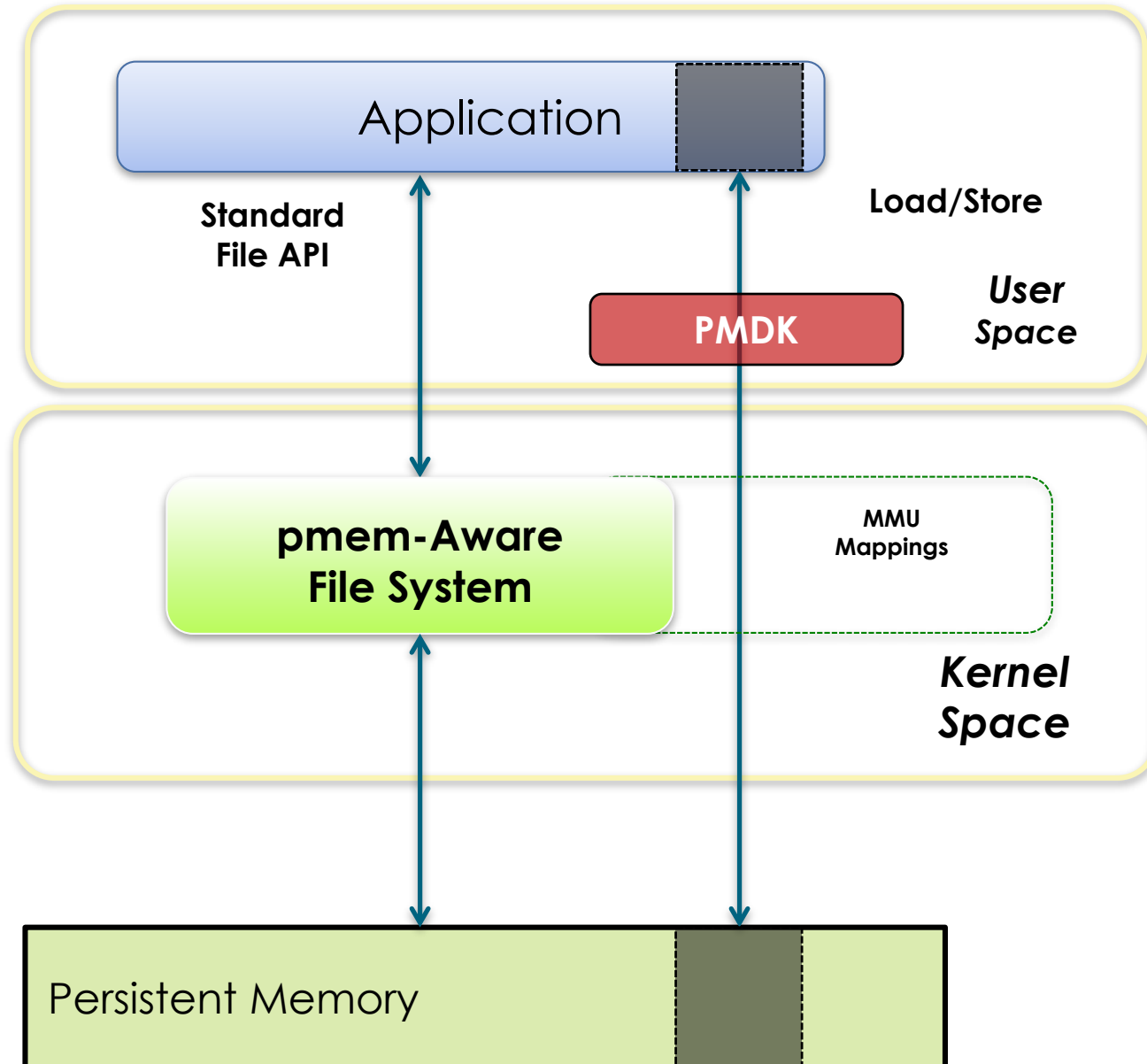
Focus of Many Talks Today

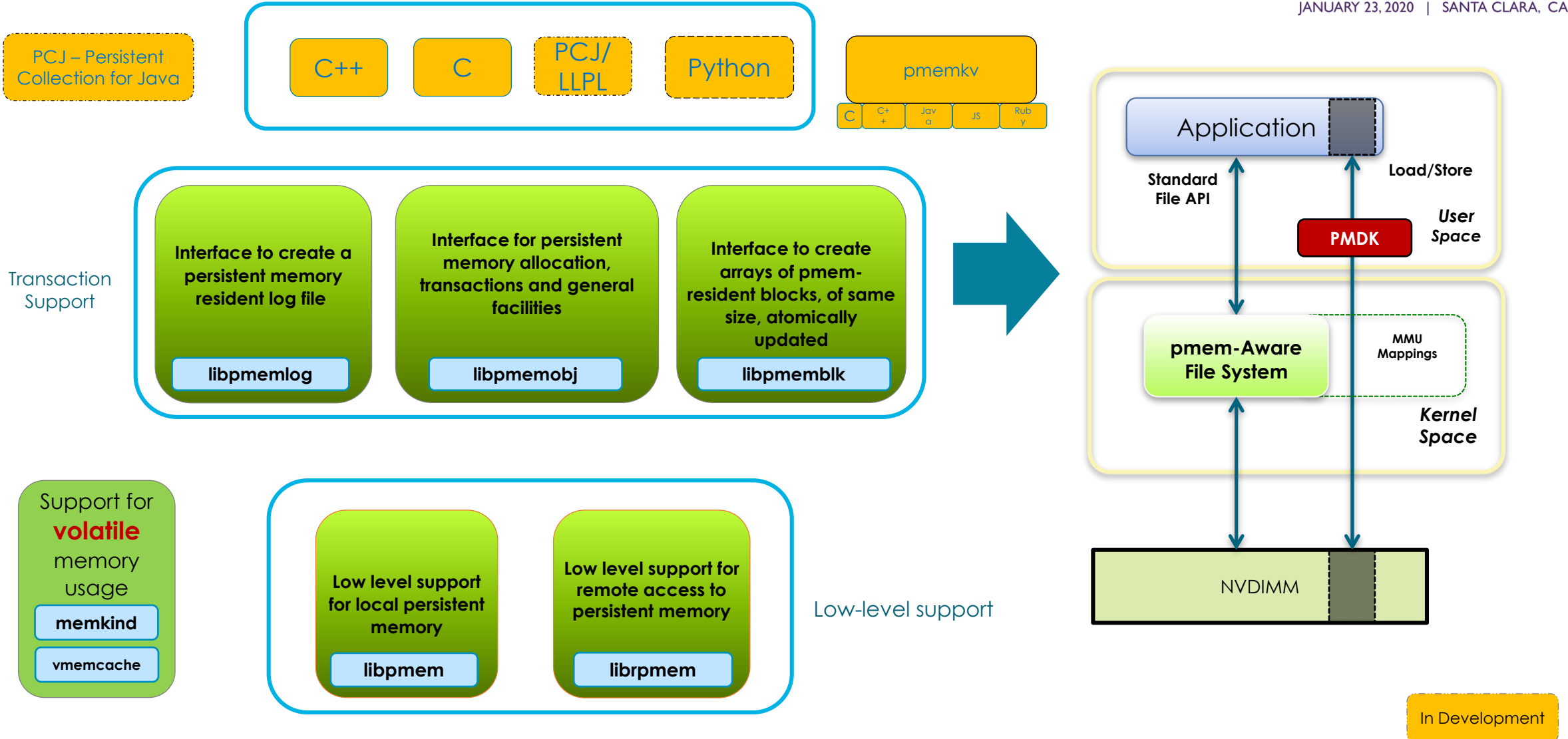


Where the Model Worked

Hard Earned Successes

Building on the Model





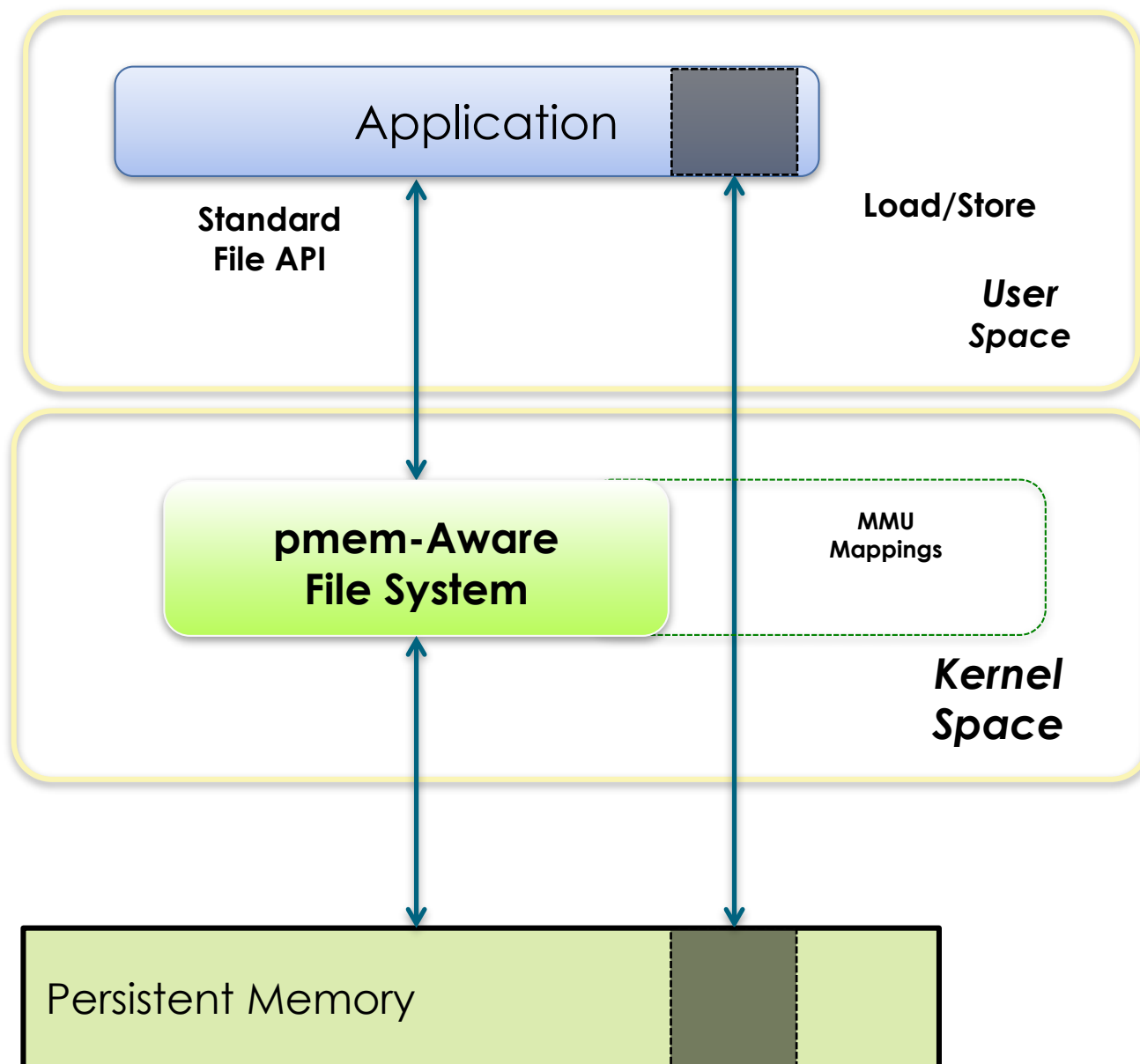
2019 – A Very Exciting Year!

- Intel launches Optane DC Persistent Memory
 - Joining multiple NVDIMM-N products in the ecosystem
- Multiple major products move beyond *demo* to *shipping*
 - More on this later today
- Number of use cases and SW enablement efforts explodes
 - More on this later today
 - Even includes some start-ups
- Academic activity continues to grow at a steady pace
 - Just two papers of note when our timeline started
 - Now I've lost count on how many pmem-related papers are out there

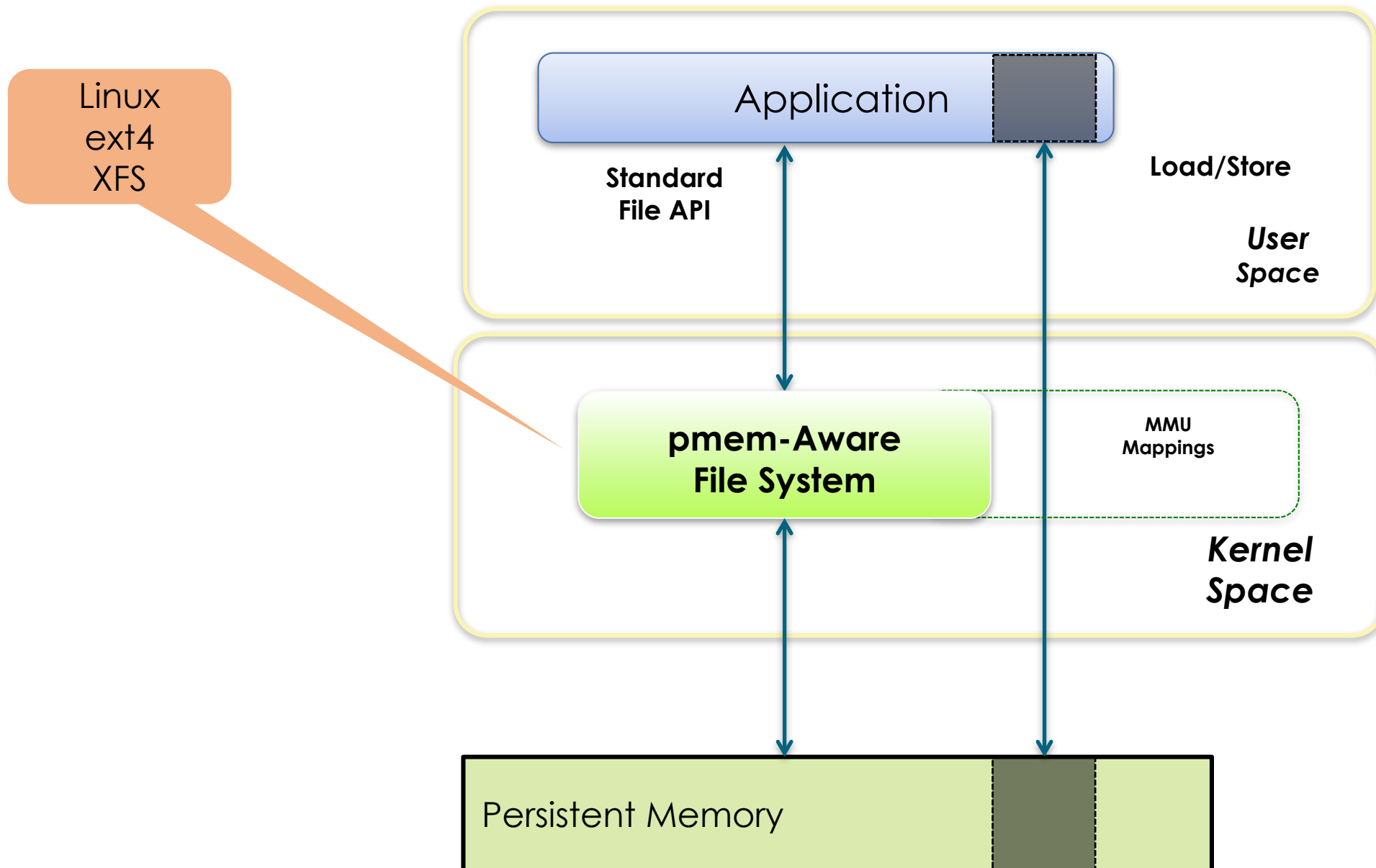
Where the Model Fell Short

Learning from the Ecosystem

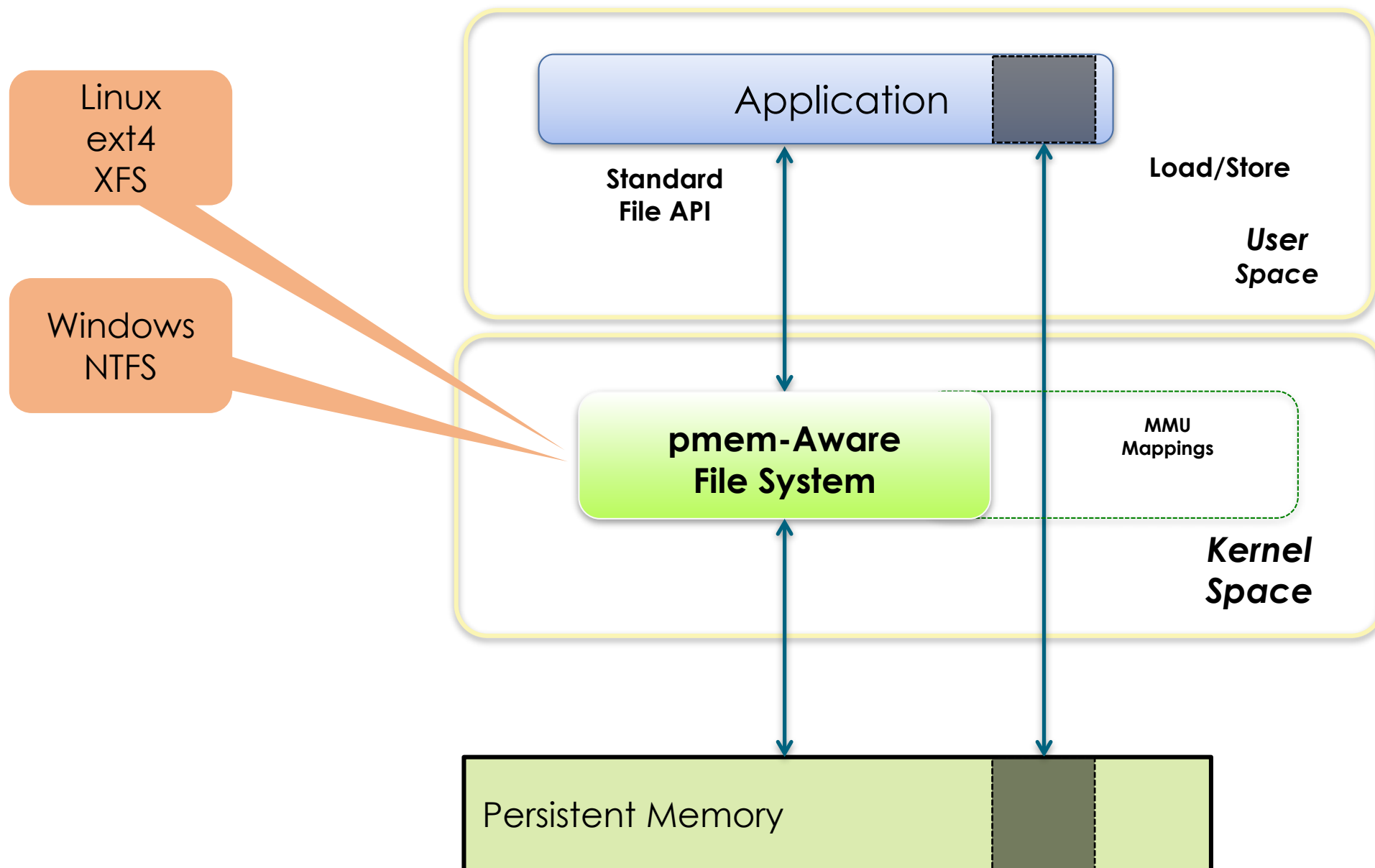
The File System



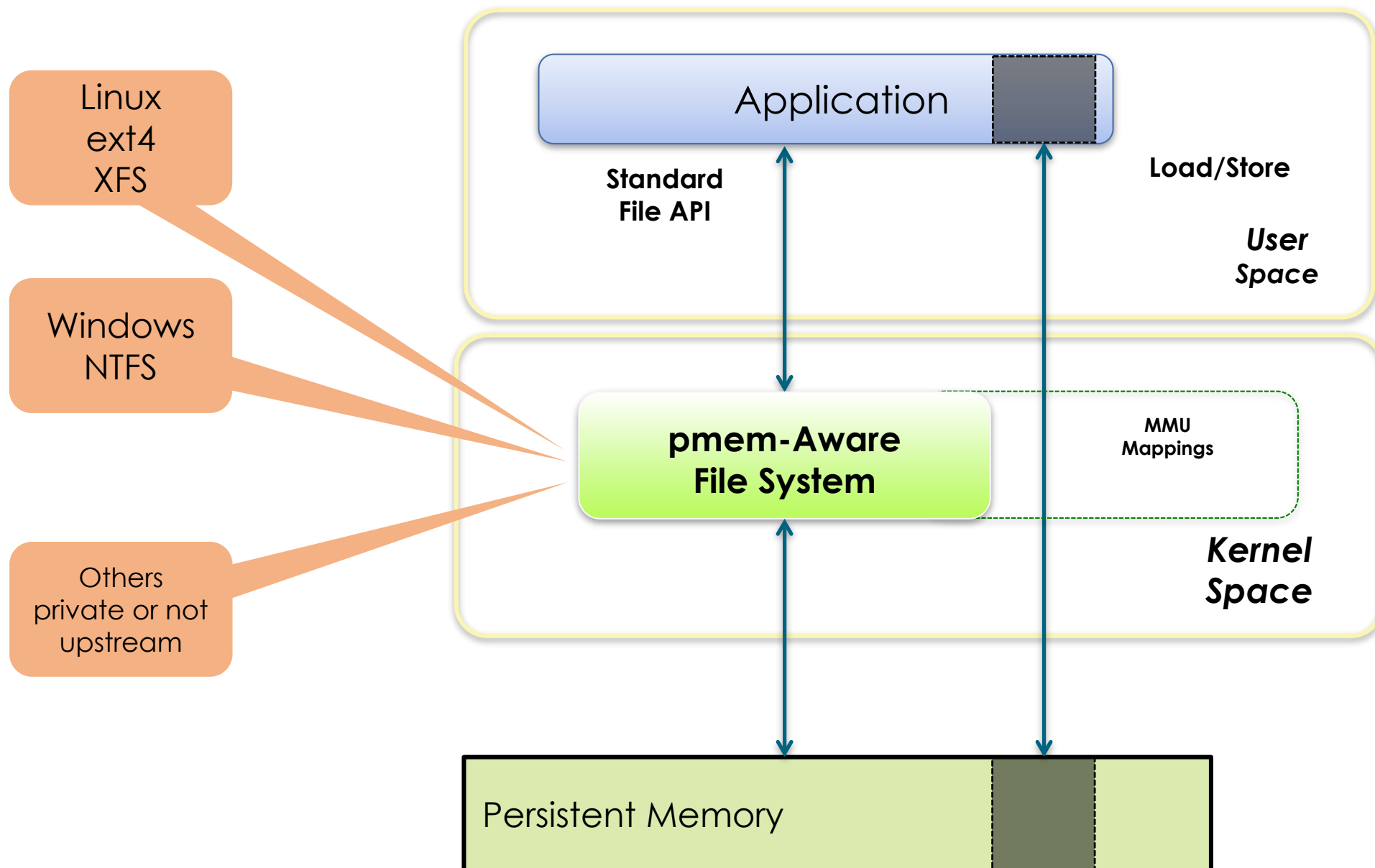
The File System



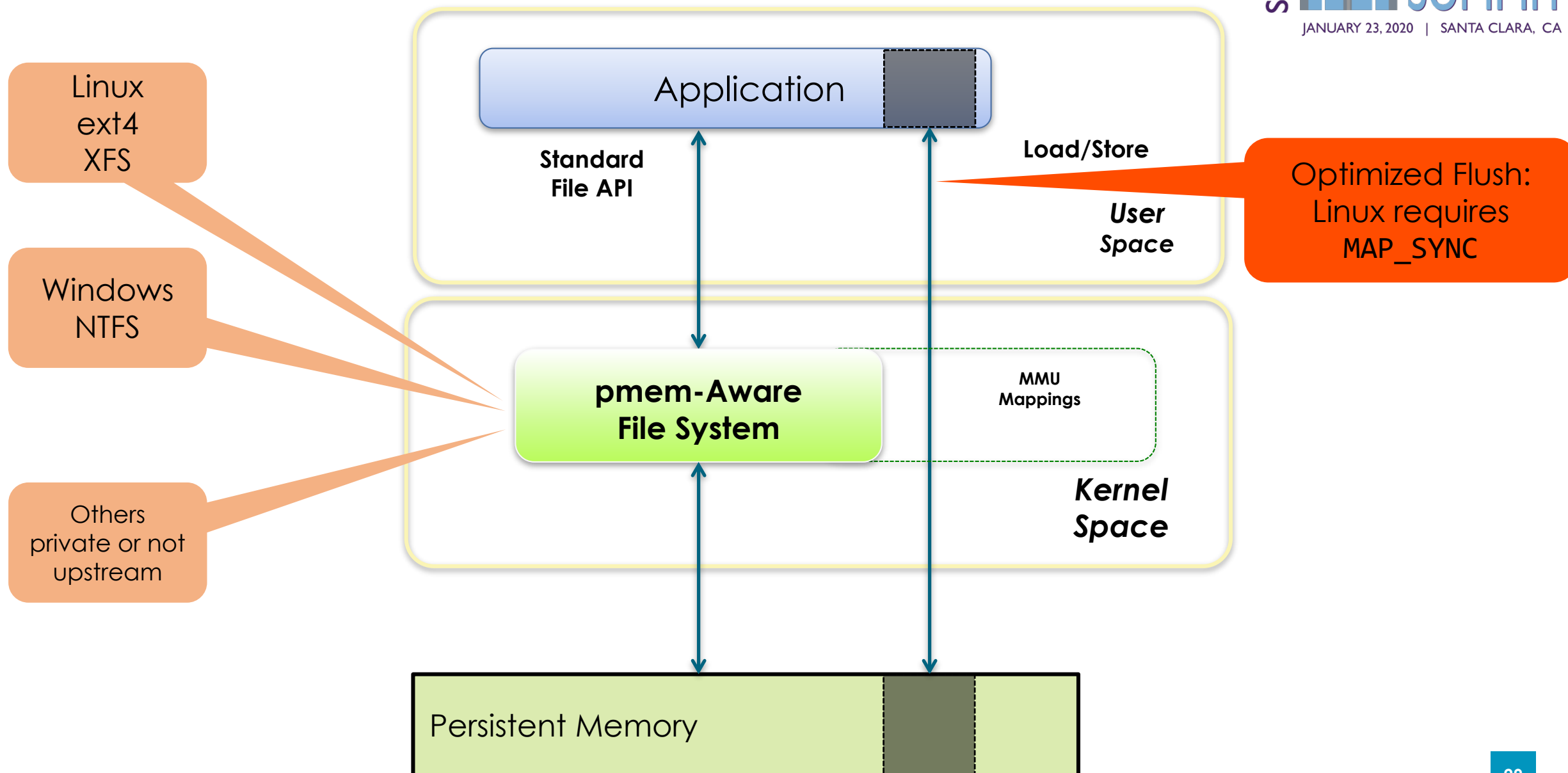
The File System



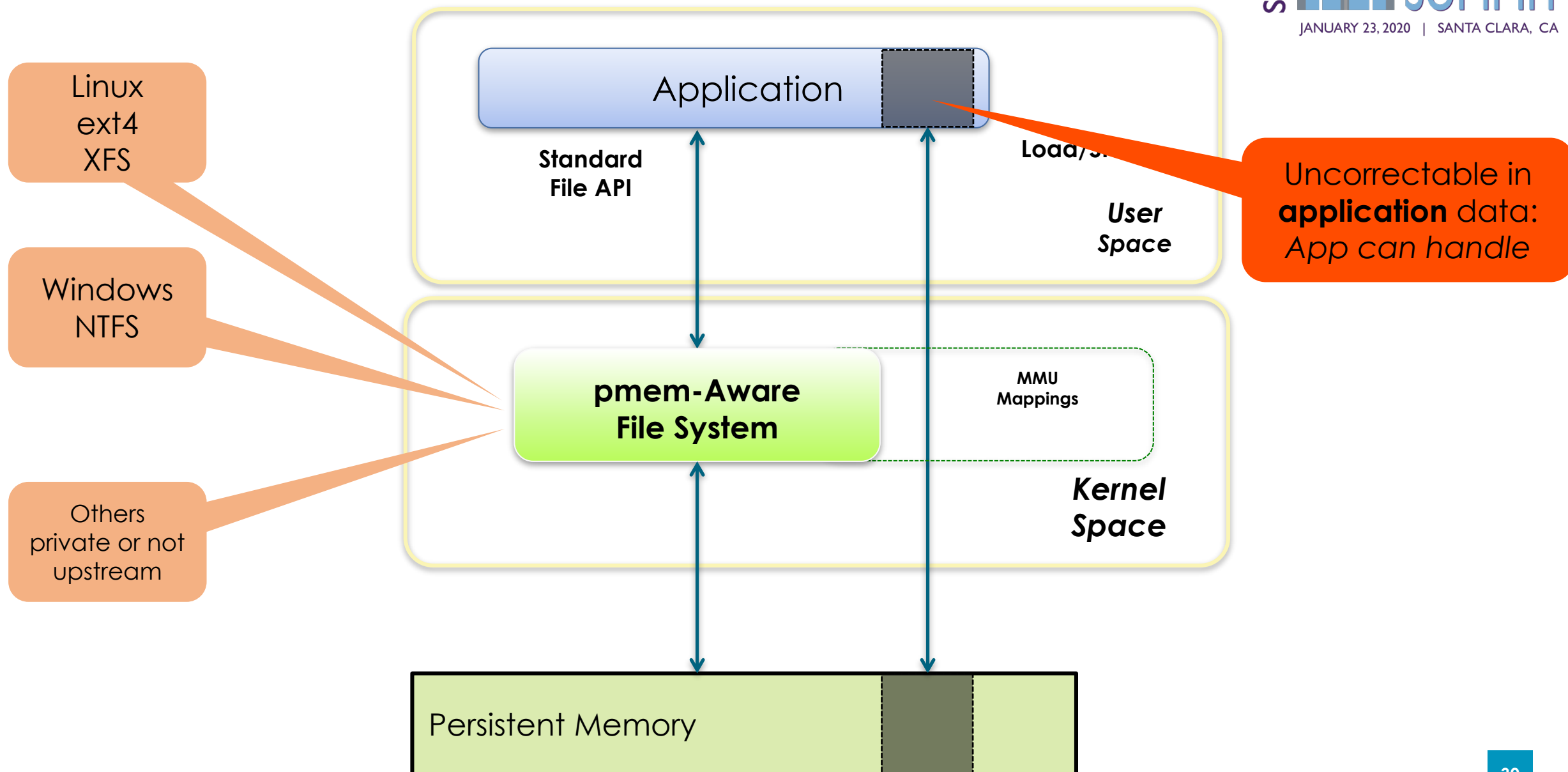
The File System



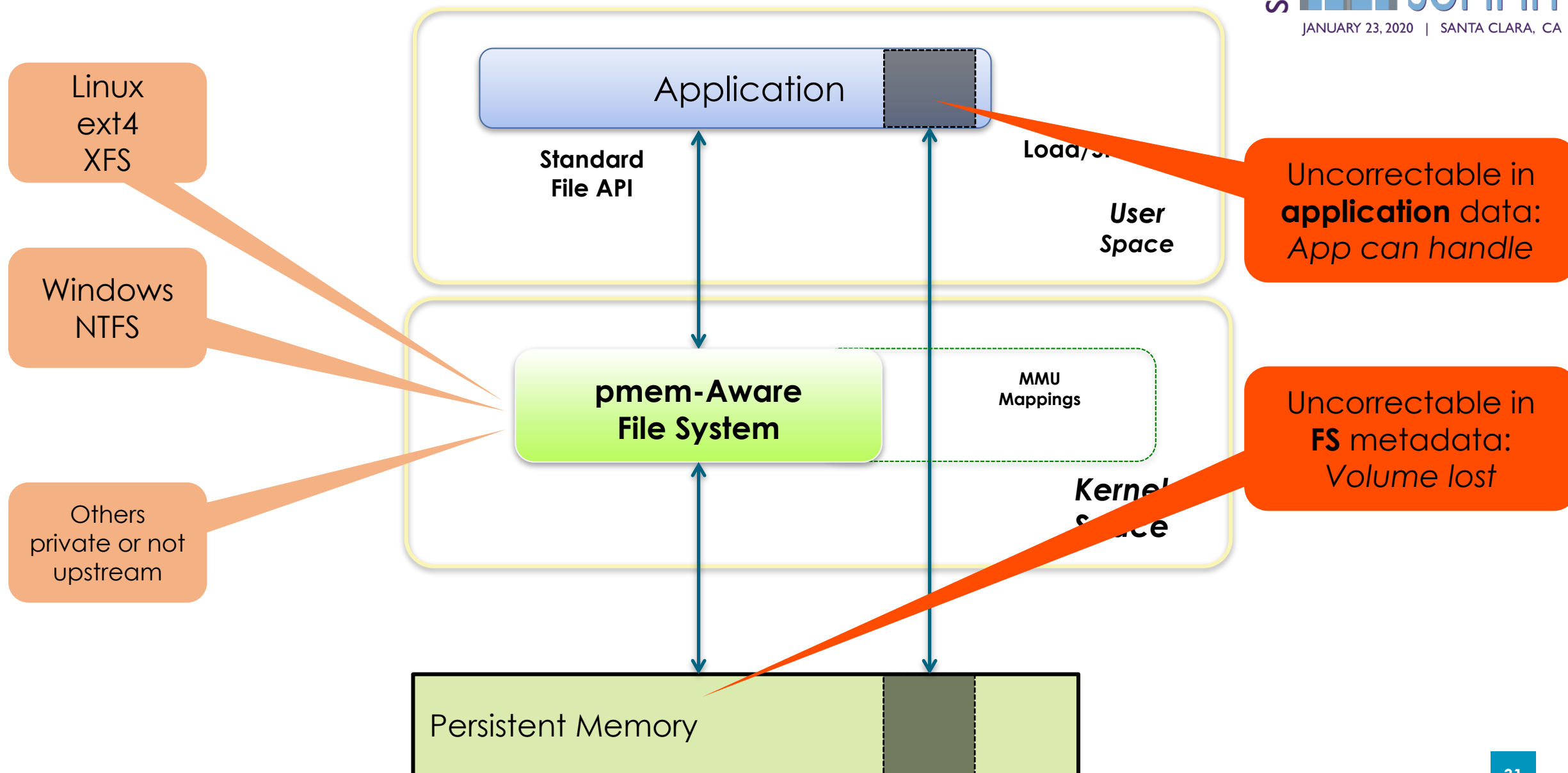
The File System



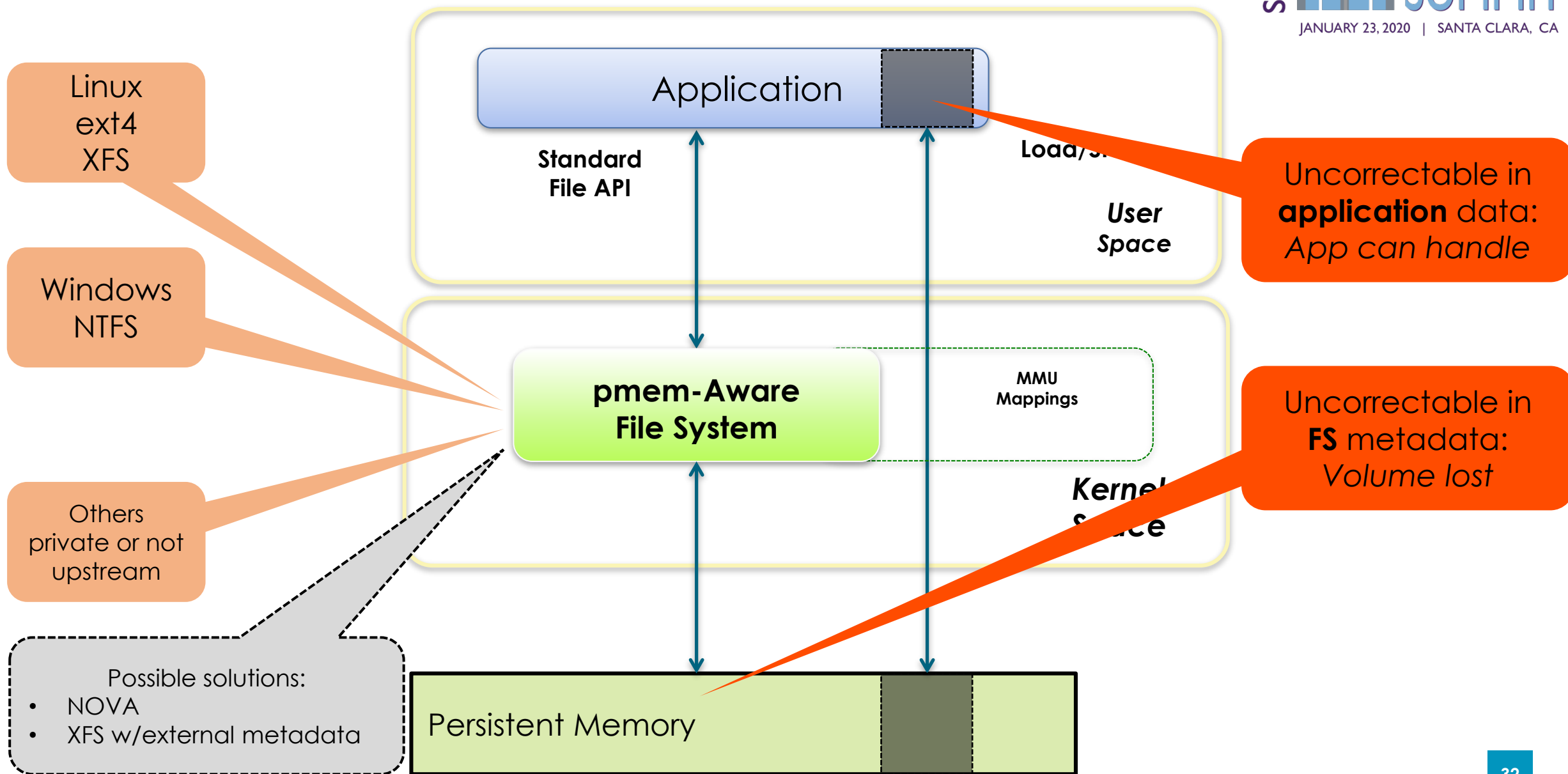
The File System



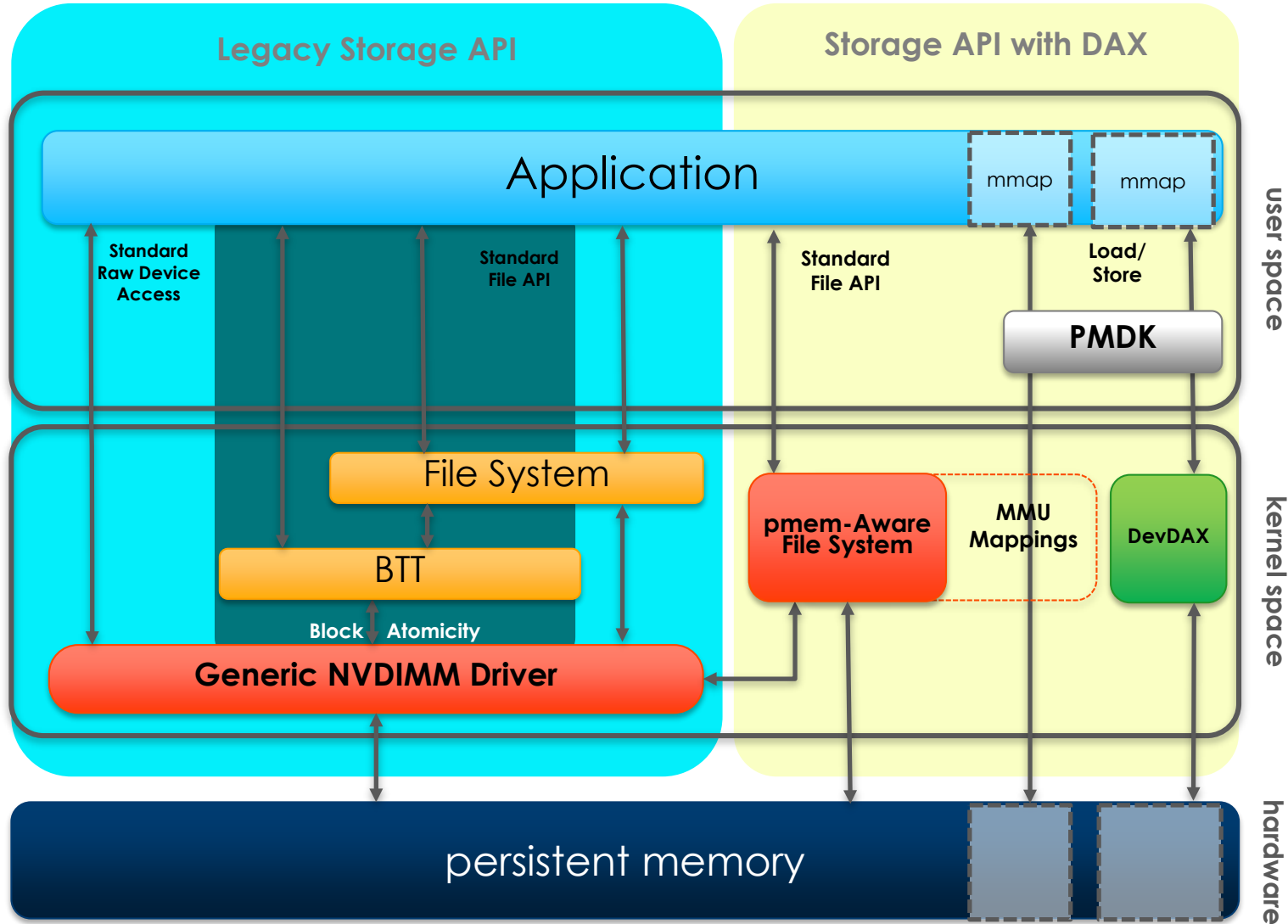
The File System



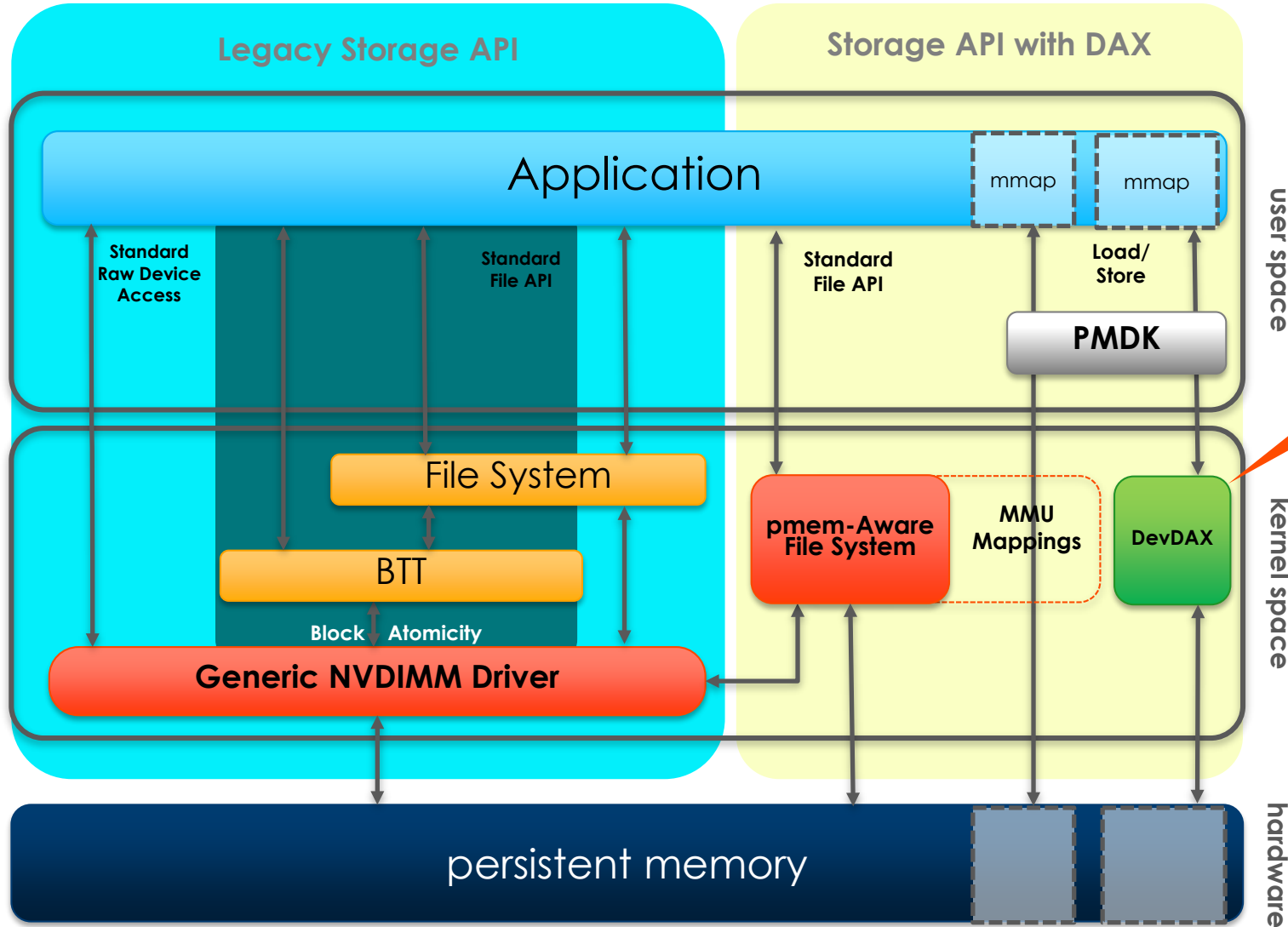
The File System



Device DAX on Linux

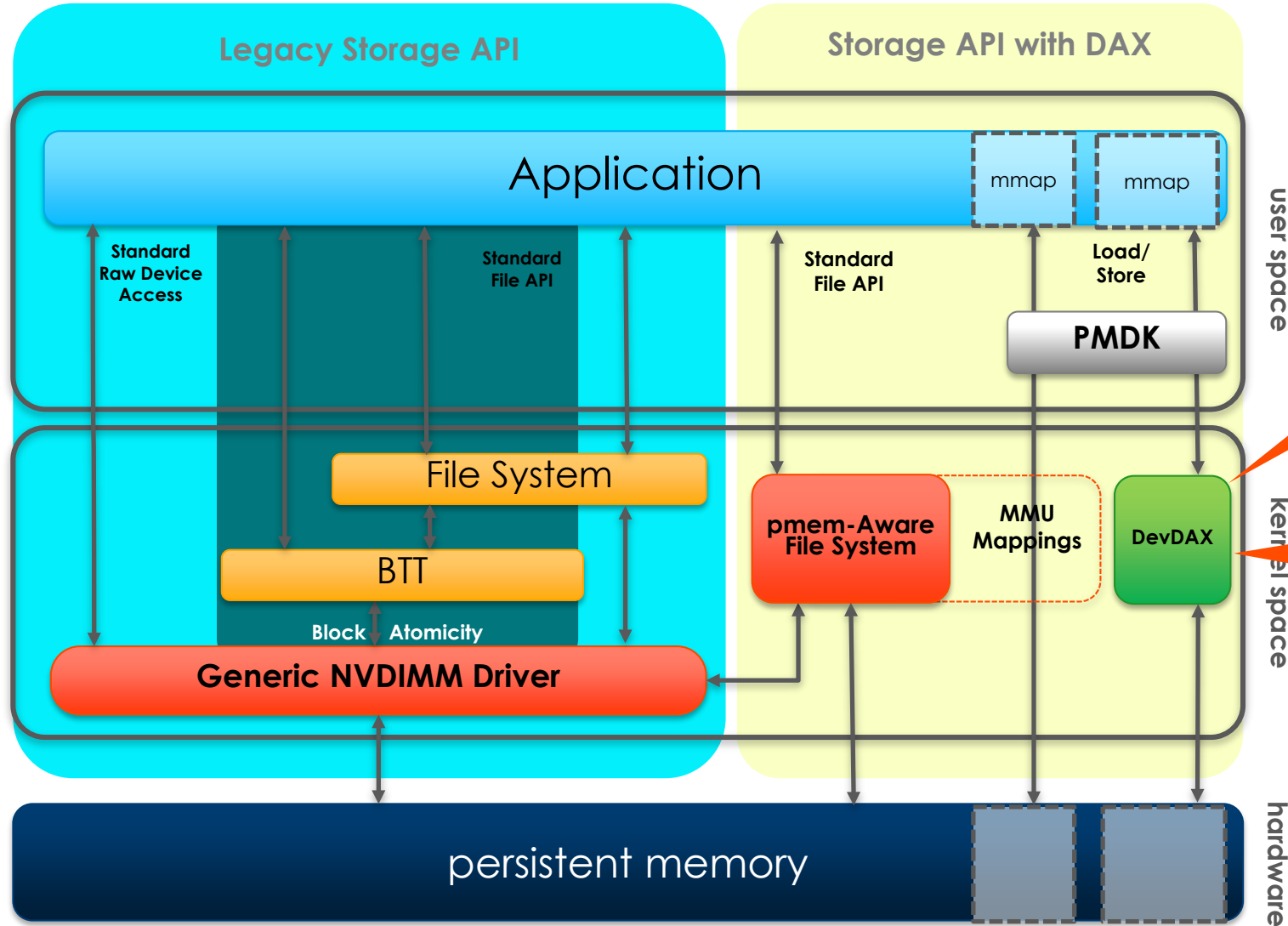


Device DAX on Linux



Raw, but powerful
(RAS, RDMA, other)

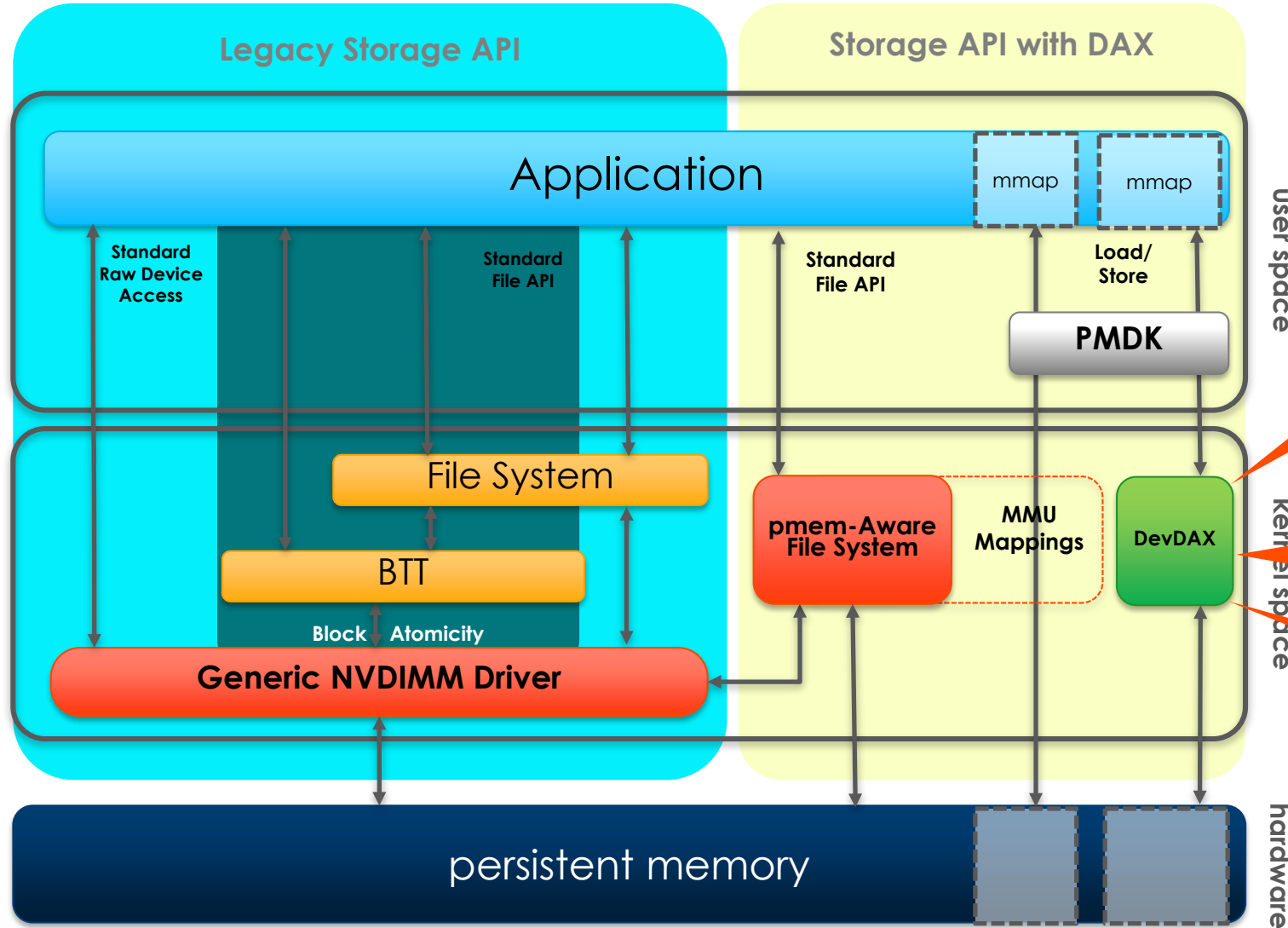
Device DAX on Linux



Raw, but powerful
(RAS, RDMA, other)

Doesn't follow
programming
model (POSIX
doesn't work)

Device DAX on Linux



Raw, but powerful
(RAS, RDMA, other)

Doesn't follow
programming
model (POSIX
doesn't work)

The fact Linux
invented it shows
the model fell short

Ongoing Work

2020 and beyond

The NVM Programming TWG...

- Should probably be called the *PM Programming TWG* now
- Continues to meet, although lower frequency
- Still heavy interest in remote persistent memory (rpmem)
- Still interest in resyncing with ecosystem
 - Publishing a 2.0 PM Programming Specification

Thank you

Please visit www.snia.org/pmsummit for presentations

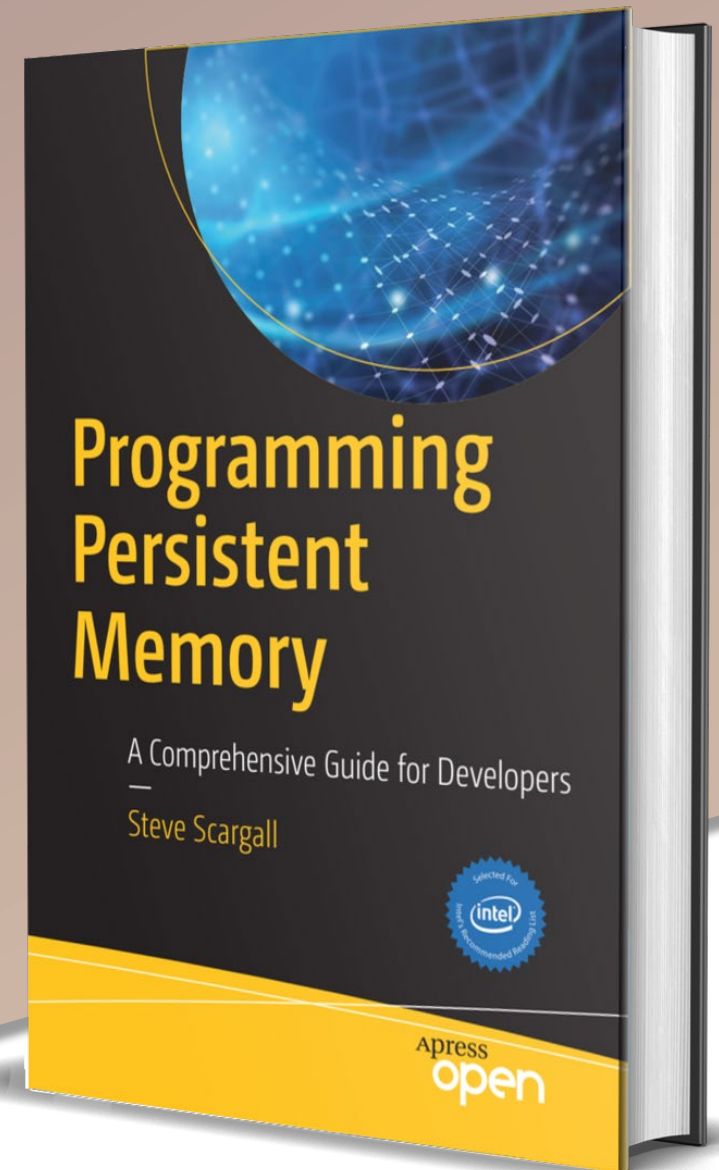
```
S1 int  
S2 main(int argc, char *argv[])  
S3 {  
S4     if (argc != 2) {  
S5         printf(stderr, "Usage: %s file-name\n", argv[0]);  
S6         return 1;  
S7     }  
S8     printf(stdout, "File: %s\n", argv[1]);  
S9     return 0;  
S10 }
```

Master Persistent Memory Programming

Are you ready to begin?



Start Reading



<https://www.apress.com/us/book/9781484249314>