ALFA\textsuperscript{[1]} is a modern C++ software framework for simulation, reconstruction and analysis of particle physics experiments. ALFA extends FairRoot\textsuperscript{[2]} to provide building blocks for highly parallelized and data flow driven processing pipelines required by the next generation of experiments, such as the upgraded ALICE detector or the FAIR experiments.

FairMQ\textsuperscript{[3]} is a C++ Message Queuing Framework that integrates standard industry data transport technologies and provides building blocks for simple creation of data flow actors and pipelines. FairMQ hides transport details behind an abstract interface and ensures best utilization of the underlying transports (zero-copy, high throughput). The framework does not impose any format on the messages.

### Concepts

Shared memory transport follows the core FairMQ concepts:

**General concepts:**
- Hide all transport-specific details from the user.
- Clean, unified interface to different data transports.
- Combinations of different transport in one device in a transparent way.
- Transport dispatch configuration only, without modifying device/user code -> same API for all transports.

**Ownership concepts:**
- Message owns data.
- Sender device (user code) passes ownership of data to framework with send call.
- Framework transfers to next device, passes ownership to receiver (no physical copy of the data with shared memory transport).
- No sharing of ownership between different devices – if the same message is needed by more than one receiver it is copied.

### Unmanaged Region

The default message creation in FairMQ hides all the memory allocation/management details from the user and simply provides a ready to use buffer for every message. However, sometimes user might have very specific memory layout requirements – typically where hardware needs to write to that memory (e.g. detector readout). Ideally this memory should still be usable with no/minimal copy by further devices in the pipeline.

- UnmanagedRegion component for full memory layout control.
- Zero-copy for shared memory transport.
- Allocates memory via the transport allocator and provides it to the user to manage.
- Message creation out of subset(s) of this region.
- Framework cleans only the entire region, not separate messages – this is in user hands.
- Callback system to notify creator when a buffer is no longer needed by transport.

### Monitoring

Because shared memory segments can outlive the process, it is important to make sure no unused shared memory is left in case devices crash and fail to cleanup used resources.

- Automatic cleanup of shared memory resources, even if all devices crash.
- Monitoring tool to debug/monitor shared memory use.

![Shared memory implementation](image)

### Implementation

- **boost:interprocess\textsuperscript{[4]}** library for management and allocation of shared memory - cross-platform shared memory implementation with many features such as different allocation algorithms, shmrm STL-like containers, shmrm smart pointers, message queues and many more.
- **ZeroMQ library** for transfer of the meta information associated with the memory.

The FairMQ message object holds meta information (handle + size) about the underlying shared memory. This meta information is transferred to other devices on the host via ZeroMQ\textsuperscript{[5]}.

### Performance

**Transfer Rate:**
Allocation of messages with shmem+ZeroMQ transport and transfer to a second device.

**CPU Usage:**
CPU usage measured at ~2.5 GB/s rate (one core) (1MB message size)

![CPU Usage](image)

**Run the tests:**

1. **Allocate/Transfer:** Prepare necessary configuration files and transport connections between the devices
2. **Notification:** Use of ZeroMQ notify to inform sender of a message transfer completion
3. **Receiver:** Use/specialize zeroMQ library for receiving messages
4. **Test:** Use device-specific output files in any convenient way

References:
- [2] https://github.com/FairRootGroup/FairRoot
- [3] https://github.com/FairRootGroup/FairMQ
- [4] [ZeroMQ](https://zeromq.org/)
- [5] [ALFA](https://github.com/ALFA-Routing)