NVM Express - Delivering Breakthrough PCIe SSD Performance and Scalability

Paul Luse
Intel Corporation

- **NVM Express (NVMe) Overview**
- New NVMe Features in Enterprise & Client
- Driver Ecosystem for NVMe
- Windows NVMe OFA Driver Architecture and Implementation Details

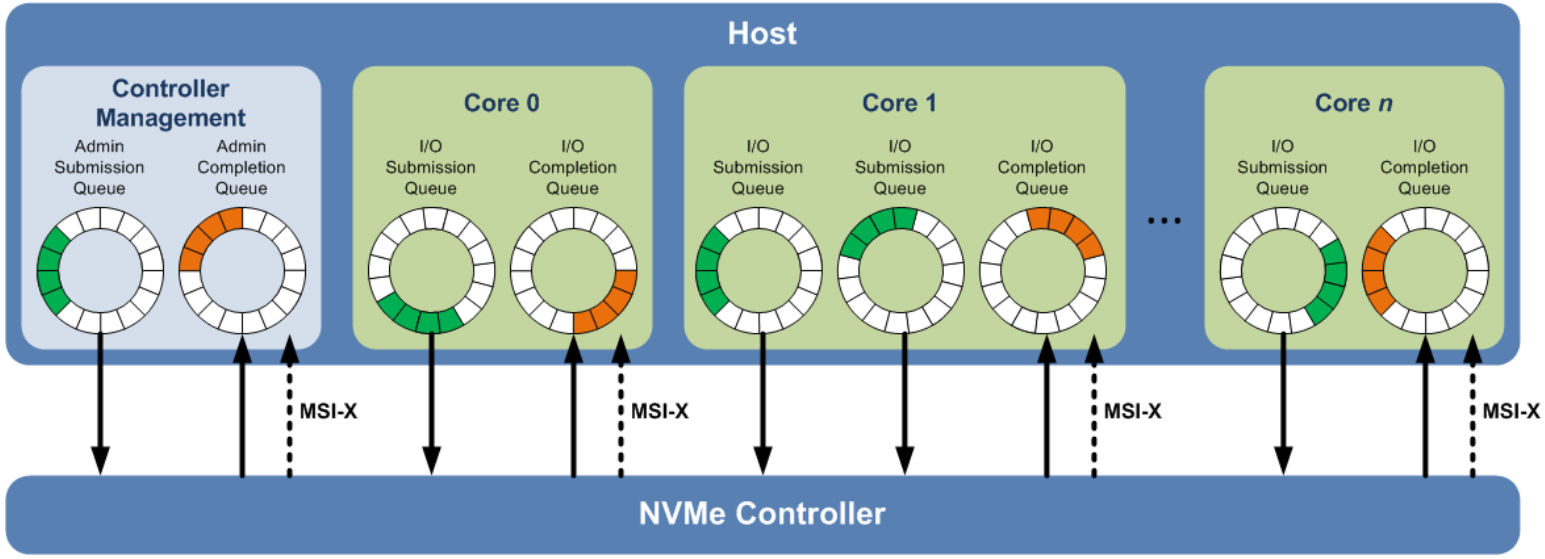
NVM Express (NVMe) Overview

- NVM Express is a scalable host controller interface designed for Enterprise and client systems that use PCI Express* SSDs
- NVMe was developed by industry consortium of 80+ members and is directed by a 13 company Promoter Group

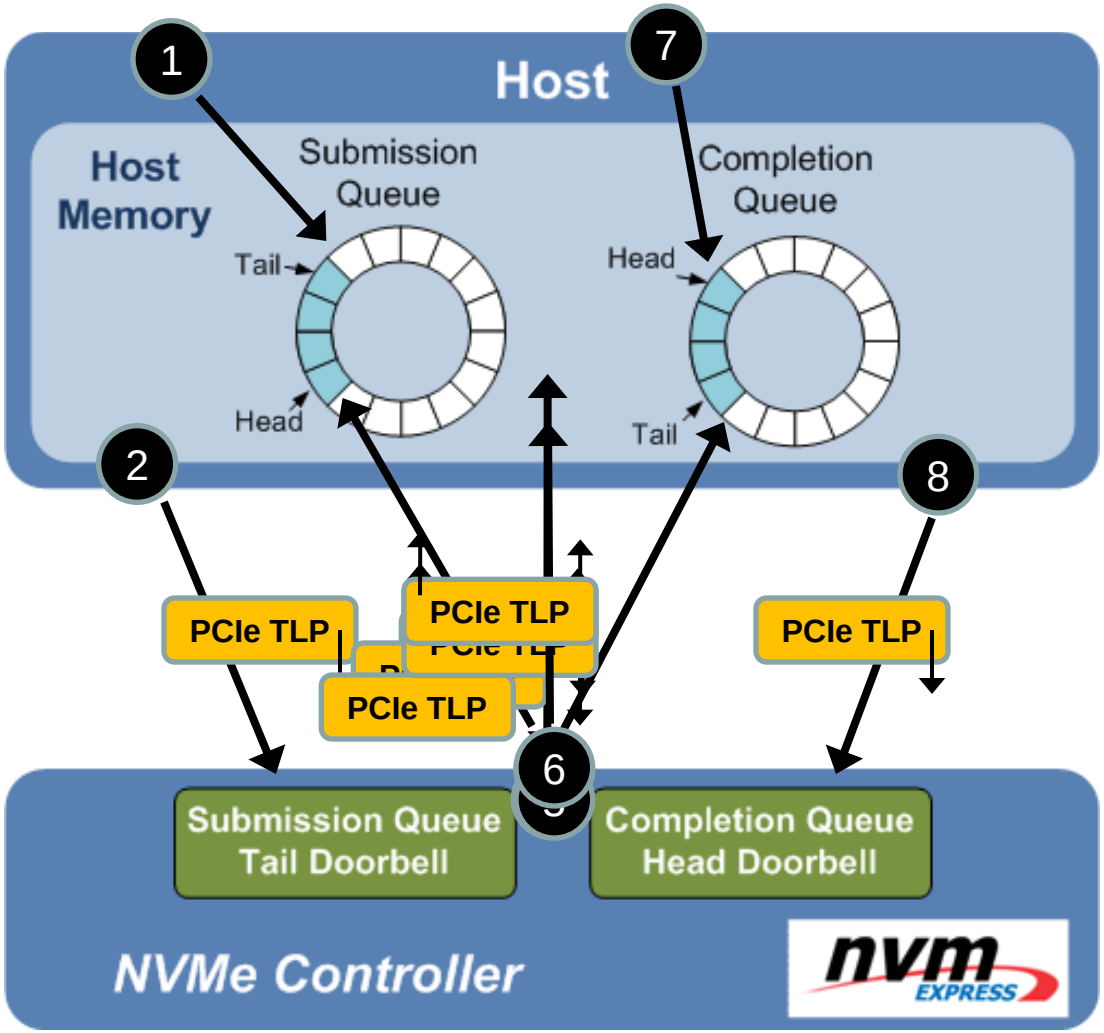
- NVMe 1.0 was published March 1, 2011
- Product introductions later this year, first in Enterprise

Technical Basics

- The focus of the effort is efficiency, scalability, and performance
  - All parameters for 4KB command in single 64B DMA fetch
  - Supports deep queues (64K commands per Q, up to 64K queues)
  - Supports MSI-X and interrupt steering
  - Streamlined command set optimized for NVM (6 I/O commands)
  - Enterprise: Support for end-to-end data protection (i.e., DIF/DIX)
  - NVM technology agnostic



NVMe Command Execution



1) Queue Command(s)
2) Ring Doorbell *(New Tail)*
3) Fetch Command(s)
4) Process Command
5) Queue Completion(s)
6) Generate Interrupt
7) Process Completion
8) Ring Doorbell *(New Head)*

Command Set Overview

Management Commands for Queues & Transport

Admin Command	Description
Create I/O Submission Queue	Queue Management
Create I/O Completion Queue	
Delete I/O Submission Queue	
Delete I/O Completion Queue	
Abort	
Asynchronous Event Request	Status & Event Reporting
Get Log Page	
Identify	Configuration
Set Features	
Get Features	
Firmware Activate (Optional)	Firmware Management
Firmware Image Download (Optional)	
Security Send (Optional)	Security
Security Receive (Optional)	
Format NVM (Optional)	Namespace Management

I/O Commands for SSD Functionality

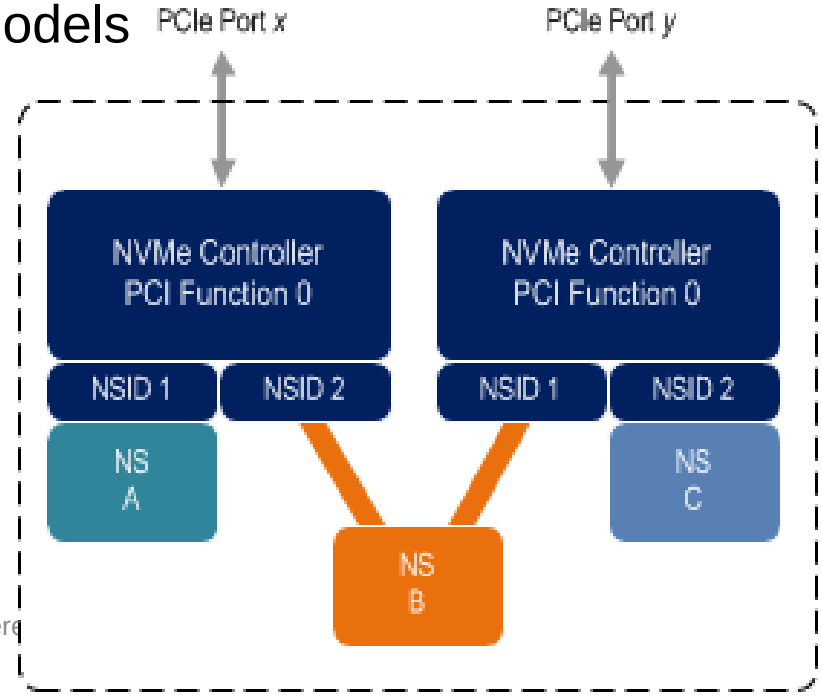
NVM Command	Description
Read	Data Transfer, Including end-to-end data protection & security
Write	
Write Uncorrectable (Optional)	
Compare (Optional)	
Dataset Management (Optional)	Data Usage Hints
Flush	Data Ordering

13 Required Commands Total (10 Admin, 3 I/O)

- NVM Express (NVMe) Overview
- **New NVMe Features in Enterprise & Client**
- Driver Ecosystem for NVMe
- Windows NVMe OFA Driver Architecture and Implementation Details

Multi-Path I/O and Namespace Sharing

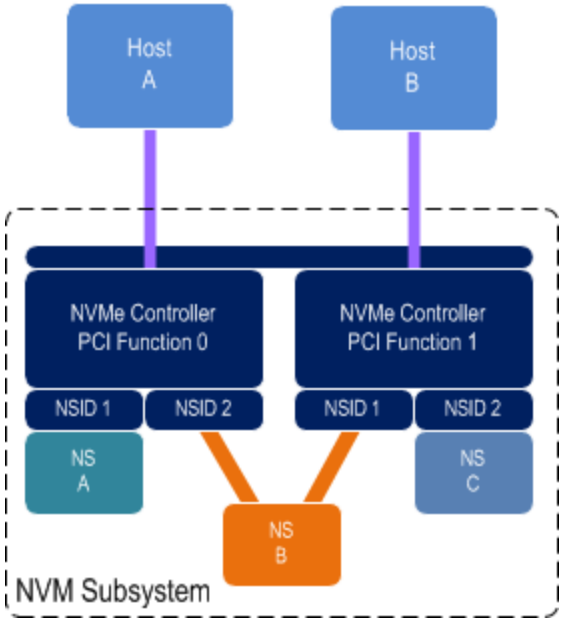
- An NVMe namespace may be accessed via multiple “paths”
  - SSD with multiple PCI Express* ports
  - SSD behind a PCIe switch to many hosts
- Two hosts accessing the same namespace must coordinate
  - A namespace is an NVMe defined range of LBAs
- NVMe is adding capabilities in NVMe 1.1 to enable Enterprise multi-host usage models



NVMe 1.1

Uniquely Identifying a Namespace

- How do Host A and Host B know that NS B is the same namespace?
- NVMe 1.1 adds unique identifiers for:
  - The NVM Subsystem
  - Each Namespace within an NVM Subsystem
- These identifiers are guaranteed to be globally unique



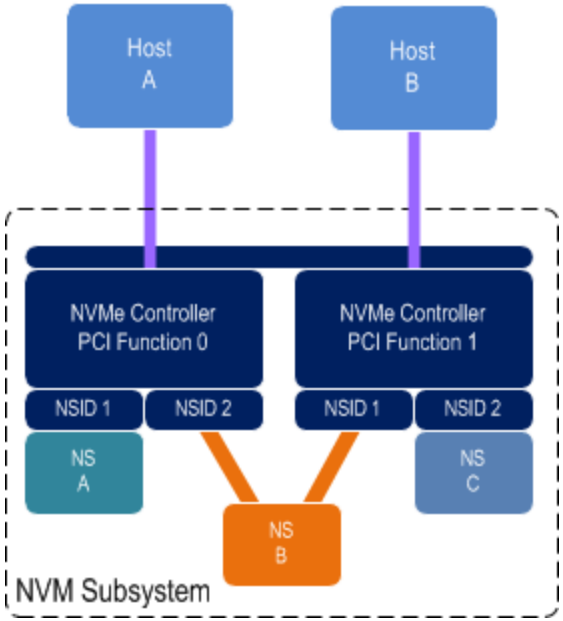
Unique NVM Subsystem Identifier =
2B PCI Vendor ID + 20B Serial Number + 40B Model Number + 2B Controller ID

Unique Namespace Identifier =
64B Unique NVM Subsystem Identifier + 8B IEEE Extended Unique Identifier

NVMe 1.1

NVM Subsystem Reset

- Resets in NVMe 1.0 are controller based
- NVMe 1.1 adds a capability to reset the entire NVM Subsystem
  - E.g., new firmware needs to be applied to both controllers
- To perform an NVM Subsystem Reset, write the value “NVMe” to the register



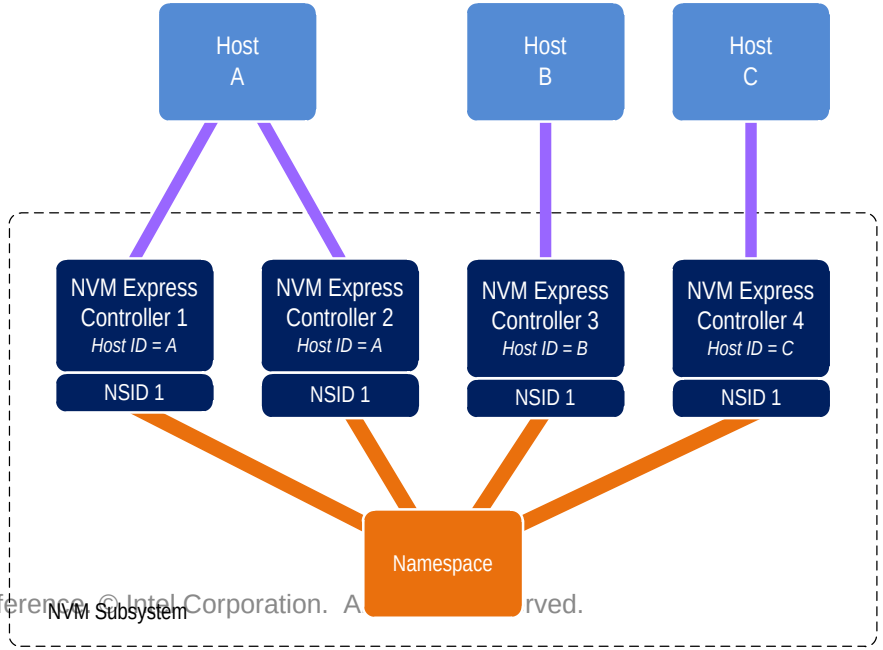
NVM Subsystem Register

Bit	Type	Reset	Description
31:00	RW	0h	**NVM Subsystem Reset Control (NSSRC):** A write of the value 4E564D65h ("NVMe") to this field initiates an NVM Subsystem Reset. A write of any other value has no functional effect on the operation of the NVM subsystem. This field shall return the value 0h when read.

NVMe 1.1

Reservations

- In some multi-host environments, like Windows clusters, reservations are used
- NVMe 1.1 includes a simplified reservations mechanism that is compatible with implementations that use SCSI reservations
- What is a reservation? Enables two or more hosts to coordinate access to a shared namespace.
  - A reservation may allow Host A and Host B access, but disallow Host C



NVMe 1.1

- NVM Express (NVMe) Overview
- New NVMe Features in Enterprise & Client
- **Driver Ecosystem for NVMe**
- Windows NVMe OFA Driver Architecture and Implementation Details

Reference Drivers for Key OSes

- Linux
  - Already accepted into the mainline kernel on kernel.org
  - Open source with GPL license
  - Refer to http://git.infradead.org/users/willy/linux-nvme.git
- Windows
  - Baseline developed in collaboration by IDT, Intel, and LSI
  - Open source with BSD license
  - Maintenance is collaboration by NVMe WG and Open Fabrics Alliance
  - Refer to https://www.openfabrics.org/resources/developer-tools/nvme-windows-development.html
- VMware
  - Initial driver developed by Intel
  - Based on VMware advice, “vmk linux” driver based on Linux version
  - NVMe WG will collaborate with VMware on delivery/maintenance

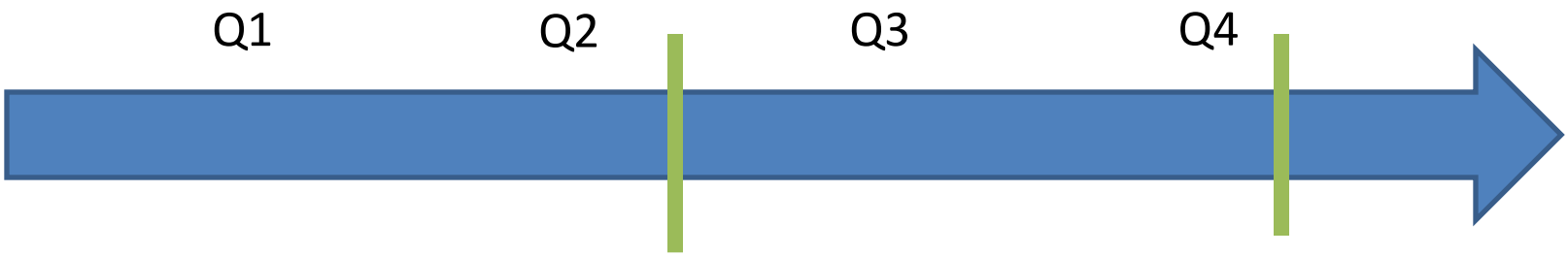
Reference Drivers for Key OSes (cont)

- Solaris
  - There is a working driver prototype
  - Planned features include:
    - Fully implement and conform to 1.0c spec
    - Efficient block interfaces bypassing complex SCSI code path
    - NUMA optimized queue/interrupt allocatoin
    - Reliable with error detect and recovery fitting into Solaris FMA
    - Build ZFS with multiple sector sizes (512B, 1KB, 2KB, 4KB) on namespaces
    - Fit into all Solaris disk utilities and fwflash(1M) for firmware
    - Boot & install on SPARC and X86
    - Surprise removal support
  - Plan to validate against Oracle SSD partners
  - Plan to integration into S12 and a future S11 Update Release

- UEFI
  - The driver is under development
  - Plan to open source the driver in Q1 ‘13, including bug/patch process
  - Beta quality in Q1’13, production quality Q2’13

OFA NVMe Driver Release Plans

- Windows* reference drivers are targeting two releases per year
- Release 1 is available, and 1.1* work underway



Windows Release 1

- Initial release
- NVMe 1.0b support
- Win7,2008R2

Windows Release 1.1*

- NVMe 1.0c support
- Public IOCTLs
- Win8 Storport specific
- Bug Fixes
- Win7,2008R2,Win8

** Not to be confused with NVMe spec 1.1*

Robust Driver Update Criteria

- The NVMe community is committed to robust reference drivers
- For the Windows NVMe driver maintained with the OpenFabrics Alliance, there is a detailed update process:

Review Criteria:

- Patches submitted by anyone, email to distribution list
- Patch submission should include time sensitivity/expectations and justification for patch (what value it will add, any tradeoffs to consider)
- Patch must be reviewed by at least three NVMe company representatives
- Reviews include compliance with coding guidelines as well as logic

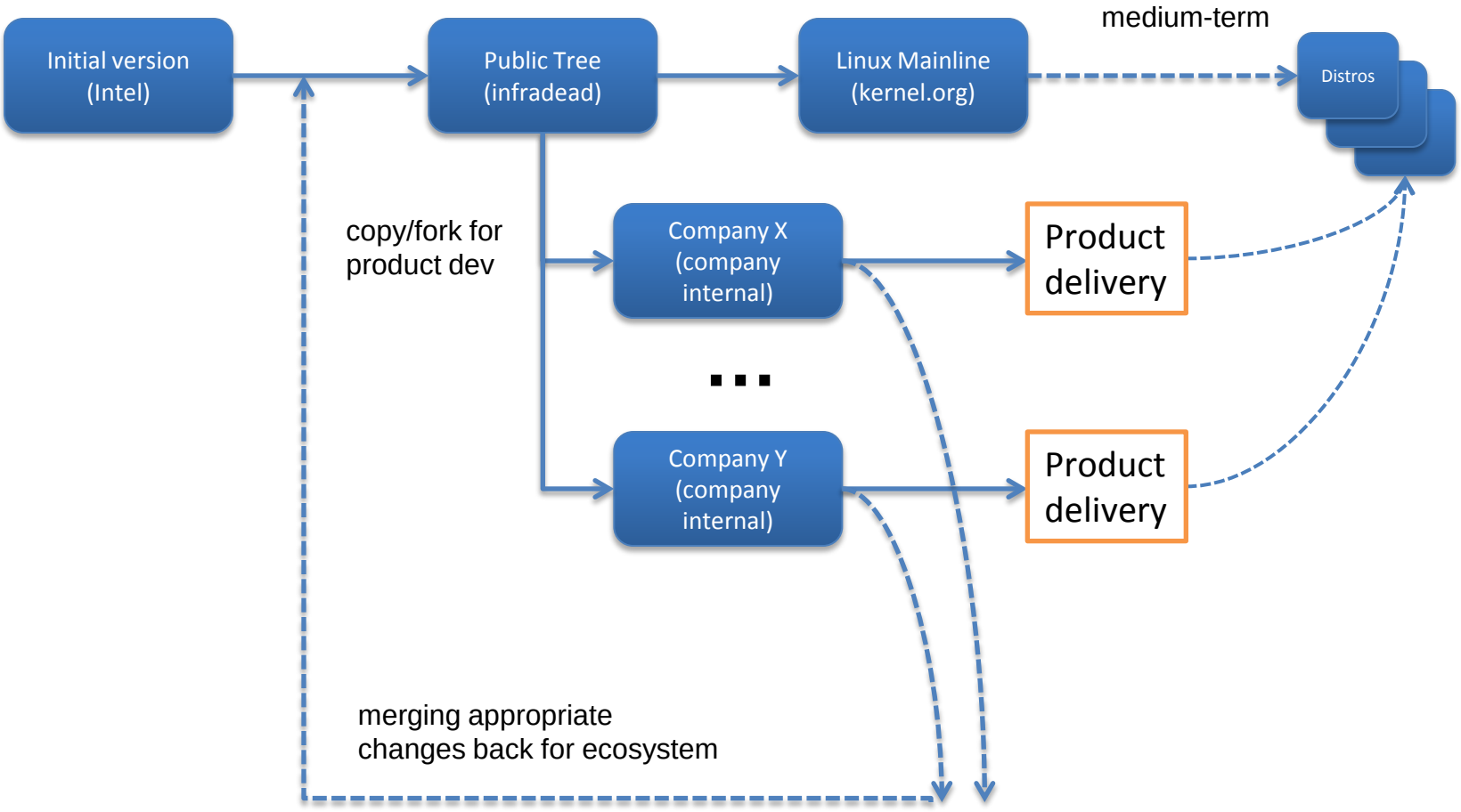
Testing Criteria:

- All patches and release candidates require, at a minimum, the following;
  - 1 hour of data integrity testing using sdstress (Microsoft Tool)
  - 1 hour of heavy stress testing using IoMeter covering, at least, 512B, 4KB and 128KB ranging from 1 OIO to 64 OIO both sequential and random
  - Quick and slow format of both MBR and GPT partitioning
  - Microsoft SCSI Compliance with no failures
- Testing done for all supported OSes for the release

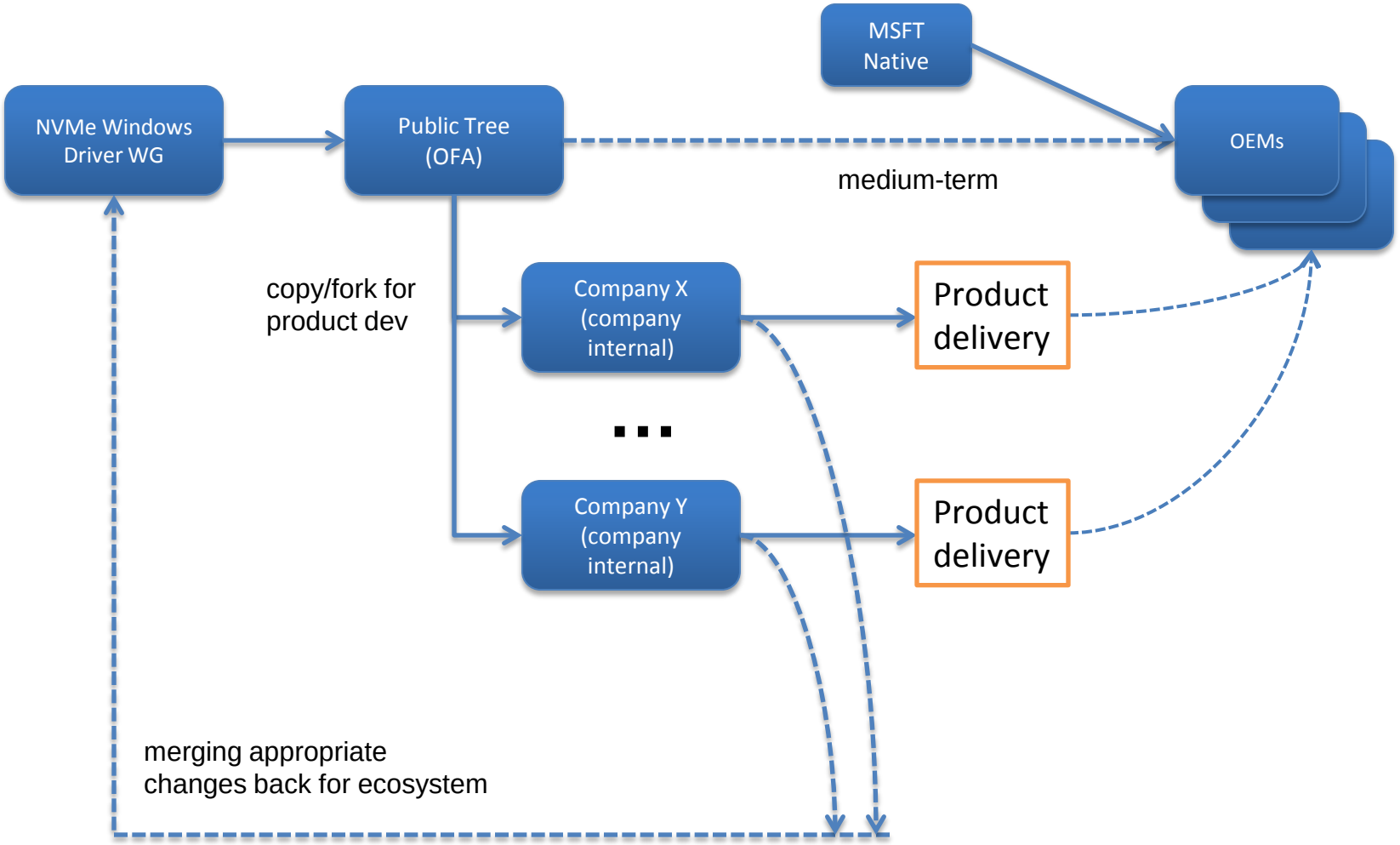
Driver Ecosystem Goals

- The long term goal is for each major OS to ship with a standard NVM Express driver
- The short term goal is to allow NVMe device manufacturers to provide the drivers they need with their products leveraging the reference drivers
- The reference drivers provide high performance, validated and fully compliant drivers to the ecosystem with reasonable licenses (e.g., GPL, BSD)
- "Fork and Merge" to achieve short term with reference drivers
  - Each NVMe device manufacturer "forks" the reference driver
  - Each NVMe device manufacturer adds in any product specific features
  - Each NVMe device manufacturer "merges" industry-wide applicable changes back to the reference driver

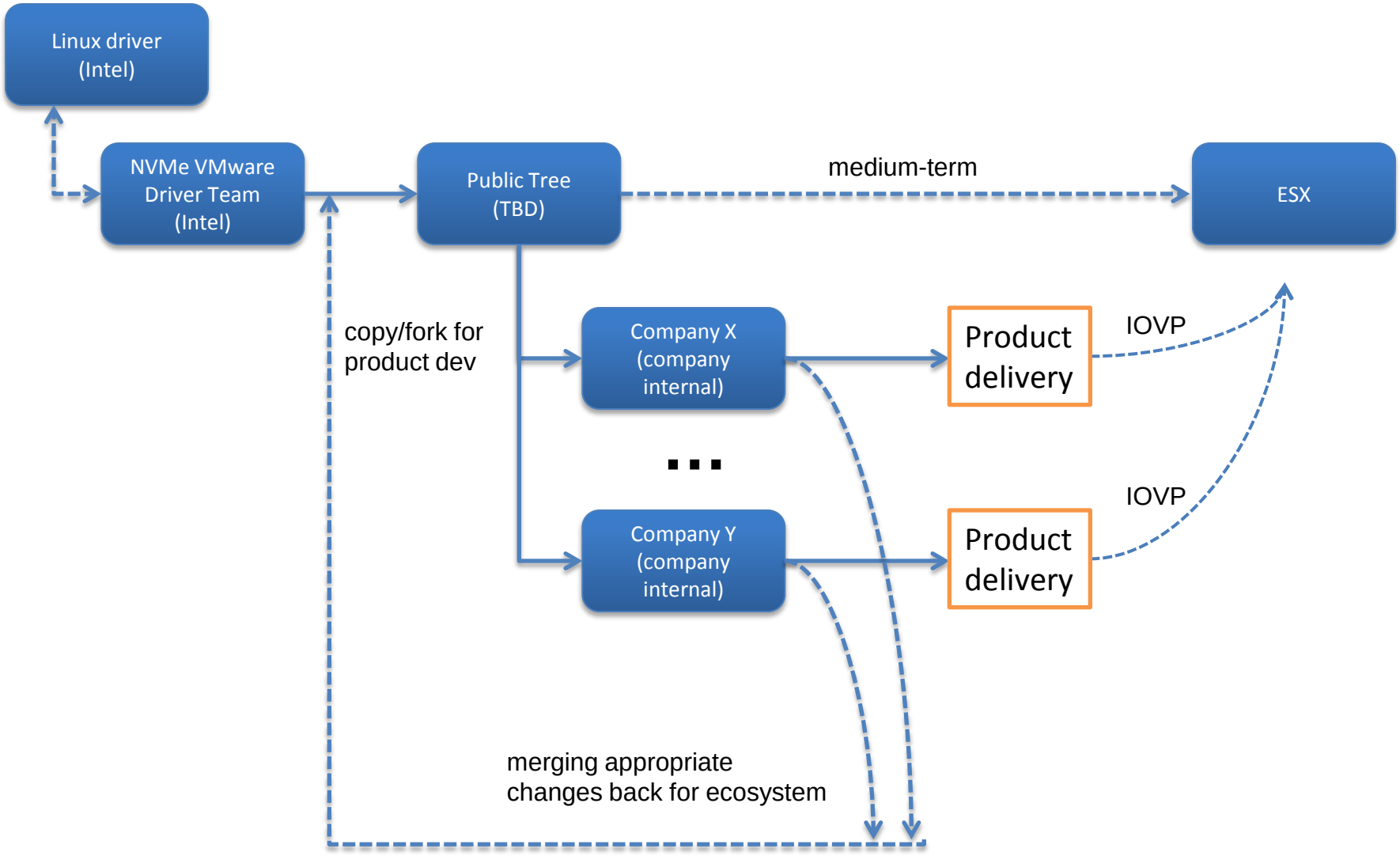
Linux* “Fork and Merge”



Windows* “Fork and Merge”



VMware* (non-native) “Fork and Merge”



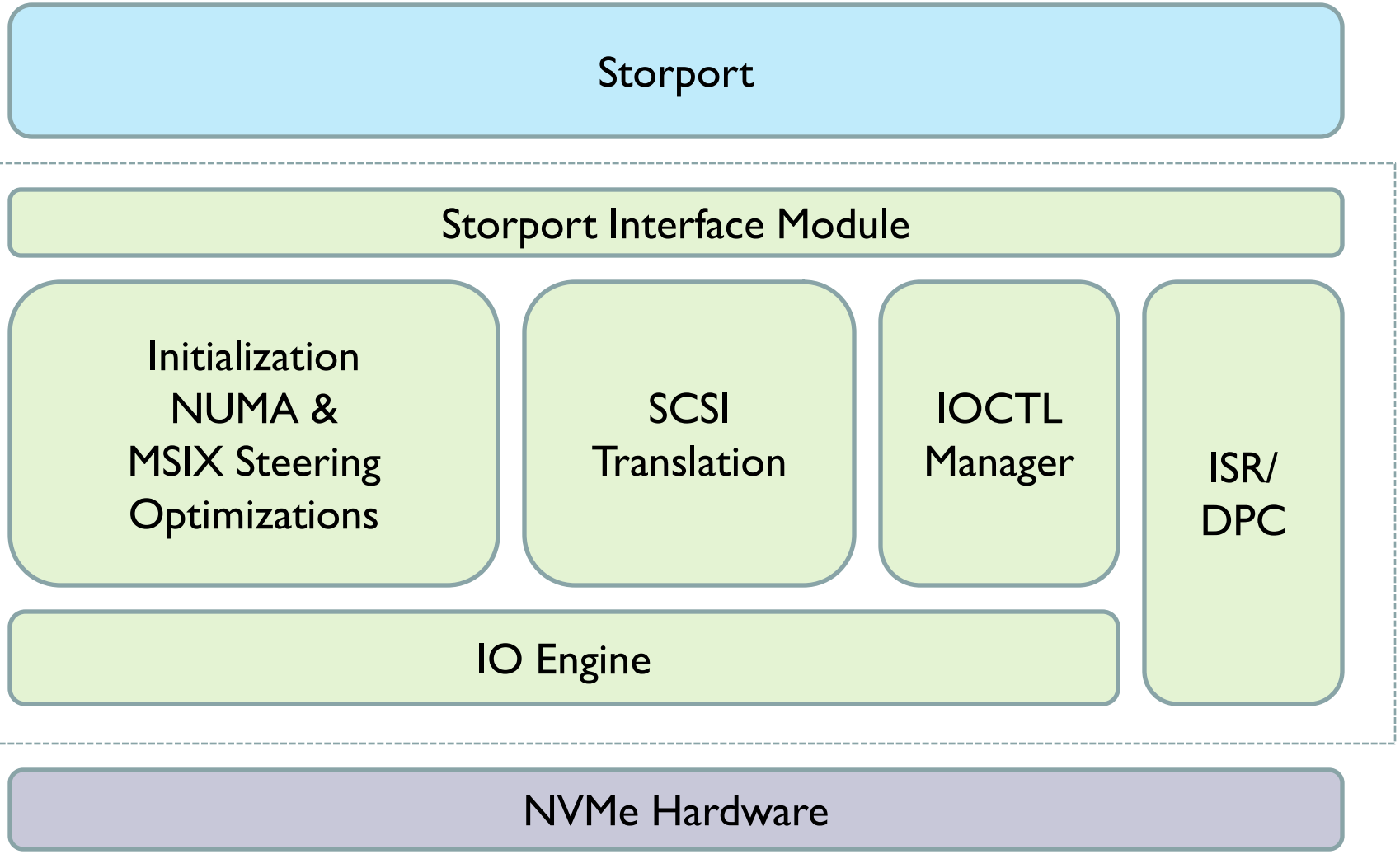
Fork and Merge Strategy Summary

- The benefits of the “Fork and Merge” strategy include:
  - Maximize re-use of reference code
  - Enable continuous improvement of the reference drivers
  - Enable product team to focus on delivery goals
- NVMe device manufacturer responsibilities:
  - Ensure drivers delivered with products have unique binary names
  - Ensure drivers delivered with products bind only to those products (e.g., Micron version of driver only loads against Micron NVMe SSDs)
  - Merge changes back to the reference driver when appropriate
- The NVMe community is available to support manufacturers:
  - Reference driver maintainers are available to review driver changes
  - Reference driver maintainers will provide guidance that maximizes the ability for a change to be general and flexible for eventual inclusion in the ecosystem driver.

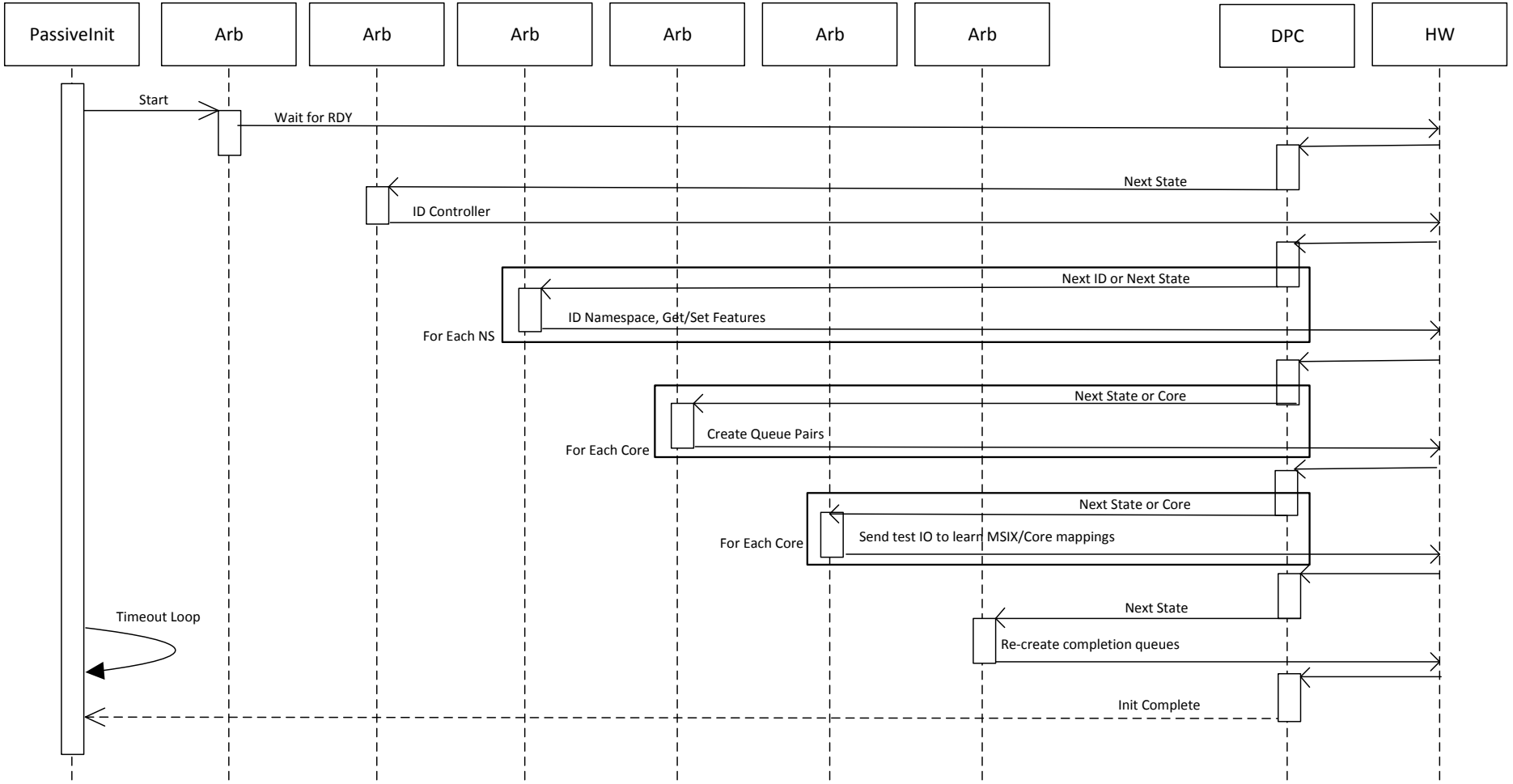
Take advantage of reference drivers, and then “give back” to further improve the ecosystem.

- NVM Express (NVMe) Overview
- New NVMe Features in Enterprise & Client
- Driver Ecosystem for NVMe
- **Windows NVMe Driver Architecture and Implementation Details**

NVMe Miniport Architecture



Initialization – Setting up for Performance



Performance Features

Taking full advantage of NVM Express

- Source Core Interrupt Steering
  - Initialization process "learns" MSIX to core mapping up front
  - IO Engine chooses the IO queue to use based on submitting core
  - Completions interrupt on the submitting core
- I/O Submission & Completion Queues Configuration
  - One queue pair dedicated per core
  - Allocated from closest NUMA node
- Pre-Allocation
  - All structs associated with IO are pre-allocated adhering to NVMe rules for alignment and are managed/re-used internally
- Use of BuildIO
  - NVMe submission queue entry is almost entirely constructed without holding locks; this includes SCSI translation and PRP construction