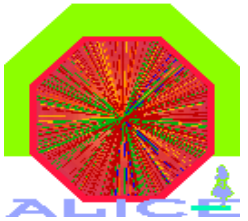


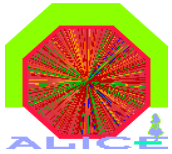
TPC Online Monitor and PubSub Debugging tools

Matthias Richter[1], T. Steinbeck[2]

[1]Department of Physics and Technology, University of Bergen, Norway
[2]Kirchhoff Institute of Physics, University of Heidelberg, Germany

ALICE HLT/ Offline workshop, CERN Dec. 6th - 8th

Motivation



How to implement an online display and how to get data out of the online framework and investigate it in an easy way?

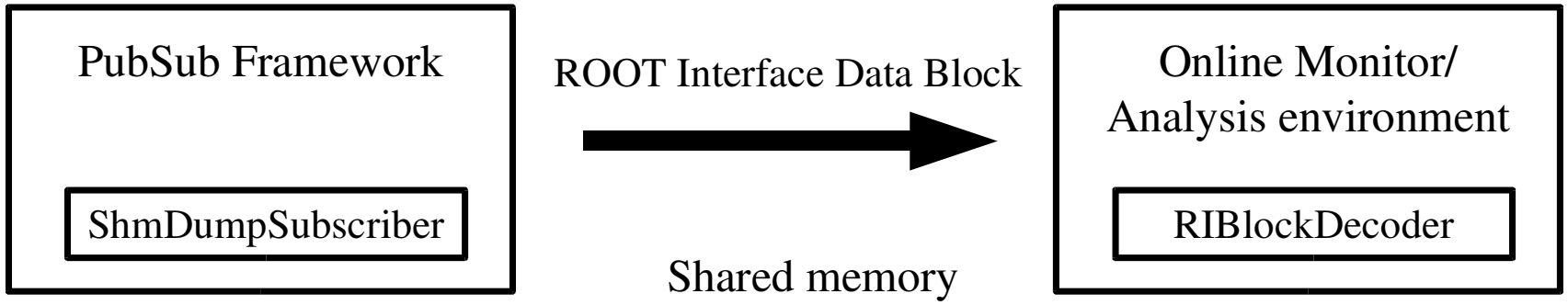
Internal: OM as Subscriber

- ✔ part of the online analysis chain
- ✗ might effect work of HLT
- ✗ little flexibility in terms of debugging tool
- ✗ difficult to handle for 'outside' users (detector groups)

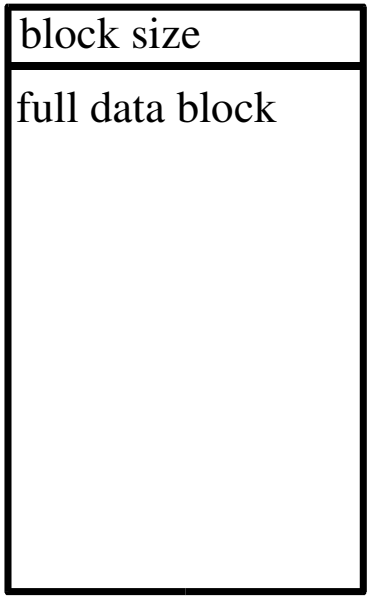
Extern: data exchange via shm

- ✔ Display and debugging tools completely separated from PubSub framework
- ✔ can use any environment to investigate the data, e.g. (Ali)ROOT interactive session
- ✔ no effects to work of HLT
- ✔ can use all features of AliROOT
- ✗ needs to handle all effects imposed by the change of program space and/or architecture

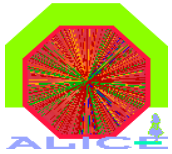
Working Scheme



- collect incoming data for one event
- prepare data block and write to shared memory
- first word of Shm reserved for blocksize – signalize valid data block
- wait for blocksize to become zero



- wait for blocksize > 0
- process data block
- reset blocksize



Data exchange format

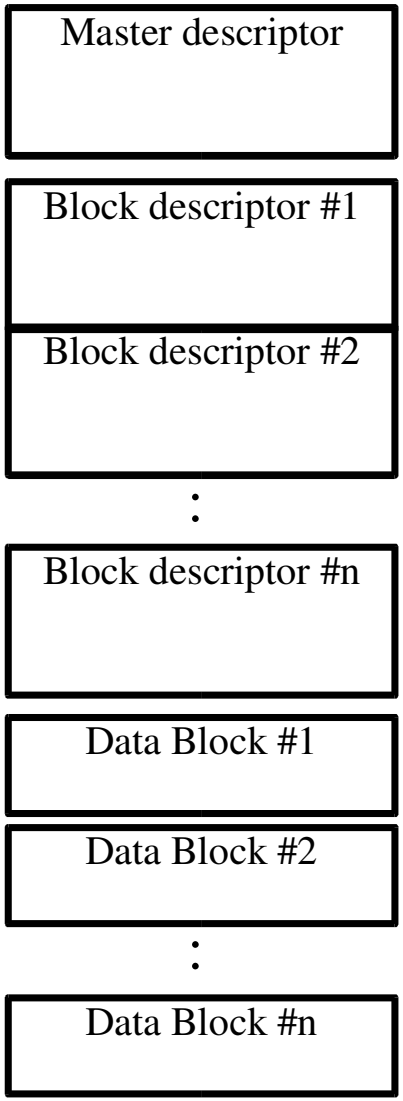
Root Interface Block Descriptor

AliHLTRIBlockDescriptor

- data structure used for data exchange
- 64 bit aligned
- applicable for header and data block descriptors

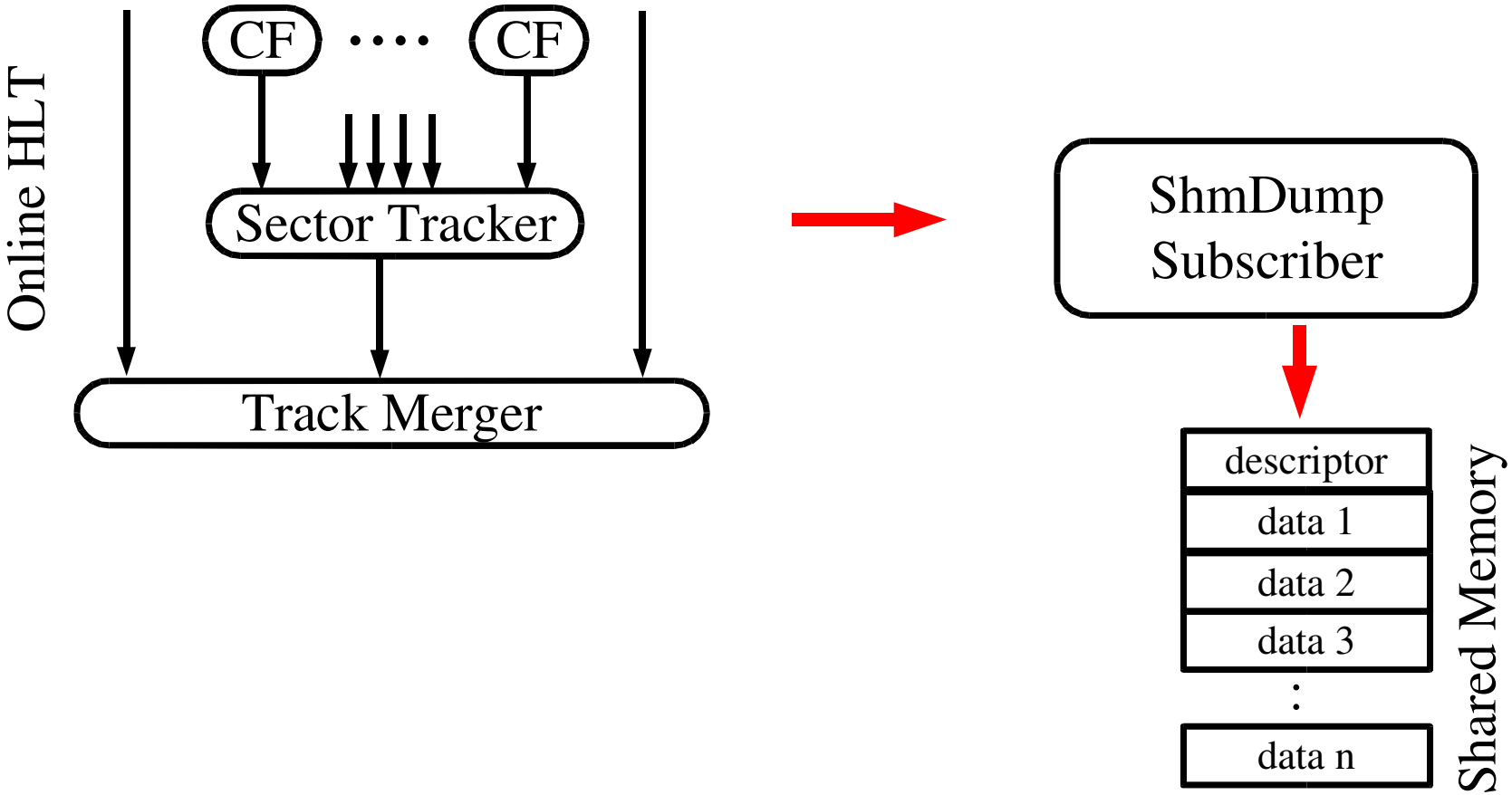
• fAttributes[8]	version, byte order
• **fLength**	length of block descriptor structure
• fAlignments[8]	alignment for different data types
• fType.fID	data type for event data blocks
• fSubType1.fID	data origin for event data blocks
• fSubType2.fID	data specification for event data blocks
• fBirth_s	Seconds part of time of block data creation
• fBirth_us	Microseconds part of time of block data creation
• fProducerNode	Node ID of block's producing node
• **fOffset**	Starting offset of described block's data
• **fSize**	Size of described block's data

Data block structure

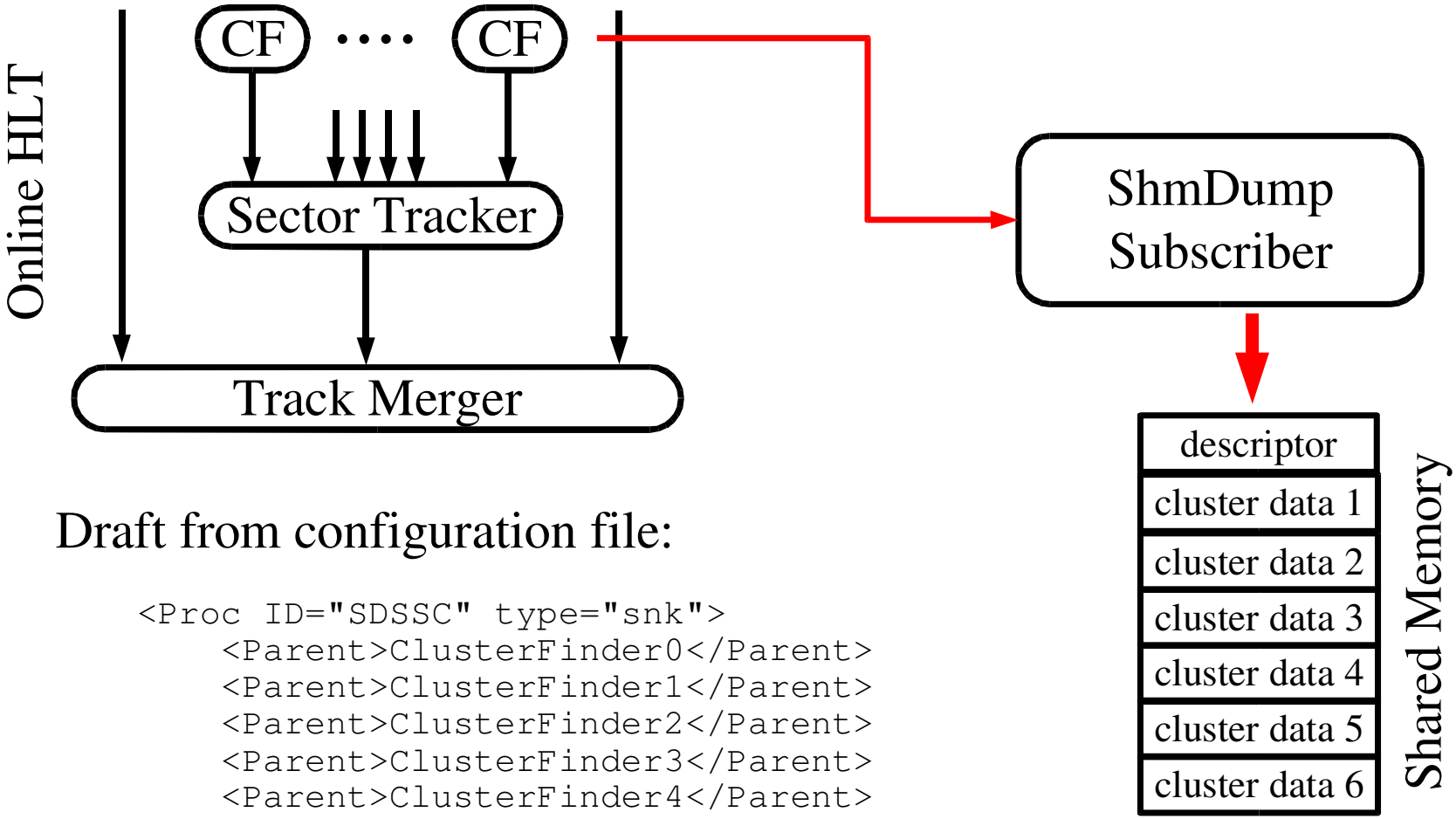


- Master Descriptor
  - attributes and alignment valid for the descriptor itself and the block descriptors
  - offset points to the first block desc.
- Block Descriptor
  - attributes and alignment of the data blocks
  - offset points to data block, all others follow subsequently
- Data Block
  - the internal data structure as it comes into the ShmDumpSuscriber

Connection to the data flow



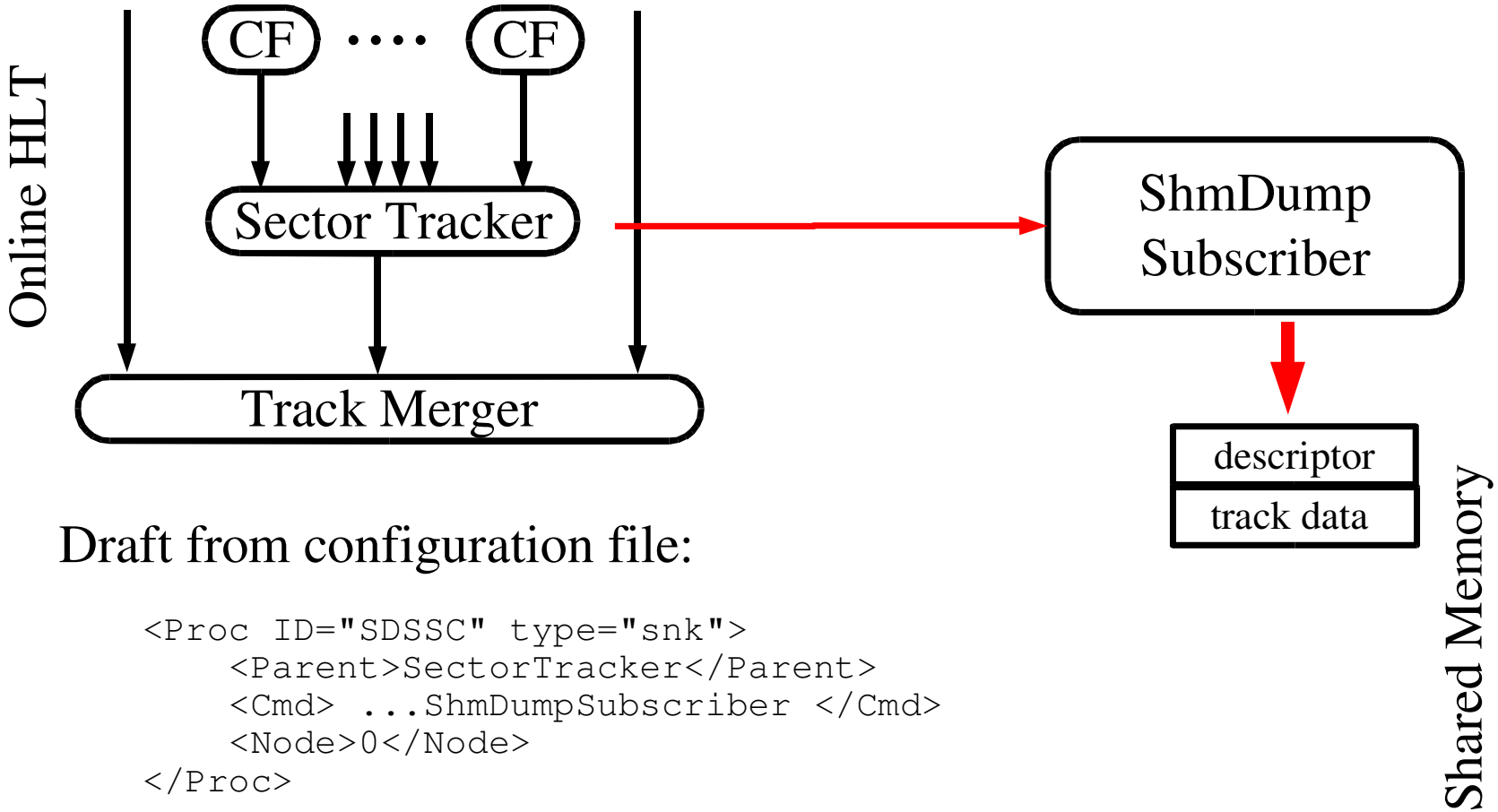
Connectivity example 1



Draft from configuration file:

```
<Proc ID="SDSSC" type="snk">
    <Parent>ClusterFinder0</Parent>
    <Parent>ClusterFinder1</Parent>
    <Parent>ClusterFinder2</Parent>
    <Parent>ClusterFinder3</Parent>
    <Parent>ClusterFinder4</Parent>
    <Parent>ClusterFinder5</Parent>
     <Cmd> ...ShmDumpSubscriber </Cmd>
    <Node>0</Node>
</Proc>
```

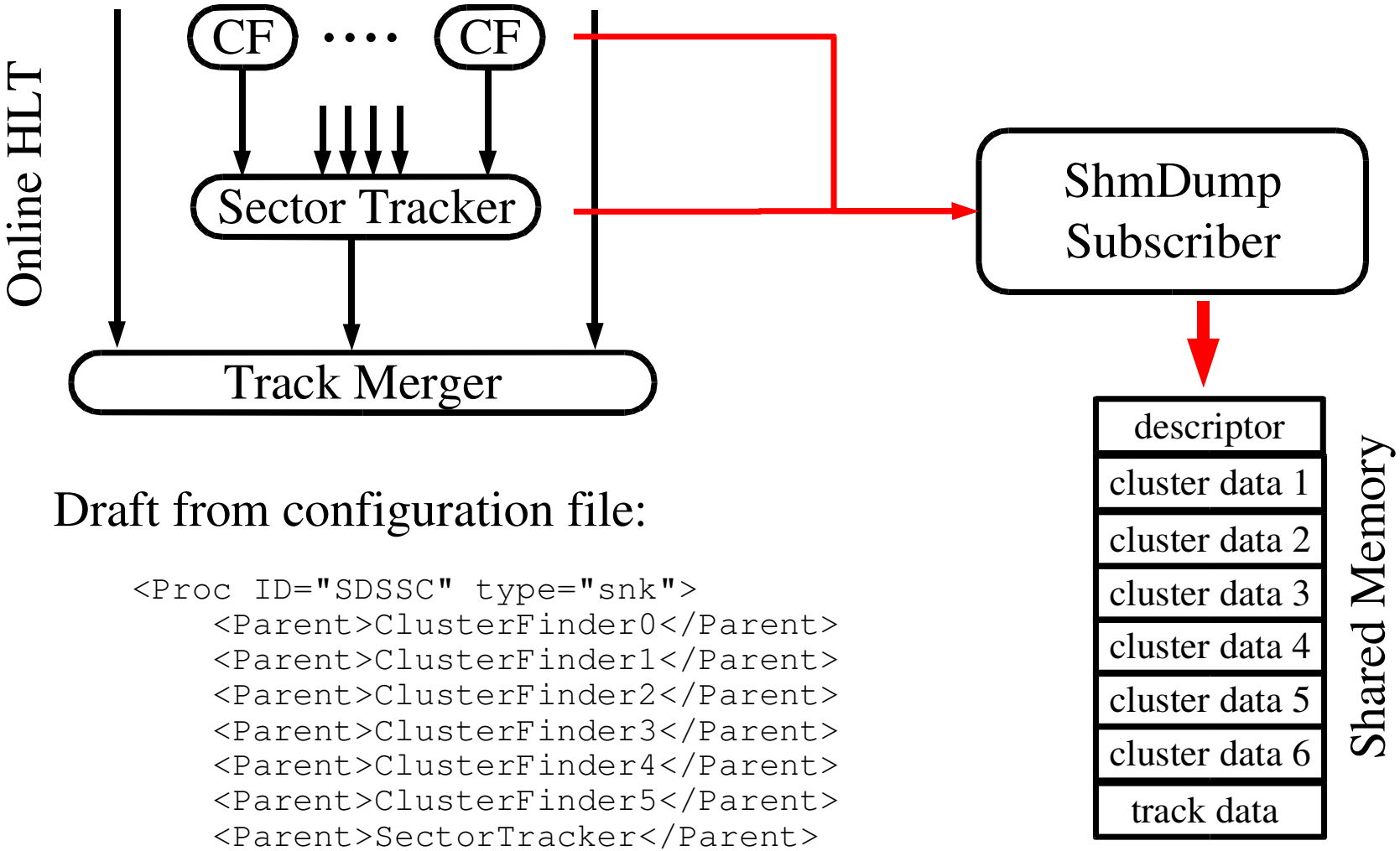
Connectivity example 2



Draft from configuration file:

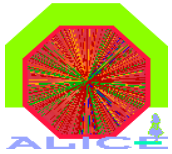
```
<Proc ID="SDSSC" type="snk">
    <Parent>SectorTracker</Parent>
    <Cmd> ...ShmDumpSubscriber </Cmd>
    <Node>0</Node>
</Proc>
```

Connectivity example 3



Draft from configuration file:

```
<Proc ID="SDSSC" type="snk">
    <Parent>ClusterFinder0</Parent>
    <Parent>ClusterFinder1</Parent>
    <Parent>ClusterFinder2</Parent>
    <Parent>ClusterFinder3</Parent>
    <Parent>ClusterFinder4</Parent>
    <Parent>ClusterFinder5</Parent>
    <Parent>SectorTracker</Parent>
    <Cmd> ...ShmDumpSubscriber </Cmd>
    <Node>0</Node>
</Proc>
```



Shared memory block decoder

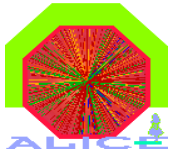
extracts data blocks from the shared memory

- based on a class – easy to use in (Ali)ROOT interactive session
- stand-alone tool dumps data to stdout
- can act as a filter

parameters of stand-alone SDSBlockDecoder (first draft)

```
--noreset              // dont reset the blocksize
--header, --tail       // additional header and tail
--filter [origin=<>] [datatype=<>] [blockno=<>] [eventno=<>] ...
--format [eventtype] [subtype] [content][origin] ...
```

Status and outlook



- ready for a TPC Online display
- interface defined
- ShmDumpSubscriber ready
- block decoder ready but common functionality has to be separated from detector specific func.
- stand-alone decoder has to be implemented
- Extension of Online display to support track and raw data investigation has to be done