Flash Memory SUMMIT

nvm EXPRESS®

Architected for Performance

NVM Express® Device Drivers

Uma M. Parepalli

August 09, 2016

SK hynix memory solutions

Agenda – NVMe® Drivers

- Session Organization
- NVM Express® Driver Eco-System
- Individual Driver Presentations
- Q&A at end of all driver presentations

- Note: All registered trademarks, logos and brands are property of their respective owners

Session Organization

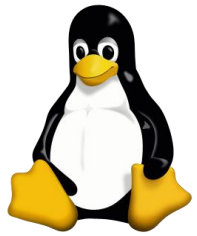
- Larry Chisvin, Broadcom (Session Organizer)
- Uma Parepalli, SK hynix memory solutions (Session Chair)
  - NVMe® Driver Ecosystem
  - UEFI NVMe® Drivers
  - Windows NVMe® Reference Driver
- Lee Prewitt, Microsoft
  - MS Windows NVMe® Inbox Drivers
- Parag Maharana, Seagate
  - Linux NVMe® Fabrics Drivers
- Jim Harris, Intel
  - FreeBSD NVMe® Driver
  - NVMe® Storage Performance Development Kit (SPDK)
- Sudhanshu (Suds) Jain, VMware
  - VMware NVMe® Driver
- Q&A

Session Organization

- Larry Chisvin, Broadcom (Session Organizer)
- Uma Parepalli, SK hynix memory solutions (Session Chair)
  - NVMe® Driver Ecosystem
  - UEFI NVMe® Drivers
  - Windows NVMe® Reference Driver (time permits)
- Lee Prewitt, Microsoft
  - MS Windows NVMe® Inbox Drivers
- Parag Maharana, Seagate
  - Linux NVMe® Fabrics Drivers
- Jim Harris, Intel
  - FreeBSD NVMe® Driver
  - NVMe® Storage Performance Development Kit (SPDK)
- Sudhanshu (Suds) Jain, VMware
  - VMware NVMe® Driver
- Q&A

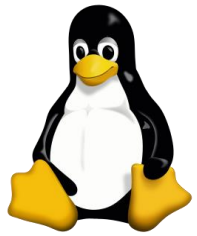
NVM Express® Driver Ecosystem

- All major OSes have stable Inbox Drivers
  - MS Windows
  - Linux
  - FreeBSD
  - VMware
  - Solaris
  - and UEFI BIOS Drivers




NVMe® Driver Ecosystem…

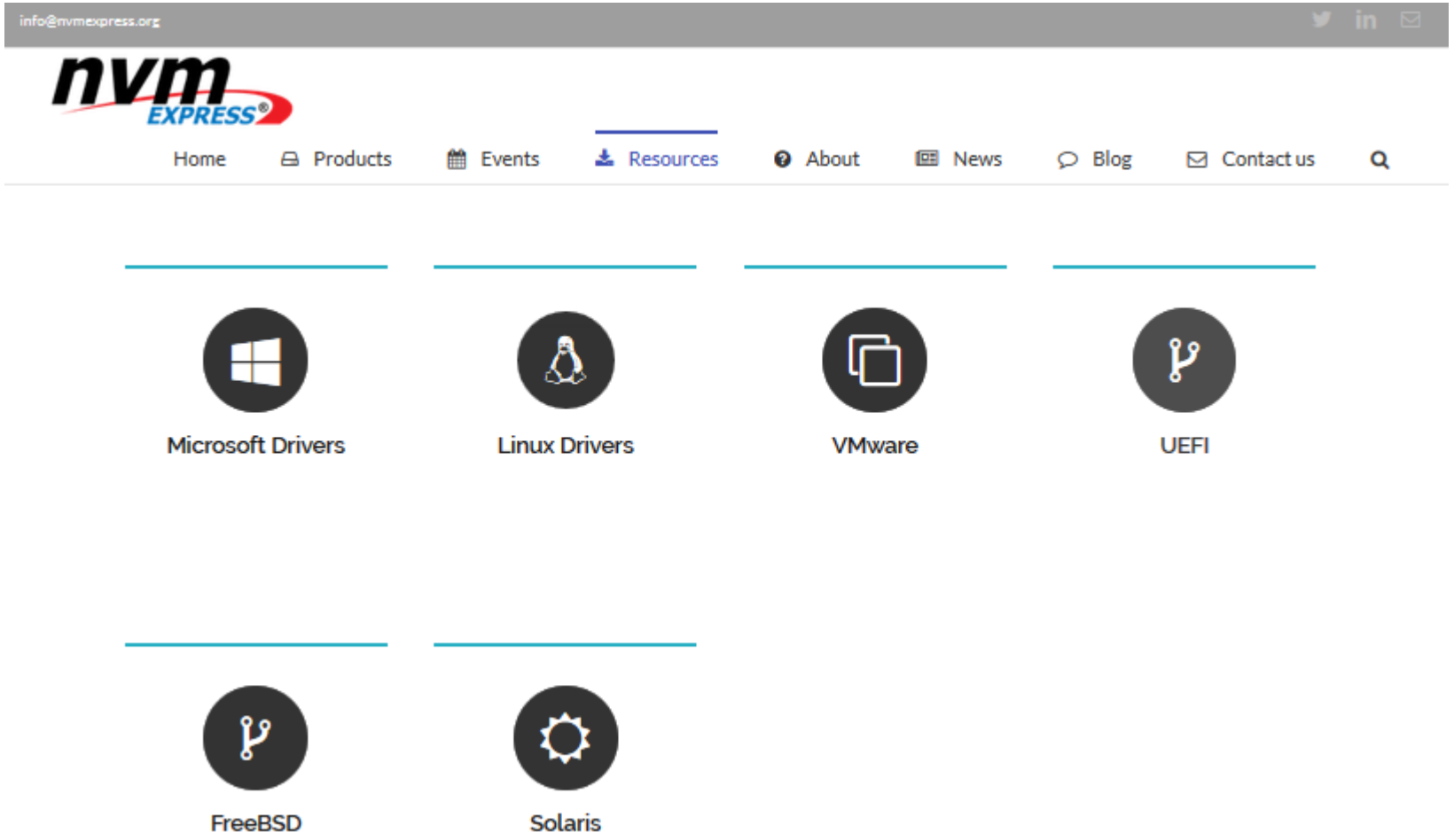
- Reference Drivers
  - Source Code is available
  - Contributions from NVM Express® members
  - Passionate driver developer contributors (unsung heroes)




Compliance / Plugfests

- The UNH-IOL - 6th NVMe® Plugfest - October 3 to 6, 2016
  - https://www.iol.unh.edu/testing/storage/nvme/grouptest
- PCI-SIG
  - https://pcisig.com/events/compliance-workshops
- Fall UEFI Plugfest in US – September 20-22, 2016, Seattle, WA
  - 2016 Fall UEFI Plugfest: http://www.uefi.org/2016FallUEFIPlugfest
  - UEFI Plugfest offers NVMe® SSD Vendors a unique opportunity to perform hardware/firmware/OS interoperability testing on future release / next generation OEM hardware, BIOS and Operating Systems all in one place

NVM Express® Website



NVMe® Drivers - Full Links

- NVM Express® Drivers: http://nvmexpress.org/drivers
- Microsoft Windows: https://support.microsoft.com/en-us/kb/2990941
- Linux: http://www.nvmexpress.org/resources/linux-driver-information/
- FreeBSD: http://svnweb.freebsd.org/base/head/sys/dev/nvme/nvme.h?view=log&pathrev=240616
- VMWare: https://github.com/vmware/nvme
- Solaris: https://docs.oracle.com/cd/E36784_01/html/E52463/makehtml-id-48.html
- UEFI Driver Source Code: https://svn.code.sf.net/p/edk2/code/trunk/edk2/MdeModulePkg/Bus/Pci/NvmExpressDxe/

Session Organization

- Larry Chisvin, Broadcom (Session Organizer)
- Uma Parepalli, SK hynix memory solutions (Session Chair)
  - NVMe® Driver Ecosystem
  - UEFI NVMe® Drivers
  - Windows NVMe® Reference Driver (time permits)
- Lee Prewitt, Microsoft
  - MS Windows NVMe® Inbox Drivers
- Parag Maharana, Seagate
  - Linux NVMe® Fabrics Drivers
- Jim Harris, Intel
  - FreeBSD NVMe® Driver
  - NVMe® Storage Performance Development Kit (SPDK)
- Sudhanshu (Suds) Jain, VMware
  - VMware NVMe® Driver
- Q&A

UEFI® BIOS NVM Express® Drivers

Uma M. Parepalli

SK Hynix Memory Solutions

Agenda – UEFI Drivers

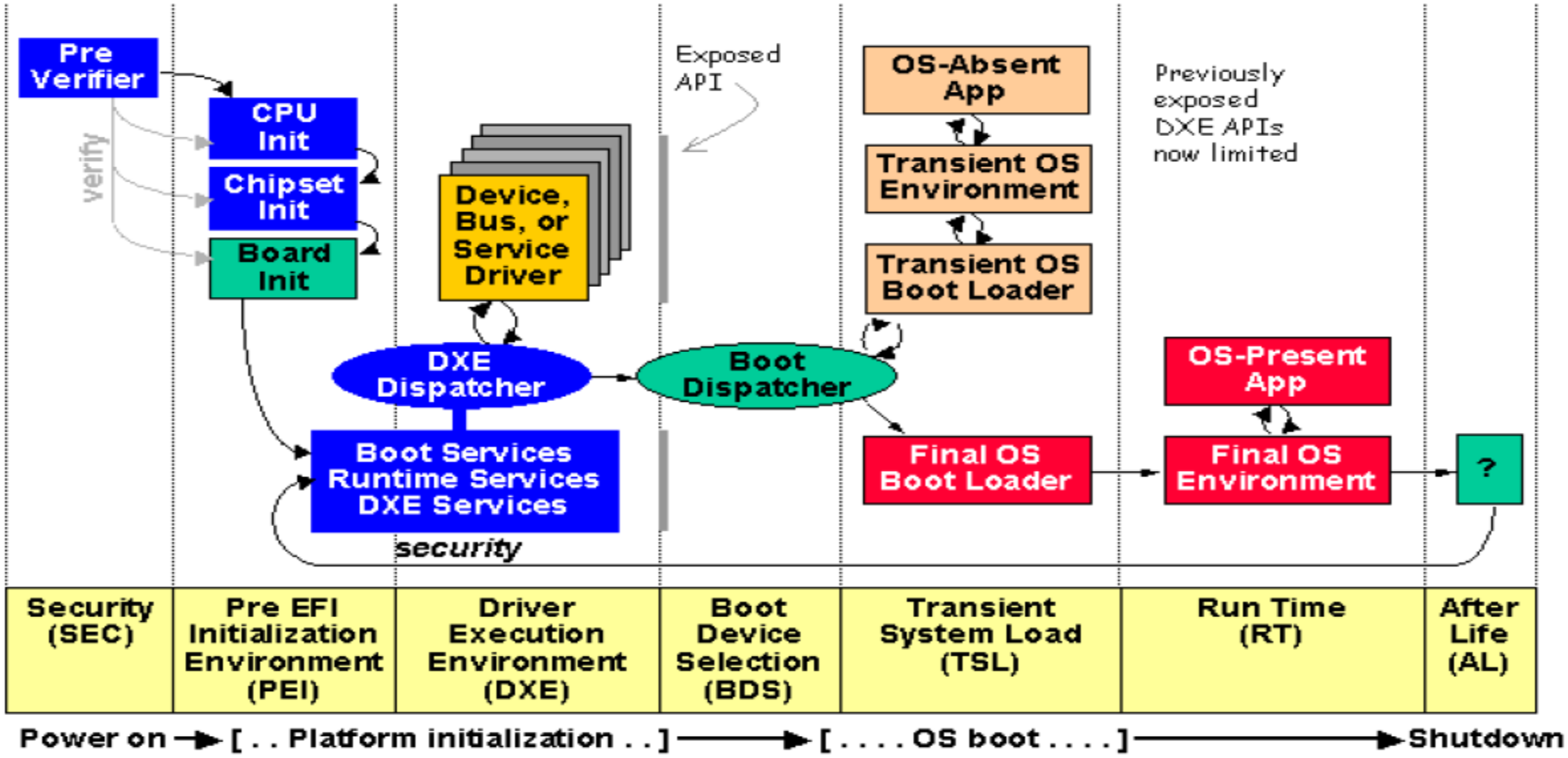
- Platform Firmware / UEFI BIOS Eco-System
- UEFI NVMe® Device Drivers
  - UEFI Reference Driver
  - OEM Custom Drivers
  - NVMe® SSD Vendor Customization / Value Additions
- Getting the most from UEFI NVMe® Drivers
- Resources

Unified Extensible Firmware Interface Forum

UEFI Eco-System

- UEFI Eco-System consists of
  - OEMs
  - OS Vendors (ISV)
  - BIOS Vendors (IBV)
  - HW Vendors (IHV)
- UEFI NVMe® Drivers are stable
- UEFI Website: http://uefi.org
- UEFI is well adopted by the industry
  - http://uefi.org/members

UEFI Platform Initialization Firmware Phases



UEFI NVMe® Drivers

- Major OEM Platform Specific Drivers
  - Built-in UEFI drivers
  - Automatic NVMe® device recognition & boot (from approved devices)
- Reference Driver
  - Useful for enabling debug, adding custom features
  - NVMe® SSD board bring-up and validation diagnostics
  - Use this on UEFI platforms that doesn't have built-in UEFI NVMe® Driver
  - Great opportunity to support latest NVMe® specification features
- IHV Drivers
  - Custom, OptionROM based and/or part of OEM Platform Firmware

Getting the most from UEFI NVMe® drivers

- UEFI driver is required for booting OS form NVMe® SSD
- Standard disk tools are available under UEFI shell
  - Create/delete/mount partitions & perform I/O
- Useful for NVMe® SSD bring-up without OS
  - From FPGA to end SSD product
- Useful for debugging from power-on to OS boot and beyond
- You can customize & implement your own features

Getting the most from UEFI NVMe® drivers - IHVs

- NVMe® SSD Vendors can implement support for generic & OEM specific requirements
  - Firmware Update Protocol
  - Driver Health Protocol
  - Diagnostics Support
  - Full BIST/POST diagnostics
  - HW Configuration / BIOS Menu Support using UEFI HII

Summary

- UEFI provides excellent environment for NVMe® SSD bring up from Power-on to OS boot, Shutdown/Restart and beyond.
- Debug and validation without OS present and at pre-OS boot level.
- For additional information contact Uma Parepalli and/or google search for “Uma Parepalli UEFI NVMe® Drivers”.
- Resources
  - http://www.uefi.org
  - http://www.tianocore.org

Got Feedback on UEFI NVMe® drivers?

Send email to:

umaparepalli@gmail.com

Session Organization

- Larry Chisvin, Broadcom (Session Organizer)
- Uma Parepalli, SK hynix memory solutions (Session Chair)
  - NVMe® Driver Ecosystem
  - UEFI NVMe® Drivers
  - Windows NVMe® Reference Driver (time permits)
- Lee Prewitt, Microsoft
  - MS Windows NVMe® Inbox Drivers
- Parag Maharana, Seagate
  - Linux NVMe® Fabrics Drivers
- Jim Harris, Intel
  - FreeBSD NVMe® Driver
  - NVMe® Storage Performance Development Kit (SPDK)
- Sudhanshu (Suds) Jain, VMware
  - VMware NVMe® Driver
- Q&A

MS Windows Reference Driver

Uma Parepalli, SK hynix memory solutions

Credits: Raymond C Robles, Intel

PCIe NVMe® Windows Reference Driver

- Next release plan - End of August 2016
- Recent patches include
  - Performance improvement & robustness
  - NVMe® Spec rev 1.2 feature compliant
- Supports MS Windows 10, 8.1, 7, Server 2012 R2, 2012 and 2008 R2
- Supports both 32 & 64-bit

PCIe NVMe® Windows Reference Driver

- What is new since last year
  - Namespace Management (Create, Delete, Attach, Detach)
  - EOL Read Only Support
  - Win 8.1 Timers
  - Surprise Removal Support in IOCTL Path
  - Disk Initialization Performance Optimization
  - Storage Request Block Support
  - StorPort Performance Options
  - StorPort DPC Redirection
  - Concurrent Channels (wrapping up review)
  - Misc. Bug Fixes
    - Security Send/Receive with Zero Data Length
    - SNTI updates for SCSI to NVMe® Translation

Thank You!

Architected for Performance

Session Organization

- Larry Chisvin, Broadcom (Session Organizer)
- Uma Parepalli, SK hynix memory solutions (Session Chair)
  - NVMe® Driver Ecosystem
  - UEFI NVMe® Drivers
  - Windows NVMe® Reference Driver (time permits)
- **Lee Prewitt, Microsoft**
  - **MS Windows NVMe® Inbox Drivers**
- Parag Maharana, Seagate
  - Linux NVMe® Fabrics Drivers
- Jim Harris, Intel
  - FreeBSD NVMe® Driver
  - NVMe® Storage Performance Development Kit (SPDK)
- Sudhanshu (Suds) Jain, VMware
  - VMware NVMe® Driver
- Q&A

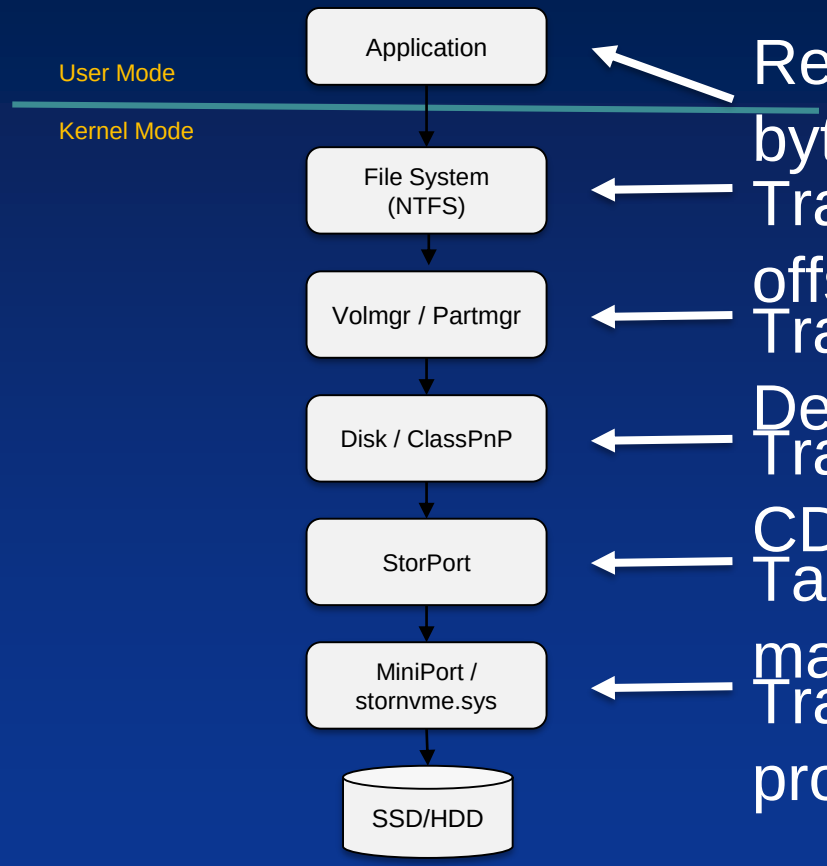
Microsoft Inbox NVMe® Driver

Lee Prewitt
Principle Program Manager
Storage and File Systems - Microsoft

Microsoft Inbox Driver

- Introduced in Windows 8.1/Server 2012r2
  - Aligned to NVMe® 1.0C
- Backported to Windows 7/Server 2008r2
- Stornvme.sys is a Storport mini-port
  - Mini-port is responsible for doing the SCSI to NVMe® translation

The Windows Storage Stack



Reads Bytes from or Writes bytes to a file buffer

Translates file offsets to byte offsets into the volume

Translates Volume offsets to Device offsets

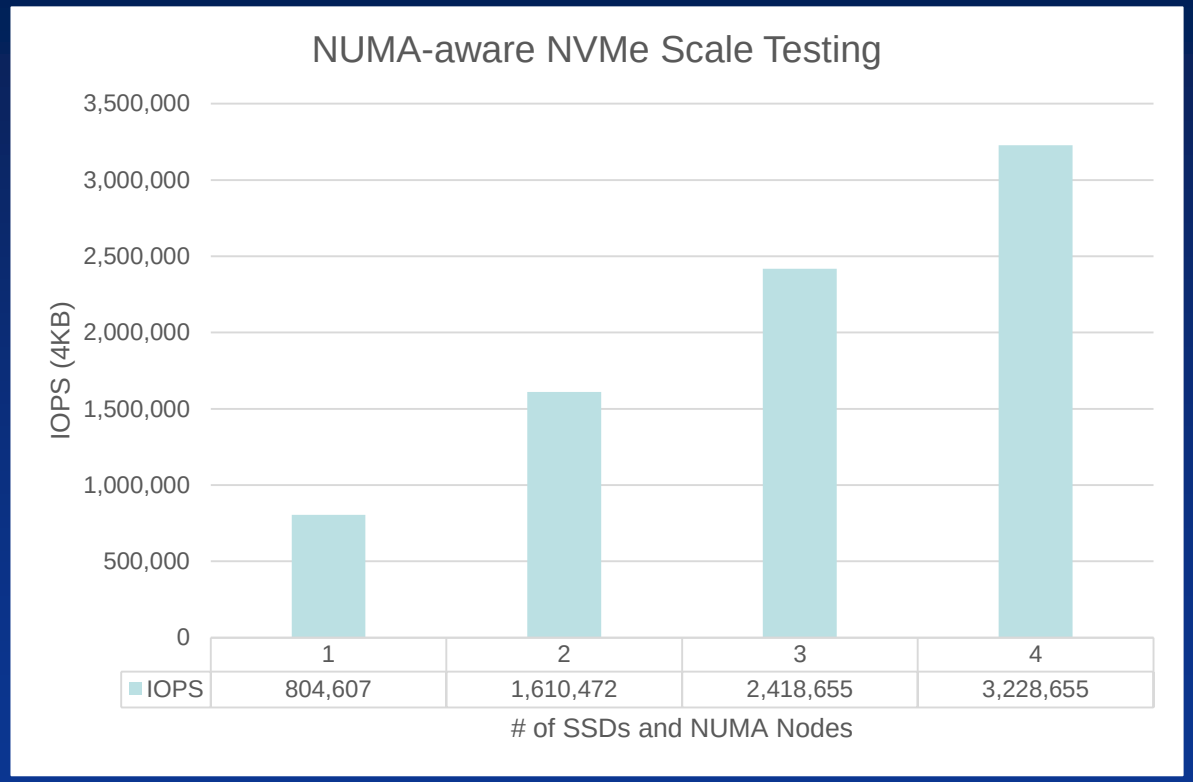
Translates IOCTLs to SCSI CDBs

Takes care of Queuing, Power management, etc.

Translates SCSI to native protocol (NVMe®)

Performance

NUMA Aware NVMe Scale Testing



System:	Intel Brickland
Cores:	60 @ 2.8GHz
SSD:	4x 800GB
Workload:	4KB Random Read 15 Threads per NUMA Node 16 I/Os per Thread

# SSDs/NUMA	1	2	3	4
Avg. CPU Util	14.50%	31.58%	49.68%	69.40%
MB/s	3,143	6,291	9,448	12,612
IOPS	804,607	1,610,472	2,418,655	3,228,655
Avg. Lat (ms)	0.298	0.298	0.297	0.297

- On par with 3rd party NVMe drivers

Reliability

- Highly robust driver
  - Telemetry shows us that the In-box driver has a very low crash rate

Driver	Unique Machine Crash Rate (%)
stornvme.sys	0.57
Vendor A	1.48
Vendor B	23.58

Power Efficient

- Tuned for Modern Standby
  - Good battery life for laptops
  - Operational power states (the device can handle IO) map to logical performance states (P-states)
  - Non-operational power states map to logical idle power states (F-States)
  - Transitions to these states is largely determined by the overall system power state
  - Can be tuned or disabled via the inbox powercfg tool

Possible Future Directions

- What would you like to see?
  - Streams
  - Write Protect and RPMB
  - Name Space Management
  - Virtualization

Not Plan of Record Yet

Call to Action

- Try out the inbox driver with your devices and give us feedback
- Contact Info:
  - leprewit@microsoft.com

Appendix

Supported Commands

Delete I/O Submission Queue	Firmware Commit
Create I/O Submission Queue	Firmware Image Download
Get Log Page	Format NVM
Delete I/O Completion Queue	Security Send
Create I/O Completion Queue	Security Receive
Identify	Vendor Specific
	Flush
Set Features	Write
Get Features	Read
Asynchronous Event Request	Dataset Management

Unsupported Commands

- Namespace Management
- Namespace Attachment
- Write Uncorrectable
- Compare
- Write Zeroes
- Reservation Register
- Reservation Report
- Reservation Acquire
- Reservation Release

Support for Pass Through

- Uses the Command Effects Log to ensure seamless IO
  - If the command is informational, then it is sent down with regular IO
  - If the command has side effects, then IO is paused, the queue is drained and then the command is sent
  - Once the command is completed, IO is resumed
- Allows for vendor-specific functionality within the inbox driver

Support for Pass Through

- Documentation on MSDN

https://msdn.microsoft.com/en-us/library/windows/desktop/mt718131(v=vs.85).aspx

Support for Host Memory Buffer

- New preview support for HMB in Windows 10 Anniversary Edition
  - Off by default
  - Can be enabled through a registry key
    - Please contact Microsoft if you are interested in testing

Thank You!

Architected for Performance

www.nvmexpress.org

Session Organization

- Larry Chisvin, Broadcom (Session Organizer)
- Uma Parepalli, SK hynix memory solutions (Session Chair)
  - NVMe® Driver Ecosystem
  - UEFI NVMe® Drivers
  - Windows NVMe® Reference Driver (time permits)
- Lee Prewitt, Microsoft
  - MS Windows NVMe® Inbox Drivers
- **Parag Maharana, Seagate**
  - **Linux NVMe® Fabrics Drivers**
- Jim Harris, Intel
  - FreeBSD NVMe® Driver
  - NVMe® Storage Performance Development Kit (SPDK)
- Sudhanshu (Suds) Jain, VMware
  - VMware NVMe® Driver
- Q&A

NVMe Over Fabrics Linux Driver Overview

Parag Maharana
Seagate

NVMe Linux Fabrics Drivers

- Linux Fabrics Drivers are based on Fabrics Spec 1.0 and Core Spec 1.2.1
- Linux Host driver is re-architected to support multiple transports (PCIe and Fabrics)
  - Reused ~90% of existing host driver code
  - Added template method to support multiple transports
    - For example – nvme_ctrl_ops have function pointers to implement reg_read32, but PCIe and Fabrics will implement differently. However, the common core will operate on reg_read32 for both PCIe and Fabrics transport in transport agnostic way
- Linux Fabrics Driver has Host and Target components:
  - Host has Core, PCIe and Fabric modules
  - Target components has Core and Fabric modules
- Linux Fabrics Driver will be part of Linux Kernel 4.8

Driver Development Environment and Methodology

- Initial Host and Target drivers were developed by multiple NVMe member companies prior the specifications becoming public
  - All Linux Driver WG members have access to the repository with exclusive NDA
  - *for-next* branch is where the latest approved code resides
  - Developers create git patches and email them to the WG reflector
    - Note: Some at super-human rates
  - WG members approve and/or comment on the new patches
  - Maintainer integrates approved patches into the for-next branch
- Now Host and Target source codes moved to infradead repository
  - Multiple members actively submitting patches
  - Adding new functionality based on latest fabrics specifications
  - Fixing bugs that have been identified during testing
  - Several rebases to latest upstream Linux kernel functionality
    - New RDMA APIs were introduced in 4.5

Current Functionality Implemented

NVMe Host Driver

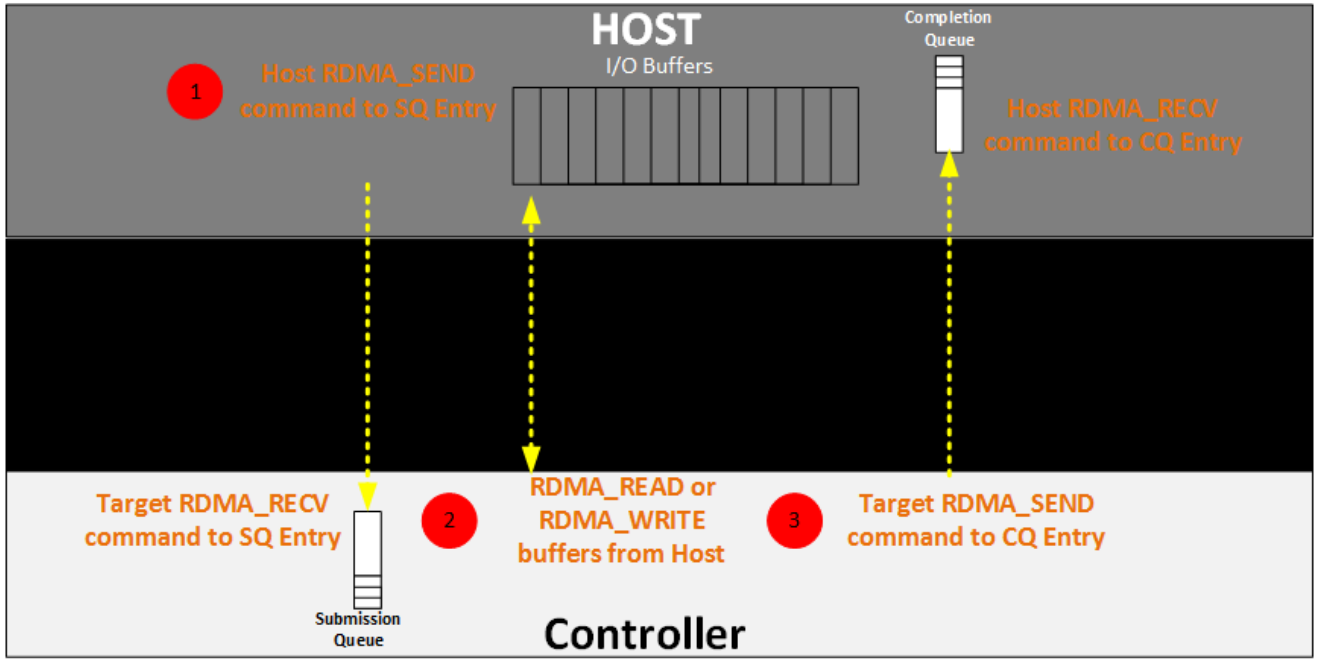
- ✓ Support for RDMA transport (Infiniband™/RoCE™/iWARP™/Intel OmniPath®)
- ✓ Connect/Disconnect to multiple controllers
- ✓ Transport of NVMe commands/data generated by NVMe core
- ✓ Initial Discovery service implementation
- ✓ Multi-Path
- ✓ Keep Alive

NVMe Target Driver

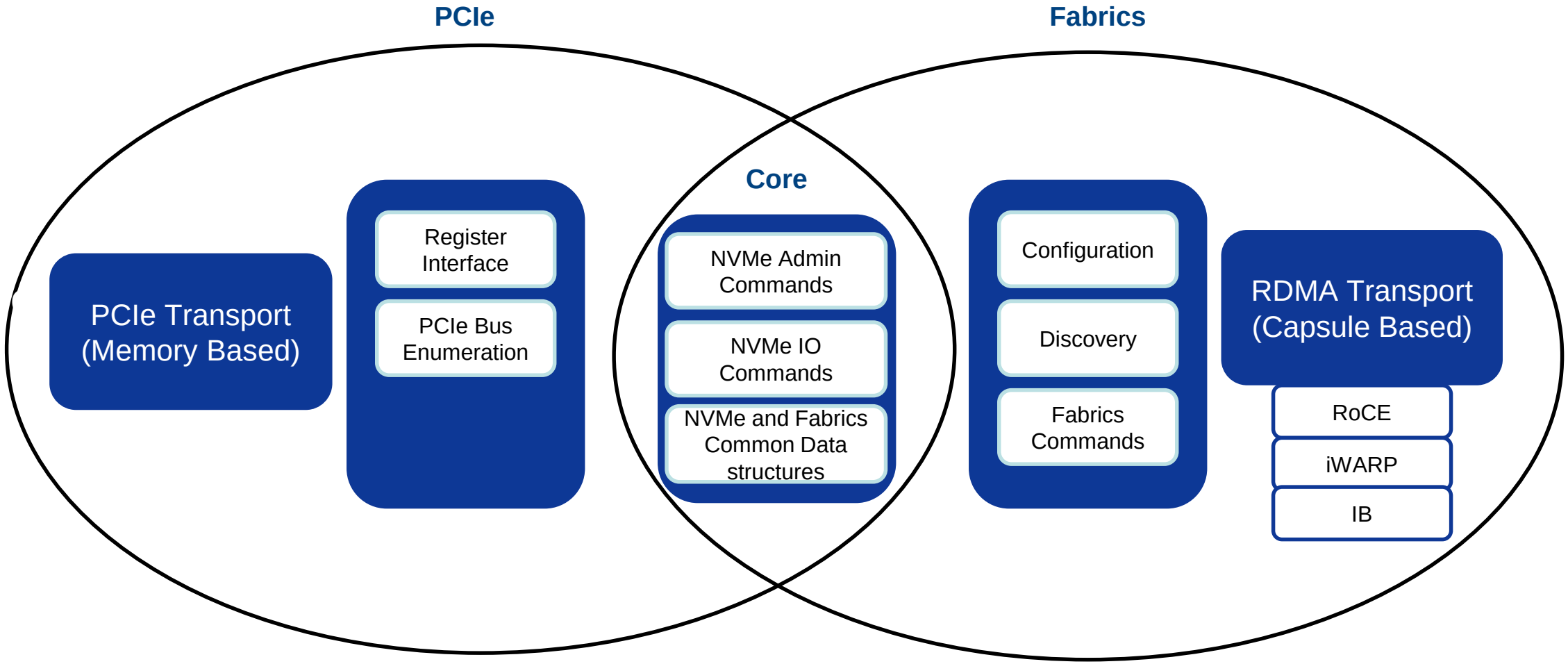
- ✓ Support for mandatory NVMe and Fabrics commands
- ✓ Support for multiple hosts/subsystems/controls/namespaces
- ✓ Namespaces backed by <any> Linux block devices
- ✓ Initial Discovery service; Discovery Subsystem/Controller(s)
- ✓ Target Configuration interface using Linux configfs
  - ✓ Create NVM and Discovery Subsystems

Fabrics Queuing Model

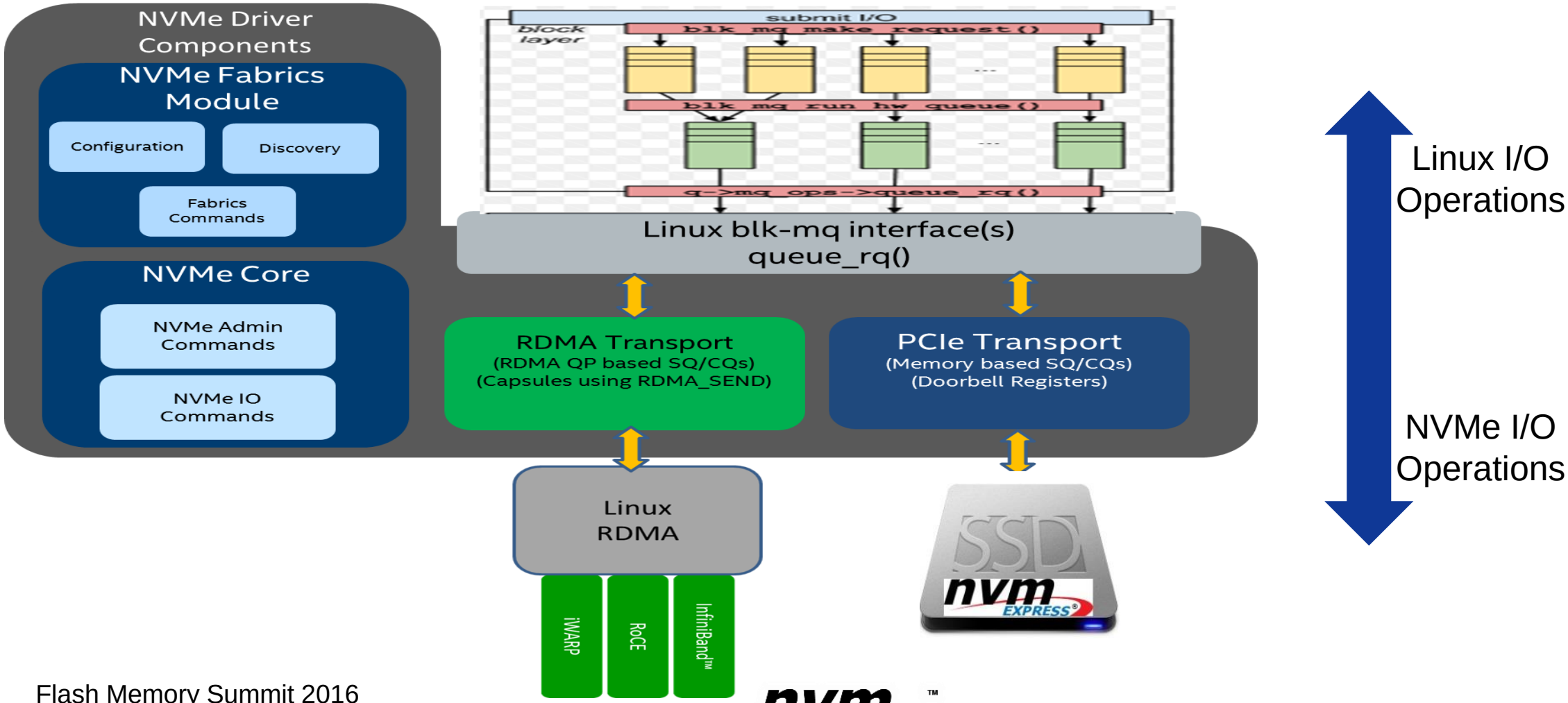
1. Host send RDMA_SEND that update in target as RDMA_RECV in target SQ
2. Target issue RDMA_READ or RDMA_WRITE to access data in host memory for Read or Write respectively
3. On completion target update in host CQ using RDMA_SEND that is received by host as RDMA_RECV
4. NVMe over Fabrics does not define an interrupt mechanism that allows a controller to generate a host interrupt. It is the responsibility of the host fabric interface (e.g., Host Bus Adapter) to generate host interrupts



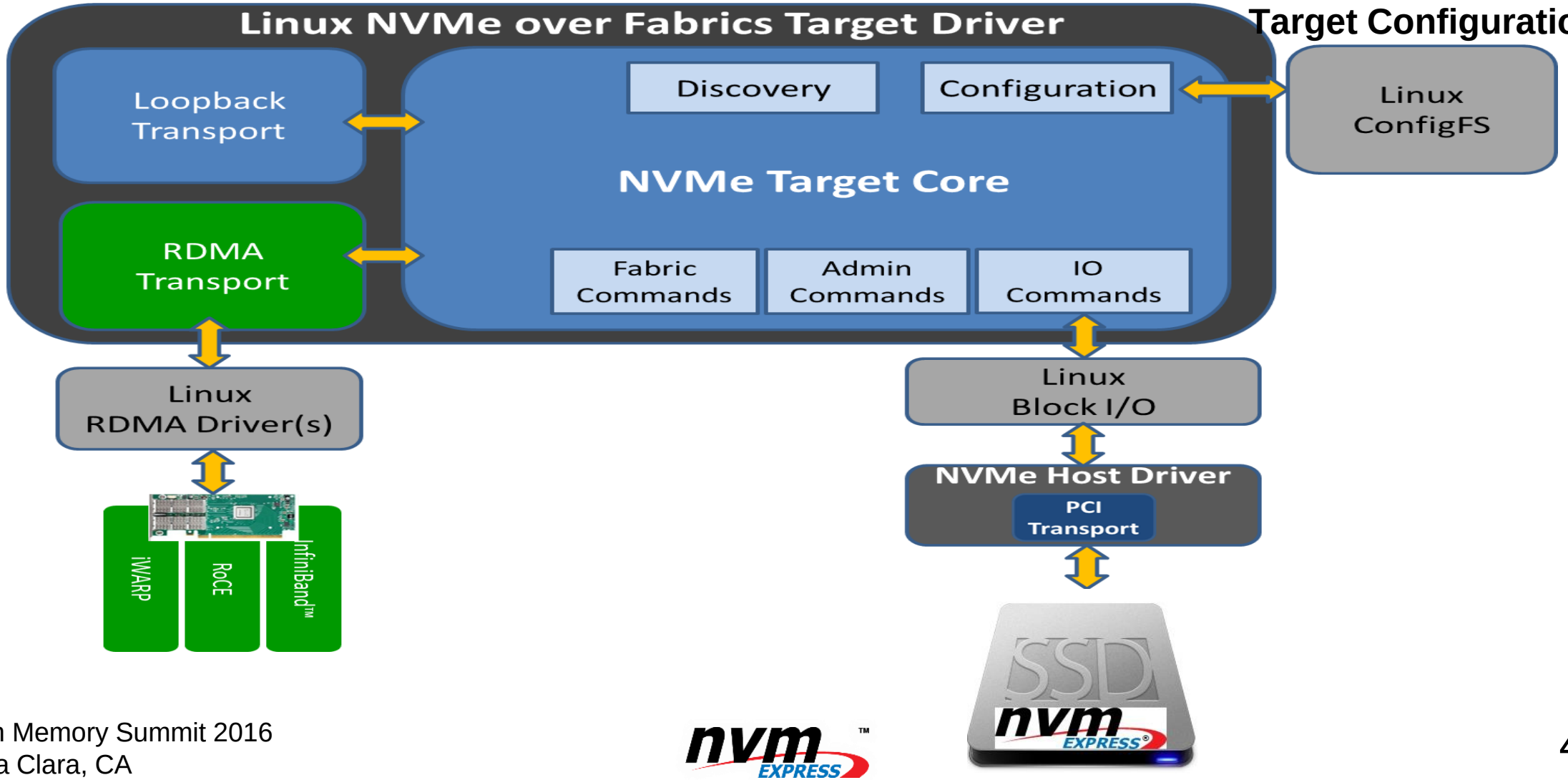
NVMe Over Fabrics Host and Target Driver Components



NVMe Over Fabrics Host Driver Components



NVMe over Fabrics Target Driver Components



Linux Driver WG Next Steps

- Next steps
  - Fibre Channel (FC) Fabric transport planned for Kernel 4.9
  - Authentication features
  - Controller Memory Buffer
  - Automated host multi-path
  - Log page support (smart log pages, error log pages, …)

- Call for Action:
  - Download driver and try it out
  - Provide suggestion/comment/feedback
  - Suggest any future enhancement

Linux Driver Reference

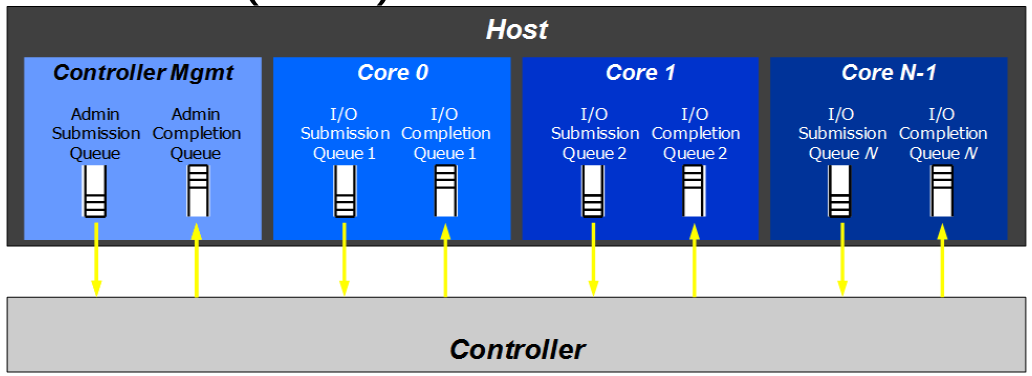
- Linux Fabrics drivers are public from June 2016
  - NVMe Specification
    - http://www.nvmexpress.org/specifications/
  - NVMe Fabric Driver Resource
    - http://www.nvmexpress.org/resources/nvme-over-fabrics-drivers/
  - NVMe Linux Fabric Drivers Source
    - http://git.infradead.org/nvme-fabrics.git
  - NVMe Linux Fabric Mailing List
    - linux-nvme@lists.infradead.org

Backup

NVMe® over Fabrics

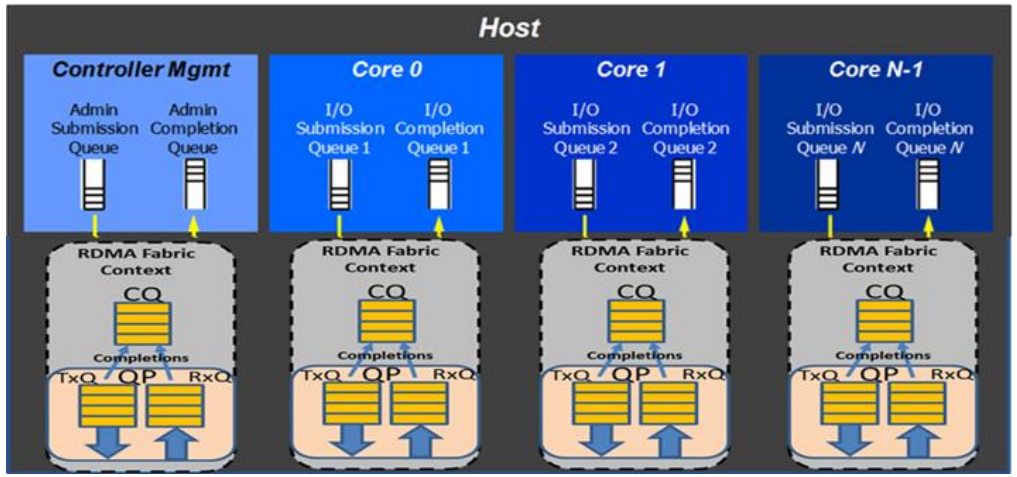
NVMe Multi-Queue Host Interface Maps Neatly to the RDMA Queue-Pair Model

Standard (local) NVMe



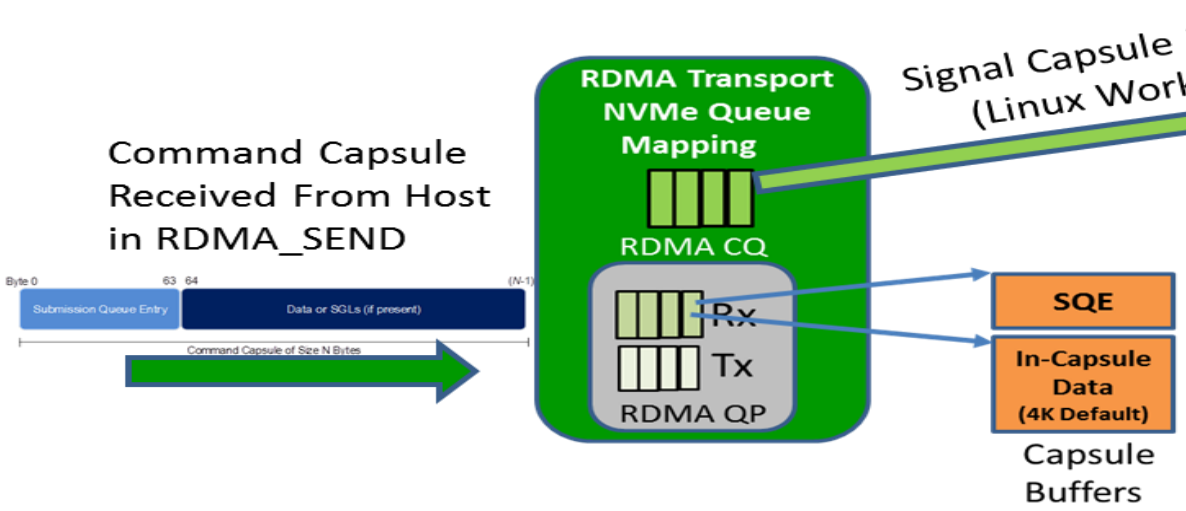
- NVMe Submission and Completion Queues are aligned to CPU cores
- No inter-CPU software locks
- Per CQ MSI-X interrupts enable source core interrupt steering

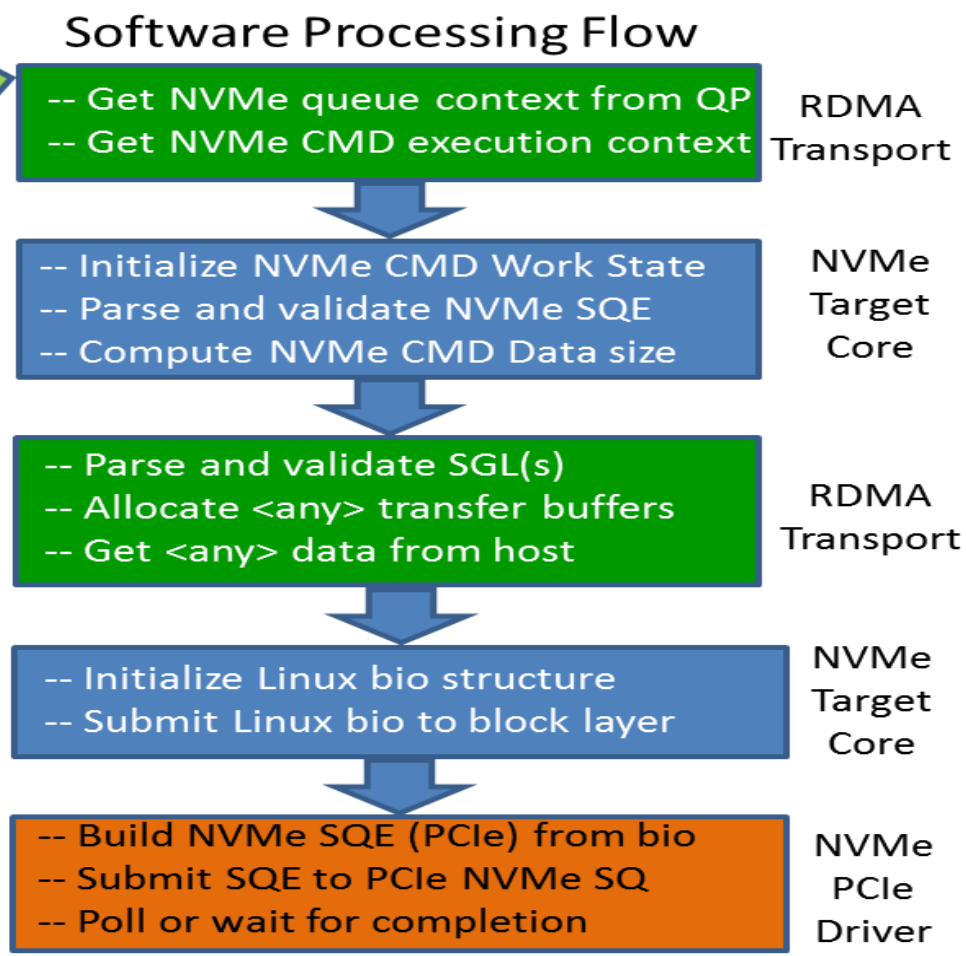
NVMe Over RDMA Fabrics



- Retains NVMe SQ/CQ CPU alignment
- No inter-CPU software locks
- Source core interrupt steering retained by using RDMA Event Queue MSI-X interrupts

NVMe Target Driver Command Capsule Rx Flow

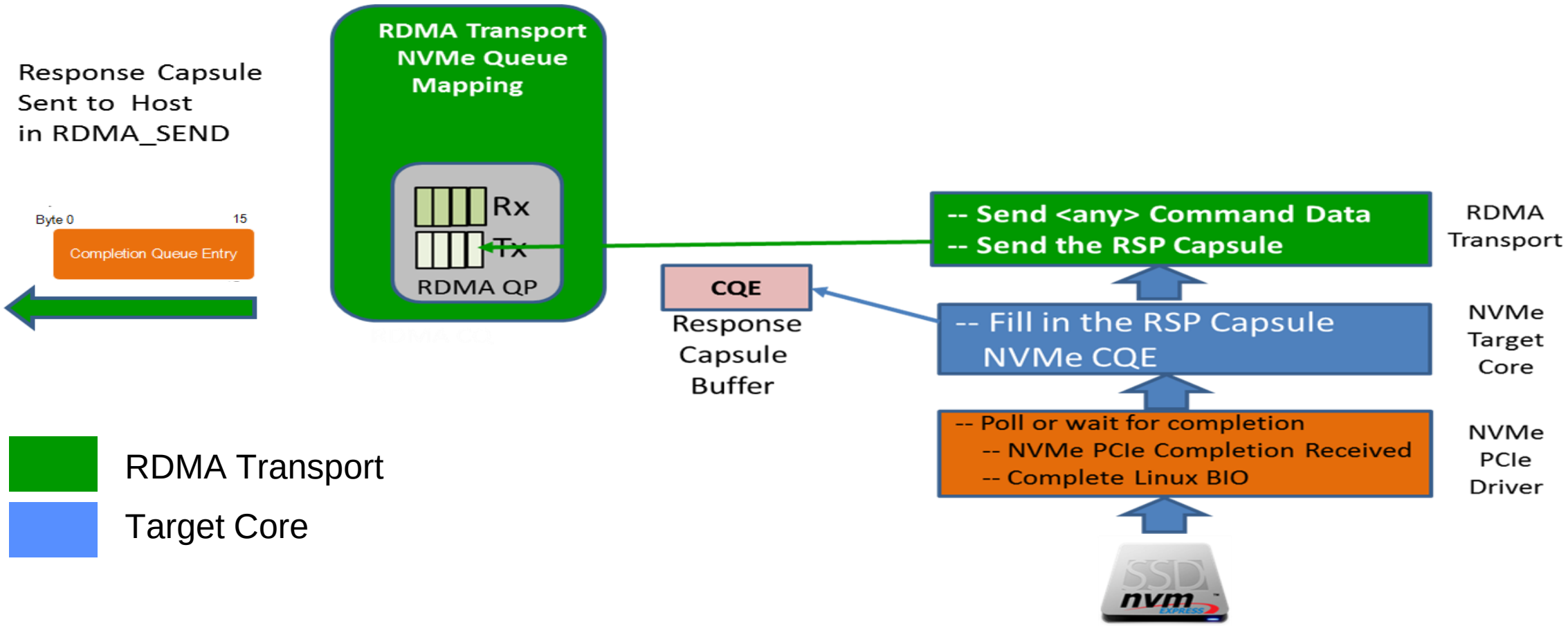




RDMA Transport

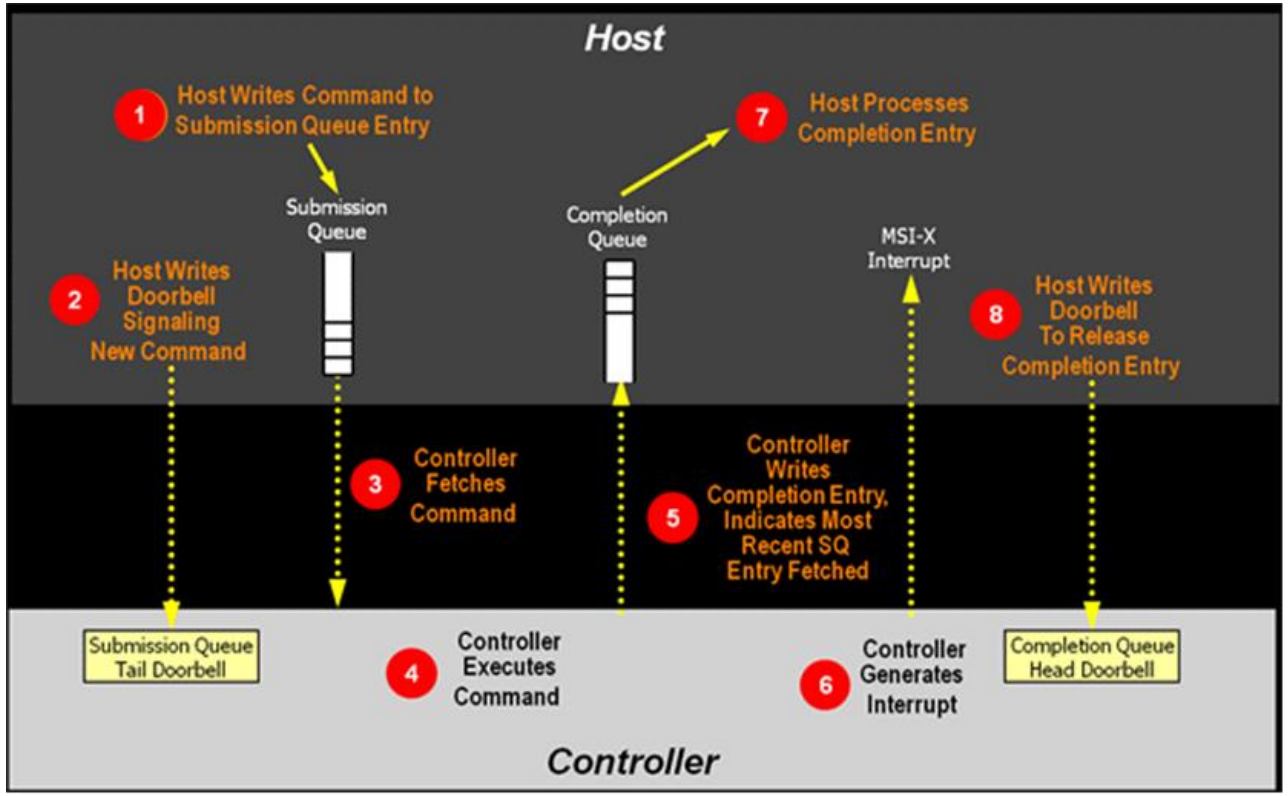
Target Core

NVMe Target Driver Response Capsule Tx Flow



PCIe Memory Queuing Model

1. Host writes command to SQ
2. Host writes SQ tail pointer for doorbell
3. Controller fetches command
4. Controller processes command
5. Controller writes completion to CQ
6. Controller generates MSI-X interrupt
7. Host processes completion
8. Host writes to CQ head pointer for doorbell



Thank You!

Architected for Performance

Session Organization

- Larry Chisvin, Broadcom (Session Organizer)
- Uma Parepalli, SK hynix memory solutions (Session Chair)
  - NVMe® Driver Ecosystem
  - UEFI NVMe® Drivers
  - Windows NVMe® Reference Driver (time permits)
- Lee Prewitt, Microsoft
  - MS Windows NVMe® Inbox Drivers
- Parag Maharana, Seagate
  - Linux NVMe® Fabrics Drivers
- **Jim Harris, Intel**
  - **FreeBSD NVMe® Driver**
  - **NVMe® Storage Performance Development Kit (SPDK)**
- Sudhanshu (Suds) Jain, VMware
  - VMware NVMe® Driver
- Q&A

NVMe®* Drivers – SPDK and FreeBSD

Jim Harris

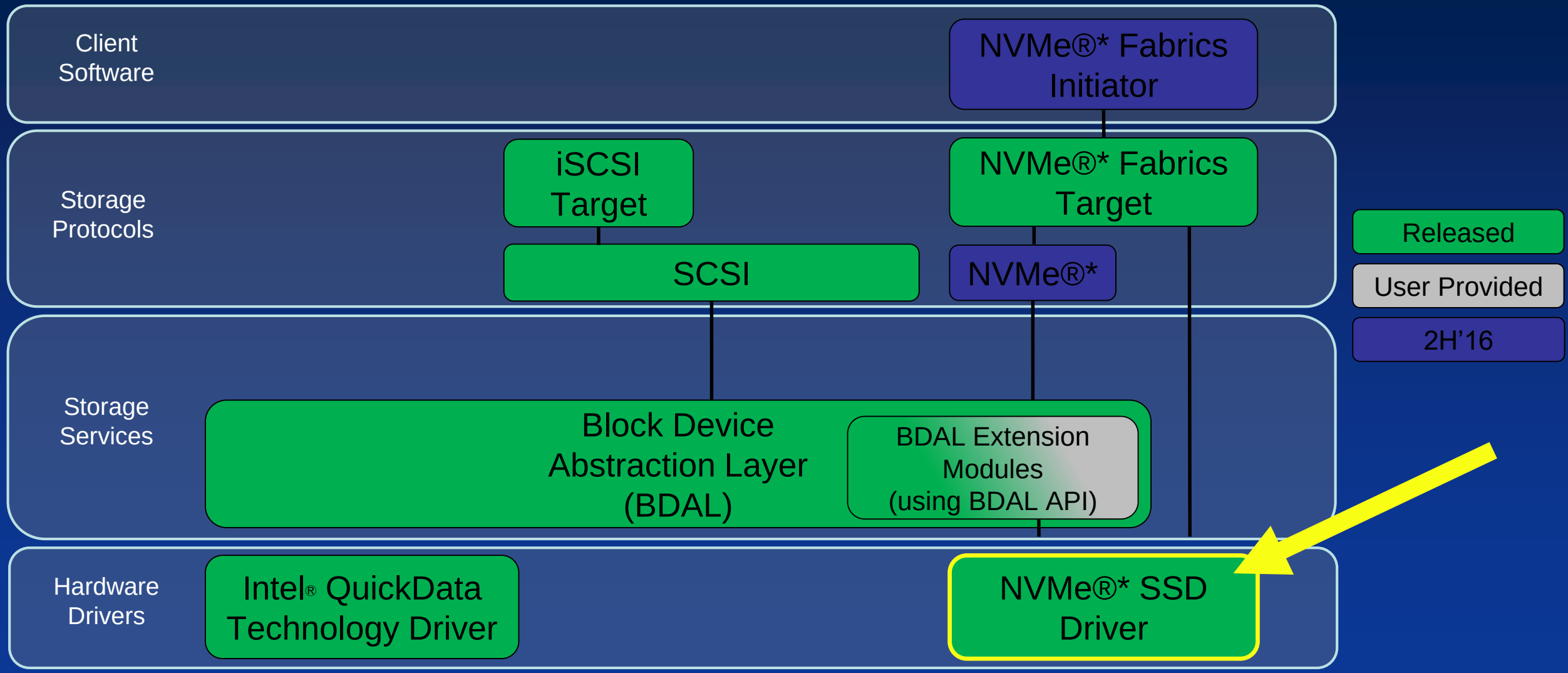
Software Architect

Intel Data Center Group

Storage Performance Development Kit (SPDK)

- Set of software building blocks for scalable efficient storage applications
  - Polled-mode and user-space drivers and protocol libraries (including NVMe®*)
  - Leverages Data Plane Development Kit (DPDK)
- Designed for next generation NVM media latencies (i.e. 3D XPoint™ media):
- BSD licensed
  - Source code: http://github.com/spdk
  - Project website: http://spdk.io

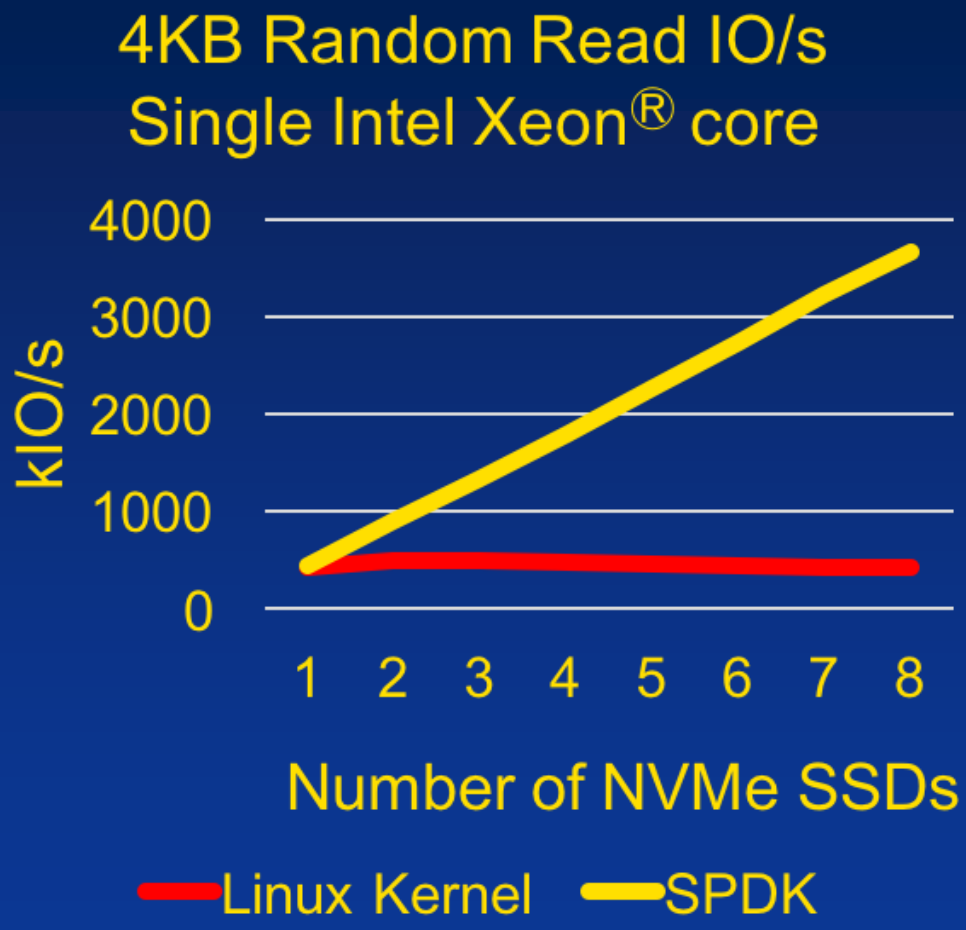
Storage Performance Development Kit (SPDK)



NVMe®* Driver Features

- NVMe®* 1.2 Spec Compliant
- Asynchronous Polled Mode operation
  - Application owns I/O queue allocation and synchronization
- Optional features implemented:
  - Weighted Round Robin
  - Controller Memory Buffer
  - End-to-end Data Protection
  - Reservations
  - Scatter Gather List

NVMe®* Driver Throughput Scalability

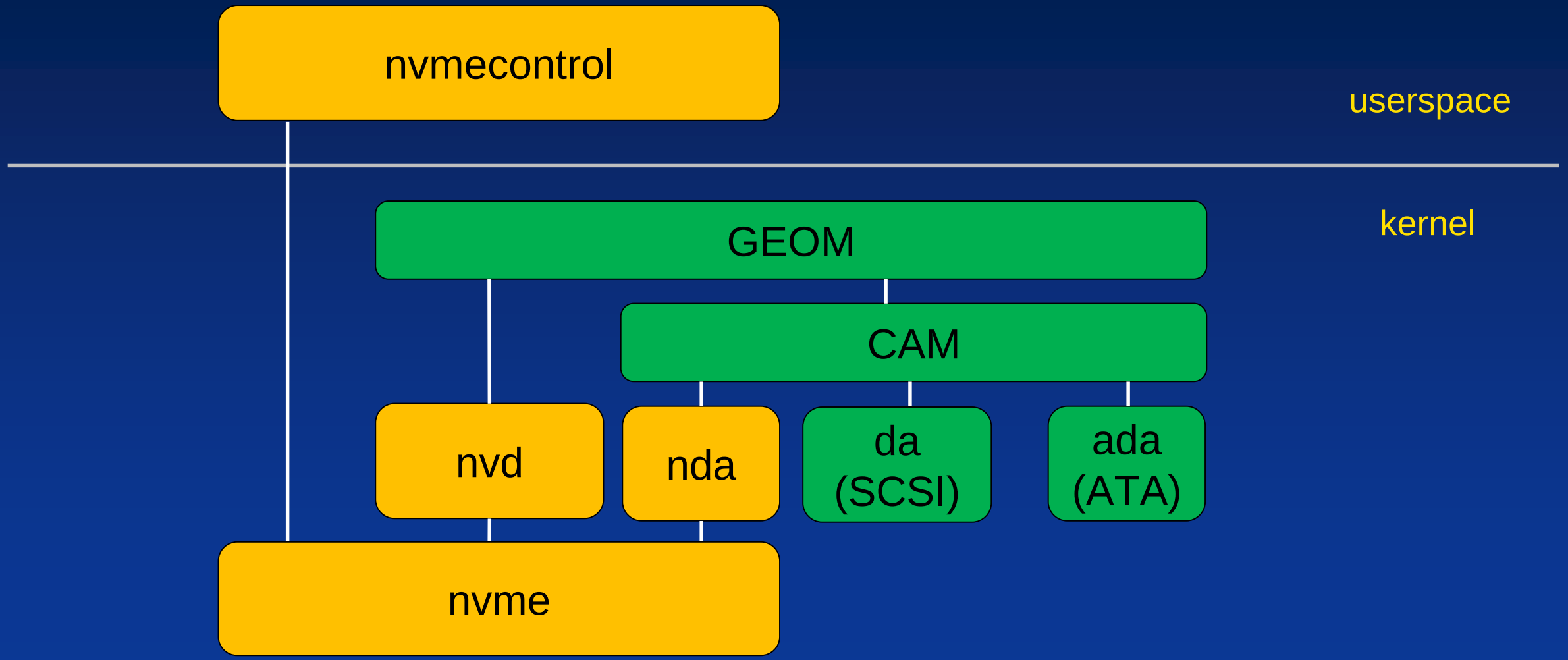


System Configuration

- 2x Intel® Xeon® E5-2695v4 (HT off)
  - Intel® Speed Step enabled
  - Intel® Turbo Boost Technology enabled
- 8x 8GB DDR4 2133 MT/s
  - 1 DIMM per channel
- CentOS Linux 7.2
- Linux kernel 4.7.0-rc1
- 8x Intel® P3700 NVMe®* SSD (800GB)
  - 4x per CPU socket
  - FW 8DV10102
  - Queue Depth 32 per SSD

Results based on SPDK NVMe®/perf application, http://github.com/spdk/spdk.git commit ID 49e0054f18

FreeBSD NVMe®* Block Diagram



FreeBSD NVMe® Overview

- NVM Express®* support added in FreeBSD 9.2 (2012)
- NVMe®* 1.0e Specification compliant
- Primary contributors: Intel, Netflix*
- FreeBSD base system includes:
  - nvme module – core NVM Express®* driver
  - nvd module - NVM Express®* GEOM disk driver
  - nvmecontrol utility – NVMe®* command line management tool
  - nda module (upcoming 11.0 release) – NVMe®* CAM disk driver

Legal Notices and Disclaimers

- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.
- No computer system can be absolutely secure.
- Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit **http://www.intel.com/performance**.
- Cost reduction scenarios described are intended as examples of how a given Intel- based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.
- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
- Intel, the Intel logo and others are trademarks of Intel Corporation in the U.S. and/or other countries.
- *Other names and brands may be claimed as the property of others.
- © 2016 Intel Corporation.

Session Organization

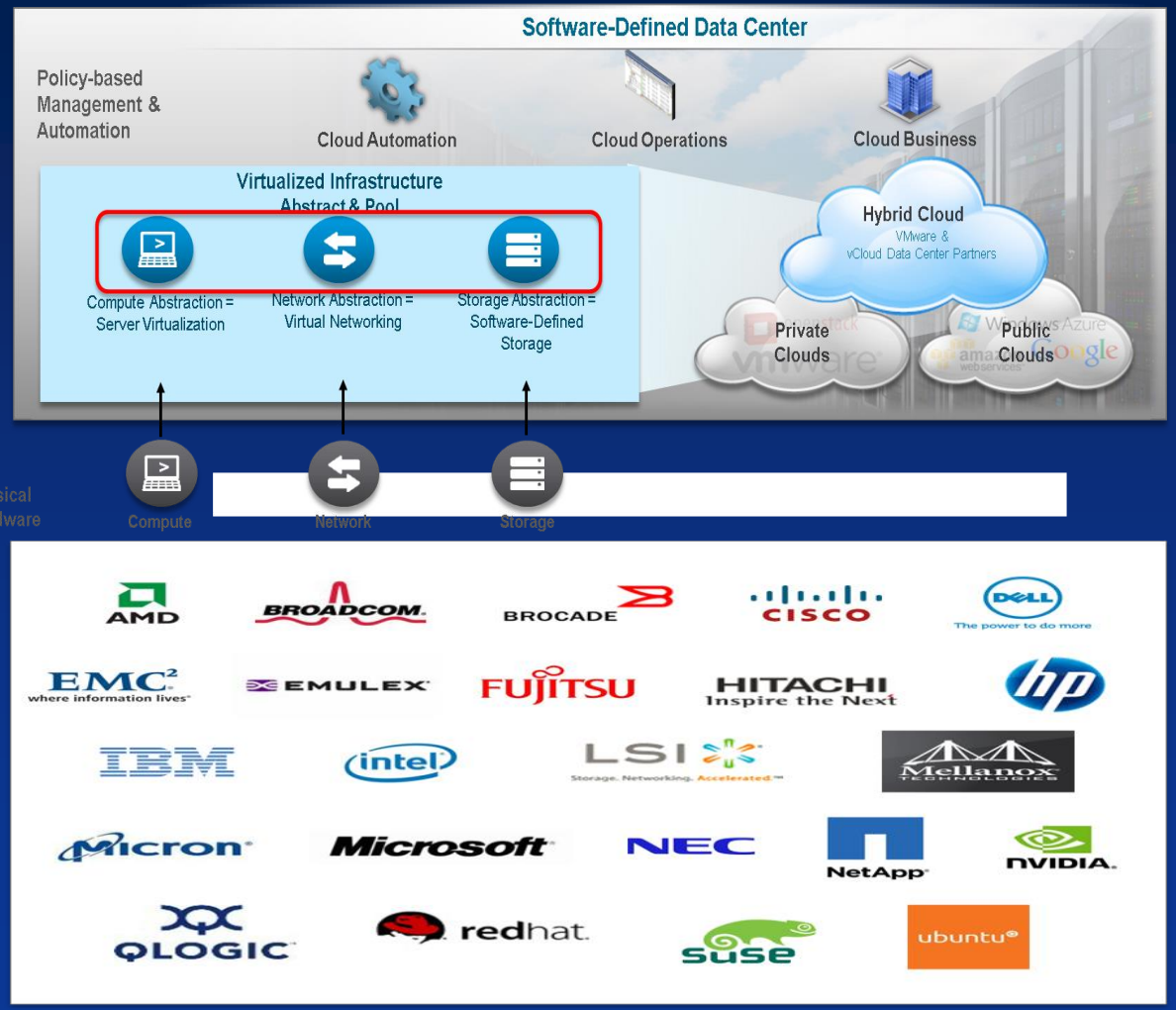
- Larry Chisvin, Broadcom (Session Organizer)
- Uma Parepalli, SK hynix memory solutions (Session Chair)
  - NVMe® Driver Ecosystem
  - UEFI NVMe® Drivers
  - Windows NVMe® Reference Driver (time permits)
- Lee Prewitt, Microsoft
  - MS Windows NVMe® Inbox Drivers
- Parag Maharana, Seagate
  - Linux NVMe® Fabrics Drivers
- Jim Harris, Intel
  - FreeBSD NVMe® Driver
  - NVMe® Storage Performance Development Kit (SPDK)
- Sudhanshu (Suds) Jain, VMware
  - VMware NVMe® Driver
- Q&A

VMware NVMe® Driver

Host Driver and Virtual Device

Sudhanshu (Suds) Jain

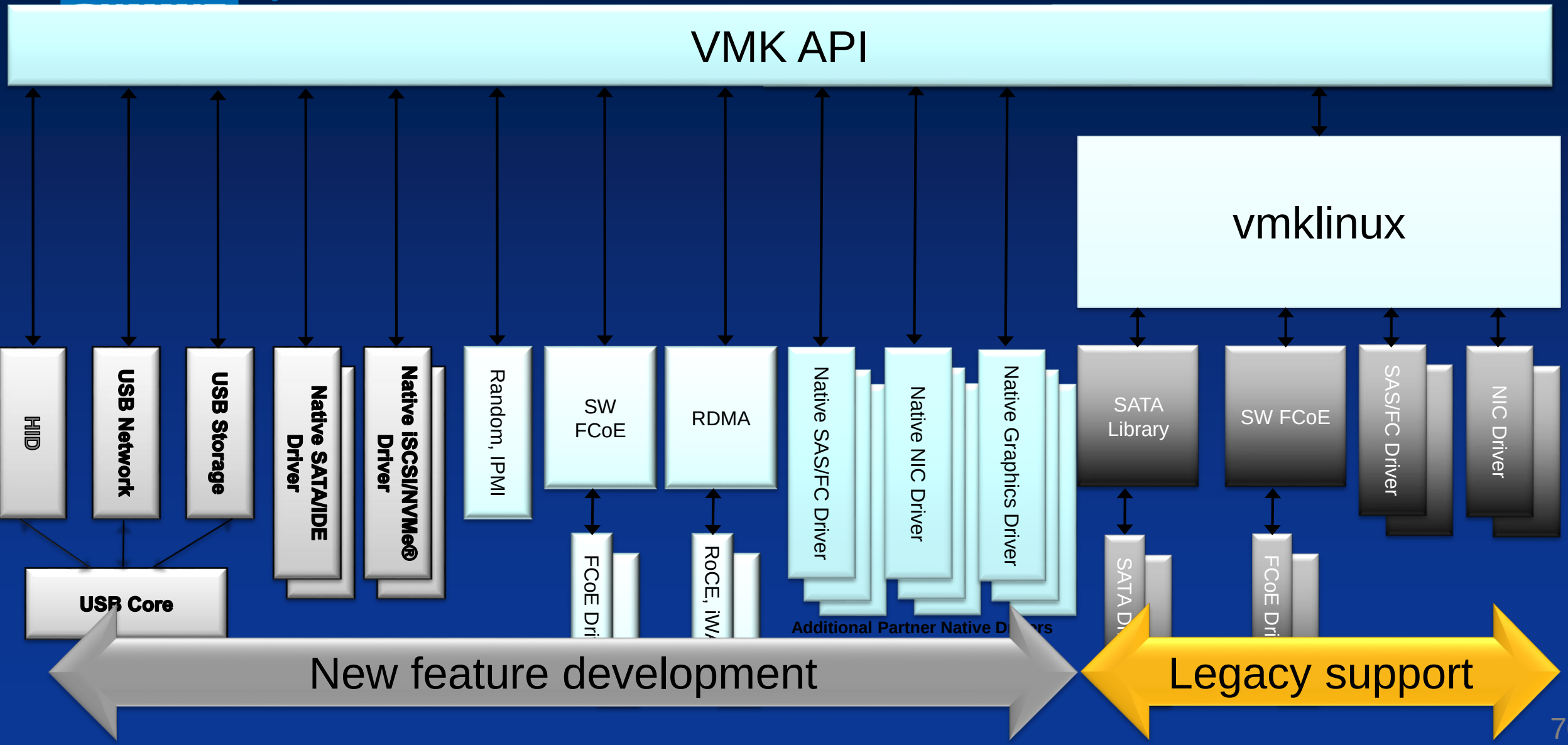
Software-Defined Infrastructure



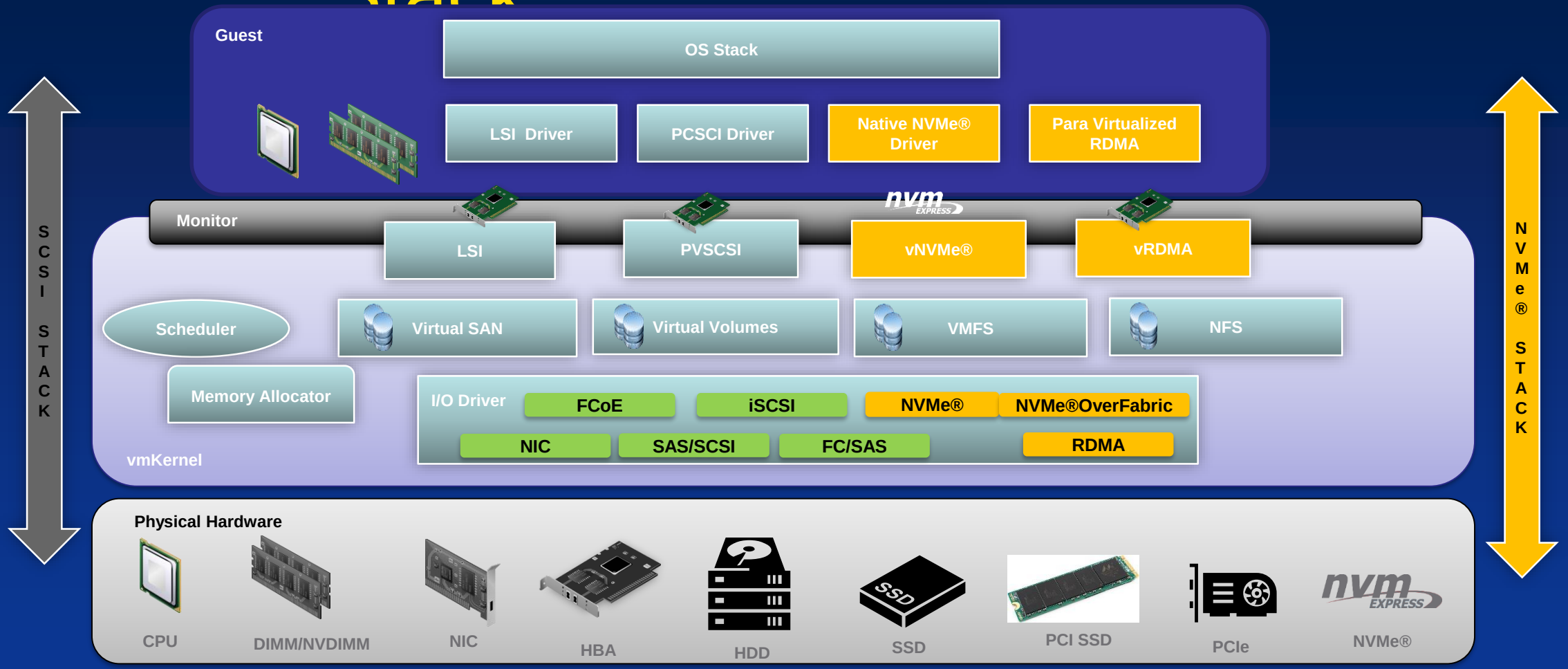
Flash and NVMe®

- A major focus area (moving forward)
- vSphere Flash Use-Cases:
  - Host swap cache
  - Regular Datastore
  - vSphere Flash Read Cache (aka Virtual Flash)
  - VSphere ESXi Boot Disk
  - VSphere ESXi Coredump device
  - VSphere ESXi Logging device
  - Virtual SAN (VSAN)

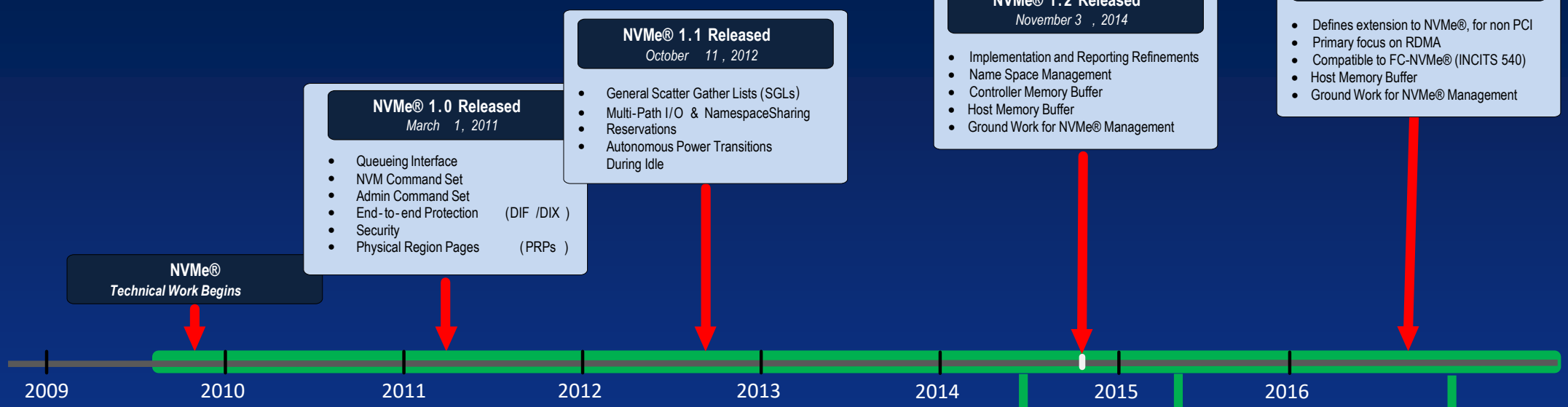
vSphere Driver Architecture Evolution



vSphere NVMe® Native Driver Stack



NVM Express® Evolution & vSphere



vSphere 5.5

- Introduce first async NVMe® driver 1.0e
- Launch IOVP cert program for NVMe®

vSphere 6.0

- Introduce first inbox NVMe® driver
- Bring broader ecosystem support

vSphere (Future)

- Multiple name spaces, Queues
- NVMe® Over Fabric
- vNVMe®
- End-to-end NVMe® Stack

Where to get more information?

- vSphere 5.5: Download VMware ESXi 5.5 Driver CD for NVM Express® (NVMe®) driver.
- vSphere 6.0: available as part of base image.
  - Also available for download VMware ESXi 5.5 nvme 1.2.0.27-4vmw NVMe® Driver for PCI Express based Solid-State Drives
- NVMe® Ecosystem:
  https://www.vmware.com/resources/compatibility/search.php?deviceCategory=io
- vSphere NVMe® Open Source Driver to encourage ecosystem to innovate
  - https://github.com/vmware/nvme

Architected for Performance

Thank You!
Q&A