OPENFABRICS ALLIANCE

14th ANNUAL WORKSHOP 2018

T10-DIF OFFLOAD

Tzahi Oved

Mellanox Technologies

Apr 2018

[LOGO HERE]

OPENFABRICS ALLIANCE

T10-DIF INTRO

IT IS ALL ABOUT INTEGRITY

- **Storage system is typically multi-layered**
  - Memory
  - OS file system
  - Remote storage adaptor (HBA/NIC/..)
  - Storage Controller (SATA/SAS/NVME/..)
- **Any of the above has it’s own integrity check**
  - CSUM/CRC
  - Parity check
- **But none protect against:**
  - OS bugs, driver bugs, disk controllers and their FW errors and storage admin errors
  - Data at rest

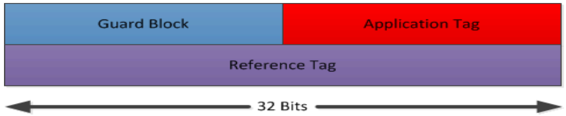
- **End to end storage integrity protection is needed**

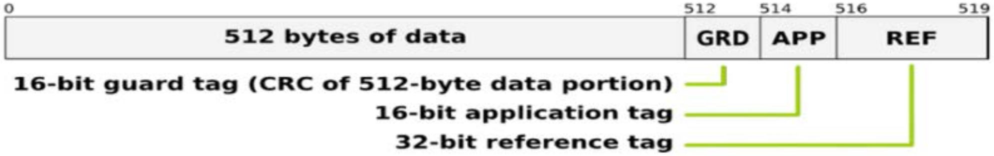
- ***Data integrity metadata should be generated together with data creation***

WHAT IS DIF

- **DIF = Data Integrity Field**
  - Aka “PI” – Protection Information
  - Aka “Signature”

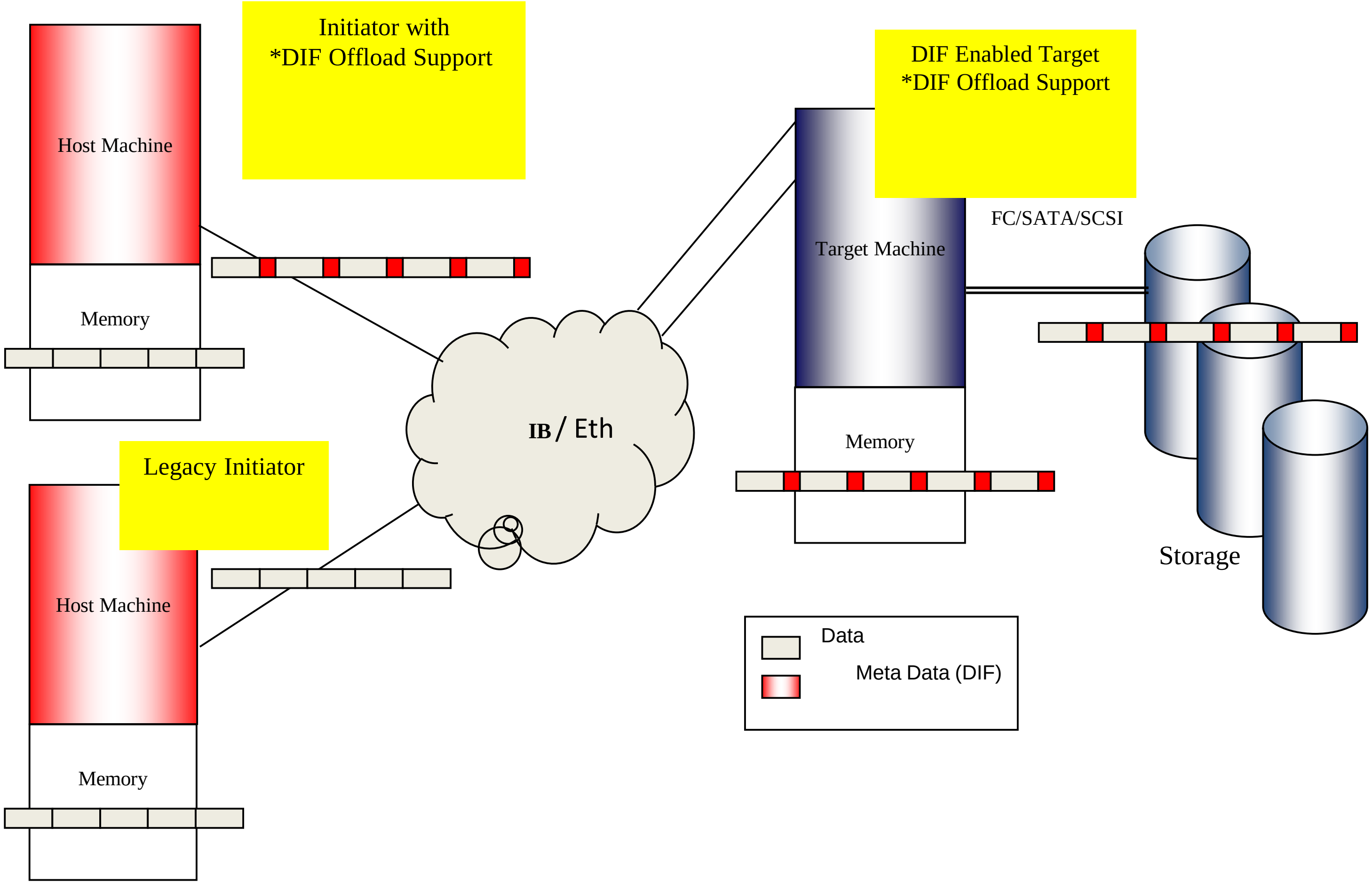
- **The T10 standard committee specify an additional 8 byte field designated for data integrity/protection for each data block (usually of size 512 bytes but not a must).**



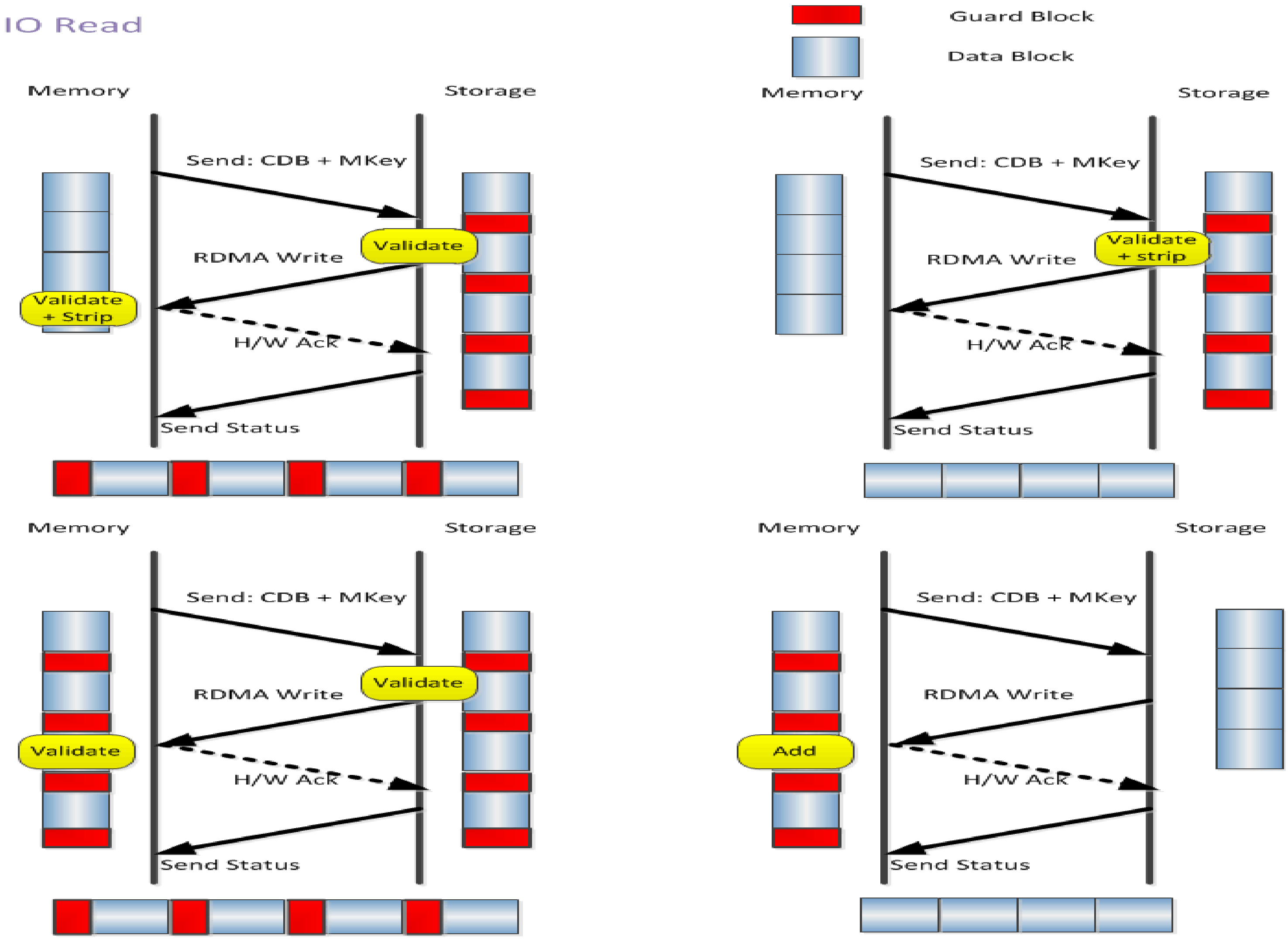


- **GUARD tag (Logical Block Guarding)**
  - 2 bytes CRC covering the 512 byte data sector
- **APPLICATION tag (Up for grabs)**
  - 2 bytes per sector - ownership negotiated with target
- **REFERENCE tag (Misdirected writes)**
  - 4 bytes – Information associated with a specific data block, typically lower 4 bytes of the Logical Block Address

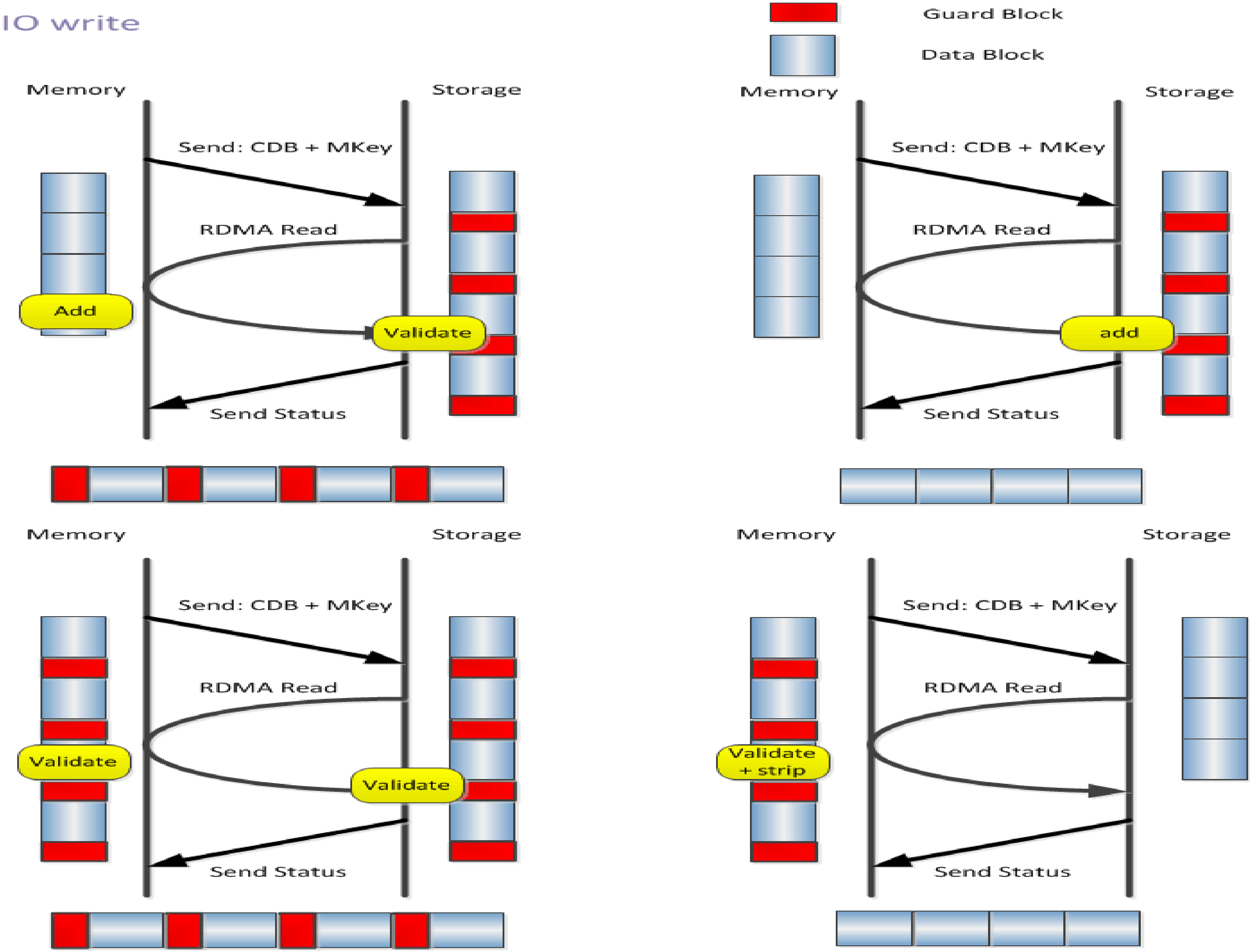
TYPICAL USE CASE



IO READ


IO Read
Guard Block
Data Block
Memory
Storage
Send: CDB + MKey
Validate
RDMA Write
Validate + Strip
H/W Ack
Send Status
Memory
Storage
Send: CDB + MKey
Validate + strip
RDMA Write
H/W Ack
Send Status
Memory
Storage
Send: CDB + MKey
Validate
RDMA Write
Validate
H/W Ack
Send Status
Memory
Storage
Send: CDB + MKey
RDMA Write
Add
H/W Ack
Send Status

IO WRITE

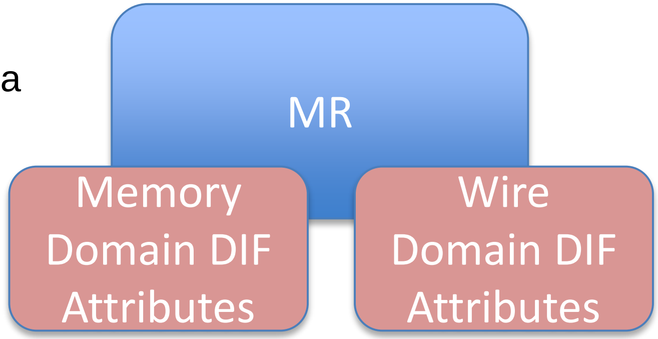

IO write
Guard Block
Data Block
Memory
Storage
Send: CDB + MKey
RDMA Read
Add
Validate
Send Status
Memory
Storage
Send: CDB + MKey
RDMA Read
add
Send Status
Memory
Storage
Send: CDB + MKey
RDMA Read
Validate
Validate
Send Status
Memory
Storage
Send: CDB + MKey
RDMA Read
Validate + strip
Send Status

VERBS

BASICS

- **T10-DIF is a property of MR (SIGMR)**
  - DIF calculations are needed when transferring data between wire and memory
  - Signature is property of the memory layout
  - Typically a single MR describes a single IO transaction
  - Well defined for passive read/write side
  - Allows single QP to serve protected and non protected transactions

- **Additional attributes to SIGMR describe**
  - How data is organized in memory
    - With/without DIF
    - DIF field is on separate buffer than data or interleaved with data
    - Block size, CRC/checksum, etc…
  - How is data organized in wire
    - With/without DIF
    - Must be interleaved
    - Block size, CRC/checksum, etc…



- **Performing IO operation using SIGMR will always go through Signature processing**
  - According to SIGMR attributes
  - No matter if access is local or remote
  - HW will add/strip/pass the DIF info and verify it if possible

T10-DIF ERROR

- **DIF error is NOT a transport error**
  - Data transfer will not fail and may complete successfully
  - Same QP may service different entities
    - QP can’t transfer to error state on Signature check error

- **Storage app should actively check for DIF error**
  - Inspect the SIGMR after transaction is finished
  - A lightweight operation (no device access)

- **Storage app should KNOW that transaction is finished**
  - Before checking DIF errors
  - Otherwise DIF error may happen later

IBV_SET_LAYOUT_SIGNATURE

```
int ibv_set_layout_signature(
     struct ibv_mr *mr, int flags,
     struct ibv_signature_attrs sig_attrs,
     struct ibv_sge *data, struct ibv_sge *sig);
```

- **sig_attrs**
  - Memory and wire DIF attributes
- **data**
  - A single SGE that points to the data buffer
  - sge.mr can result from previous set_layout_*() calls
    - Hence can describe sophisticated memory layouts
- **sig**
  - Needed in case DIF is not interleaved with data buffers
  - A single SGE that points to the data buffer
  - sge.mr can result from previous set_layout_*() calls
    - Hence can describe sophisticated layouts

SIGNATURE ATTRIBUTES

- **Two sets of DIF parameters**
  - Memory
  - Wire
- **Currently one type of DIF: T10-DIF**
- **T10-DIF parameters:**
  - bg_type: CRC or IPCHECKSUM
  - pi_interval: sector (block) size
  - bg: block guard seed
  - app_tag: application tag
  - ref_tag: first block LBA
  - check_mask: what to check, bit-per-byte of the T10-DIF 8 bytes
  - apptag_check_mask: what to check in apptag, bit-per-bit

```
struct ibv_signature_attrs {
          struct ibv_signature_domain mem;
          struct ibv_signature_domain wire;
      };

struct ibv_signature_domain {
          enum ibv_signature_type sig_type;
          union {
              struct ibv_t10dif_domain t10dif;
          };
      };

struct ibv_t10dif_domain {
          enum ibv_t10dif_bg_type bg_type;
          enum ibv_t10dif_flags flags;
          uint16_t    pi_interval;
          uint16_t    bg;
          uint16_t    app_tag;
          uint32_t    ref_tag;
          uint8_t     check_mask;
          uint16_t    apptag_check_mask;
      };
```

IBV_CHECK_MR_STATUS()

- **Check if there was T10-DIF error on the Signature MR**
- **If T10-DIF failed, get all needed info**
  - What has failed: GUARD, REFTAG or APPTAG
  - What were the expected and actual values
  - At which offset the error occurred
    - Will be block granularity…
  - The mkey of the data block

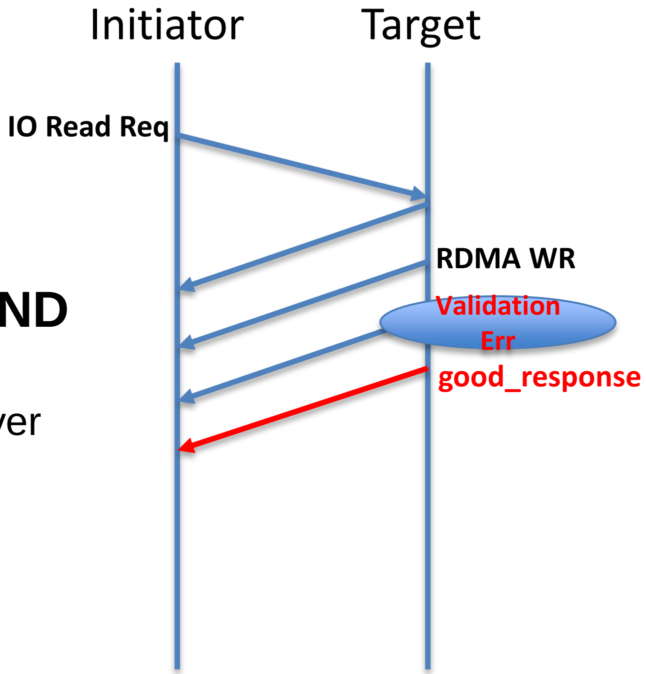
```
int ibv_check_mr_status(
        struct ibv_mr *mr,
        u32 check_mask,
        struct ibv_mr_status *mr_status)
```

```
struct ibv_mr_status {
      u32            fail_status;
      struct ibv_sig_err   sig_err;
};
```

```
struct ibv_sig_err {
      enum ibv_sig_err_type err_type;
      u32             expected;
      u32             actual;
      u64             sig_err_offset;
      u32             key;
};
```

PIPELINING ISSUE

- **ibv_check_mr_status() must happen at the end of transaction**
- **Storage systems used to respond to READ IO request by posting**
  - RDMA_WRITE(data)
  - SEND(good_status)
- **No wait needed between RDMA_WRITE and SEND**
  - Because nothing can go wrong unless transport error
  - Transport error kills QP, hence SEND(good_response) will never execute
- **With T10-DIF, can't do that anymore**
  - If RDMA_WRITE generates SIG ERROR, can't stop SEND(good_response) from happening – BUG!
  - Must wait for RDMA_WRITE to complete before posting SEND(good_response)



PIPELINING ISSUE RESOLVED

- **If SIG ERROR happens, QP revert to SQD state**
  - After RDMA_WRITE operation with bad signature check has finished, before SEND(good_status) started
- **Storage app will get notified (SQD async QP affiliated event)**
  - Now can check MR status and see the DIF error
- **Two new verbs**
  - ibv_get_current_wrid()
    - To get the next WRE of the SEND(good_response)
  - ibv_cancel_current_wr()
    - To cancel SEND(good_response) from happening
- **modify_qp(RTS)**
  - SEND(good_response) WQE was removed from the WQ
  - SEND(bad_response) at a later time when convenient

SUMMARY

- **End to end data integrity check is mandatory to avoid corruption**
- **It is way better to drop corrupted data than receive it**
- **Network storage systems are great use case for NIC integrity check offload**
- **Perfect fit for memory context attribute**

- **T10-DIF status and future work**
  - For kernel verbs, upstream already
  - For user verbs
    - Upstreaming to rdma-core - WIP
    - Older API supported since MLX_OFED 4.1 UR

OPENFABRICS ALLIANCE

14th ANNUAL WORKSHOP 2018

THANK YOU

Tzahi Oved, Oren Duer

Mellanox Technologies

[LOGO HERE]