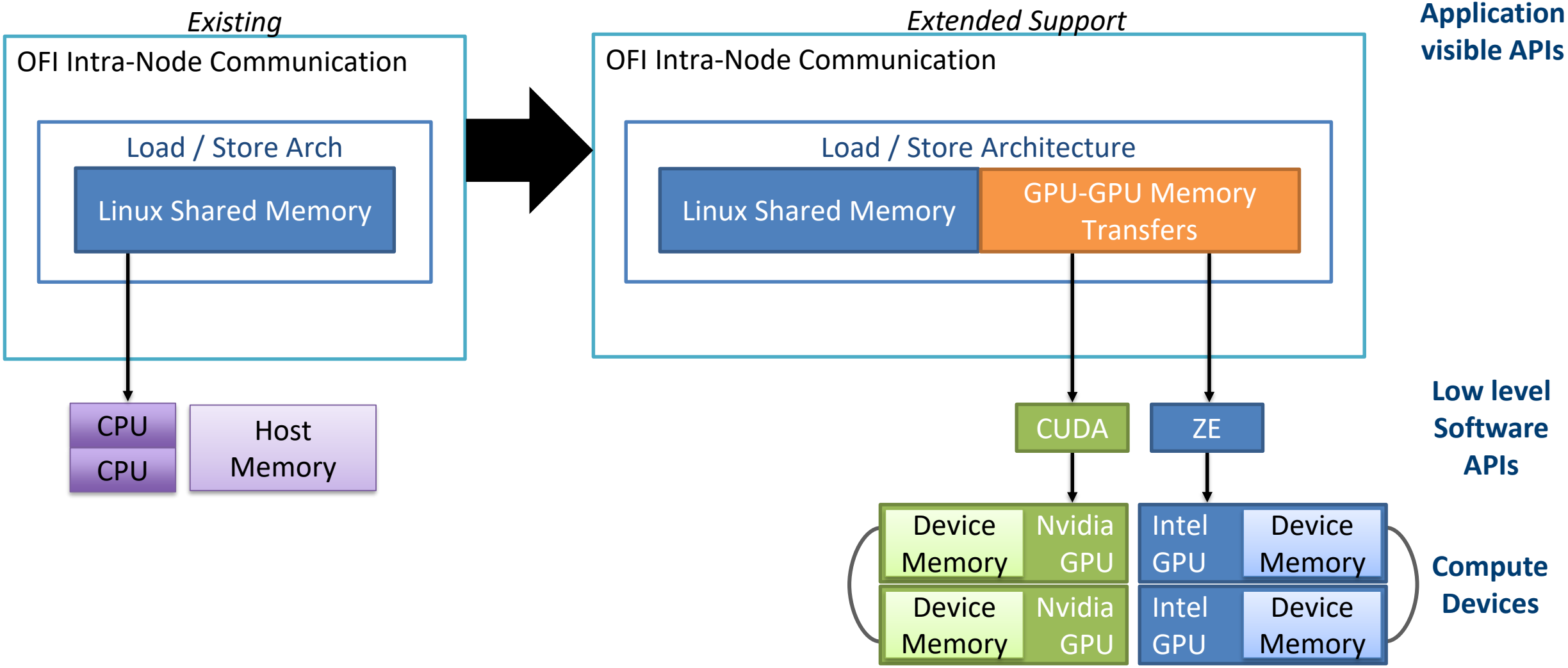
2020 OFA Virtual Workshop

LIBFABRIC INTRANODE DEVICE SUPPORT

Alexia Ingerson

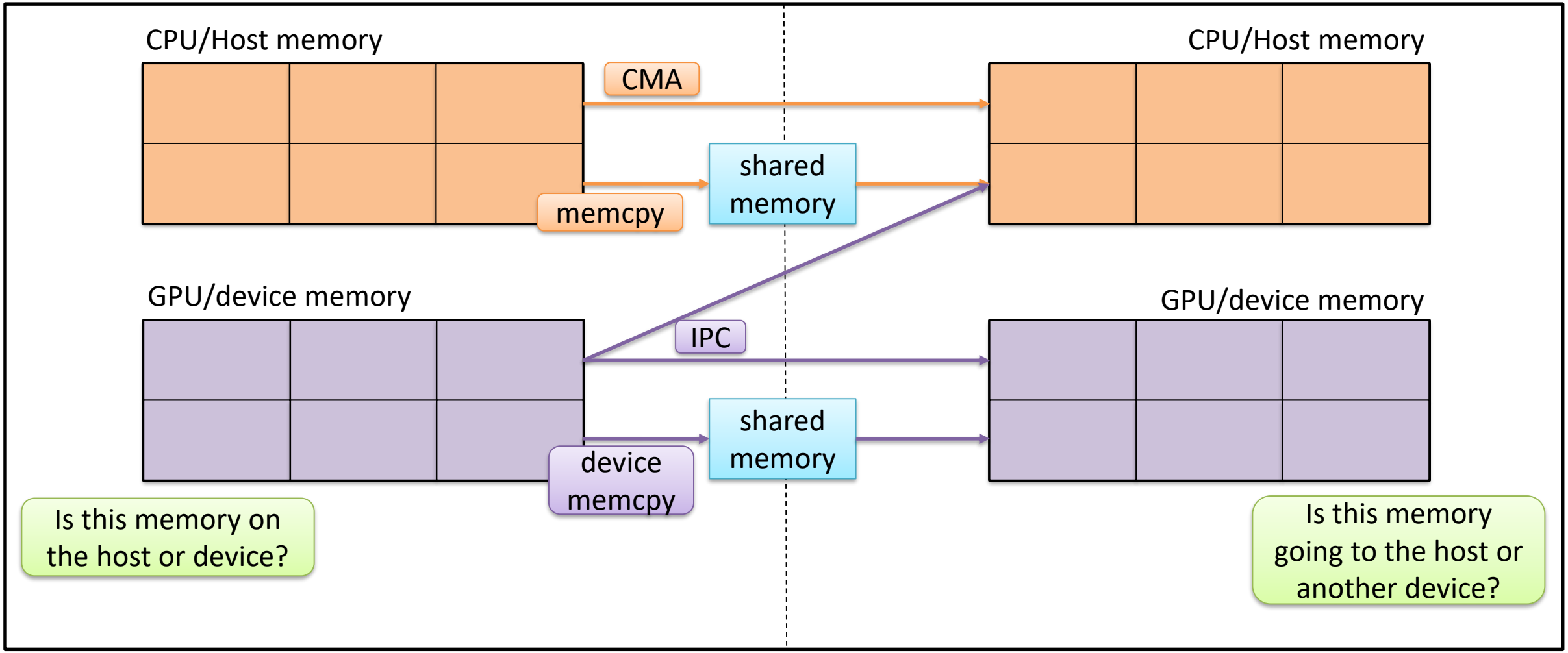
Intel Corp.

GPU-GPU COMMUNICATION OVERVIEW



© OpenFabrics Alliance

OVERVIEW



© OpenFabrics Alliance

OFI API CHANGES

- **info->caps: FI_HMEM**
  - Requests support for transfers to and from device memory
- **domain_attr->mr_mode: FI_MR_HMEM**
  - Specifies that the application should register all device memory with proper interfaces
  - Eliminates the need for a provider to query devices in order to determine memory location (expensive)
- **fi_mr_attr->iface**
  - Indicates software interface used to manage memory region

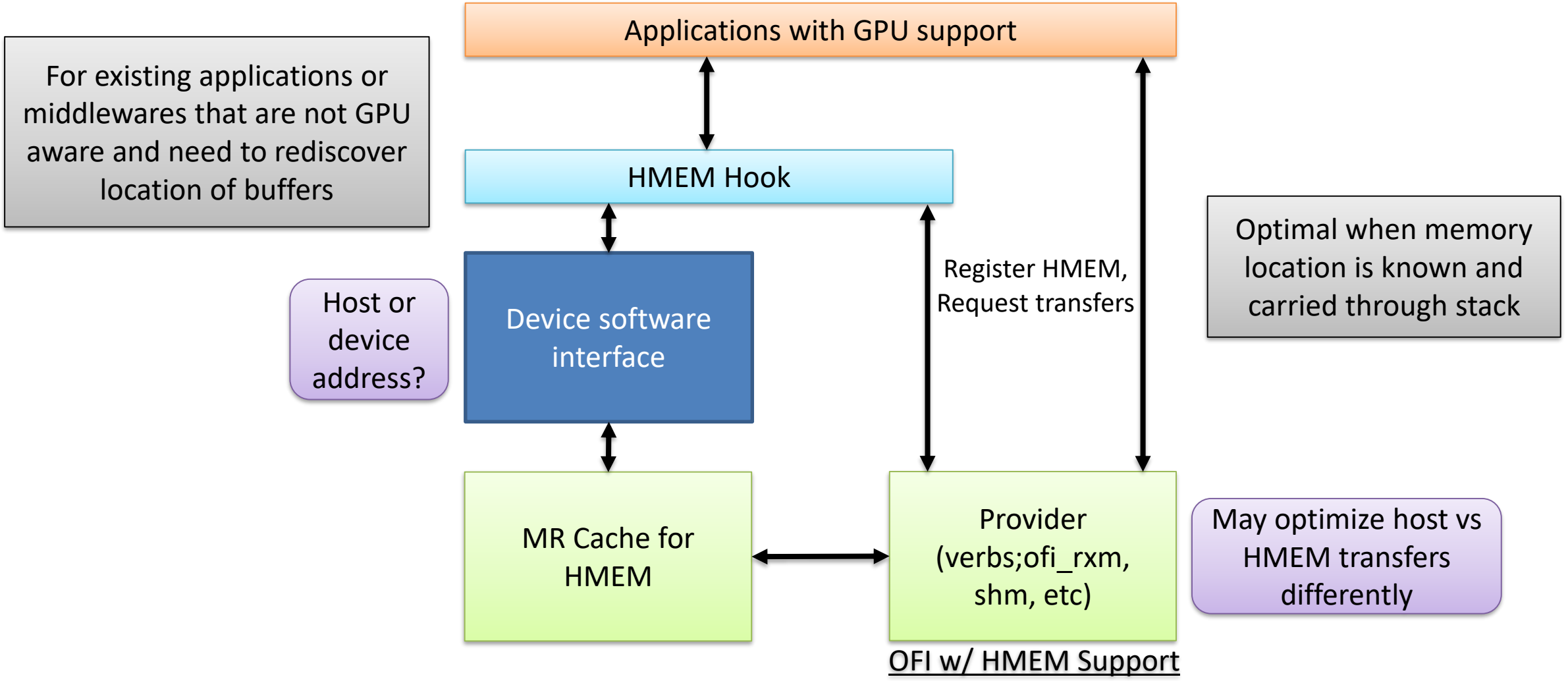
```
enum fi_hmem_iface {
        FI_HMEM_SYSTEM = 0, //system/host memory
        FI_HMEM_CUDA, //Nvidia/CUDA memory (libcuda)
        FI_HMEM_ZE, //Intel/Ze memory (libze_loader)
        …
}
```

  - Tells provider which API calls to use when copying to and from device
- **fi_mr_attr->device**
  - Device identifier for HMEM memory
  - Indicates on which device the memory is located on (type varies by interface) when multiple devices are present

```
fi_mr_attr {
    …
    enum fi_hmem_iface  iface;
    union {
        uint64_t        reserved;
        int             cuda;
        void            *ze;
    } device;
}
```

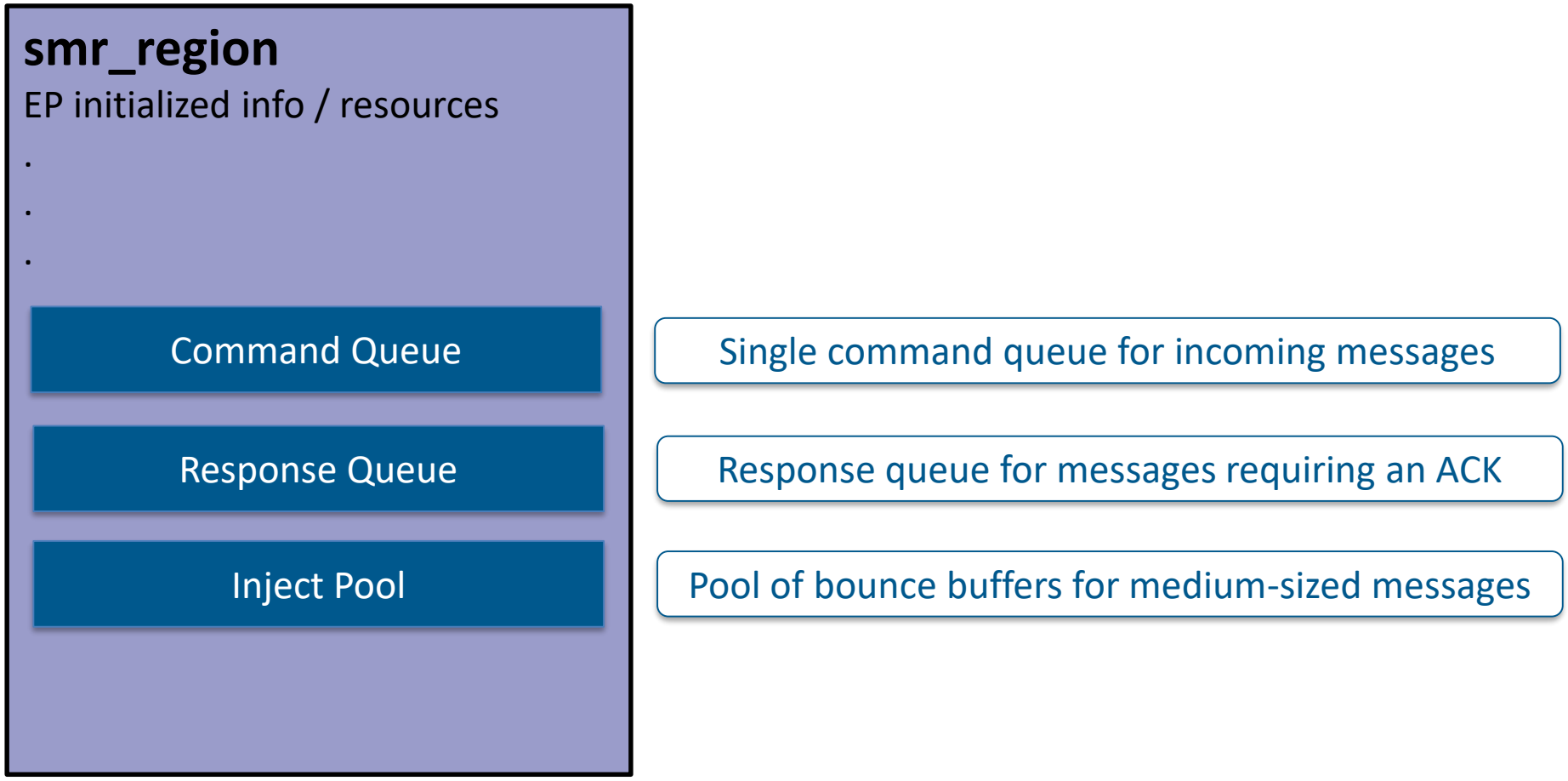

© OpenFabrics Alliance

HMEM HOOK



© OpenFabrics Alliance

SHM OVERVIEW



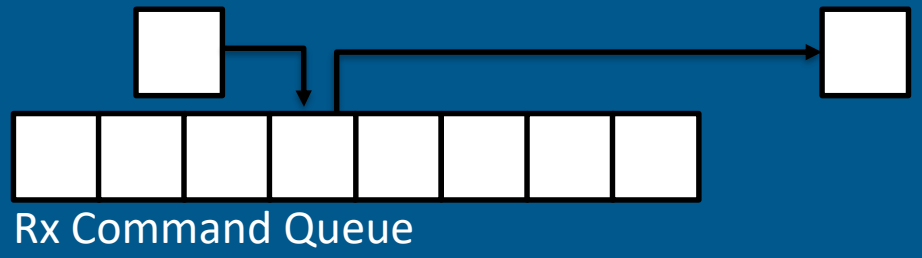
Single command queue for incoming messages

Response queue for messages requiring an ACK

Pool of bounce buffers for medium-sized messages

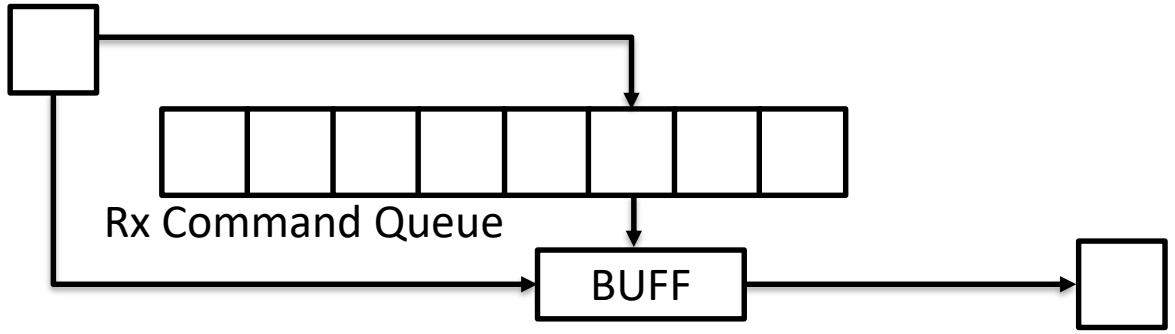
© OpenFabrics Alliance

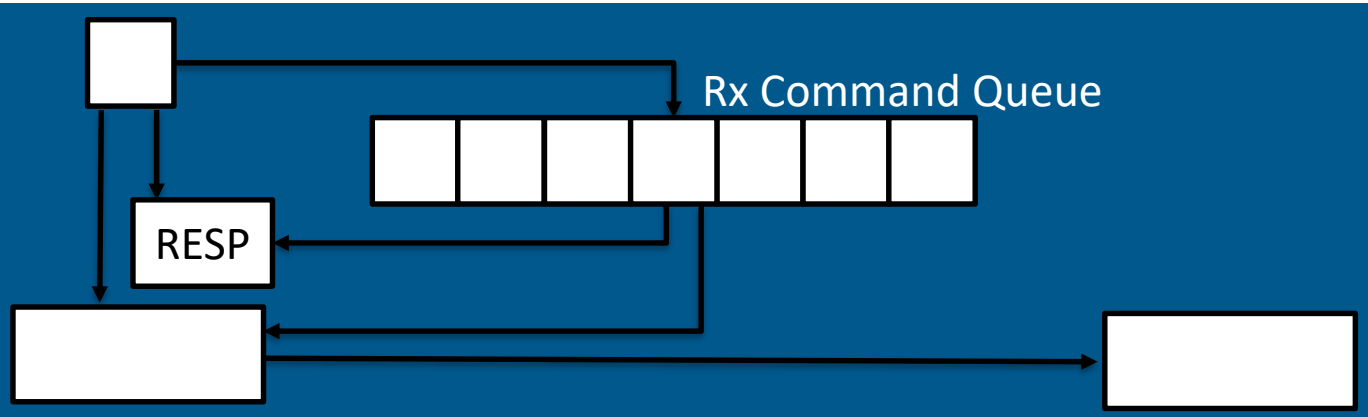
SHM PROTOCOLS



Inline

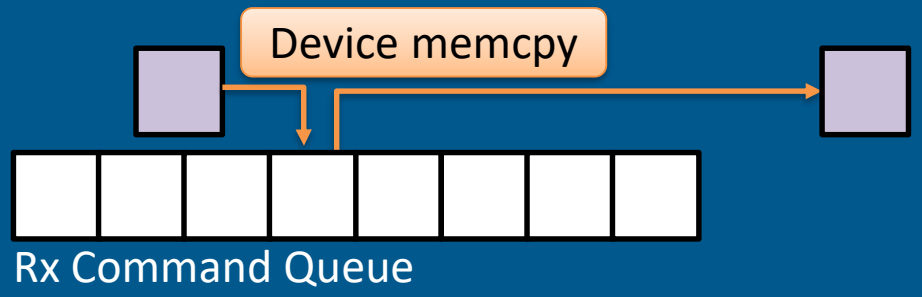
Inject





IOV

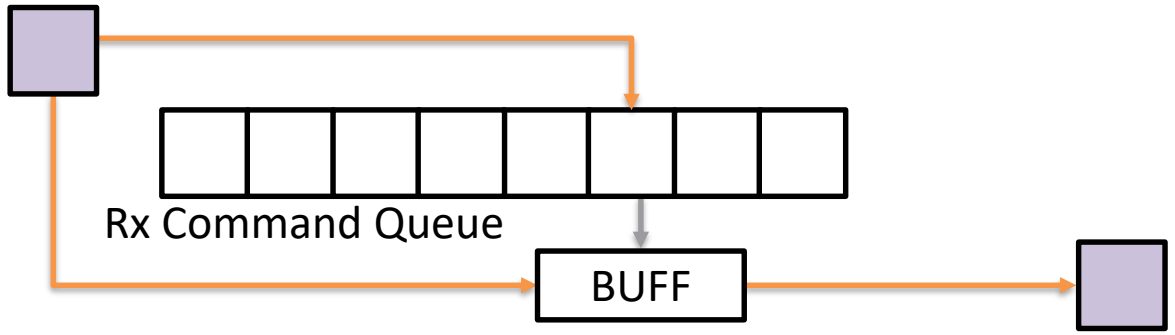
SHM PROTOCOLS

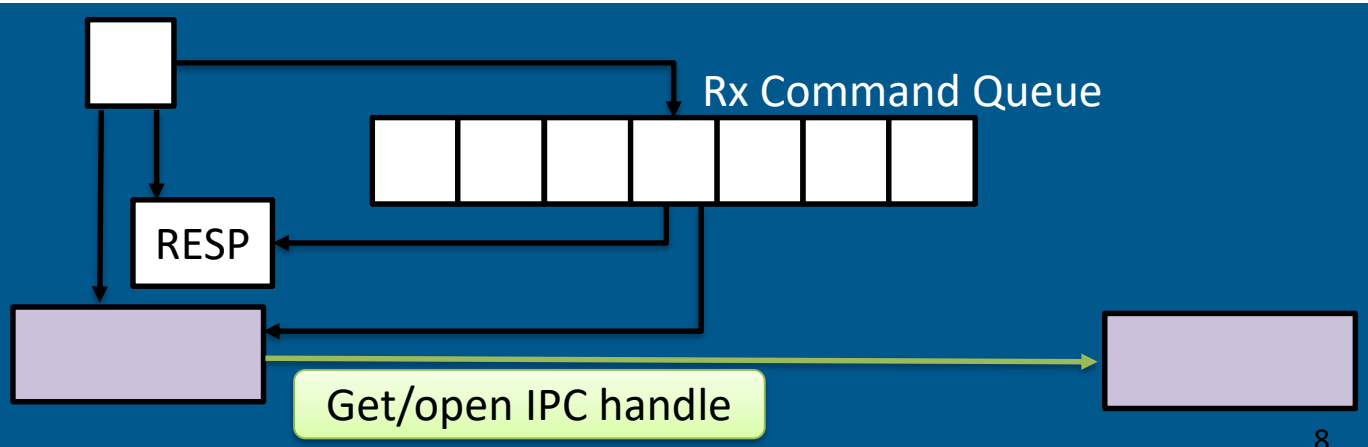


Inline

! Device->host/ host->device copies tend to be expensive

Inject





IOV

DEVICE BOUNCE BUFFERS

smr_region

EP initialized info / resources

.

.

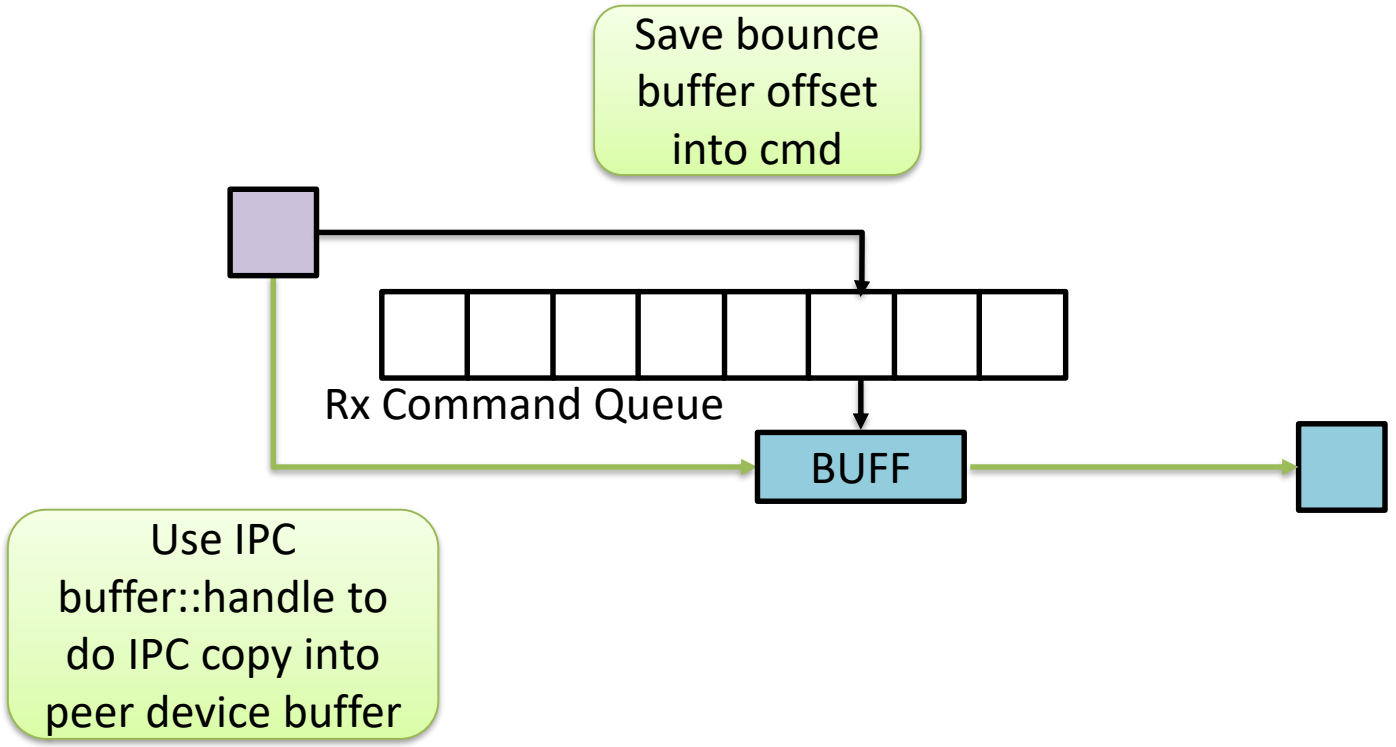
.

- Command Queue — Single command queue for incomings messages
- Response Queue — Response queue for messages requiring an ACK
- Inject Pool — Pool of bounce buffers for medium-sized messages
- IPC Handle Pool — Pool of device buffer IPC handles

```
smr_ipc_handle {
    void        *buf;
    uint8_t     handle[64];
}
```


© OpenFabrics Alliance

DEVICE BOUNCE BUFFER PROTOCOL



© OpenFabrics Alliance

GENERIC BUFFER OPS

Initialization/cleanup of any device-specific resource

Memory copy functions for copying to and from device memory

IPC handle functions for copying device memory across processes

Alloc and free calls for use in device bounce buffer protocol

```
struct ofi_hmem_ops {
    (*init)();
    (*cleanup)();
    (*copy_to_hmem)();
    (*copy_from_hmem)();
    (*get_handle)();
    (*open_handle)();
    (*close_handle)();
    (*alloc)();
    (*free)();
};
```


© OpenFabrics Alliance

SAMPLE HMEM OPS

	Intel GPU Level 0 API (Ze)	Nvidia GPU CUDA API

INIT	`zeInit()`
	`zeDriverGet()`
	`zeCommandQueueCreate()`
	`zeCommandListCreate()`
		`cuInit()`

COPY	`zeCommandListAppendMemoryCopy()` `zeCommandQueueExecuteCommandLists()`	`cudaMemcpy()`

IPC	`zeDriverGetMemIpcHandle()`
	`zeDriverOpenMemIpcHandle()`
	`zeDriverCloseMemIpcHandle()`
		`cudaIpcGetMemHandle()`
		`cudaIpcOpenHandle()`
		`cudaIpcCloseMemHandle()`

MEM	`zeDriverAllocSharedMem()` `zeDriverFreeMem()`	`cudaMalloc()` `cudaFree()`

© OpenFabrics Alliance

2020 OFA Virtual Workshop

THANK YOU

Alexia Ingerson

Intel Corp.