2021 OFA Virtual Workshop

PERFORMANCE SCALED MESSAGING V3 (PSM 3) ARCHITECTURE OVERVIEW

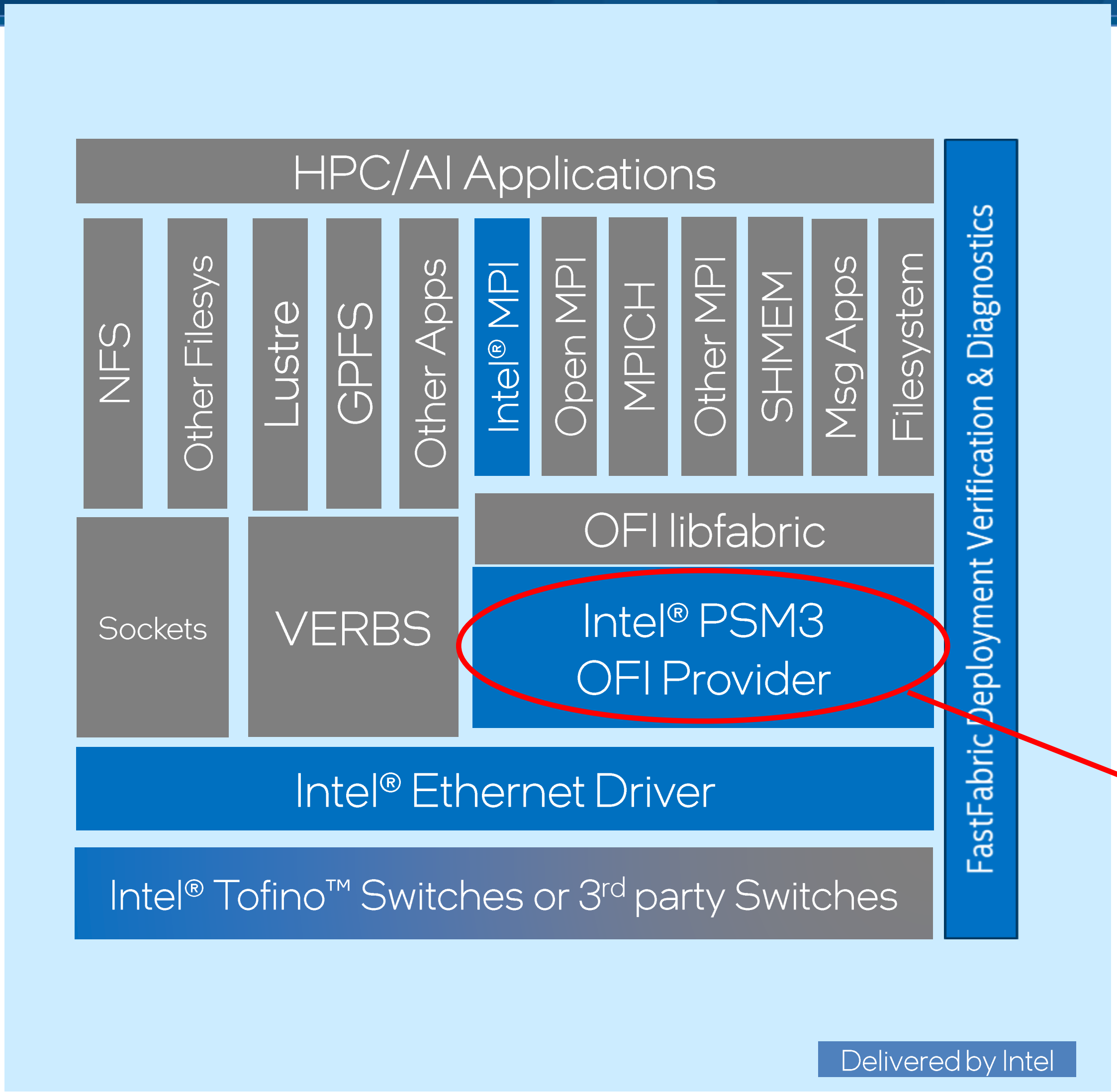
Todd Rimmer, Director Software Architecture

Intel Corp

WHAT IS PSM3?

- **PSM3 is a new libfabric provider**
  - Leverages concepts and code from Intel® Omni-Path Architecture (OPA)
  - Mature and Feature rich
- **PSM3 is designed for RoCE**
  - Optimizes performance and scalability
  - Uses standard RoCE protocols and APIs
- **PSM3 is available upstream now**
  - Integrated into libfabric
  - Out of Tree code for older distros available on github

HPC/AI COMMS ARCHITECTURE WITH PSM3



3rd generation Performance Scaled Messaging (PSM3)

- Evolution of PSM (TrueScale) & PSM2 (OPA)
- Enhanced for Ethernet and RoCE v2

Compatible with existing MPI applications

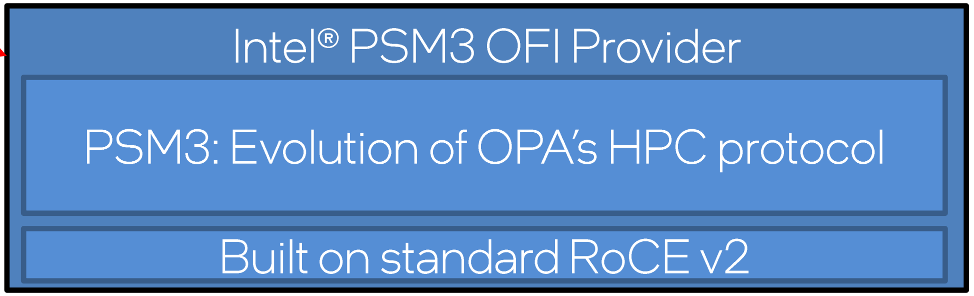
- No code changes necessary

Leverages the OpenFabrics Alliance*

- Standards based software

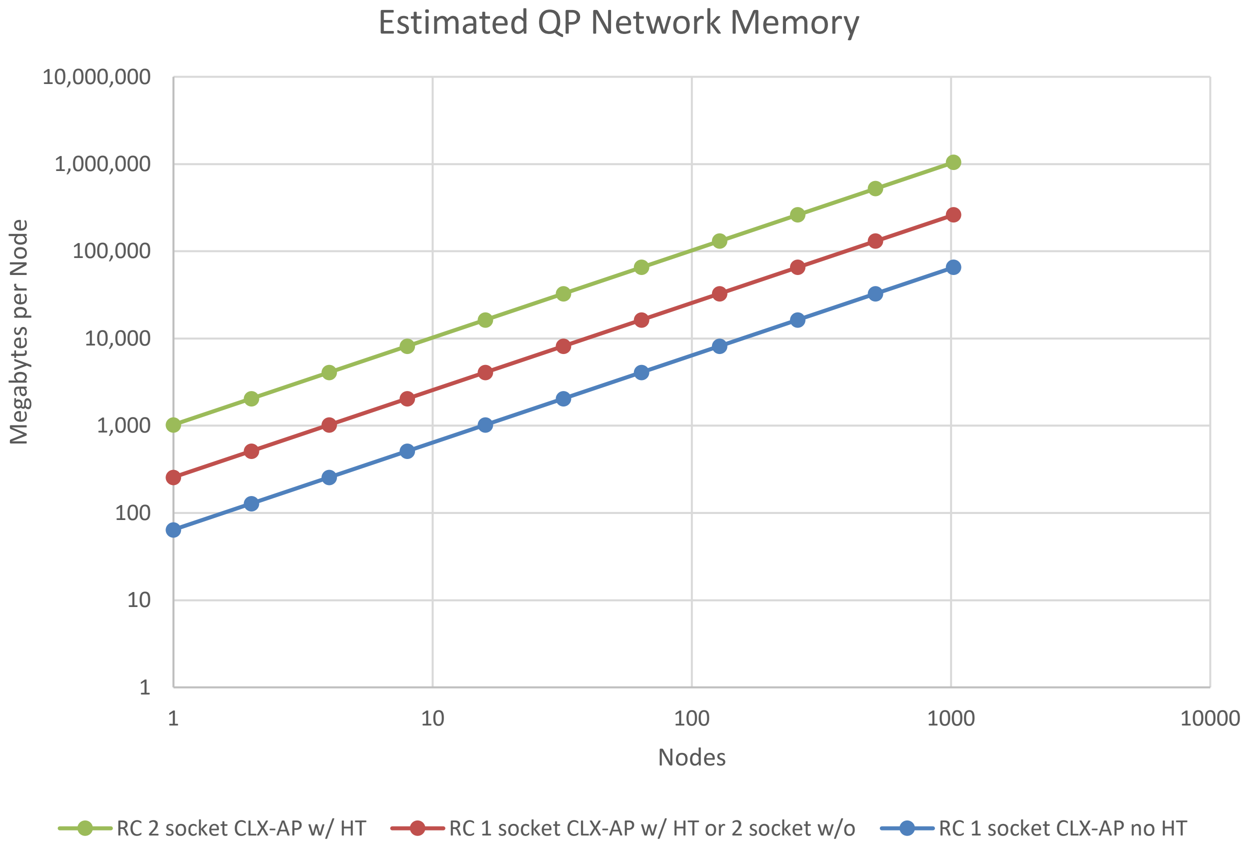
Provides communications for OneAPI

- Open common environ for CPU/GPU/Accel



Intel® PSM3 OFI Provider
PSM3: Evolution of OPA's HPC protocol
Built on standard RoCE v2

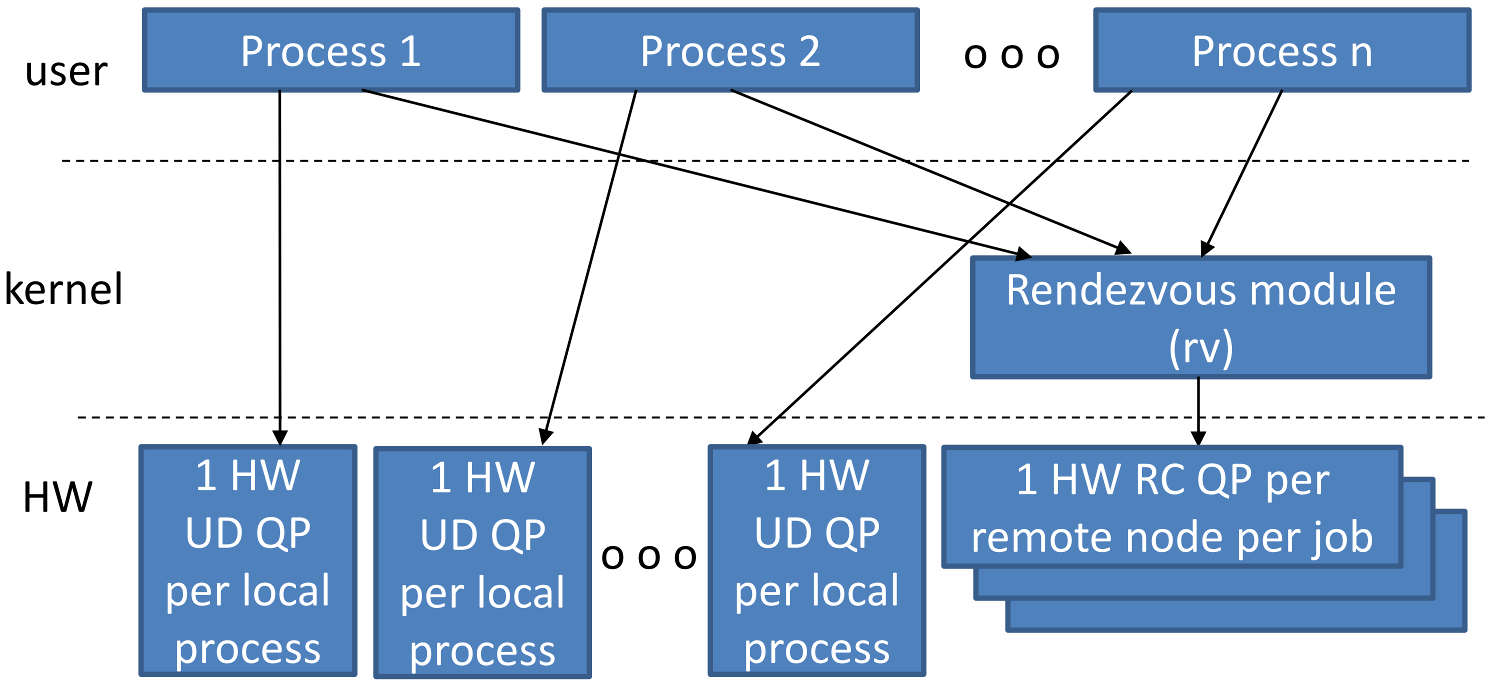
THE QP AND MEMORY CHALLENGE AT SCALE



1 MPI rank per core.
~20KB state (WQE+Buffer+QP state) per RC QP
CLX-AP socket w/56 cores (112 with HT)

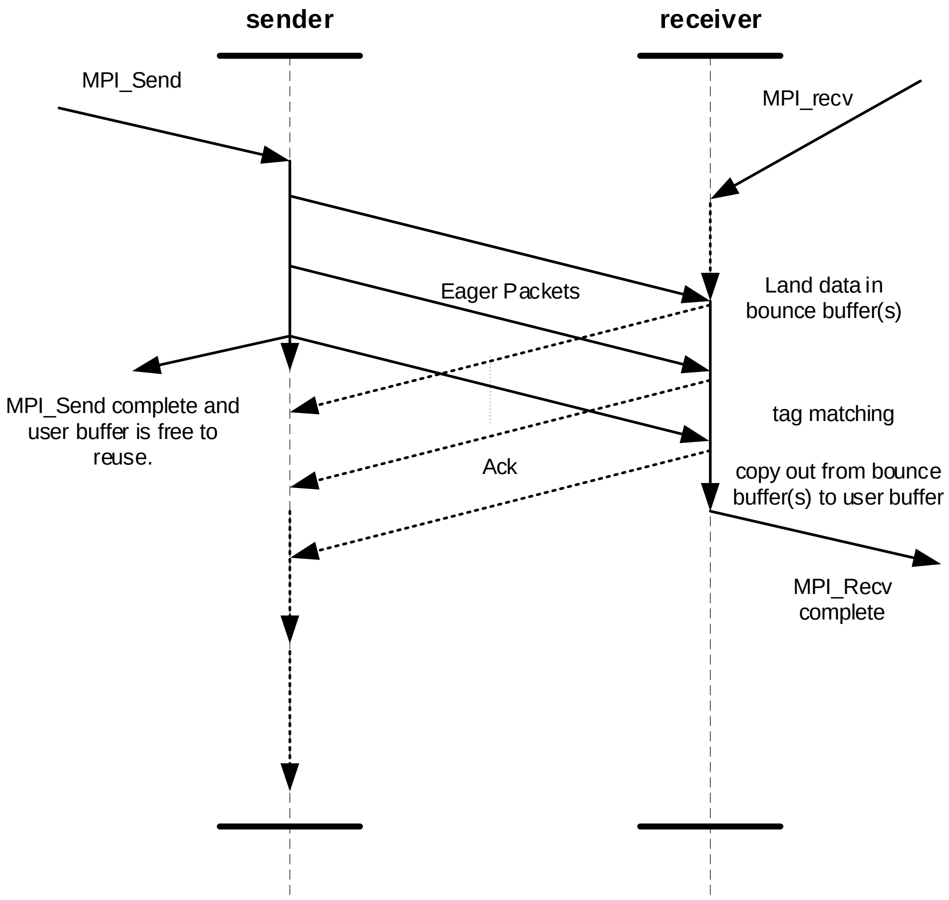
- **RC QP WQE + Buffer + QP State excessive at >100 Nodes**
  - 10GB-100GB per server @ 100 nodes
- **Driving Factors**
  - Per RC QP WQE and recv buffer space
  - High core count servers with 1 MPI rank per core
    - quadratic component in memory footprint
- **PSM3 Solution**
  - UD based eager and control protocols
    - per process UD QP WQE and recv buffer
      - linear component in memory footprint
    - reduced per connection state
      - still a quadratic component, much smaller coefficient
  - Shared Node to Node RDMA QPs
    - linear QP scaling with RDMA for rendezvous

QP MODEL AND RENDEZVOUS MODULE

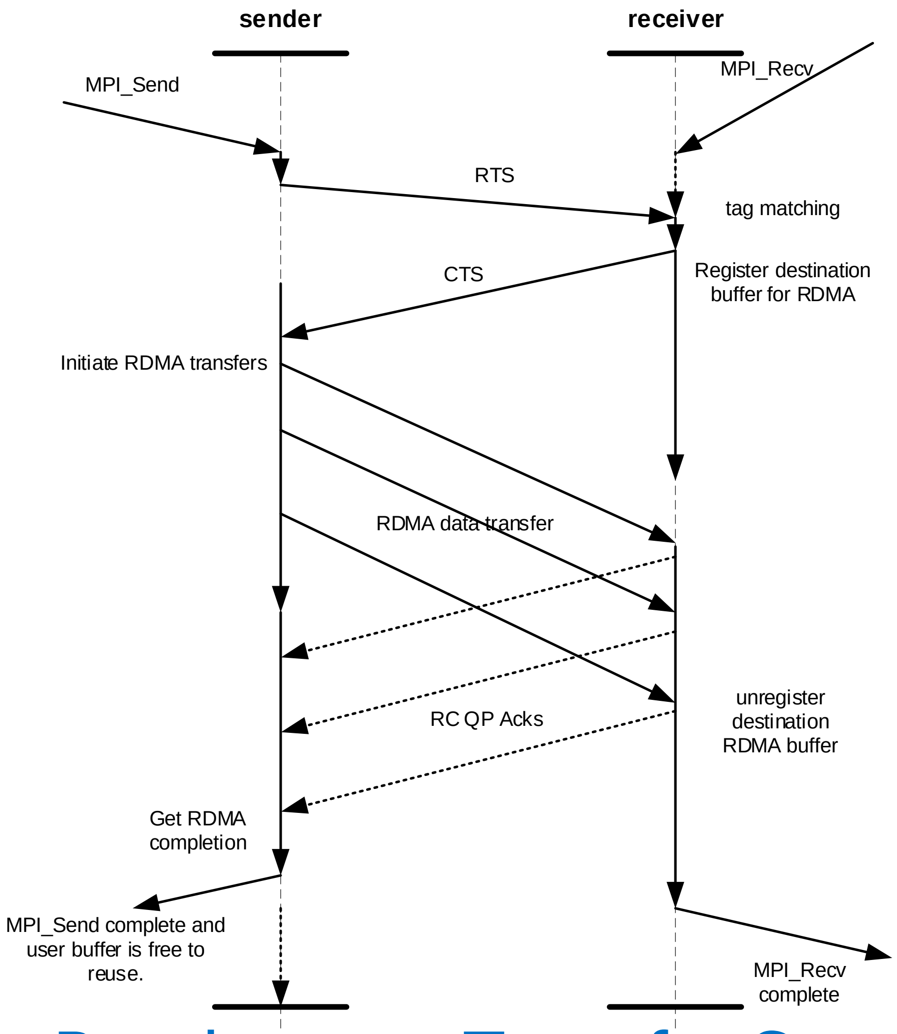

user
kernel
HW
Process 1
Process 2
o o o
Process n
Rendezvous module (rv)
1 HW UD QP per local process
1 HW UD QP per local process
o o o
1 HW UD QP per local process
1 HW RC QP per remote node per job

- **Scalable latency benefits of UD**
- **Use RDMA for Rendezvous**
- **Keep memory footprint in line**
  - O(nodes+ppn) vs O(nodes*ppn^2) memory and QP scaling
  - Keeps QP caches hot @ scale
- **Node-Node shared RC QPs**
  - Shared across processes in job
  - multi-QP striping option (default 4)
- **MR caching**
- **Automatic QP Recovery**
  - Restores disrupted connections
- **Leverages concepts from OPA**

BASICS OF DATA MOVEMENT STRATEGIES


sender
receiver
MPI_Send
MPI_recv
Land data in bounce buffer(s)
Eager Packets
MPI_Send complete and user buffer is free to reuse.
tag matching
Ack
copy out from bounce buffer(s) to user buffer
MPI_Recv complete

Eager Transfer Strategy



Rendezvous Transfer Strategy

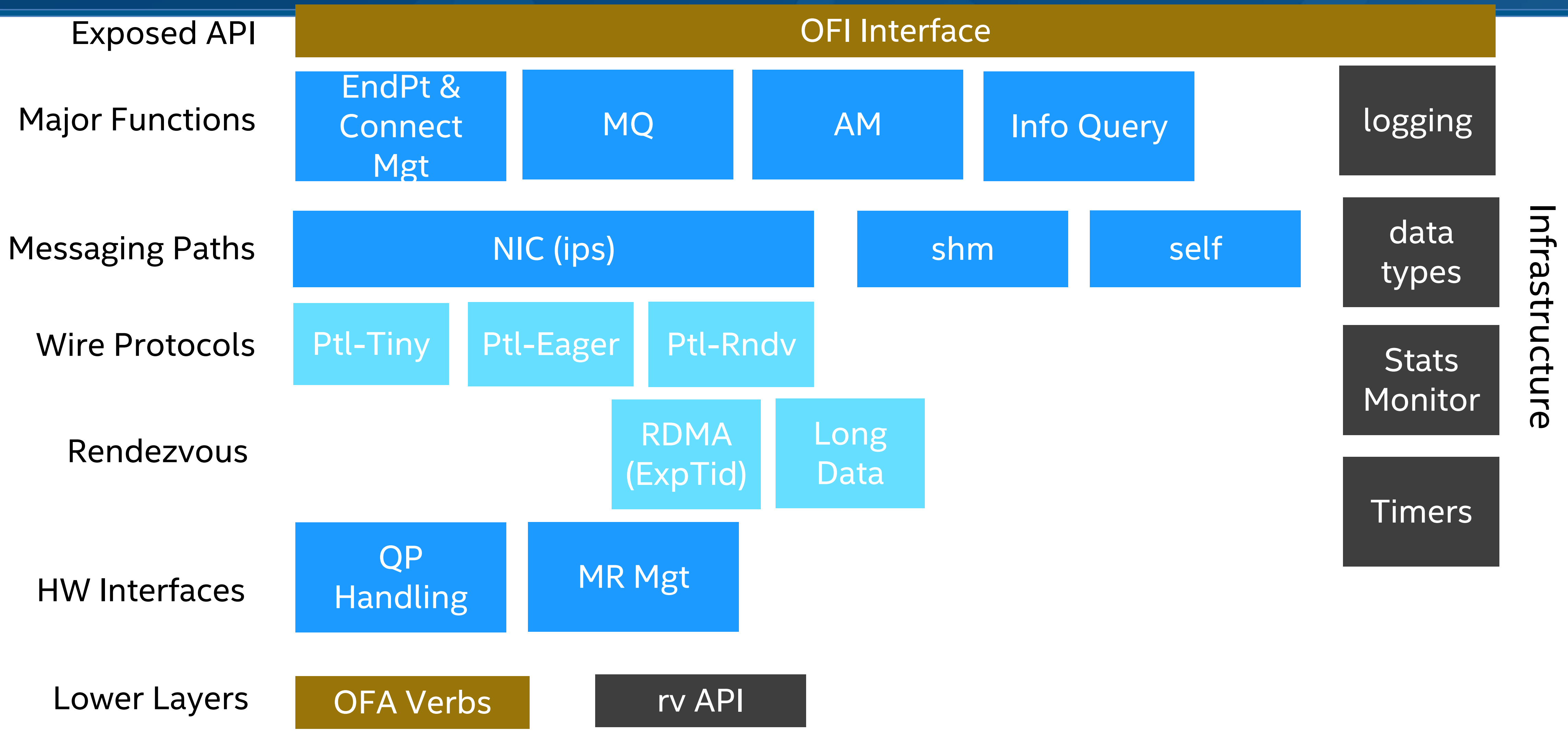
PSM3 ADVANCED CAPABILITIES

- **Multi-Rail, especially for AI**
  - 1 NIC/proc
  - Multi-NIC/proc single plane
  - Multi-NIC/proc multi-plane
- **Multi-Endpoint**
- **Tunable strategies**
  - eager/rendezvous, load balancing, etc
- **Resilient to fabric disruptions**
- **Dispersive routing**
- **Independent progress option**
- **Scalable tag matching algorithms**
- **Credit based flow control**
- **Receiver side Rendezvous pacing**
- **Multi-CTS Rendezvous pipelining**
  - Striping of large messages (rails and/or QPs)

- **PSM3_RDMA modes**
  - Mode 0 – UD QP only
    - Most scalable, lowest memory footprint
  - Mode 1 – RV shared RC QP for Rendezvous RDMA
    - >64,000 bytes by default
    - Next most scalable, Best BW
  - Mode 2 – User Space RC QP for Rendezvous RDMA
    - Slightly less latency for large messages (~5% less)
    - Higher memory footprint
  - Mode 3 – User Space RC QP for Eager and Rendezvous
    - Control on UD
    - Least latency, least scalable, highest memory footprint
- **Multiple Connections load balancing**
  - RV (Mode 1) – multiple QPs per remote endpoint
  - QP_PER_NIC – multiple UD & RC QP endpoints per NIC
- **MR Caching**
  - For modes 1-3, kernel MR w/ MMU notifier hooks

Mature Features and Optimizations Brought Forward from Omni-Path

PSM3 USER SPACE OFI PROVIDER ARCHITECTURE



Exposed API: OFI Interface

Major Functions: EndPt & Connect Mgt, MQ, AM, Info Query

Messaging Paths: NIC (ips), shm, self

Wire Protocols: Ptl-Tiny, Ptl-Eager, Ptl-Rndv

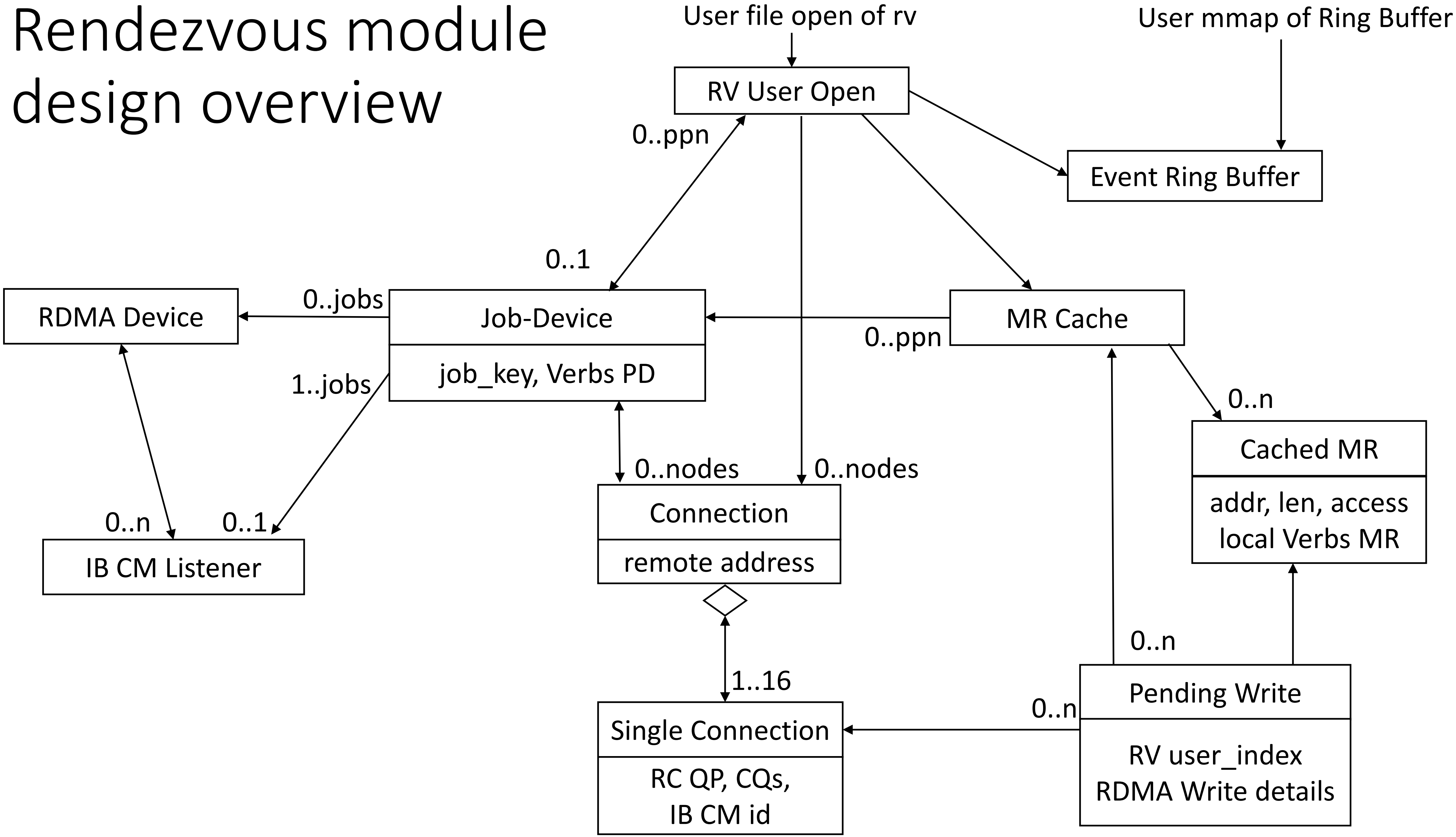
Rendezvous: RDMA (ExpTid), Long Data

HW Interfaces: QP Handling, MR Mgt

Lower Layers: OFA Verbs, rv API

Infrastructure: logging, data types, Stats Monitor, Timers

Rendezvous module design overview



UPSTREAM REPOS

- **https://ofiwg.github.io/libfabric - Includes PSM3 OFI (libfabric) provider**
  - Code fully in libfabric 1.12.0
  - https://github.com/intel/eth-psm3-fi - out of tree avail now
    - runs with pre-existing stock libfabric, including RHEL7.9-8.3
    - OOT build mechanism co-designed with libfabric maintainer
- **http://kernel.org – rv kernel driver – scalably enables zero-copy**
  - Community engagement in progress
  - https://github.com/intel/iefs-kernel-updates – out of tree avail now
    - runs with pre-existing in-distro OFA, including RHEL7.9-8.3
- **https://github.com/intel/eth-fast-fabric - FastFabric Admin Tools**
  - Avail now
- **https://github.com/intel/eth-mpi-apps - 3rd party benchmarks, for ref**
  - Avail now

SUMMARY

- **PSM3 is a new libfabric provider**
  - Leverages concepts and code from Intel® Omni-Path Architecture
  - Uses an optional kernel module to optimize rendezvous RDMA transfers and scalability
  - Mature and Feature rich
- **PSM3 is designed for RoCE**
  - Optimizes performance and scalability
  - Uses standard RoCE protocols and APIs
- **PSM3 is available upstream now**
  - Integrated into libfabric 1.12.0
  - Out of Tree code for older distros available on github

OPENFABRICS ALLIANCE

2021 OFA Virtual Workshop

THANK YOU

Todd Rimmer, Director Software Architecture

Intel Corp

intel