2025 OFA Webinar Series

CODING TO LIBFABRIC

Alexia Ingerson

Intel Corporation

MEET THE PRESENTER: Alexia Ingerson, Intel

Alexia Ingerson is a Senior Software Engineer at Intel and a maintainer of libfabric working on HPC and AI software stacks, including MPIs and CCLs. She primarily focuses on optimized intranode communication, including leveraging accelerators and use of local offload in collectives.

LIBFABRIC OVERVIEW

MPI | SHMEM | Libfabric-enabled Applications | DAOS | CCL

Libfabric API

Control
- Discovery

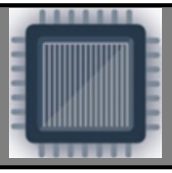
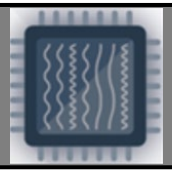
Communication
- Connection Management
- Address Vectors

Completion
- Event Queues
- Completion Queues
- Counters

Data Transfer
- Messaging
- Atomics
- RMA
- Collectives
- Tag Matching

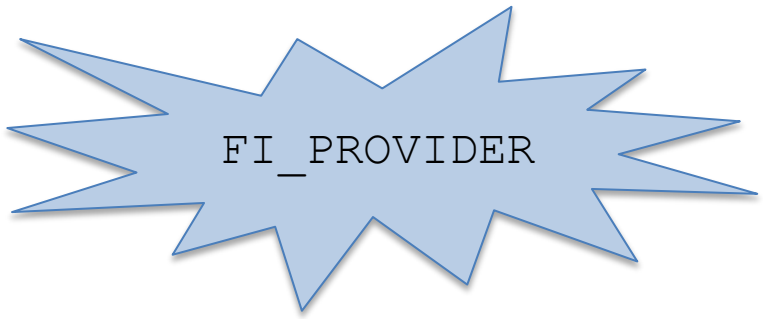
OFI provider

- Verbs (InfiniBand)
- EFA (AWS)
- TCP (sockets)
- SHM (shared memory)
- OPX (Cornelis Omni-Path)
- CXI (Cray Slingshot)
- UCX

NIC | GPU  | CPU 

QUERYING OFI

```
int fi_getinfo(uint32_t version,
               const char *node,
               const char *service,
               uint64_t flags,
               const struct fi_info *hints,
               struct fi_info **info)
```



OFI API version
`FI_VERSION(major, minor)`

Node name/IP address (string)

Port number (string)

Input flags

Requested capabilities/settings

Returned list of available providers

QUERYING OFI

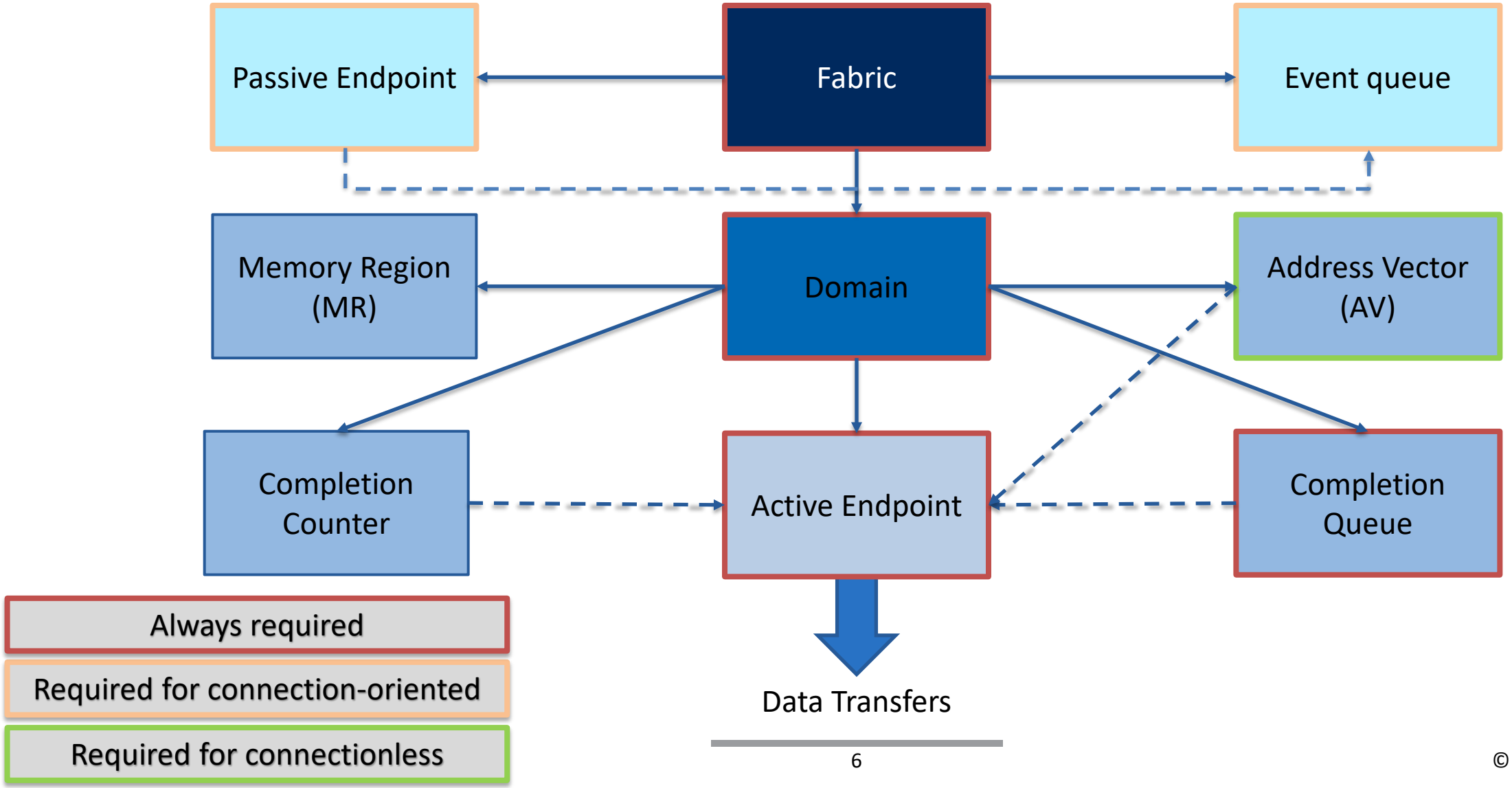
caps: what the **app** requires from the **provider**

mode: what the **provider** requires from the **app**

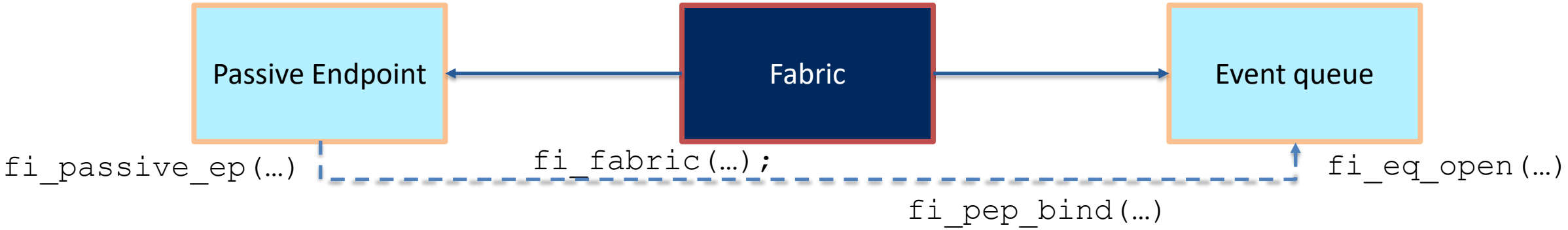
```
const struct fi_info *hints,
```

```
uint64_t                caps;
uint64_t                mode;
uint32_t                addr_format;
void                    *src_addr;
void                    *dest_addr;
struct fi_tx_attr       *tx_attr;
struct fi_rx_attr       *rx_attr;
struct fi_ep_attr       *ep_attr;
struct fi_domain_attr   *domain_attr;
struct fi_fabric_attr   *fabric_attr;
struct fid_nic          *nic;
```

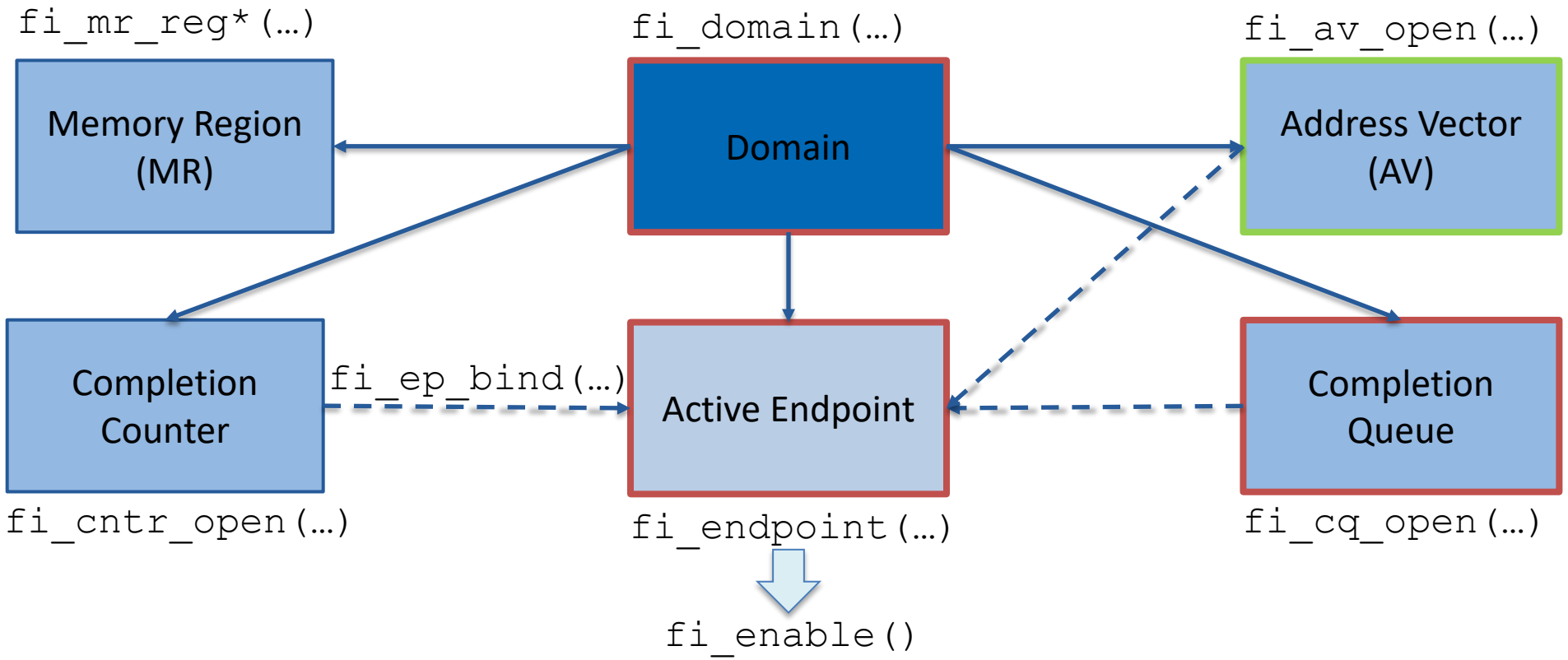
OFI RESOURCES



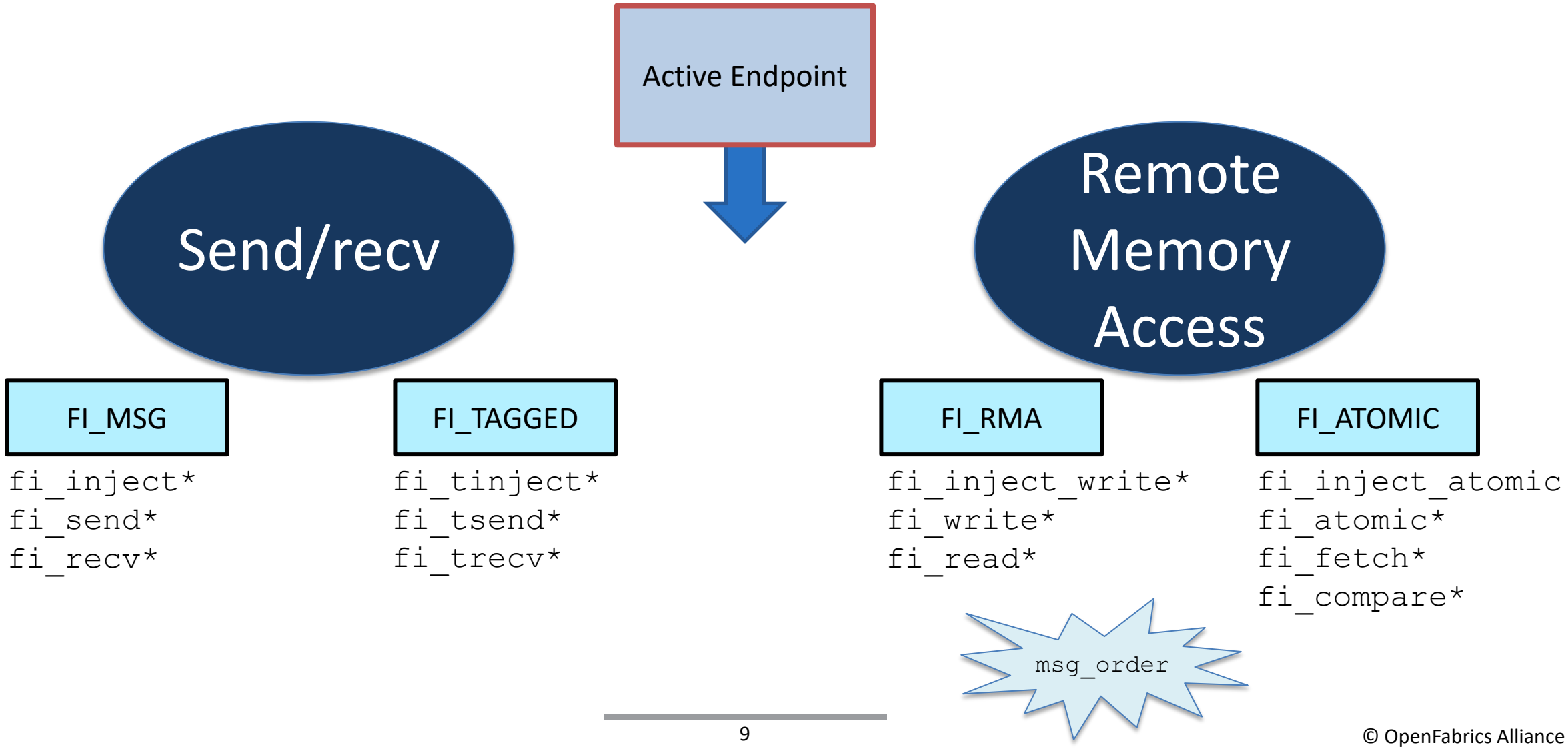
OFI RESOURCES



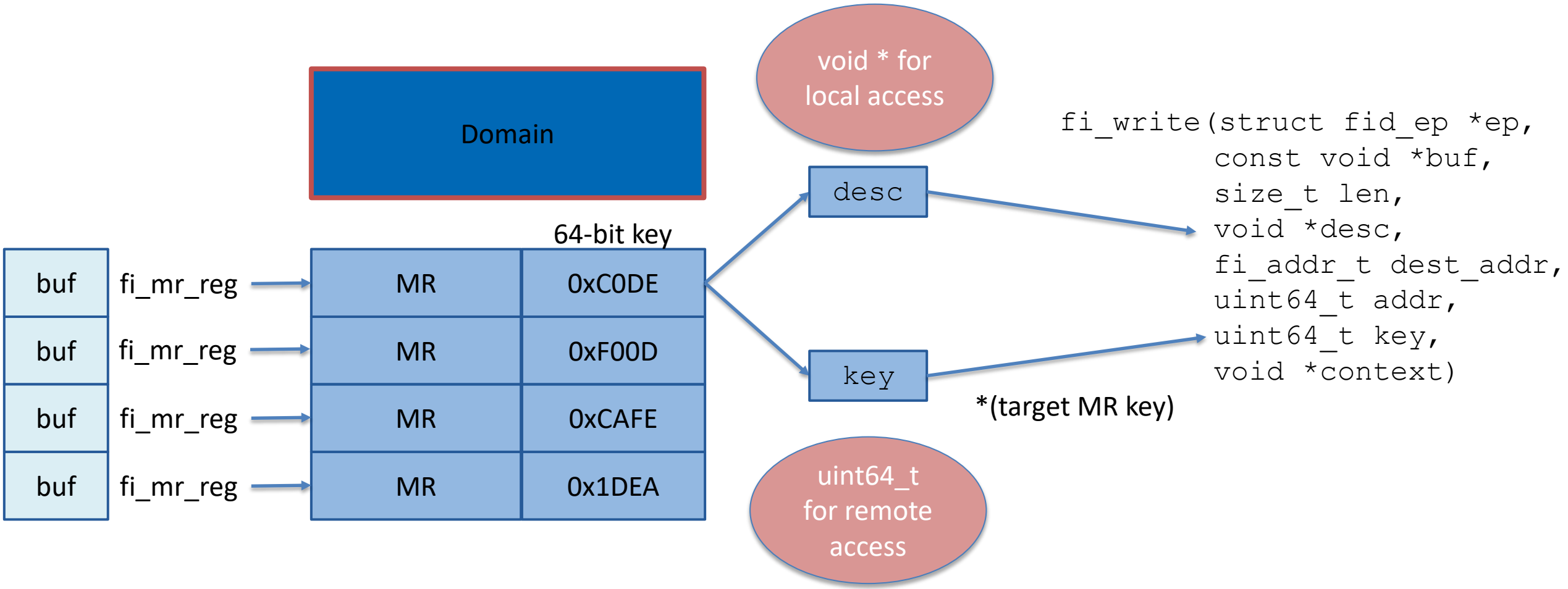
OFI RESOURCES



DATA TRANSFER INTERFACES



MEMORY REGISTRATION (THE BASICS)



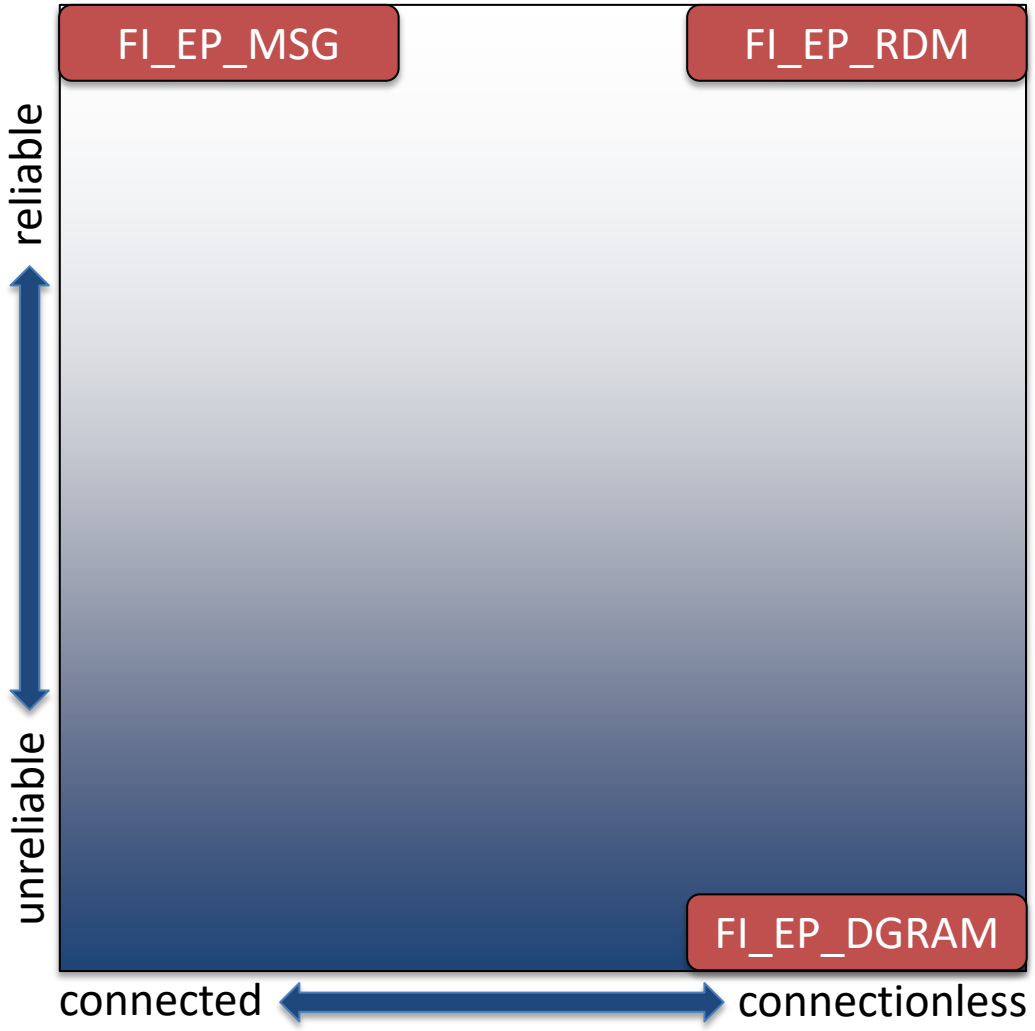
MEMORY REGISTRATION (ADVANCED)

MR mode	What the app must do/support
FI_MR_LOCAL	Register local buffers
FI_MR_RAW	MR keys larger than 64 bits
FI_MR_VIRT_ADDR	Use RMA target virtual addresses (instead of offsets)
FI_MR_ALLOCATED	Backed by physical pages at the time of registration, physical page must not change
FI_MR_PROV_KEY	MR key chosen by provider
FI_MR_ENDPOINT	MRs associated by EP, not domain; bind to EP
FI_MR_HMEM	Register all memory allocated by HMEM (GPU) interfaces

CONNECTION MANAGEMENT PROTOCOLS

```
hints->ep_attr->type
```

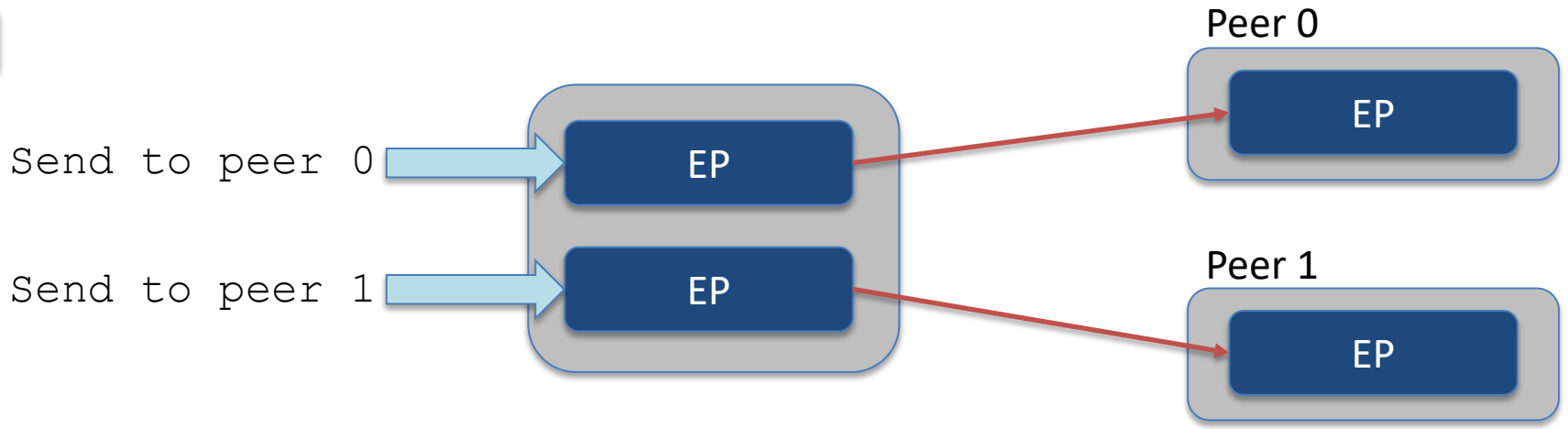
```
fi_listen(…)
fi_connect(…)
fi_accept(…)
fi_reject(…)
```



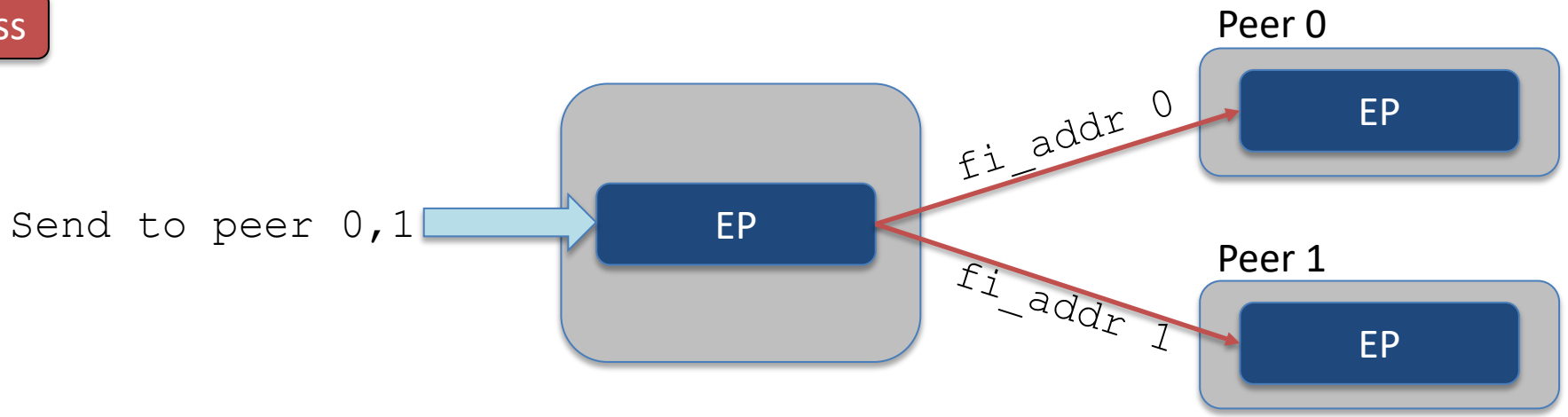

```
fi_av_insert(…)
    ->in: addr
    ->out: fi_addr
```

CONNECTION MANAGEMENT PROTOCOLS

connected



connectionless



POSTING DATA TRANSFERS

```
fi_sendmsg(struct fid_ep *ep, const struct fi_msg *msg, uint64_t flags);
```

```
fi_recvmsg(struct fid_ep *ep, const struct fi_msg *msg, uint64_t flags);
```

```
struct fi_msg {
    const struct iovec   *msg_iov;
    void                 **desc;
    size_t               iov_count;
    fi_addr_t            addr;
    void                 *context;
    uint64_t             data;
};
```

Data buffer(s) to send or receive into

MR descriptors for send/receive buffers (only if specified in mr_mode)

Number of elements in msg_iov AND desc

Peer to send to (only required for FI_EP_RDM and FI_EP_DGRAM)

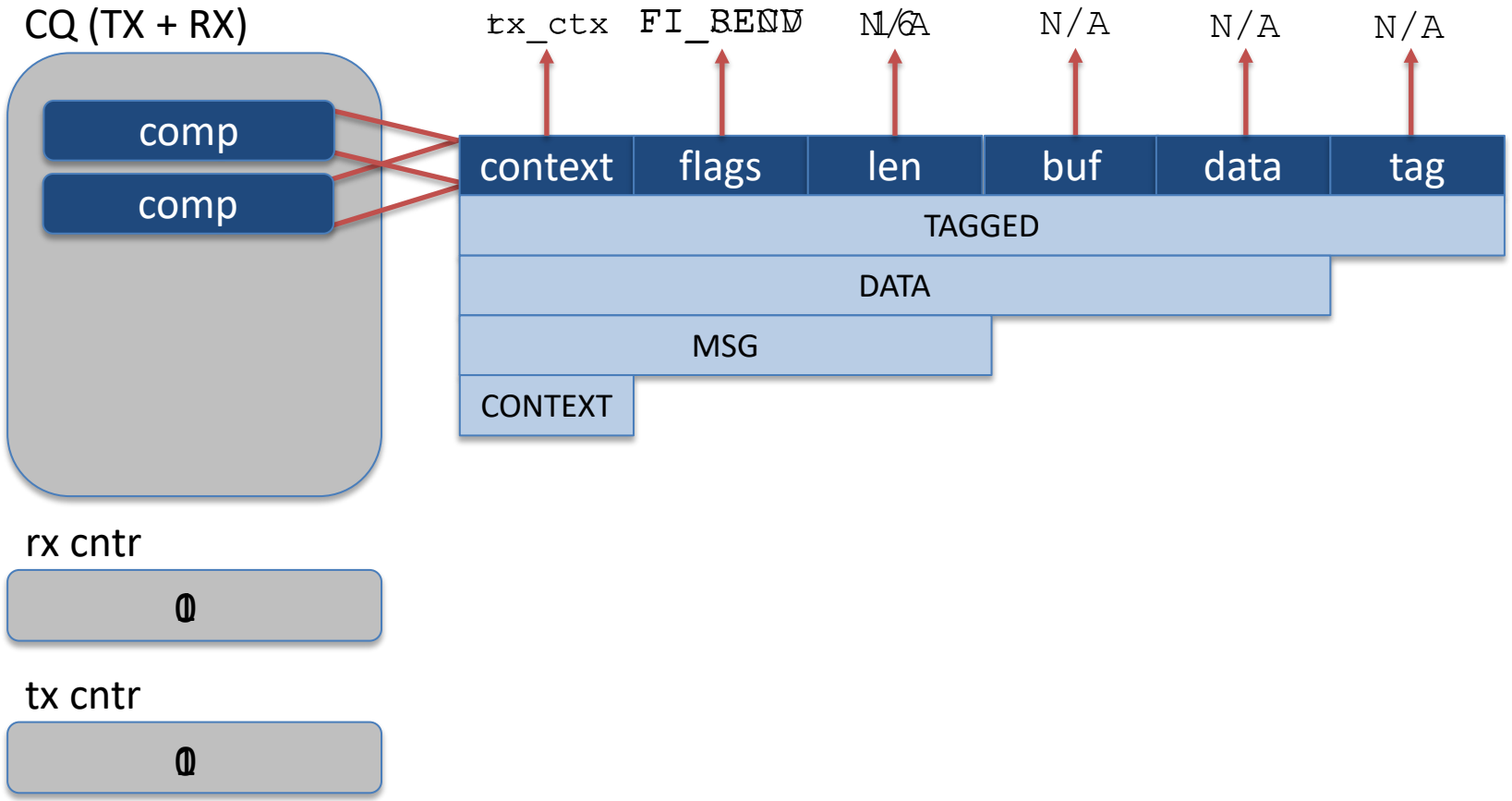
Context for tracking operation. Required when using FI_CONTEXT(2) mode. Returned in CQ completion entry

Optional CQ data to send (only for sends; set flag FI_REMOTE_CQ_DATA)

COMPLETIONS

```
fi_send(ep, tx_buf,
        16, tx_desc,
        dest_addr,
        &tx_ctx)
```

```
fi_recv(ep, rx_buf,
        128, rx_desc,
        src_addr,
        &rx_ctx)
```

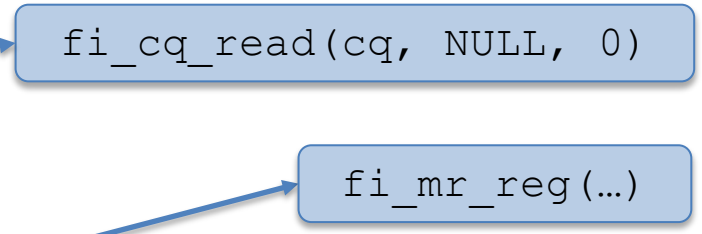


COMPLETION CONSIDERATIONS

- **Things that could affect completion flow:**
  - Completion semantics (potential performance impact at higher levels)
    - FI_INJECT_COMPLETE – buffer can be reused
    - FI_TRANSMIT_COMPLETE – data is on the wire
    - FI_DELIVERY_COMPLETE – data has been delivered to the target (not necessarily in target location)
  - “Unexpected” messaging
    - No matching receive buffer for incoming message
  - API usage that suppress completions
    - fi_inject* API calls (limited by `info->tx_attr->inject_size`)
    - FI_SELECTIVE_COMPLETION / FI_COMPLETION flags

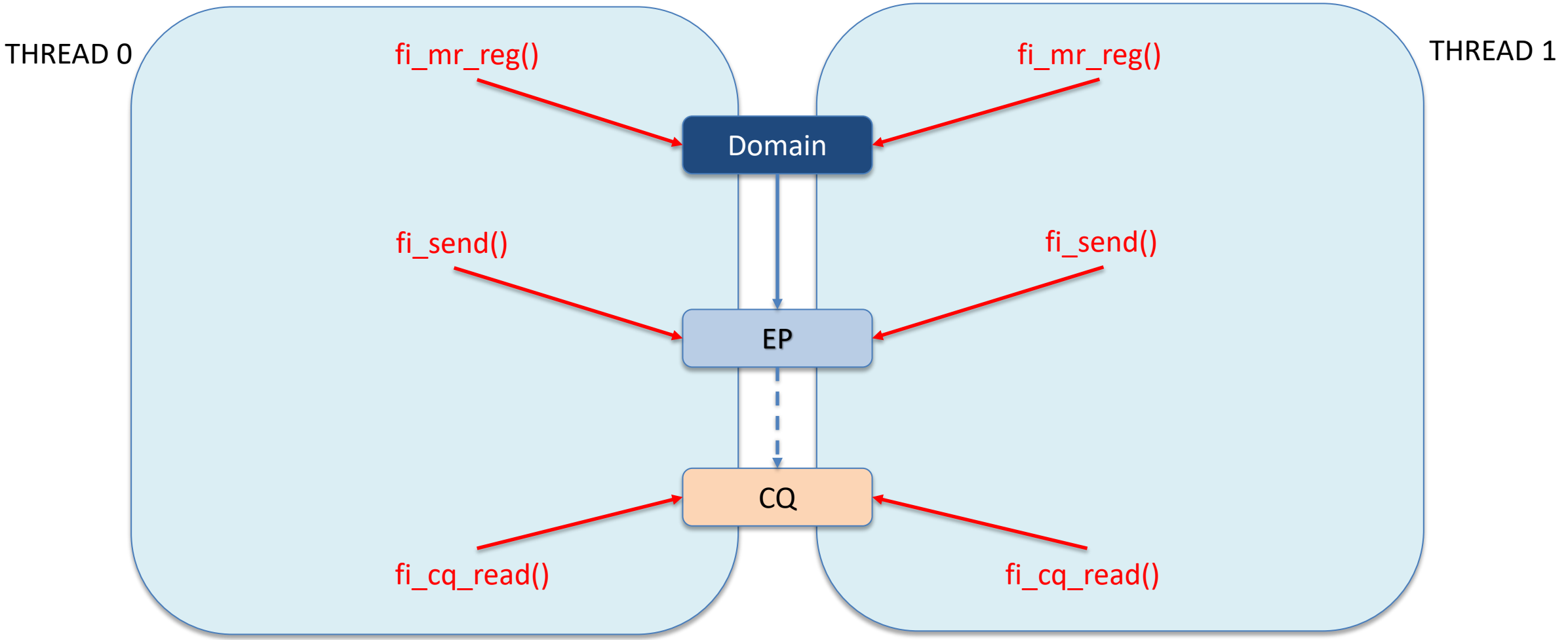
PROGRESS

- **info->domain_attr->control_progress / data_progress**
  - FI_PROGRESS_AUTO
    - provider progresses data automatically without requirement from application
  - FI_PROGRESS_MANUAL
    - provider requires application to drive progress for all data operations
  - FI_PROGRESS_CONTROL_UNIFIED
    - provider requires application to drive progress for all data AND serialize control operations
- **Not all providers support AUTO progress, requires more locking for progress thread**
- **Selecting FI_PROGRESS_CONTROL_UNIFIED may be able to reduce most locking**



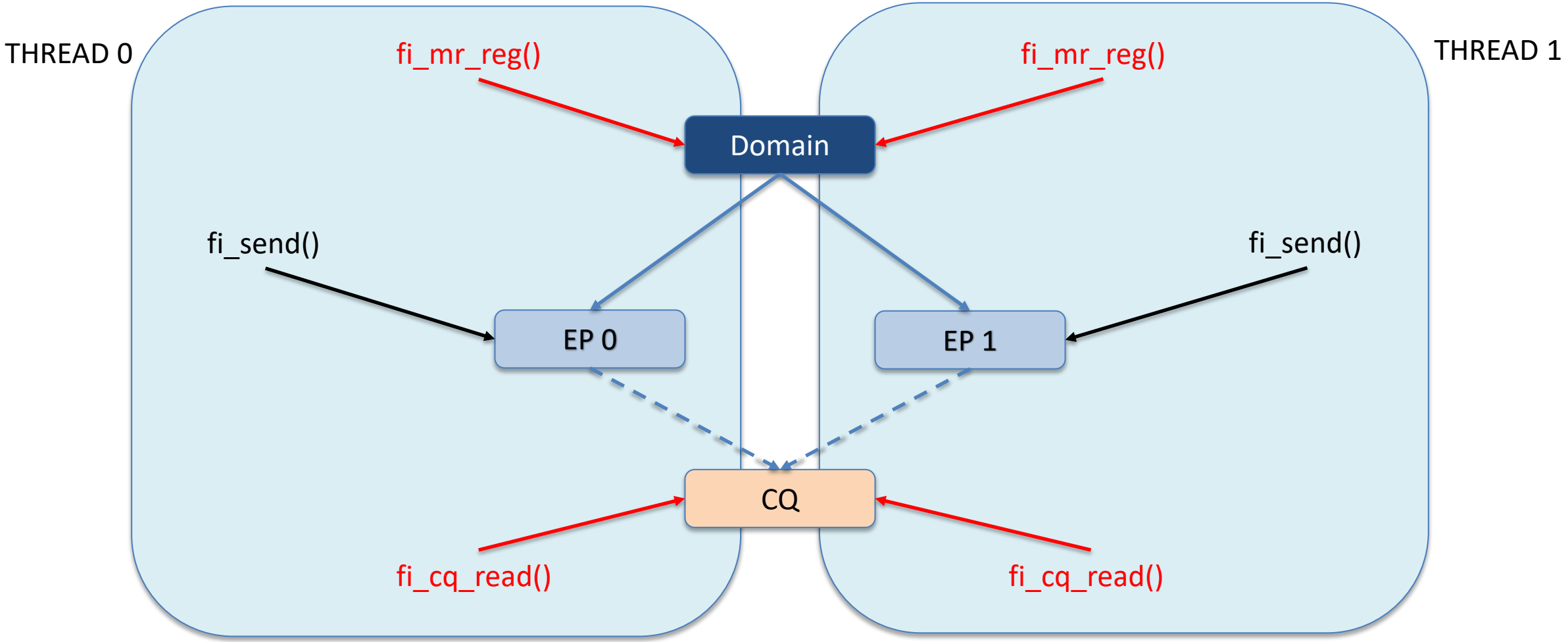

```
struct fi_domain_attr {
        enum fi_progress   control_progress;
        union {
                enum fi_progress data_progress;
                enum fi_progress progress;
        };
```

THREADING



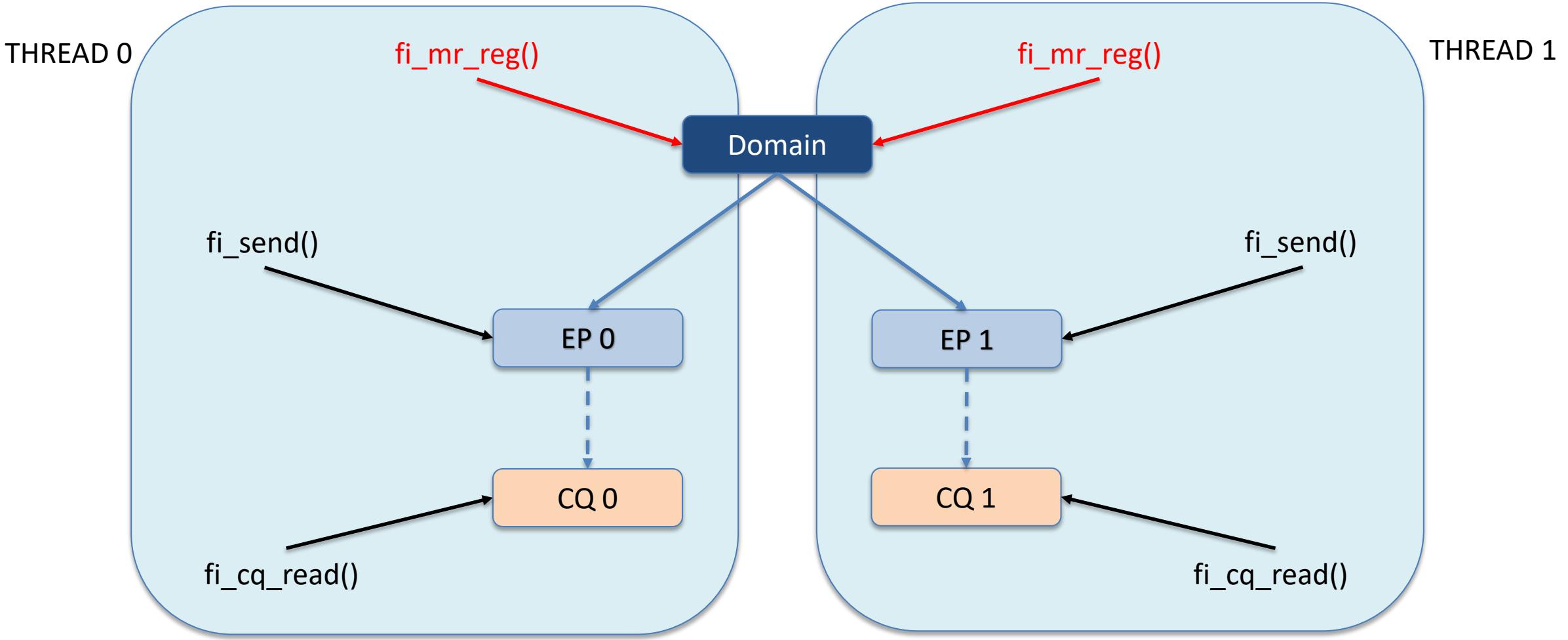
FI_THREAD_SAFE

THREADING



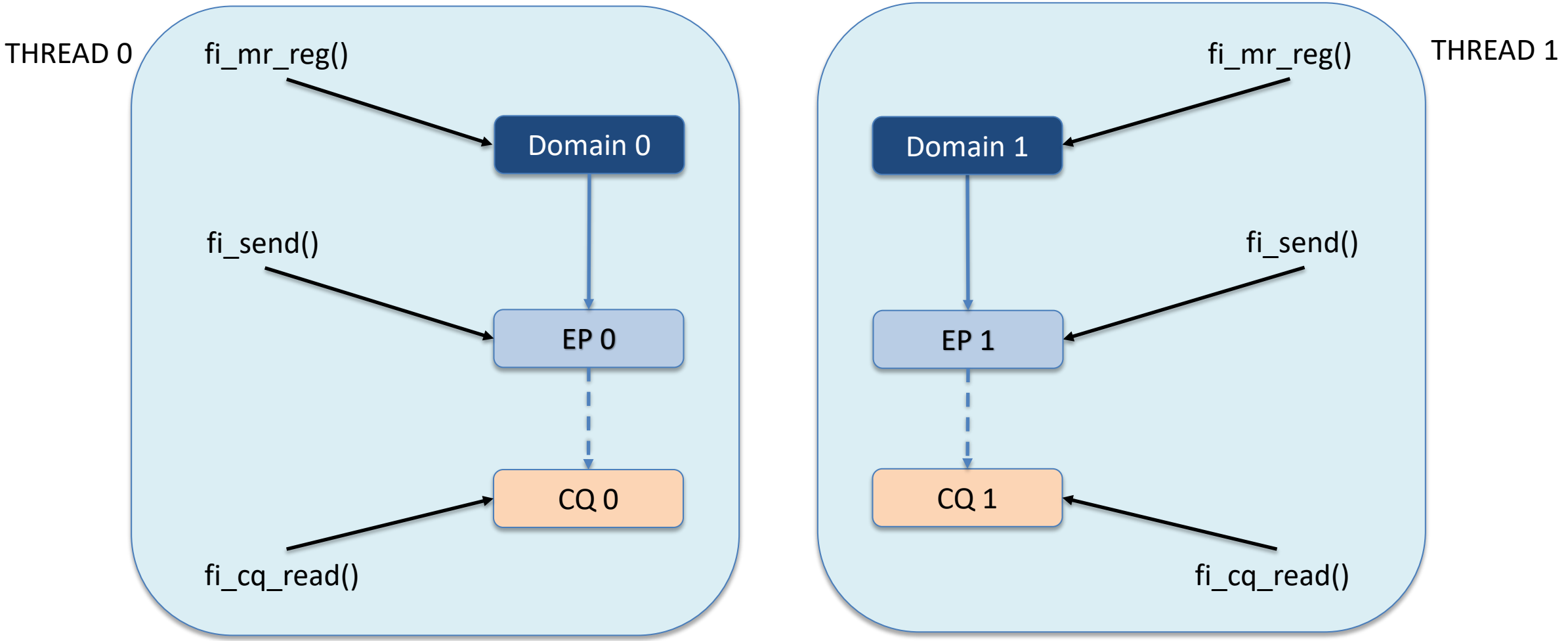
FI_THREAD_SAFE
*FI_THREAD_ENDPOINT (deprecated)

THREADING



FI_THREAD_COMPLETION

THREADING



FI_THREAD_DOMAIN

TAKEAWAYS

- **What hardware are you targeting?**
  - Is there a limitation or optimization? If so, make sure to support that use case or be able to leverage that optimization
- **Keep required capabilities as simple as possible to maximize provider usage**
- **Support as many modes as possible to maximize provider usage**
  - And make sure you actually support them when needed!
- **Call fi_getinfo with NULL hints to return providers' most optimized sets of capabilities and modes**
- **If using threads, recommended to use FI_THREAD_DOMAIN and FI_PROGRESS_CONTROL_UNIFIED (FI_PROGRESS_MANUAL)**

2025 OFA Webinar Series

THANK YOU

Alexia Ingerson

Intel Corporation