STORAGE DEVELOPER CONFERENCE
SDC 22
Fremont, CA
September 12-15, 2022
BY Developers FOR Developers

A SNIA Event

NVMe-oF™ Boot

Specification and Reference Implementation

Doug Farley (Dell) & Rob Davis (NVIDIA)

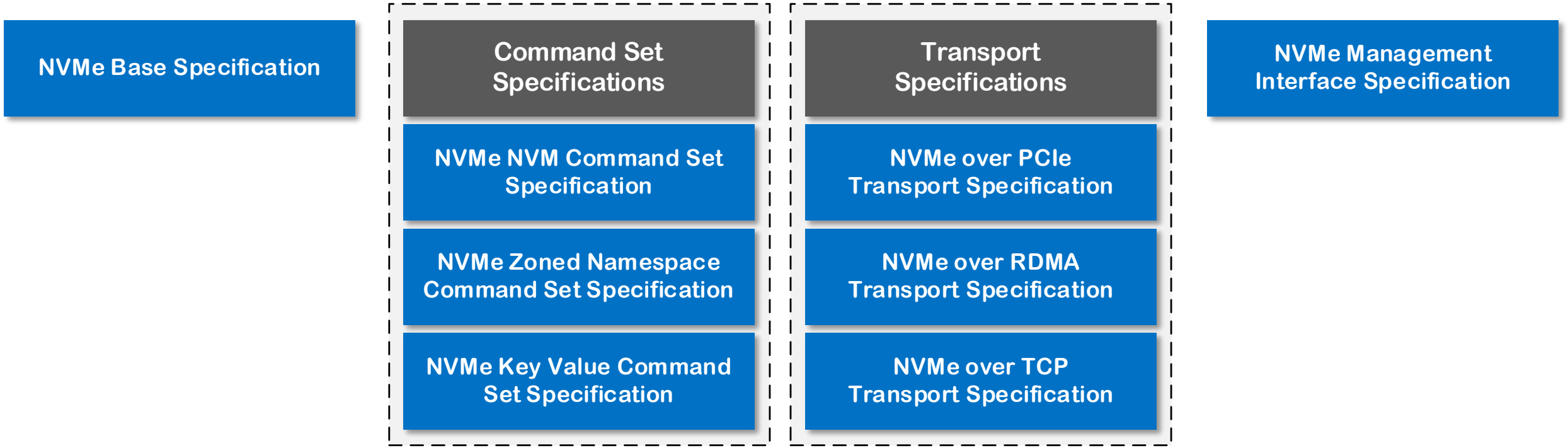
NVM Express, Inc. Overview

- NVM Express is **110+ members** strong and was created to expose the benefits of non-volatile memory in all types of computing environments
- NVMe technology delivers high bandwidth, low latency storage and overcomes bottlenecks
- NVM Express technology includes the below specifications:
  - **NVM Express® (NVMe®) Base Specification**
  - **NVM Express Command Set Specifications**
  - **NVM Express Transport Specifications**
  - **NVMe Management Interface (NVMe-MI™)**
- Markets enhanced by NVM Express technology include:
  - Artificial Intelligence
  - Virtual Reality
  - Machine Learning
  - Cloud/Data Center
  - SSD Controllers
  - Storage
  - PC/Mobile/IoT
  - Healthcare

Promoter Group
2022 - 2023

DELL EMC, Google, Hewlett Packard Enterprise, intel, KIOXIA, MARVELL, Meta, Micron, Microsoft, NetApp, SAMSUNG, SEAGATE, Western Digital

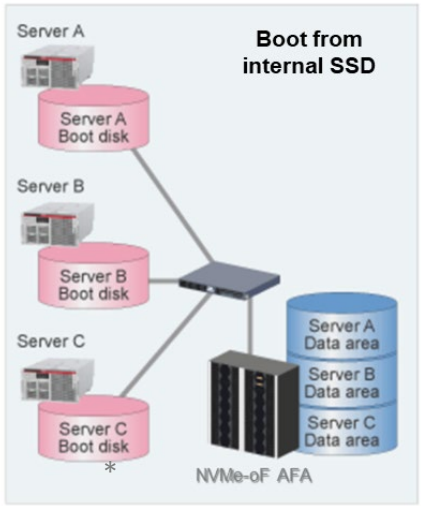
NVMe® 2.0 Family of Specifications

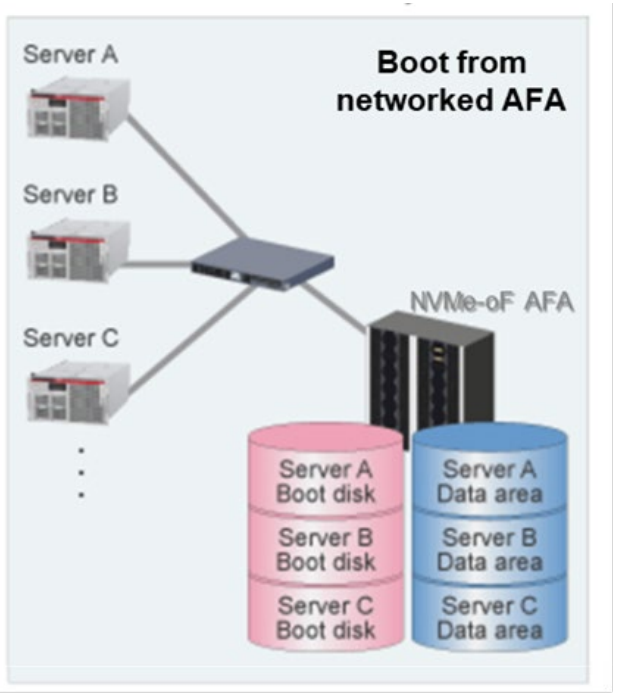


NVMe 2.0 specifications were released on June 3, 2021
Refer to nvmexpress.org/developers

Why Does NVMe® Technology Need a Boot Spec

- Currently successful storage networking technologies such as Fibre Channel and iSCSI have standardized solutions that allow attached computer systems to boot from OS images stored on attached storage nodes
- The lack of this capability in NVMe-oF™ architecture presents a barrier for adoption
- This is a missing requirement for a networked storage technology





*AFA = All Flash Array Storage System

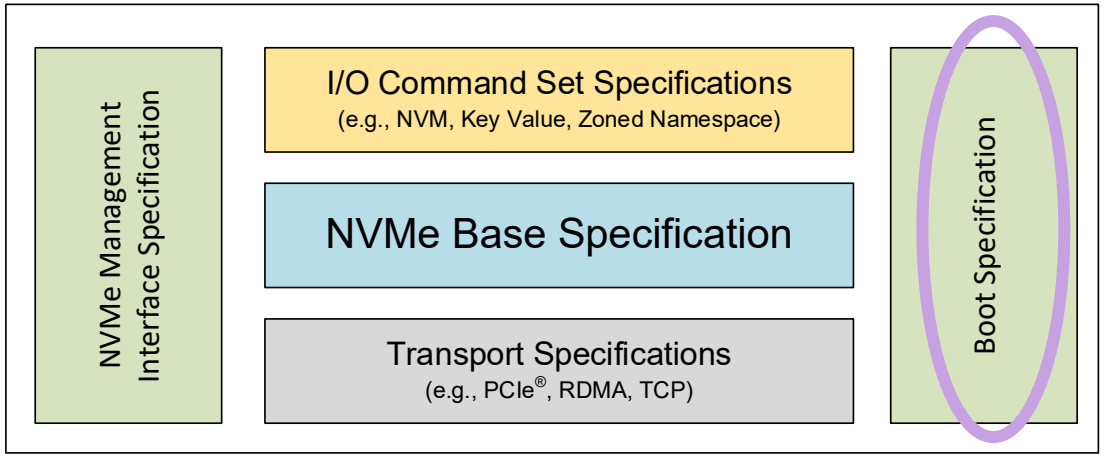
NVMe® Boot Goal: Standardize Booting from NVMe and NVMe-oF™ Namespaces

- The proposed standardization and ecosystem enablement covers
  - NVMe/PCIe® Boot– Already widespread precedent for booting from the PCIe Transport
  - NVMe over Fabrics Boot - Evolutionary step from previous remote disk boot technologies
    - fabric agnostic booting capability supporting all NVMe-oF transports
    - lower latency due to elimination of SCSI command translation via direct NVMe command execution
    - increased scalability and flexibility through the NVMe-oF specification defined discovery mechanism

NVMe® Boot Goal: Standardize Booting from NVMe and NVMe-oF™ Namespaces

Defines constructs and guidelines for booting from NVM Express® interfaces over supported transports

- Initial version defines extensions to the NVMe interface for booting over NVMe/TCP transport
- Normative content describes
  - generic requirements of information sharing
  - domain specific mechanics (e.g., ACPI tables to enable Pre-OS and OS drivers to share configuration context)
- Informative content Introduces
  - Pre-boot configuration mechanisms and best practices
  - Mechanics to provide reliable OS consumption of pre-boot configuration (e.g., ACPI NVMe Boot Firmware Table)
  - OS-specific best practices
  - Fabric transport specifics (e.g., transport address)

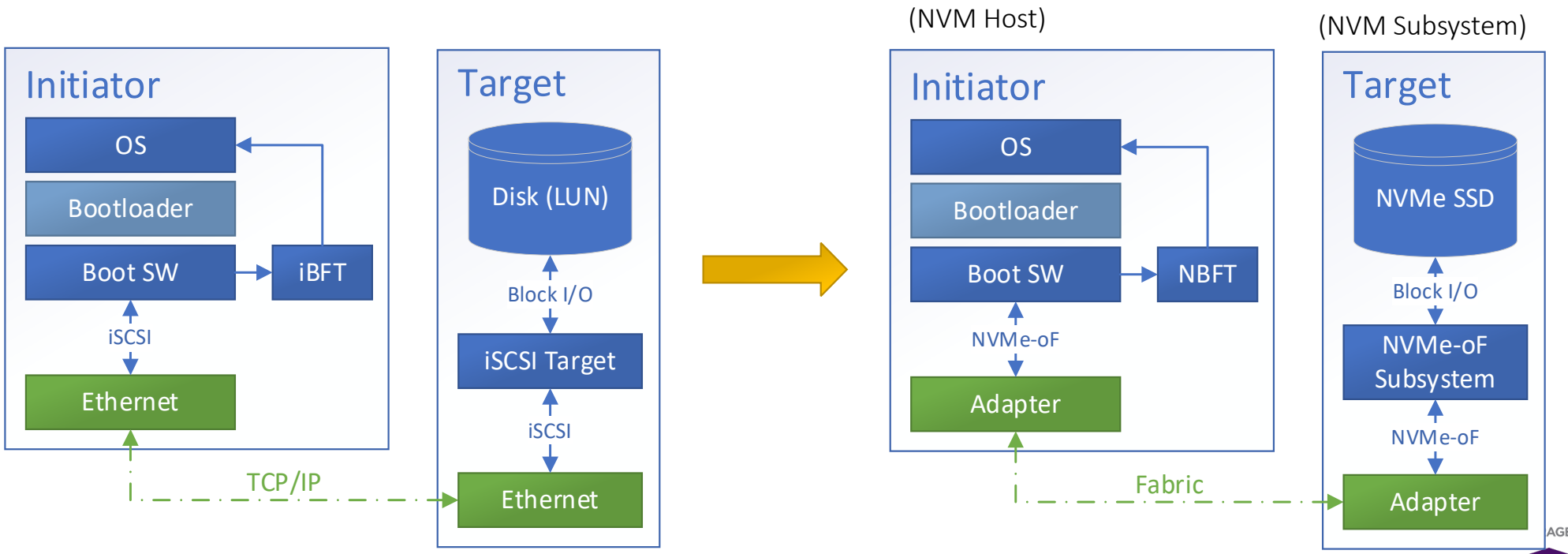


Ecosystem Artifacts to Enable Standardization

1. **ACPI Spec:** Added ACPI NVMe® Boot Firmware Table (NBFT) to ACPI.org
2. **UEFI System Spec:** Device path extension for NVMe-oF™ boot
3. **NVM Express® Boot Specification:** Contains Normative and Informative content including the ACPI NBFT XSDT definition and a wide range of ecosystem and interoperability guidance for implementing and productively using NVMe-oF boot functionality
4. **Public reference implementation:** Tianocore EDK2 + Linux PoC

NVMe® Boot Goal: Standardize Booting from NVMe and NVMe-oF™ Namespaces

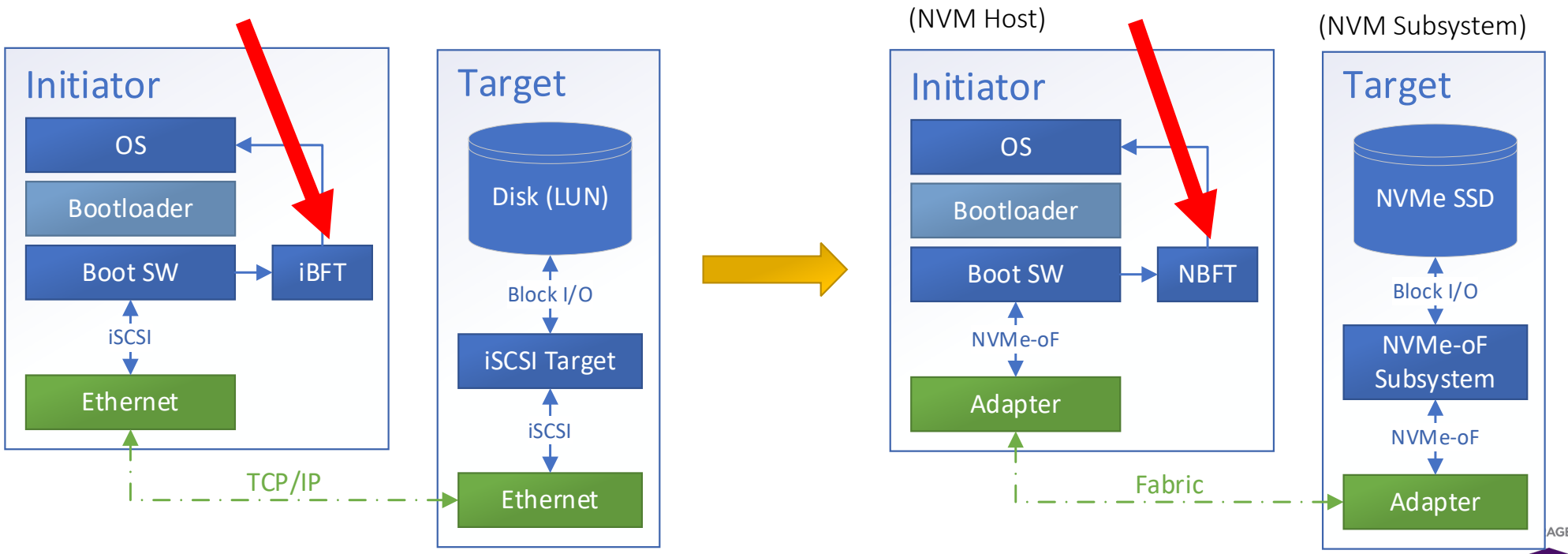
- Reference implementation plus specifications ease ecosystem adoption
- NVMe/TCP boot enabled standardization to leverage past iSCSI lessons and ecosystem enablement
- Boot from NVMe/TCP technology main concepts (boot flow and handover mechanism) are similar to booting from iSCSI



Delta between booting from iSCSI and booting from NVMe-oF

NVMe® Boot Goal: Standardize Booting from NVMe and NVMe-oF™ Namespaces

- Reference implementation plus specifications ease ecosystem adoption
- NVMe/TCP boot enabled standardization to leverage past iSCSI lessons and ecosystem enablement
- Boot from NVMe/TCP technology main concepts (boot flow and handover mechanism) are similar to booting from iSCSI
- iSCSI used the iBFT to share Pre-OS context to the OS
- NVMe technology needs a similar configuration mechanism, NBFT (NVMe Boot Firmware Table)



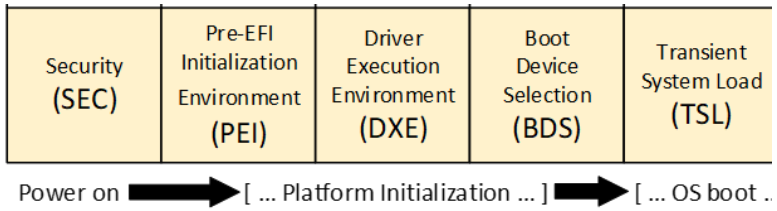
Delta between booting from iSCSI and booting from NVMe-oF

Configuring NVMe-oF™ Boot (UEFI-based example): Pre-Operating System Boot

New functionality

Driver Execution Environment phase: DXE driver supporting NVMe-oF boot is loaded and executed:

- Namespaces used as boot devices are discovered using the information stored during configuration
- Driver connects to the NVMe-oF subsystem, discovers the namespace (boot device) for OS boot



Existing functionality

Boot Device Selection phase: The Namespace can then be selected as final boot device for OS boot

Transient System Load phase:

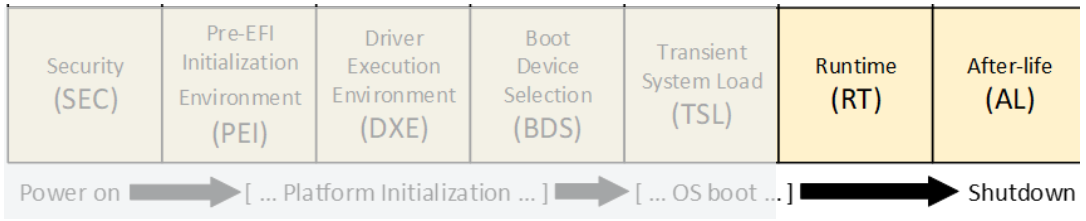
- OS image loaded from boot device
- UEFI hands over execution to OS specific boot loader
- OS Boot Loader continues the OS boot

At this point, the NBFT has been generated, stored in main memory, and can be accessed by the OS as an ACPI table

Configuring NVMe-oF™ Boot (UEFI-based example): Pre-Operating System Boot

Operating system installation:

- A user may either:
  a) use the NBFT provided host NQN as its own host NQN
  b) set a separate host NQN (if NVMe-oF subsystem supports multiple host NQNs)



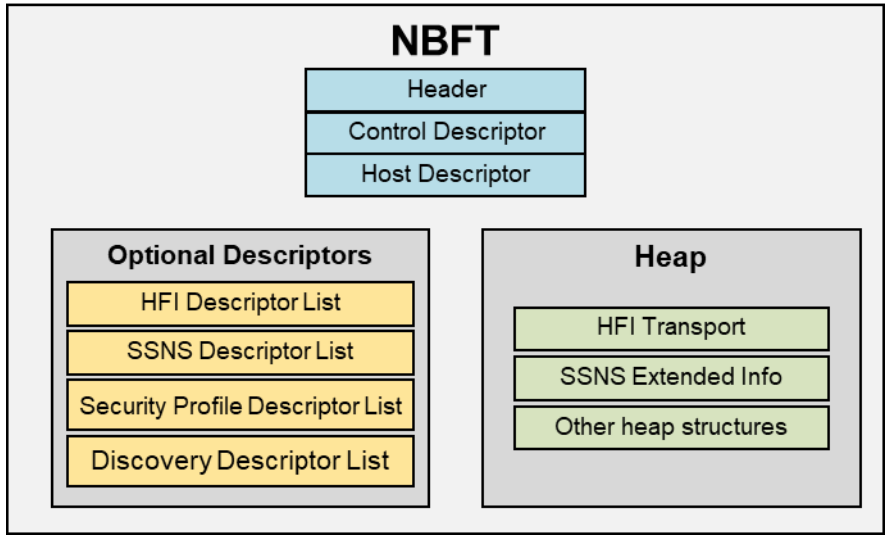
Normal operating system boot:

- To persist info to restore NVMe-oF technology connections, OS may either:
  - continue using the NBFT
  - use OS specific mechanism

NBFT: Pre-OS to OS Configuration Handoff

Administrator configures Pre-OS driver:

- target subsystem NQN
- target IP address
- target port #
- target namespace
- host NQN
- security related info



Information presented to the OS using an ACPI XSDT Table at OS boot:

- provides a standardized means of passing configuration and connection context from a pre-OS Boot environment to an administratively configured OS runtime

Element	Description
Header	An ACPI structure header with some additional NBFT specific info.
Control Descriptor	Indicates the location of host, HFI, SSNS, security, and discovery descriptors.
Host Descriptor	Host information.
HFI Descriptor	An indexable table of HFI Descriptors, one for each fabric interface on the host.
Subsystem Namespace Descriptor	An indexable table of SSNS Descriptors.
Security Descriptor	An indexable table of Security descriptors.
Discovery Descriptor	An indexable table of Discovery Ddescriptors.
HFI Transport Descriptor	Indicated by an HFI Descriptor, corresponds to a specific transport for a single HFI.
SSNS Extended Info Descriptor	Indicated by an SSNS Descriptor if needed.

Reference Implementations

Pre-OS time of boot:

- EDK2 NVMe-oF™ UEFI Driver for the NVMe®/TCP transport
  - ACPI NBFT will be produced by this UEFI implementation prior to OS boot

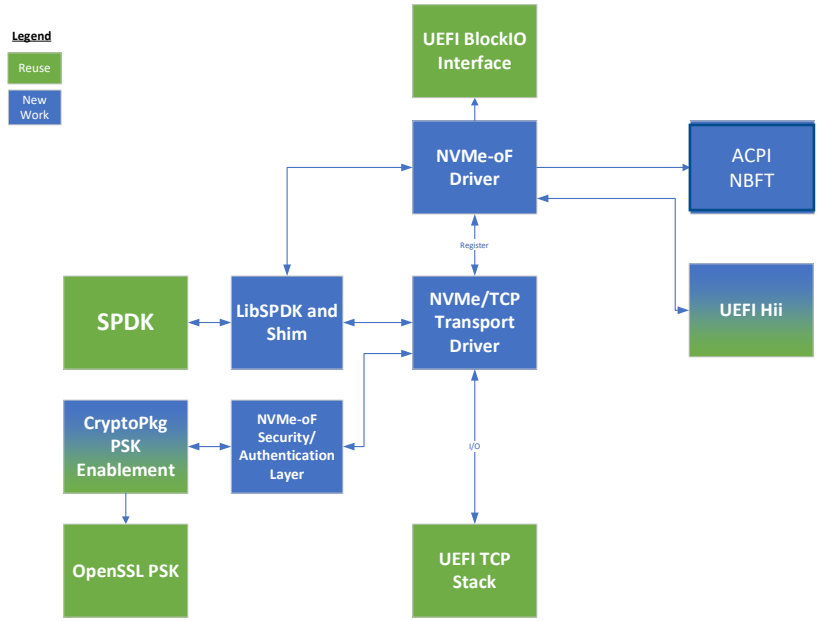
OS Boot and Runtime:

- Linux® reference implementation that:
  - Exposes the NBFT to the user-space
  - Consumes the NBFT contents to connect to configured namespaces
- Enables common tools (e.g., dracut, nvme-cli) to use the NBFT

UEFI Pre-OS Driver: Initial Concept to Reference Implementation

To be submitted for up-stream staging review after NVMe Boot Specification published

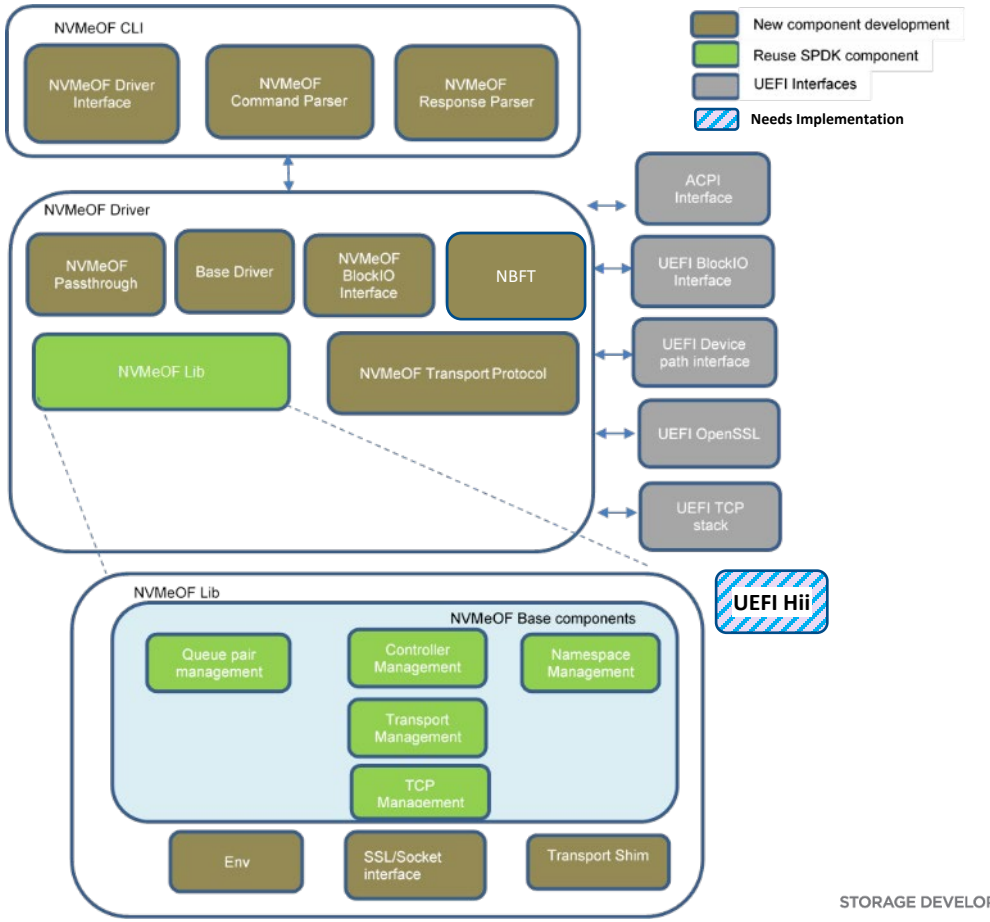
EDK2 Concept Architecture



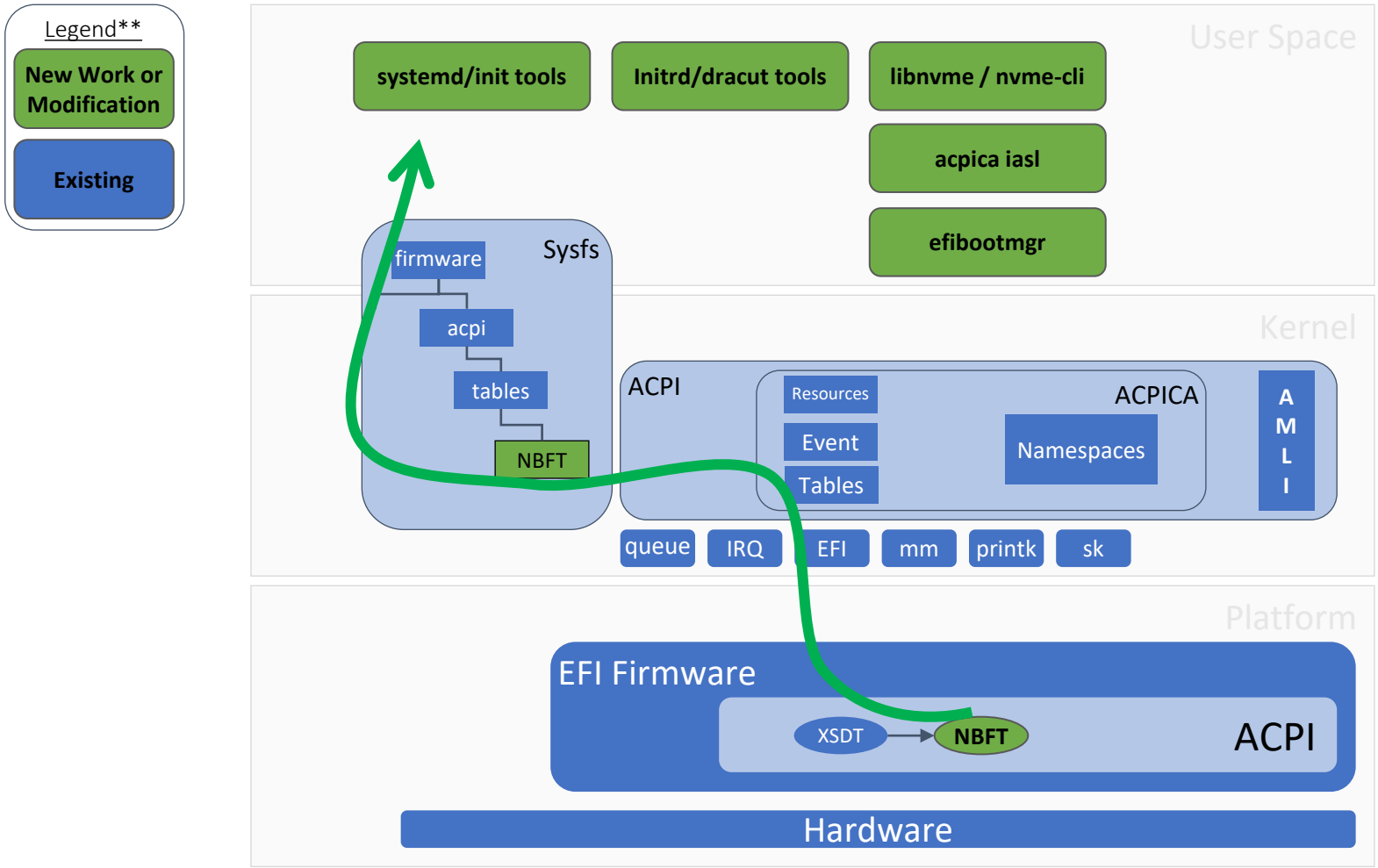
Development

EDK2 Reference Architecture as implemented

(Not yet published upstream for review)



Linux® Current Reference Implementation for NVMe® /TCP Boot Enablement*



* Not yet accepted into respective projects
** Diagram concept adapted from Joey Lee, SUSE

Ecosystem Cooperation and Standardization

- NVMe-oF™ boot required ecosystem cooperation and standardization
- Collaboration with the following ecosystem and industry partners was key
  - NVM Express, Inc.
  - UEFI
  - ACPI
  - TianoCore
  - DMTF/Redfish
  - Linux® Kernel
  - Linux Common Userspace

Work Products

Specification Material

- ACPI Spec ECR into 6.5*
- UEFI System Spec ECR into 2.10*
- Draft NVM Express® Boot Specification**

Reference Implementation**

- Linux® Kernel Patch for NBFT
- Feature PRs for:
  - edk2
  - spdk
  - dracut
  - libnvme, nvme-cli

* See references slide for publication locations
** Not yet publicly released

Call to Action

- UEFI HII for EDK2 driver
- Support for additional transports
- OEM BIOS support
- Additional OS and installer support

References and Repositories

- **NVM Express®:** https://nvmexpress.org/developers/nvme-specification/
- **UEFI 2.10:** https://uefi.org/specs/UEFI/2.10/10_Protocols_Device_Path_Protocol.html
- **ACPI 6.5:** https://uefi.org/specs/ACPI/6.5/05_ACPI_Software_Programming_Model.html
- **Open-Source Software Repos:** https://github.com/timberland-sig

Thank You

Please take a moment to rate this session.

Your feedback is important to us.

Backup

Terminology and Acronyms, Further Reading

- DC – NVM Discovery Controller, hosts NVM Subsystem Indicies
- DLP – NVMe-oF Discovery Log Page
- ESP – EFI System Partition; i.e. (/boot/efi)
- Namespace - NVMe flavor of SCSI LUN, block device representation
- NID – Namespace Identification Descriptor; a globally unique ID (def. NVMe CNS 03h)
- PDU – Protocol Data Unit
- SQE – Submission Queue Entry
- SGL – Scatter Gather List

