OPENFABRICS ALLIANCE

12th ANNUAL WORKSHOP 2016

INTRODUCING SANDIA* OPENSHMEM WITH MULTI-FABRIC SUPPORT USING LIBFABRIC:

Kayla Seager, Software Engineer, **Intel**

*Ryan Grant, Senior Member of Technical Staff, **SNL***

Special thanks to Sayantan Sur and Jim Dinan, Intel

[April 5th, 2016]

OUTLINE

- **Background**
  - Libfabric and SHMEM co-design
  - Sandia SHMEM support and software structure
- **Scalable Design**
  - Elements leveraged from Libfabric
- **Portability**
  - SOS support for many fabrics
- **Results**
  - Initial performance results

BACKGROUND

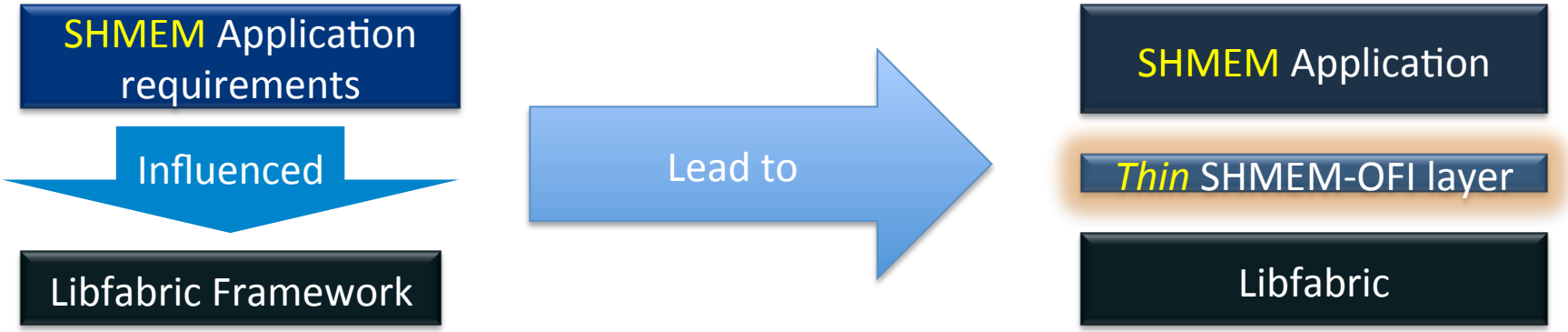
OFIWG LIBFABRIC DESIGN

- **OFA's OFIWG Goals for Libfabric:**
  - "tight semantic map between applications and underlying fabric services" – OFIWG https://ofiwg.github.io/libfabric/

App Centric Design = Performance, Scalability, Portability

SANDIA-SHMEM-LIBFABRIC METHODOLOGY

Application mapping example embodies Libfabric's goal:

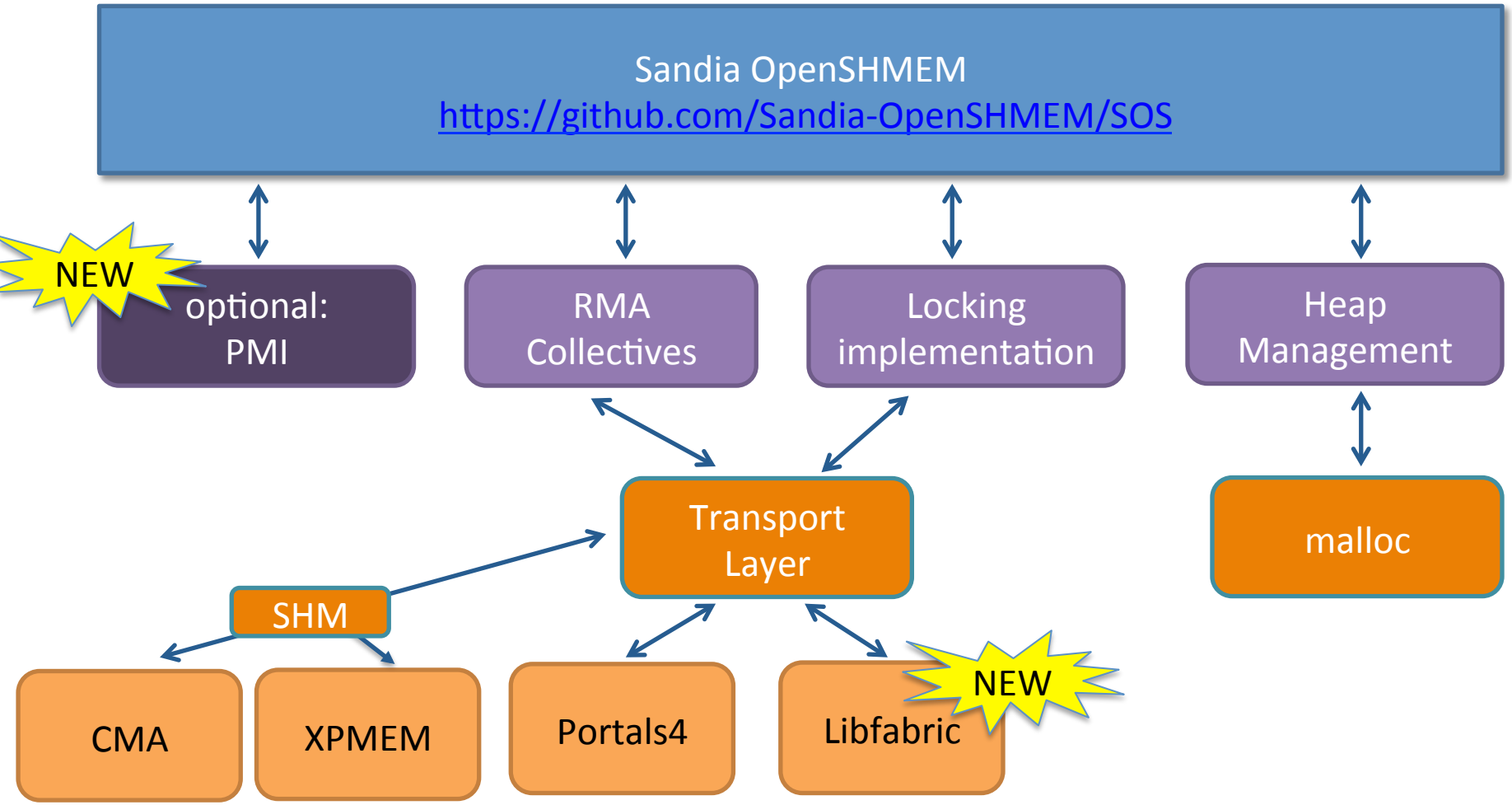


SHMEM-Libfabric = Performance, Scalability, & Portability

WHAT IS OPENSHMEM?

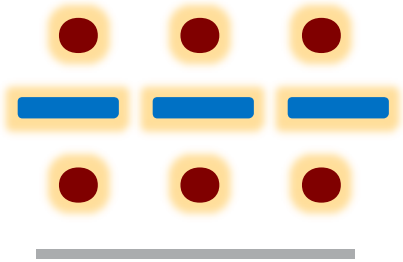
- **HPC Communication Programming Model API**
  - RMA & Atomic Pt-Pt
  - Distributed shared memory model (symmetric addressing)
  - Collectives
    - barrier, broadcast, reduce, all-to-all, strided all-to-all

SANDIA OPENSHMEM



SANDIA-SHMEM CODEBASE PART 2

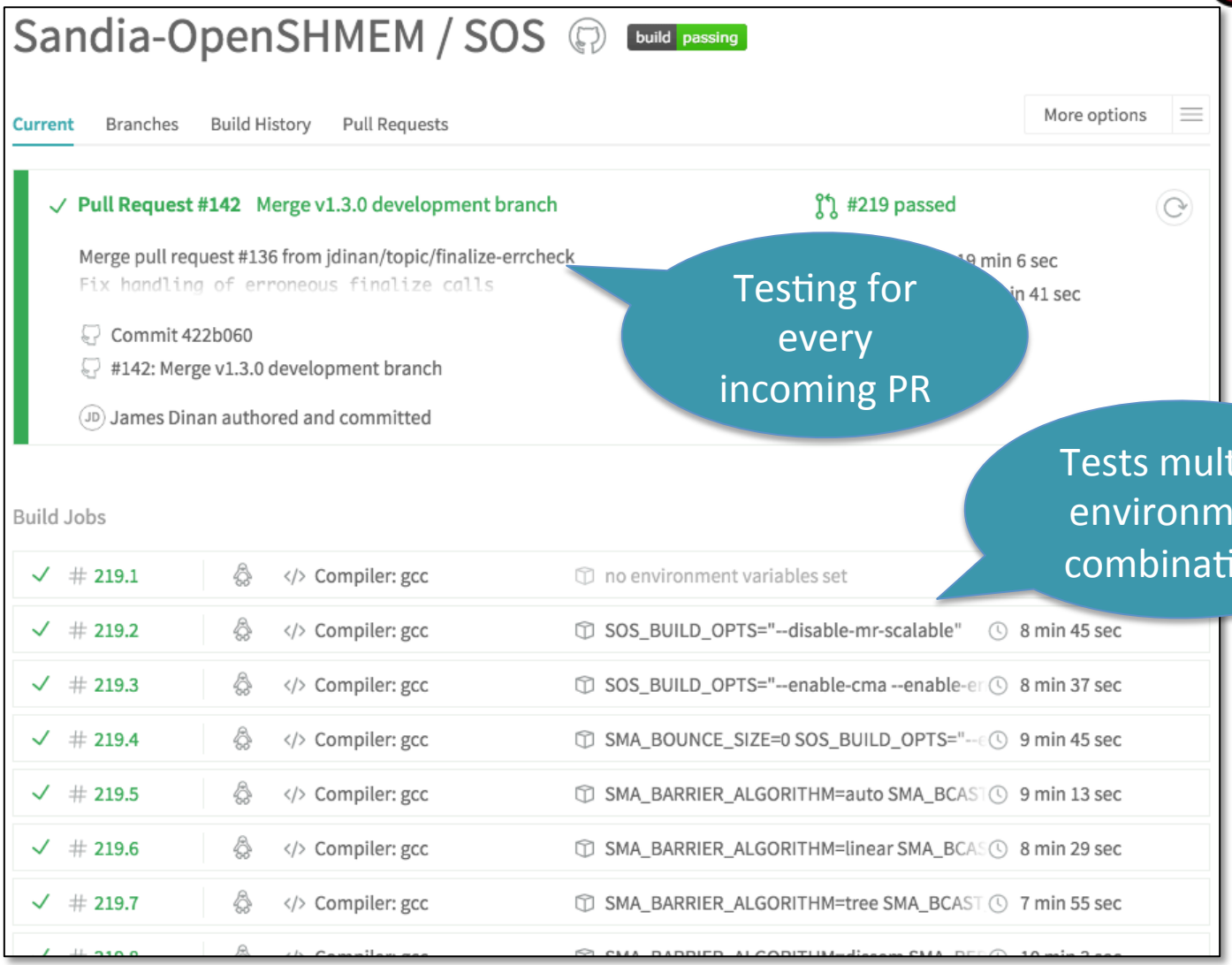
- **OpenSHMEM 1.2.0 compliant release**
- **OpenSHMEM 1.3.0 compliant release (1st)**
  - includes true non-blocking API and all-to-all collectives
- **Open Source BSD license**
- **Vehicle to…**
  - establish interface between Libfabric and OpenSHMEM
  - Drive both specifications



PR TESTING FRAMEWORK: TRAVIS CI

- **Software Quality**
  - Continuous integration testing
  - Rich test suite x Multiple interfaces
  - Supports independent run environments
  - Easy issue tracking -> bug resolution

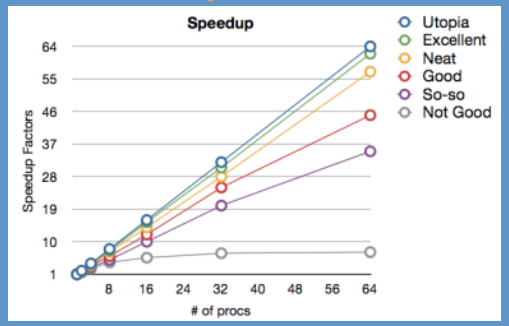
PR TESTING FRAMEWORK: TRAVIS CI



SCALABILITY AND PERFORMANCE

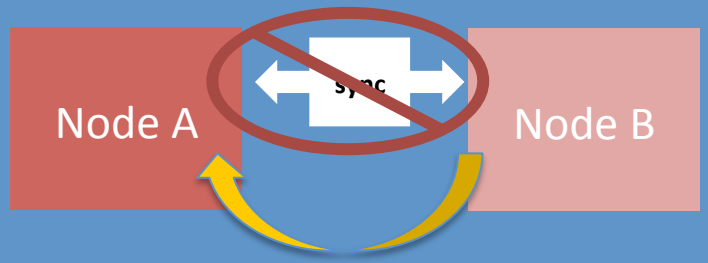
OFIWG'S SHMEM REQUIREMENTS BAKED INTO OFI USED BY SHMEM-OFI

Scalable Endpoint Enumeration



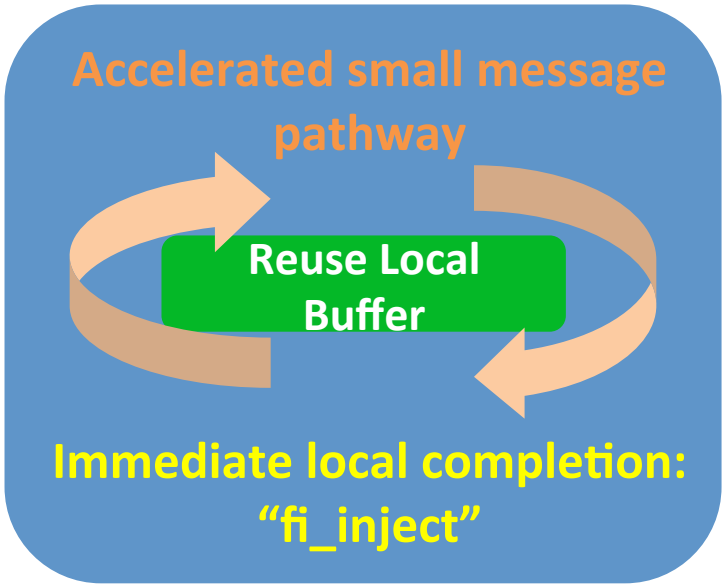
Reliable connectionless endpoints with logical addressing

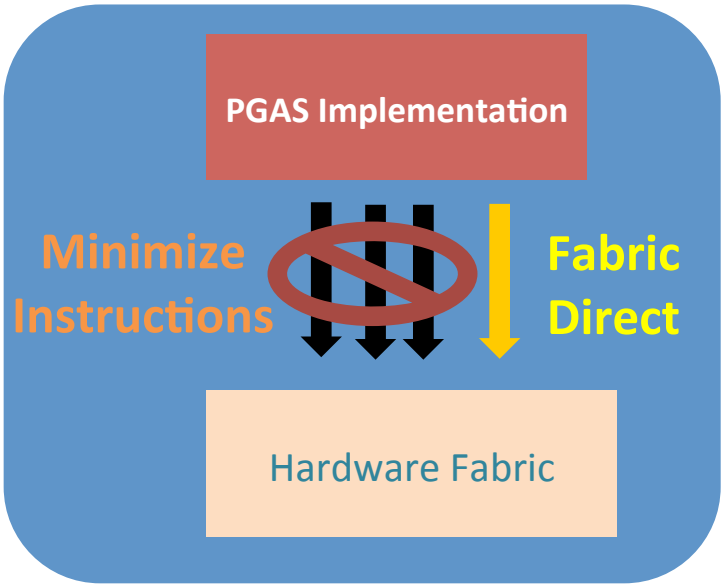
Efficient Remote Completion for Put/Get



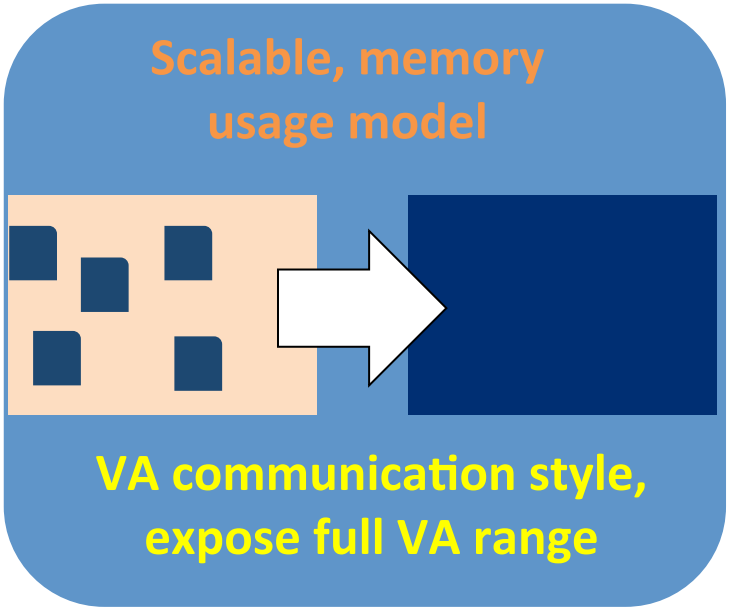
enabled counters for lightweight remote completion

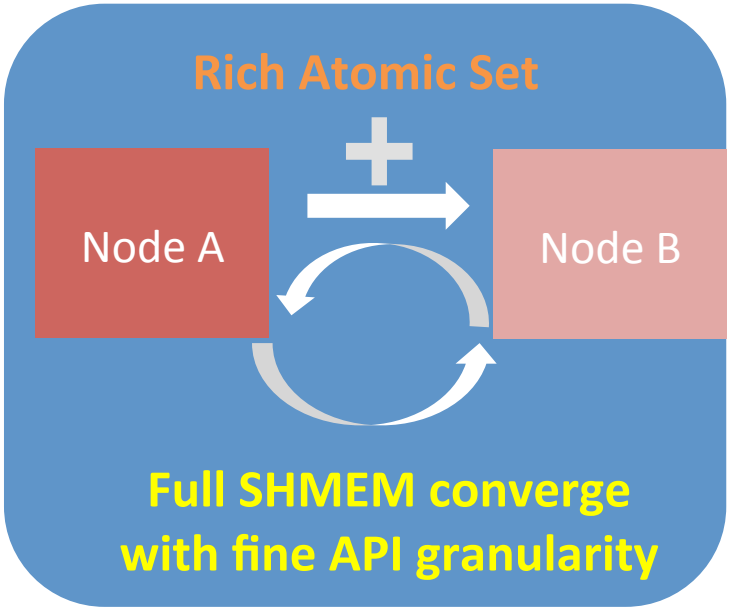
OFIWG'S SHMEM REQUIREMENTS BAKED INTO OFI USED BY SHMEM-OFI





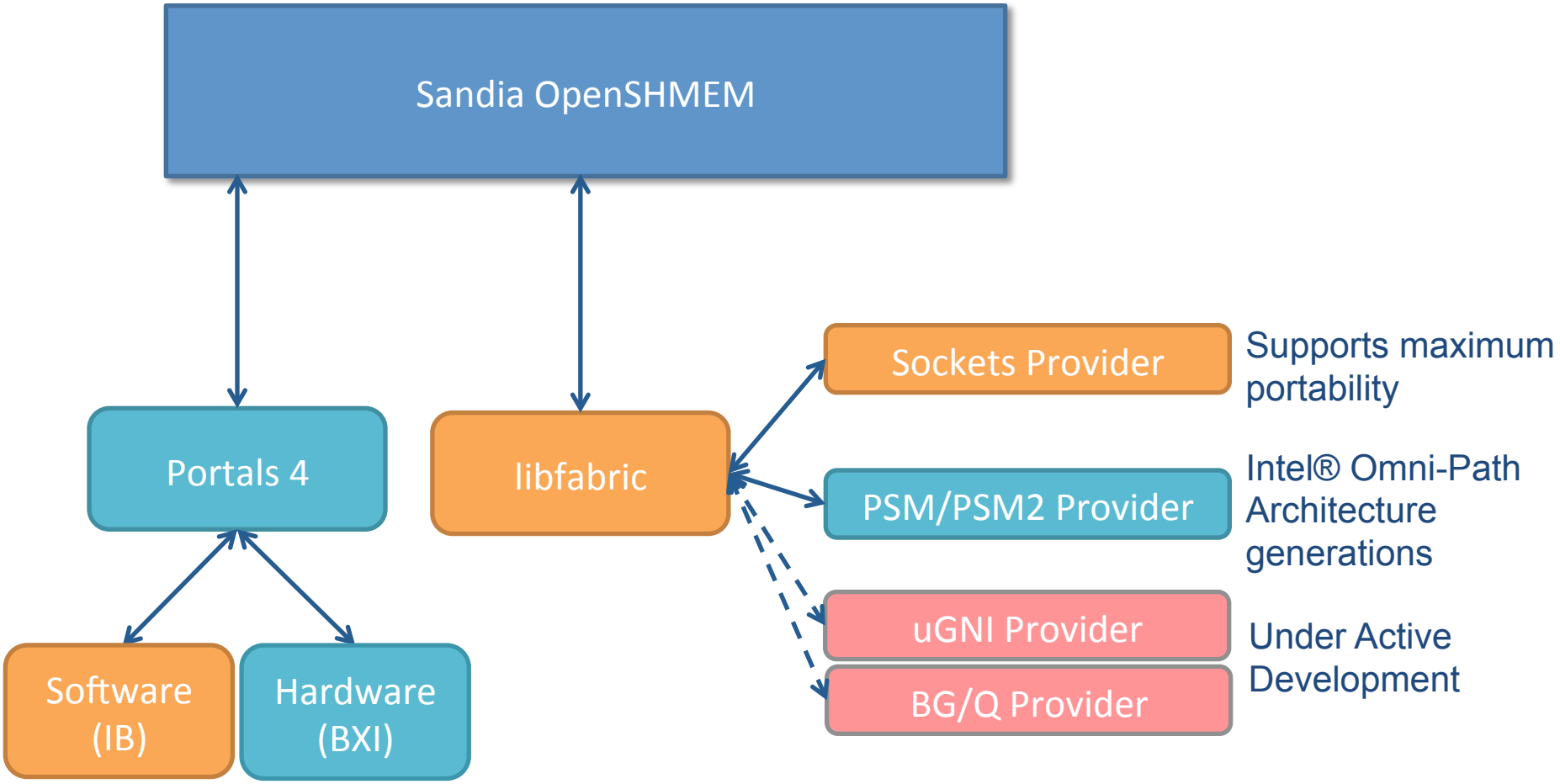
OFIWG'S SHMEM REQUIREMENTS BAKED INTO OFI USED BY SHMEM-OFI





PORTABILITY

PORTABILITY SCOPE



MEMORY MODEL BEFORE

Libfabric Semantic	Remote Virtual Addr. (OS support)
MR_SCALABLE *user defined keys *Full VA Space +no key exchange +single registration for full VA space +minimal footprint (don't track keys)	**Symmetric Addressing** (Address on Node A = Address on Node B) +full VA space addressing +minimal footprint (don't track addresses)

MR_SCALABLE and symmetric VA = good SHMEM semantic match

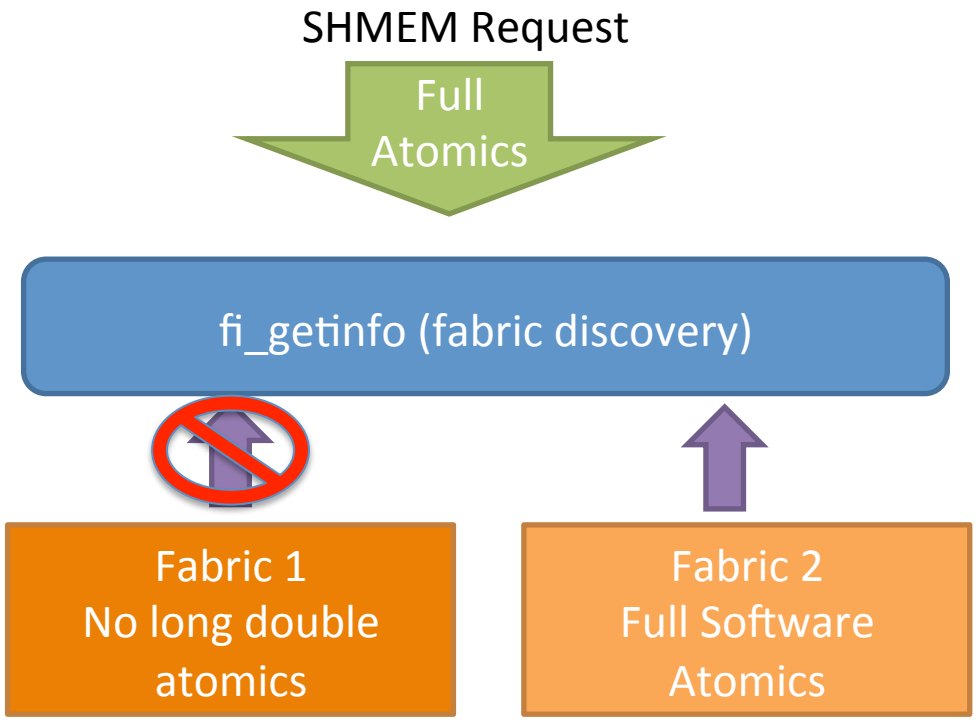
MEMORY MODEL AFTER

OFI Semantic	Remote Offset	Remote Virtual Addr. (OS support)
MR_SCALABLE *user defined keys	Offset addressing, -track and exchange offset	Symmetric addressing
MR_BASIC *fabric defined keys	Offset Addressing -track and exchange offset -track and exchange keys	Symmetric Addressing -track and exchange keys

Adding MR Basic & offset enables broader provider support

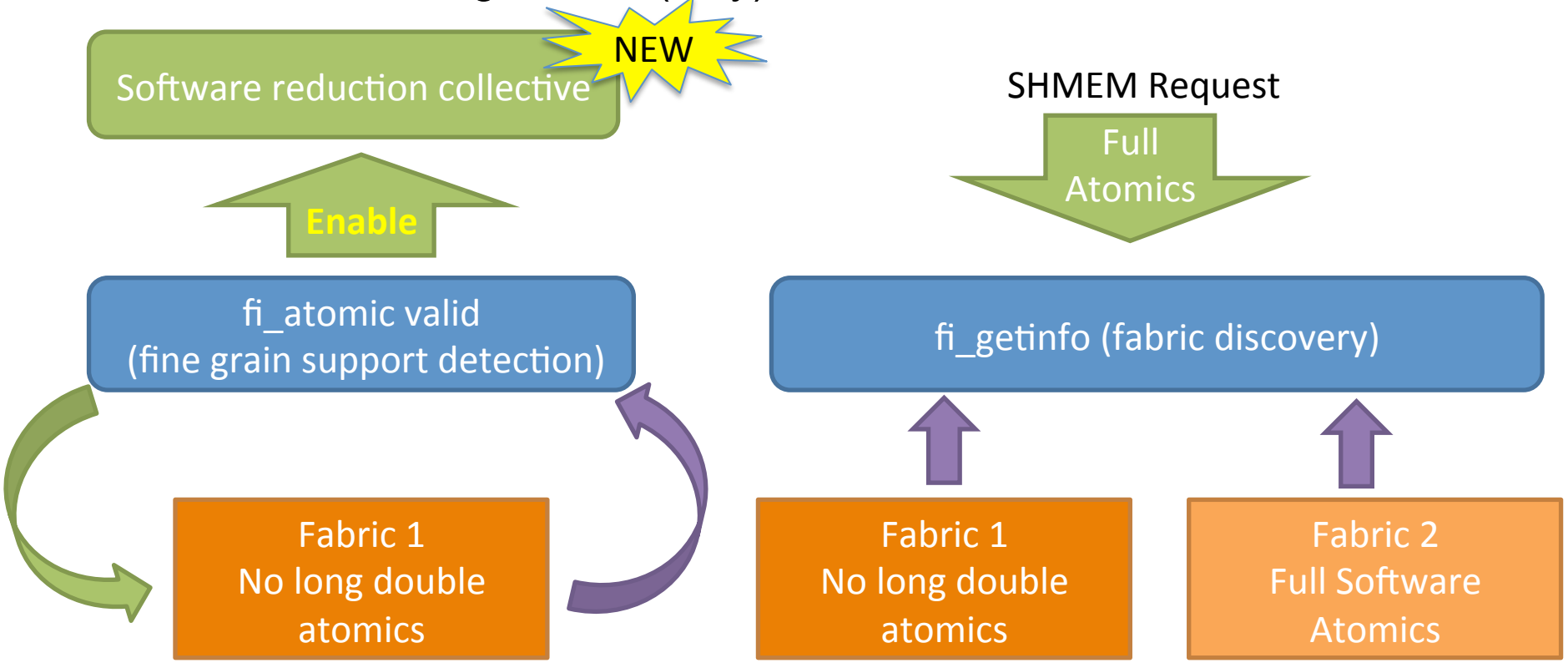
ATOMICS MODEL BEFORE

- **Require full atomic fabric support**
  - No current hardware supports long double atomics



ATOMICS MODEL AFTER

- **Enabled "hybrid" atomic support for portability**
  - detect and allow long double (only) limitation

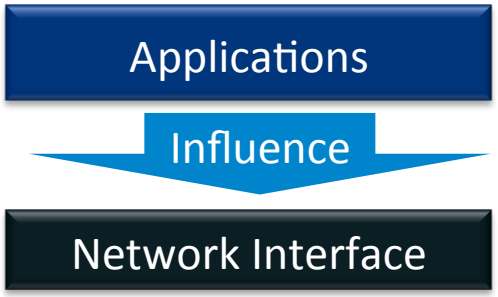


EARLY RESULTS

- **Scalability:**
  - Sandia-SHMEM-OFI-PSM2: scaling to 2,048 PEs (16ppn)
    - Using ISx (integer sort) with 134,217,728 keys per PE
  - Sandia-SHMEM-OFI-Sockets: scaling 512 PEs (14ppn)
    - Full OSU* SHMEM test-suite
- **Performance:** *Thin* SHMEM-OFI layer
  - Shmem_int_p ->fi_inject_write: **16 instructions**
  - Shmem_fence -> fi_cntr_wait: **11 instructions**

SANDIA-SHMEM-LIBFABRIC SUMMARY

- **Performance and Scalability**
  - Enabled by SHMEM requirements baked into libfabric framework
  - shmem_int_p: 16 instructions
  - scaled to 2,048 processes



- **Portability**
  - 4 different fabrics/providers
  - SHMEM-libfabric changes for portability
    - Flexible memory model (compile time choice)
    - Software long double atomic support

OPENFABRICS
ALLIANCE

12th ANNUAL WORKSHOP 2016

THANK YOU

Kayla Seager, Software Engineer, **Intel**
Ryan Grant, Senior Member of Technical Staff, **SNL**

Special thanks to Sayantan Sur and Jim Dinan

Part of this work was funded through the Computational Systems and Software Environment sub-program of the Advanced Simulation and Computing Program funded by the National Nuclear Security Administration