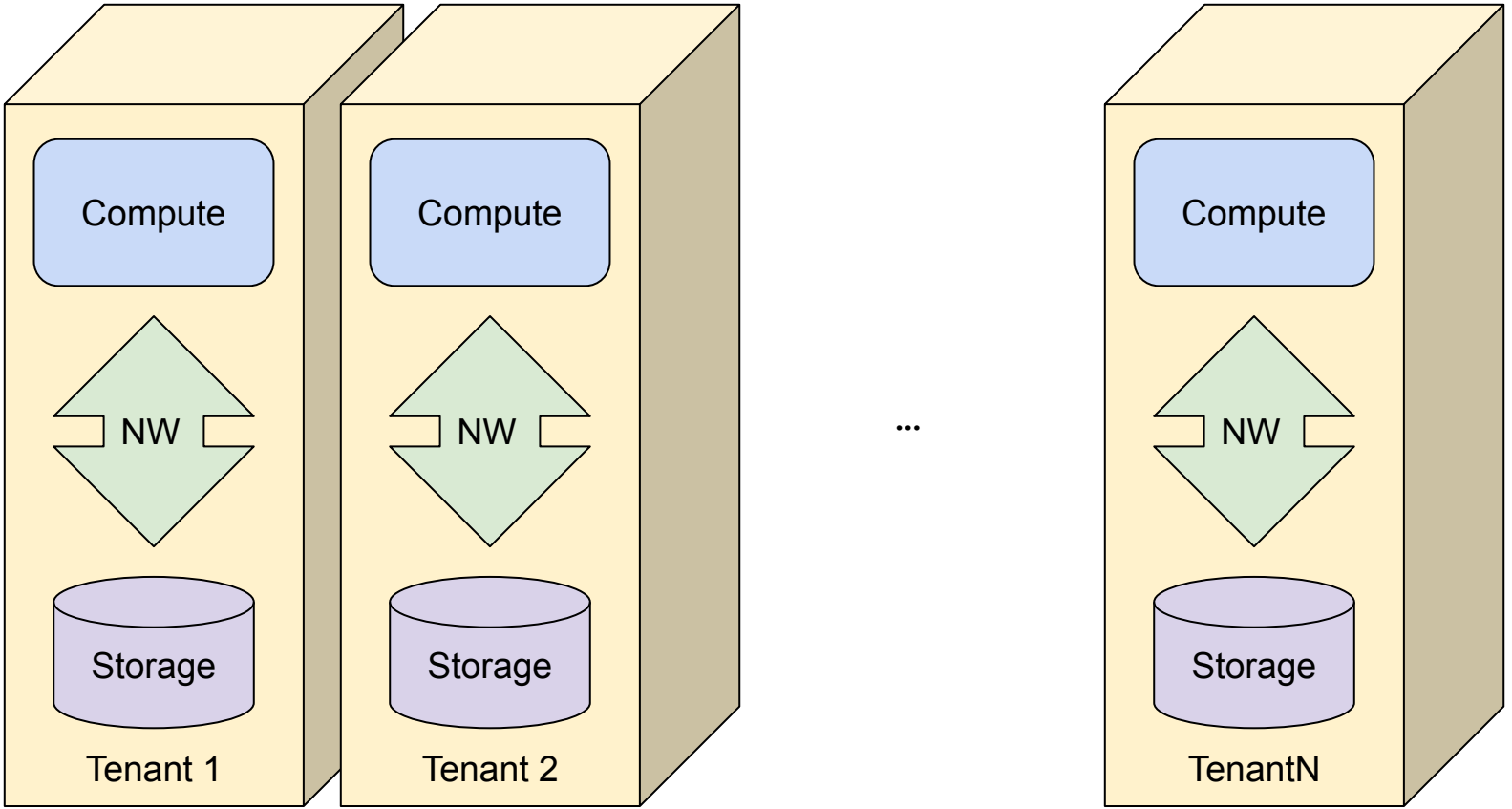
MAKING STORAGE MORE SECURE

CJ Newburn, Zero Trust and Magnum IO Architect

PARTITIONED OR UNIFIED?



PROBLEM

- The modern data center has multiple tenants
  - Explosion in demand for large-scale systems
  - Dedicated compute/storage/network resources per tenant/dept is inefficient
    → spatial and temporal sharing of resources
- Sharing involves trust
  - Other users - same time/diff resources (spatial), later use on same resources (temporal)
  - Integrity of infrastructure, which is only as strong as the weakest link
    - Consider supply chain attacks like Solar Winds vs. protecting against external entry
- Be wary
  - Assume that a rogue agent has root on a compromised OS on the compute node
    - Least trusted
  - Assume that every agent in the system could be compromised

STORAGE CONCERNS

- Storage is expensive → strong motivation to share
- Storage is persistent → spans many users and tenants over time
- Storage responsiveness matters → subject to DOS attacks
- Storage IO traffic is differentiable → may benefit from a unique NW service level

INTEGRITY

- Filer availability
- Filer contents
- Network to filer
- Data in motion and at rest

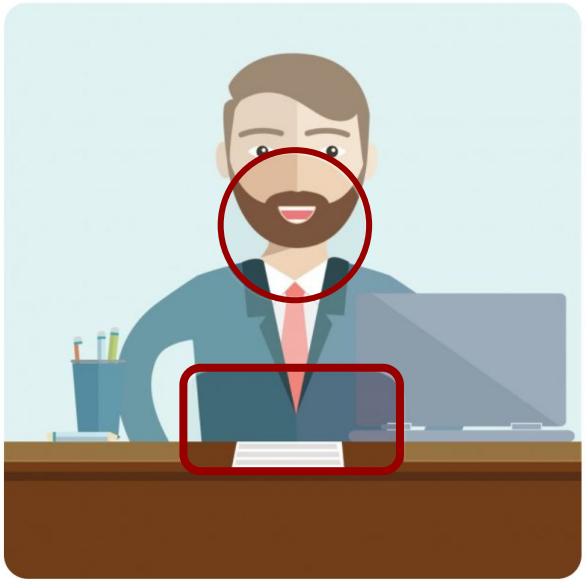
ATTACKS

- Filer availability - denial of service
- Filer contents - access/mod/rm what you shouldn’t
- Network to filer - hogging shared resources
- Data in motion and at rest - snooping network or storage

RESTRICTED INTERFACES

Teller window

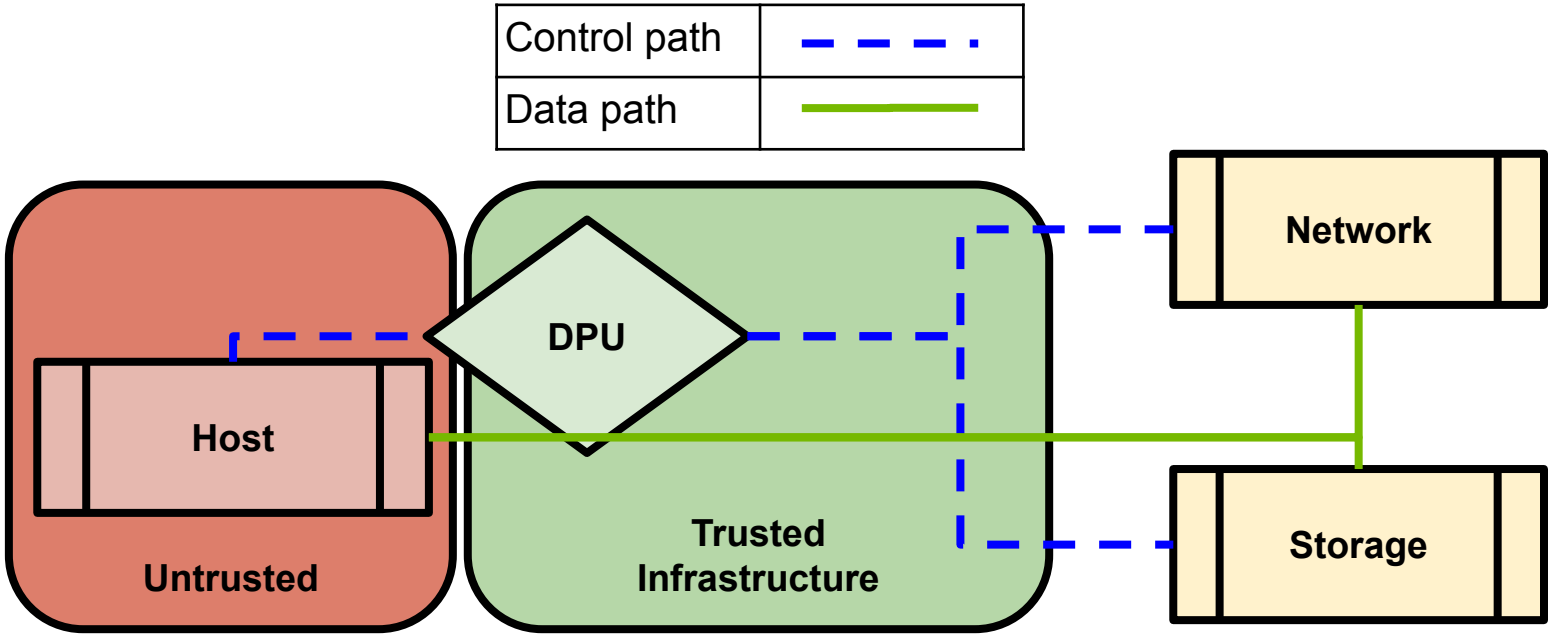
- Untrusted agent makes request
- Using restricted interface for control path
- Uninhibited data path



TRUST ZONES

Control path enabled by trusted infrastructure

Uninhibited data path (RDMA)



DATA PROCESSING UNIT

- Part of more-trusted infrastructure control plane
  - Not controlled by untrusted compute node, separate BMC
- At threshold of compute node
  - High-bandwidth, low-latency communication over PCIe
- Acceleration
  - SoC with Si dedicated to accelerating network flow, encryption, compression, etc.
  - Arm cores running standard OS capable of running protected services

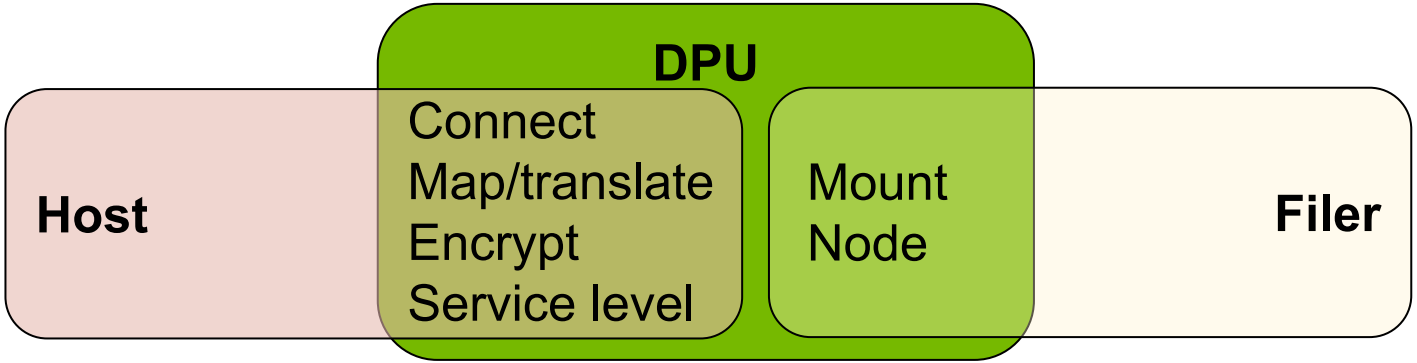
DPU TO THE RESCUE

DPU cost vs. efficiency and manageability loss from hard partitioning

- Filer availability - denial of service
  - Untrusted compute node only talks to DPU, only DPU talks to filer
- Filer contents - getting what you shouldn’t
  - Credentials and capabilities handled by DPU, not supplied by untrusted compute node
- Filer network priority - hogging shared resources
  - Service levels enforced outside of the control of untrusted compute node
- Data in motion and at rest
  - Encryption/decryption in DPU as data goes off/on the wire; stored encrypted
  - Can do virus checking with deep packet inspection

TRADE-OFFS

My goals: frame problems, explore technologies as a community



- Choices in where to implement a solution
  - Outsource client work to DPU (left)
  - Proxy actions of the filer at threshold of compute host (right)
    - May be able to reduce the BOM cost by using DPU instead of server with expensive CPU
- Trade-offs regarding the effort, robustness, generality of the solution

FILER AVAILABILITY

Connection management

- Attack - untrusted compute node
  - Rogue compute node floods filer with more requests than it can handle
- Meager mitigations - filer
  - Rate limit requests per IP, but IPs can be faked
- More robust - DPU
  - Only trusted agents can negotiate new connections
  - Host preps connection but it goes nowhere without being routed by the DPU
- Benefits with DPU
  - Avoid flooding filer and the network to it, only flood node-local DPU
  - Potentially reduce DPU-filer traffic since filtered closer to the compute node
- Established connections can use credit schemes to avoid flooding - filer

FILER CONTENTS: MOUNTING

Kernel-level example vs. user-level

- Scheduler specifies what mounts should happen more-trusted infrastructure
- Kernel normally requests a mount of the filer directly - compute node
- Respond to that request - filer
- Kernel normally completes the mount - compute node

- More robust - DPU
  - Scheduler tells DPU which mounts are licit
  - Request of a proxy on DPU vs. filer directly
- Benefits with DPU
  - More robust than trusting compute node
  - Transparent proxy leaves compute node kernel unmodifed

FILER CONTENTS: TRANSLATION

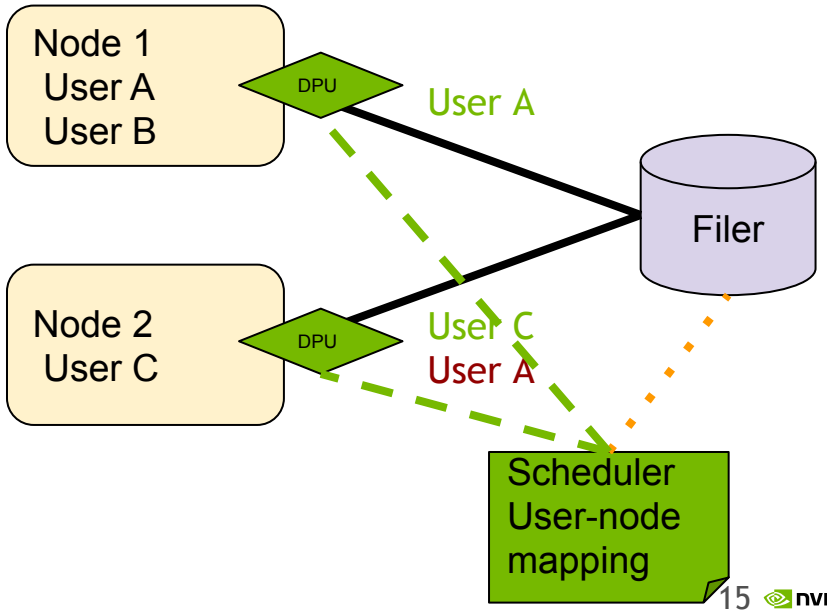
Avoid trusing the host for access checking

- Application provides file reference - <file path, offset, size>
- Passed down thru virtual filesystem (VFS) to driver
  - Uses an existing mount
  - May need to establish a new connection
  - NFS, virtiofs: passed on toward filer with uid/gid
  - Other: translated into key or blocks
- More robust - DPU
  - Proxy for NFS filer, can check gid/uid
  - Virtiofs runs reconstituted file reference through another VFS
  - Offloaded translation for modified FS client
- Benefits with DPU
  - More robust than trusting compute node
  - Transparent proxy may leave compute node kernel unmodifed

FILER CONTENTS: AUTHORIZED NODE

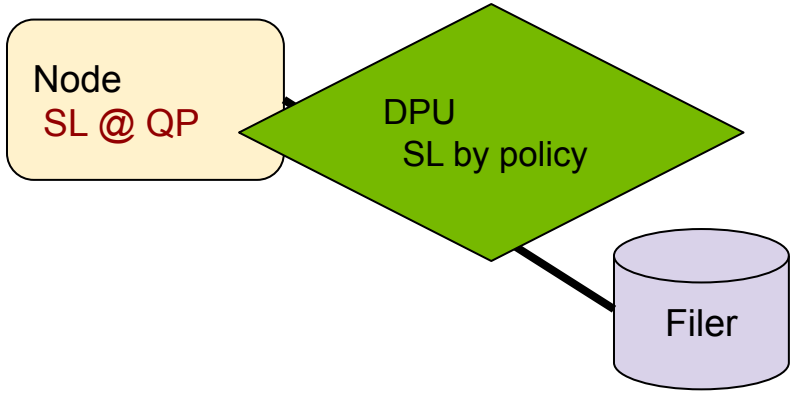
- Application provides file reference, uid, gid
- Scheduler provides user-node mapping

- Kerberos (NFSv4) checks identity but would also need to check location
- No existing standard for filers to accept and use that guidance
- Scheduler input could be introduced first at DPU in NVIDIA's data center mgt SW



FILER NETWORK PRIORITY

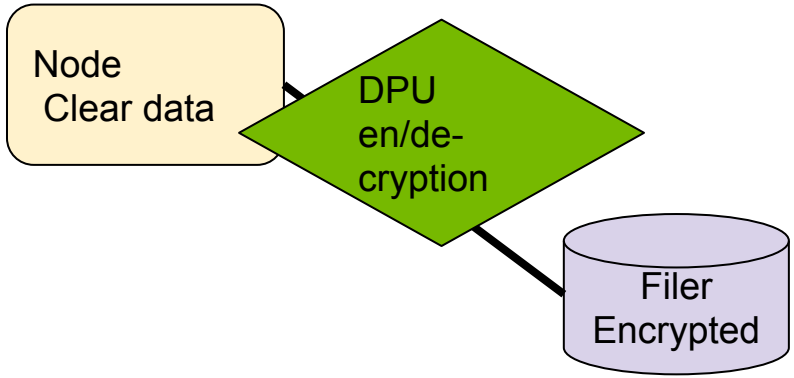
- Quality of service on network enforced by service levels (SL)
- SL encoded in queue pair
- Apps can query available SLs but there's no enforcement on what they use
- SL enforcement
  - Storage service on DPU
  - Subnet manager for host and filer HCAs
- Challenge: shared QP/node



DATA IN MOTION

- Data is en/decrypted in DPU
  - Configure data to flow thru SoC
  - Need not share encryption keys with host
- Can also do virus detection on DPU
  - Done with deep packet inspection

- At network line rate → no impact on BW



DISCUSSION

- Are these security concerns of interest to you?
- What conditions need to change to make them more important?

- Where do you see an intermediary like a DPU adding value for you?

- What security protocols do you or can you use?
  - LDAP, kerberos, NIS
  - Valid IP endpoints

- Are you interested in collaboration to evaluate options, drive toward standards?
  cnewburn@nvidia.com