STORAGE DEVELOPER CONFERENCE

SDC22

Fremont, CA
September 12-15, 2022

BY Developers FOR Developers

A SNIA Event

SPDK and Infrastructure Offload

Jim Harris
Principal Software Engineer
Intel

Notices and Disclaimers

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing on certain dates using certain configurations and may not reflect all publicly available updates. Reach out to Intel for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

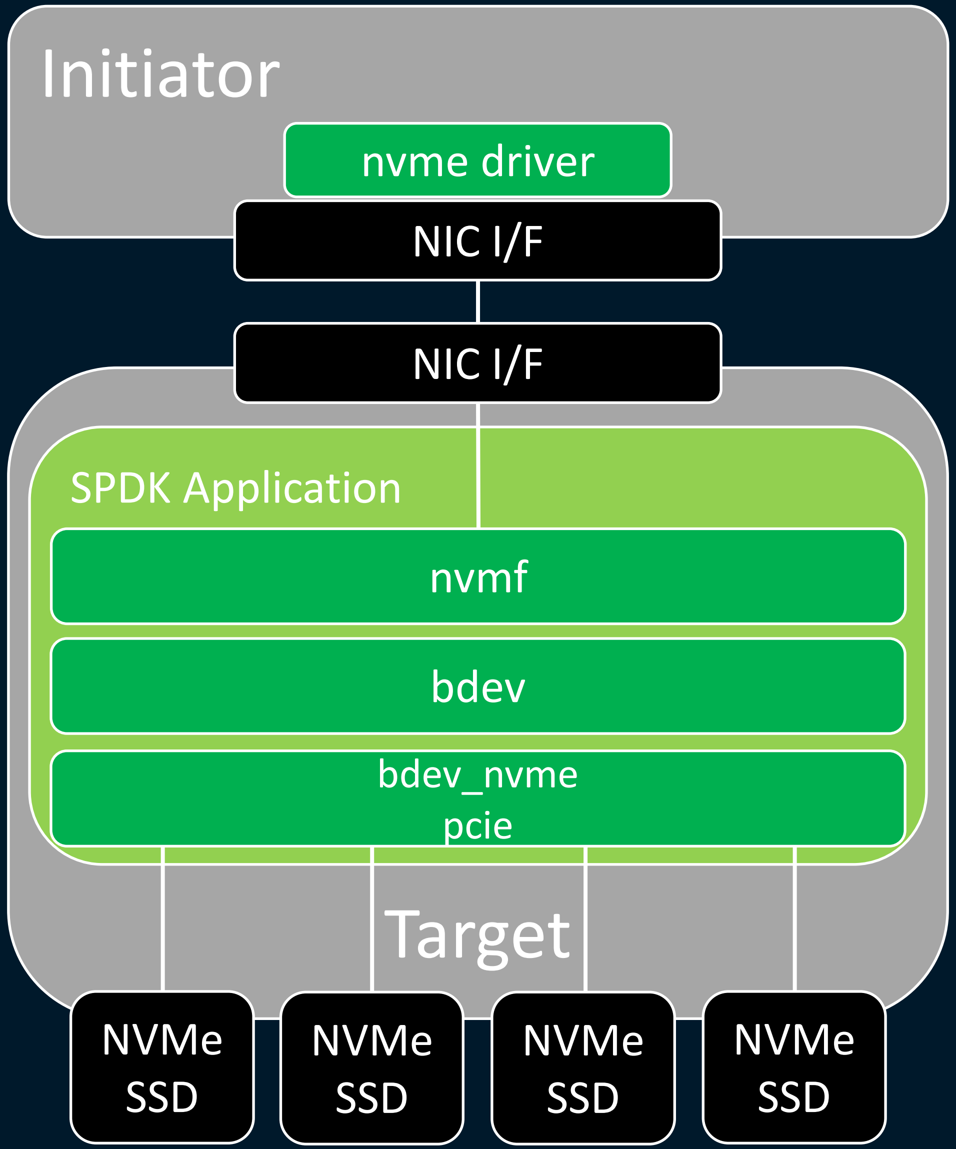
Agenda

- SPDK NVMe Target and Infrastructure Offload
  - SPDK NVMe Target History and Background
  - How does this apply to Infrastructure Processing Units (IPUs)?
  - NVMe Target Transport Abstraction
  - Extensions for non-fabrics use cases
- Bonus: Storage Management Agent

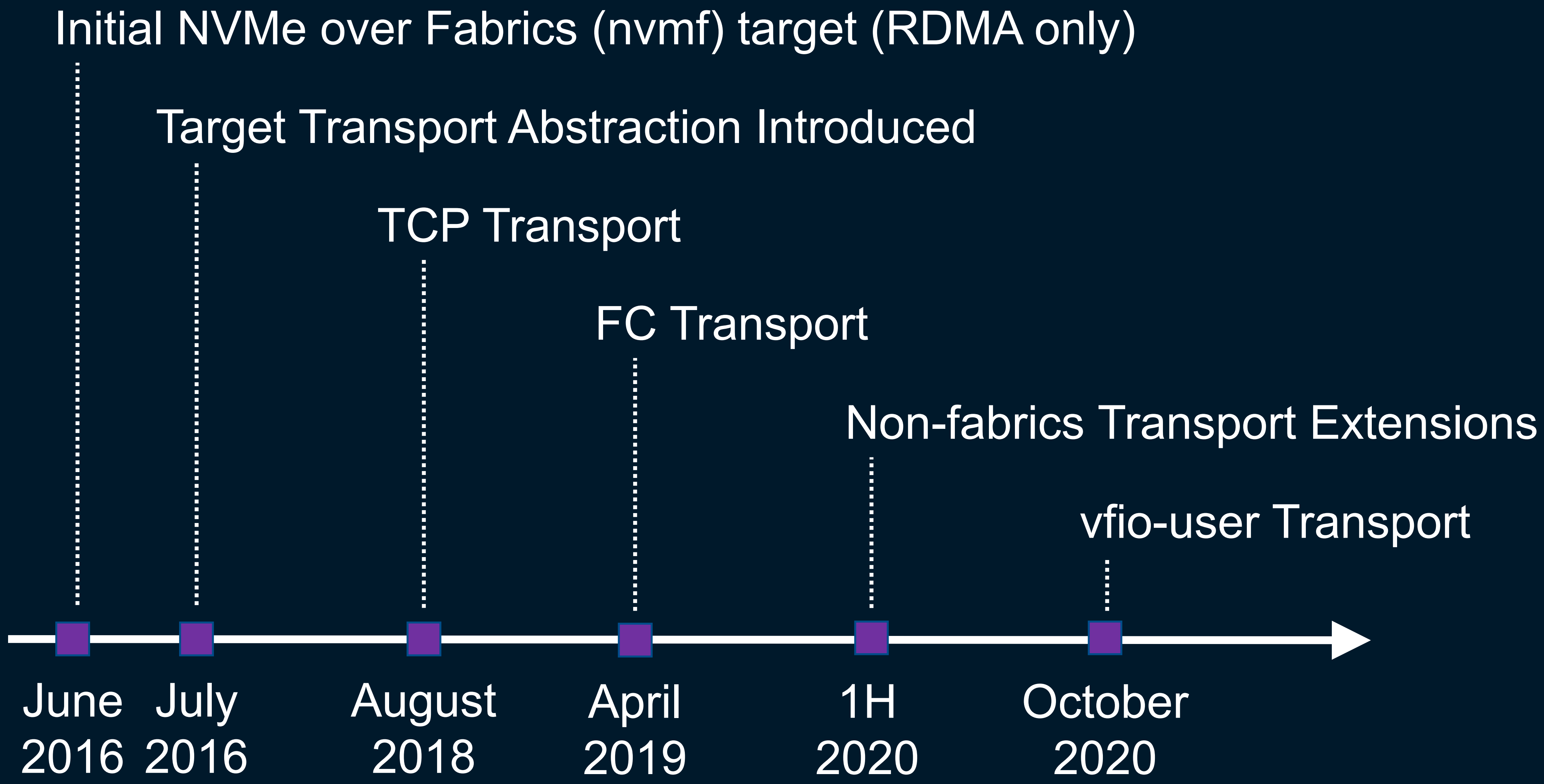
SPDK NVMe Target and Infrastructure Offload

SPDK NVMe Target Primer

- Accepts NVMe commands over network fabric
  - RDMA, TCP, FC
- Forwards commands to SPDK block device (bdev) layer
- SPDK block devices backed by storage
  - Examples
    - Local NVMe SSD
    - Logical volume on NVMe SSD
    - Remote storage (NVMe, iSCSI, Ceph)



SPDK and NVMe Target Timeline



Initial NVMe over Fabrics (nvmf) target (RDMA only)

Target Transport Abstraction Introduced

TCP Transport

FC Transport

Non-fabrics Transport Extensions

vfio-user Transport

June 2016

July 2016

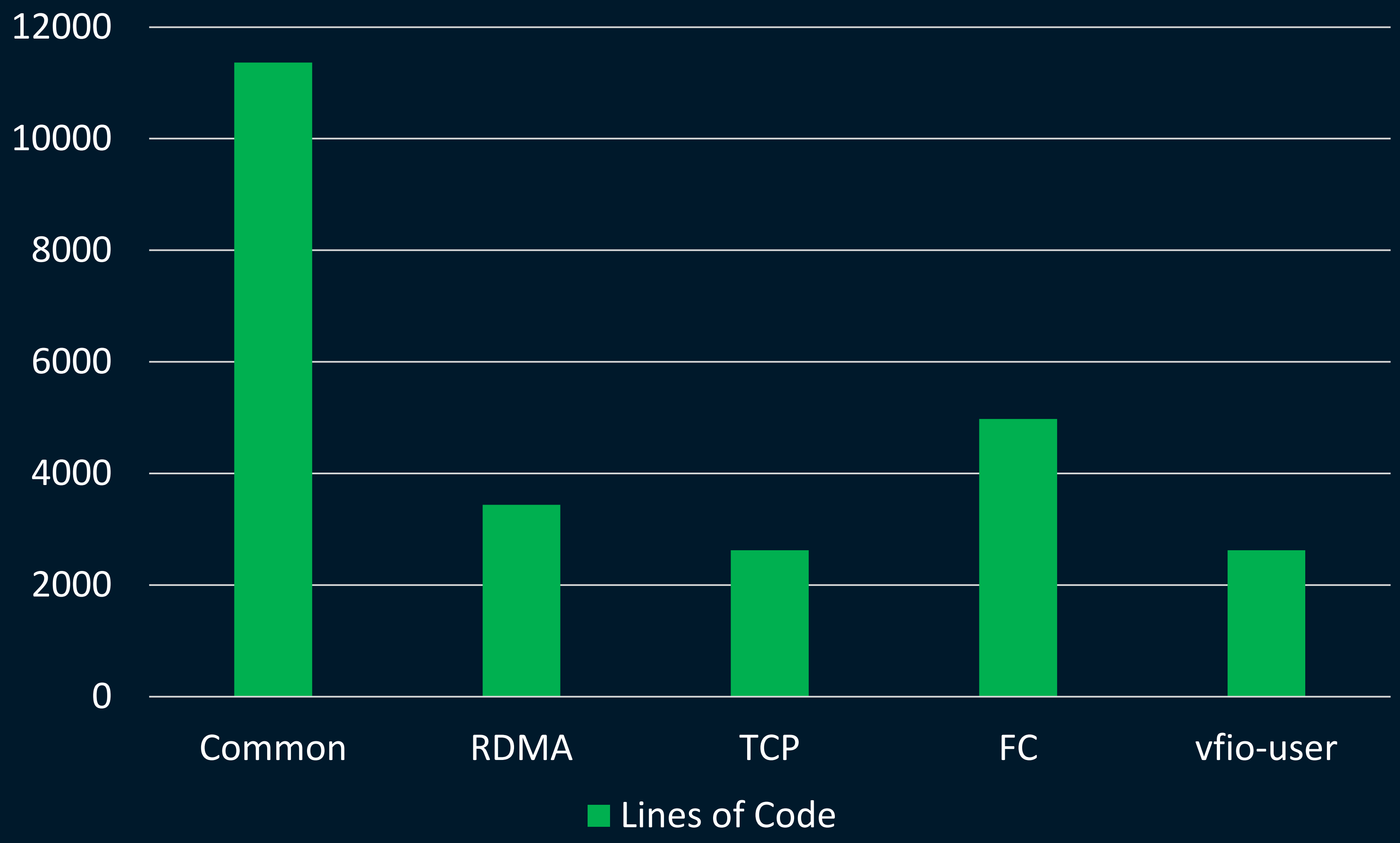
August 2018

April 2019

1H 2020

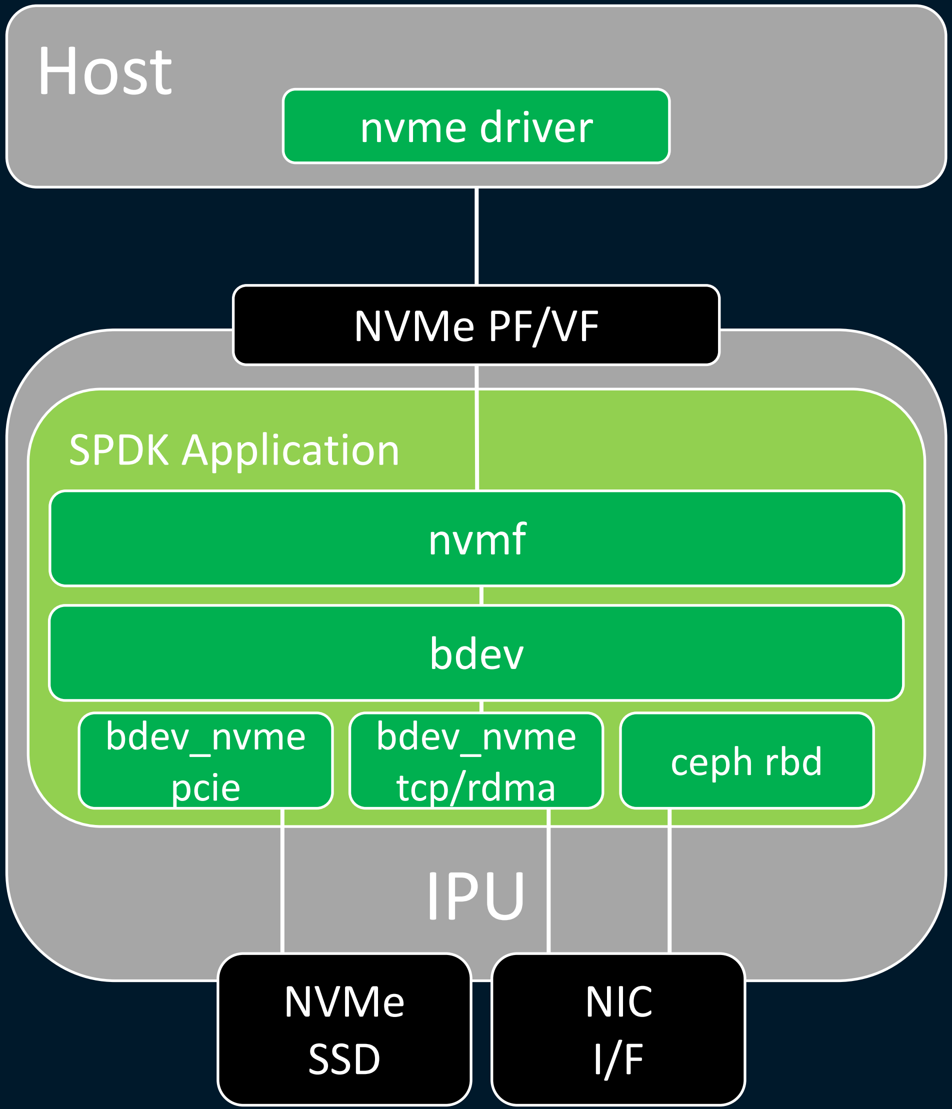
October 2020

Code Breakdown



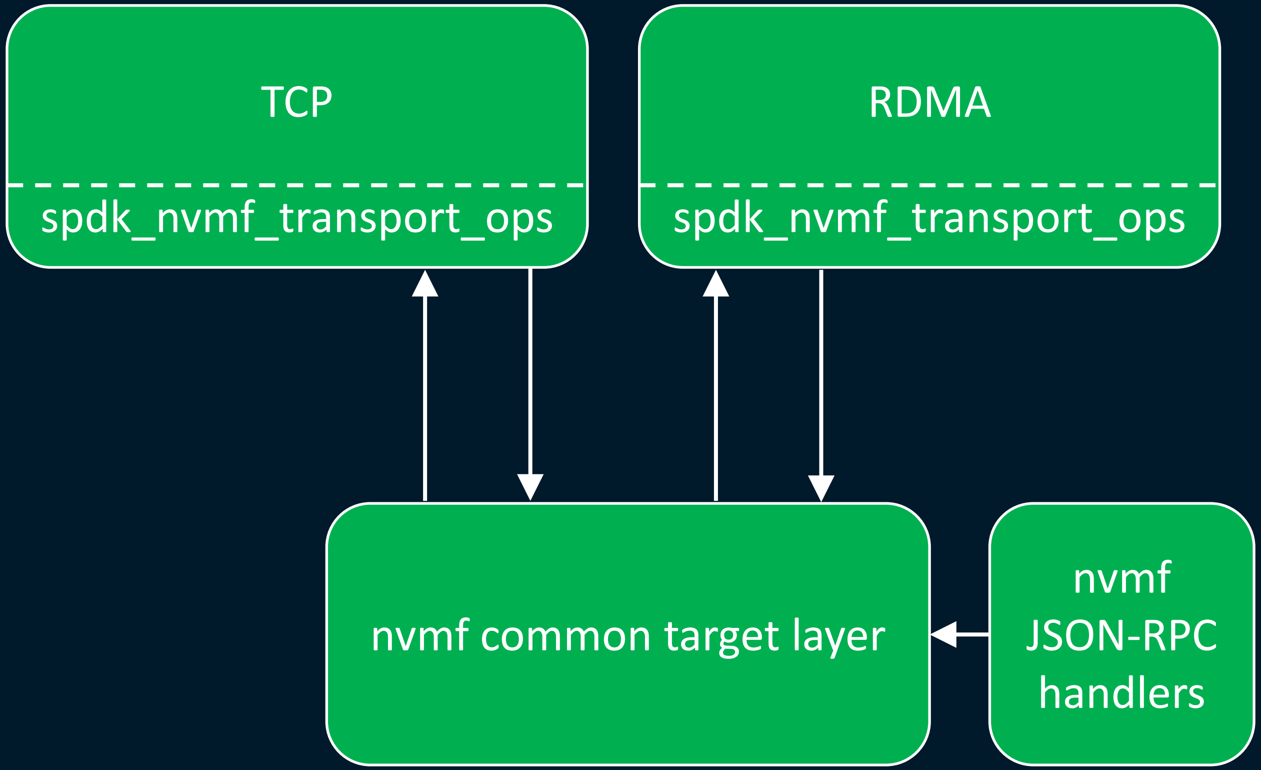
How does this apply to IPUs?

- IPUs with NVMe interface are an NVMe target!
  - But with an IPU-specific transport
- Goal: share all of the NVMe target common code for non-fabrics use cases



NVMe Target Transport Abstraction

- Transports implement spdk_nvmf_transport_ops
- Operations include
  - create, destroy
  - listen, stop_listen
  - poll_group_create, poll_group_add, poll_group_remove, poll_group_poll
- Common layer calls transport to perform transport-specific operations
- Transports notify common layer of new connections and new requests
  - spdk_nvmf_tgt_new_qpair
  - spdk_nvmf_request_exec



TCP

spdk_nvmf_transport_ops

RDMA

spdk_nvmf_transport_ops

nvmf common target layer

nvmf JSON-RPC handlers

Extensions for non-fabrics use cases

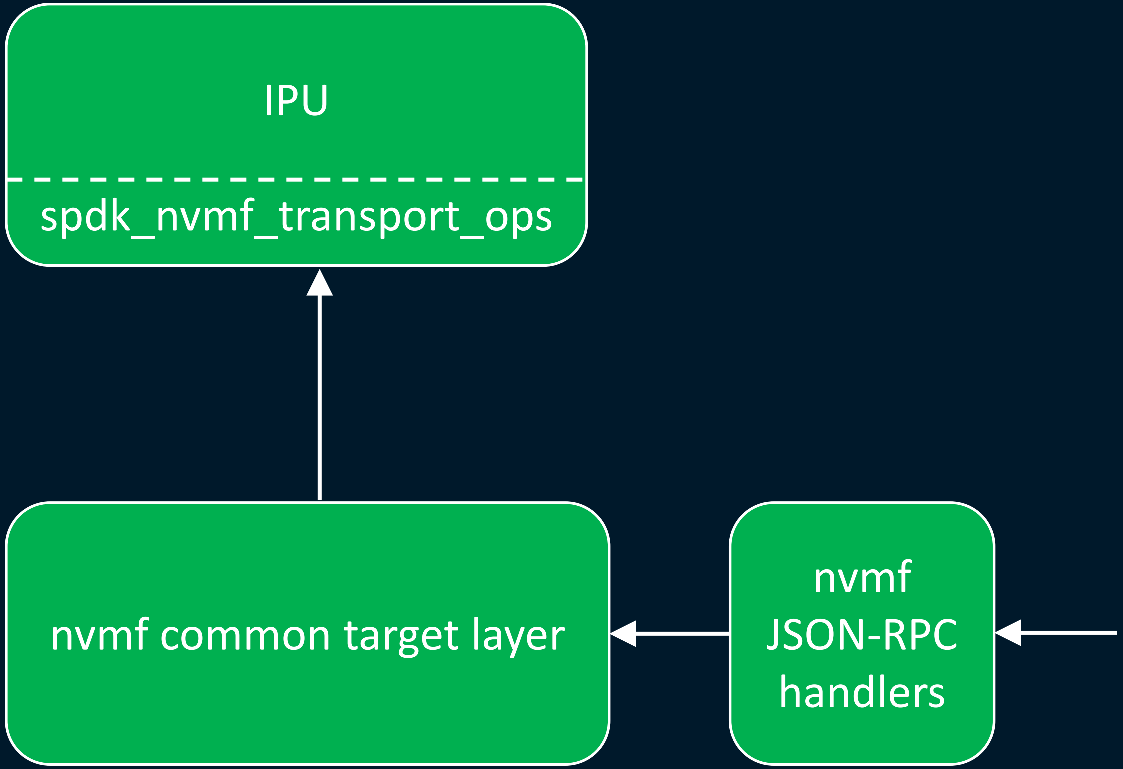
- What’s different about non-fabrics use cases?
  - Listen and connect
  - Namespace notifications
  - Register reads/writes v. Property get/set
  - Various IDENTIFY feature reporting (i.e. SGL support)

Listen and Connect

- Fabrics (i.e. TCP)
  - Listen = Listen for new TCP connections on given address/port
  - Accept = Accept new TCP connection and start processing NVMe capsules
  - Connect = NVMe Fabrics command that specifies which subsystem to associate with the connection
- IPU
  - No explicit Connect command from the host!

Listen and Connect

- nvmf_subsystem_add_listener RPC
- spdk_nvmf_tgt_listen_ext()
  - Calls ops::listen
  - Listen on IPU-specified “address”
- spdk_nvmf_subsystem_add_listener()
  - Calls ops::listen_associate
    - Pass listener and subsystem as parameters
- When host “connects”, IPU transport sends common layer a fake CONNECT command to associate the connection with specified subsystem



Namespace Notifications

- Fabrics
  - All namespace enumeration and notifications handled in-band
  - IDENTIFY NAMESPACE
  - ASYNCHRONOUS EVENT REQUEST
- Non-fabrics
  - IPU may require special handling when namespaces are added or removed
  - ops::subsystem_add_ns, ops::subsystem_remove_ns added to notify transport
    - These operations are optionally implemented by each transport

Register Reads/Writes

- Fabrics
  - NVMe uses “property” instead of “register”
  - Fabrics commands PROPERTY_GET, PROPERTY_SET
  - Host sends fabrics commands to get/set CC, CSTS, CAP, VS, etc. during init
- Non-fabrics
  - PCIe host doesn’t send Fabrics commands to PCIe controller
    - It read/writes registers in PCI BAR using load/store instructions
  - Non-fabrics transports sends common layer fake PROPERTY_GET/SET commands to simulate PCIe register accesses

IDENTIFY feature reporting

- Fabrics
  - Always reports SGLs are enabled
- Non-fabrics
  - Some IPUs may only support PRP
  - ops::cdata_init called during controller creation to allow transport to override fields in IDENTIFY CONTROLLER data structure
  - Enables common layer to handle IDENTIFY CONTROLLER requests same for all transports

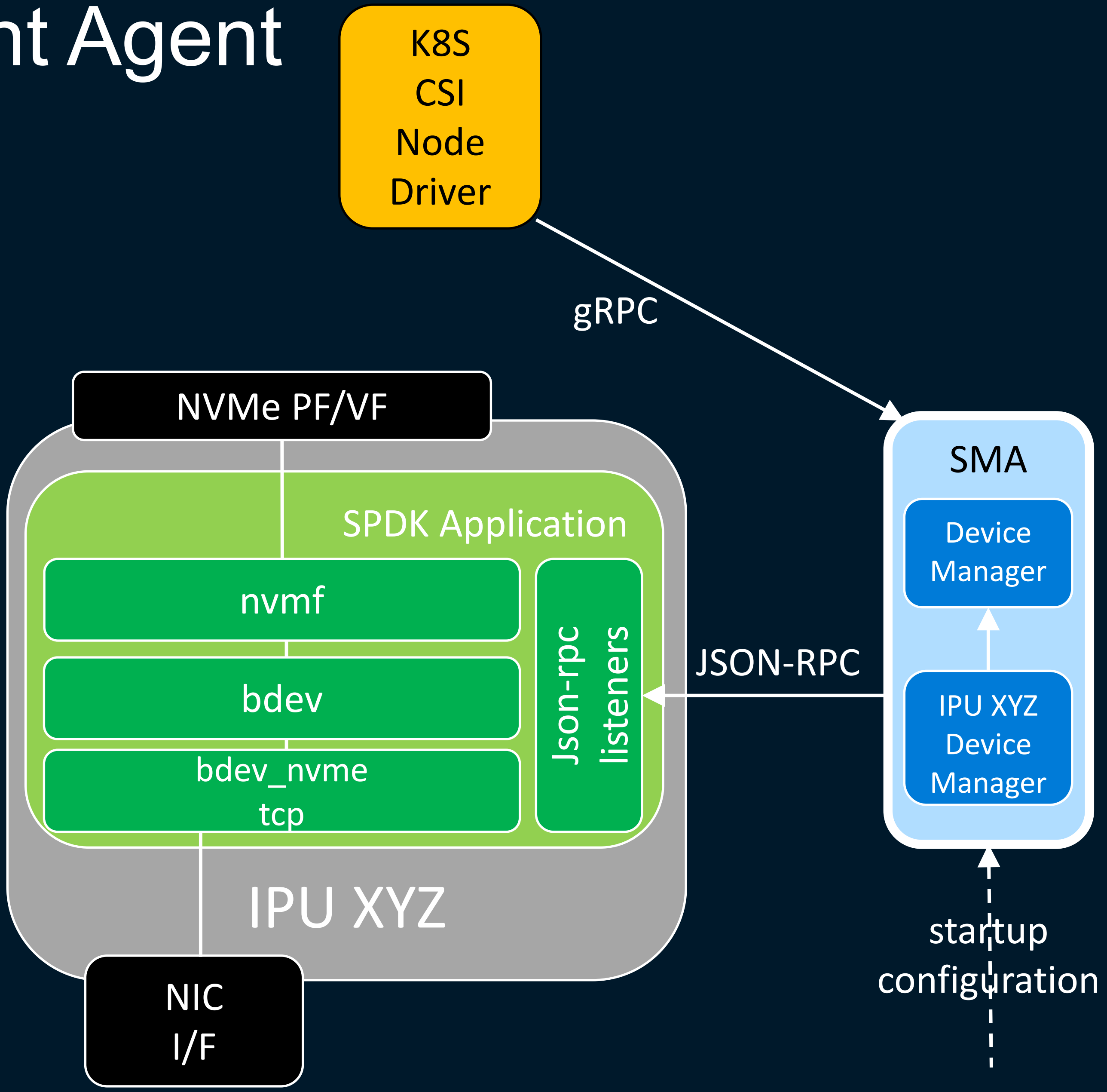
Storage Management Agent

Orchestrating SPDK-based IPUs

- SPDK JSON-RPCs are intentionally very low level
- Connecting an IPU host to its backend storage requires many steps (and error handling if any of those steps fail!)
  - bdev_nvme_attach_controller
  - nvmf_create_subsystem
  - nvmf_subsystem_add_ns
  - nvmf_subsystem_add_listener
- IPU-specific RPC parameters
- How do we make it easier to write something like a K8S CSI node driver?

SPDK Storage Management Agent

- gRPC based application
- Attach existing storage to host
  - Provisioning currently out of scope
- Basic operation set
  - CreateDevice
    - With optional volume
  - DeleteDevice
  - AttachVolume
  - DetachVolume
- IPUs implement DeviceManager interface



Current Status

- DeviceManagers:
  - nvmf_vfiouser, nvmf_tcp, vhost_blk
- Supports connecting to NVMe/TCP volumes
- In progress (targeted for SPDK v22.09 release)
  - Data encryption keys
  - Quality of service parameters

Thank you!

Please take a moment to rate this session.

Your feedback is important to us.