Detector Data Links to DMSC

DG-DMSC EF Data Link Meeting
Copenhagen 21st Feb 2018

FROM LAST MEETING
Key points for discussion...

EUROPEAN SPALLATION SOURCE

Network selection.
Raw Ethernet, basic UDP, something else????

Timing resolution and timestamp sizes
Official ESS timestamps are 64 bit. 32 bit timestamps rollover at:

- 429 secs for 100nsec
- 43 secs for 10nsec
- 47 secs for 11.4 nsec (corresponds to ESS global cock of 88Mhz)

We can guarantee latency significantly below that, but we probably need some additional timestamp reference data, eg send data every time a rollover occurs? Or just 64 bit?

Data ordering
Multiple streams of input data, may be out of time.

Time sorting would consume significant resources.

Data Distribution
Multiple EF units, need to understand what goes where. In what ways are we able to partition systems?
Which instruments are the most likely to generate big data rates, and what technology do they use?
(Eg Mutliblade, blades are essentially independent)

Ancillary Data
What ancillary data is needed for processing (config/status/monitoring/calibration) and where should that be sent?

FROM LAST MEETING
Architecture determines what is possible...

The internal architecture used in the FPGA determines the 'envelope of the possible'

Multiple streams of input data, may be out of time.
One request (Martin) was to ensure data arrive in chronological order.

Time sorting would consume significant resources – for discussion

We make the assumption that for some instruments at full luminosity, multiple EF units may be required and that some load balancing may be needed.

Data can be directed to different engines, and potentially engines can transmit to different destinations.

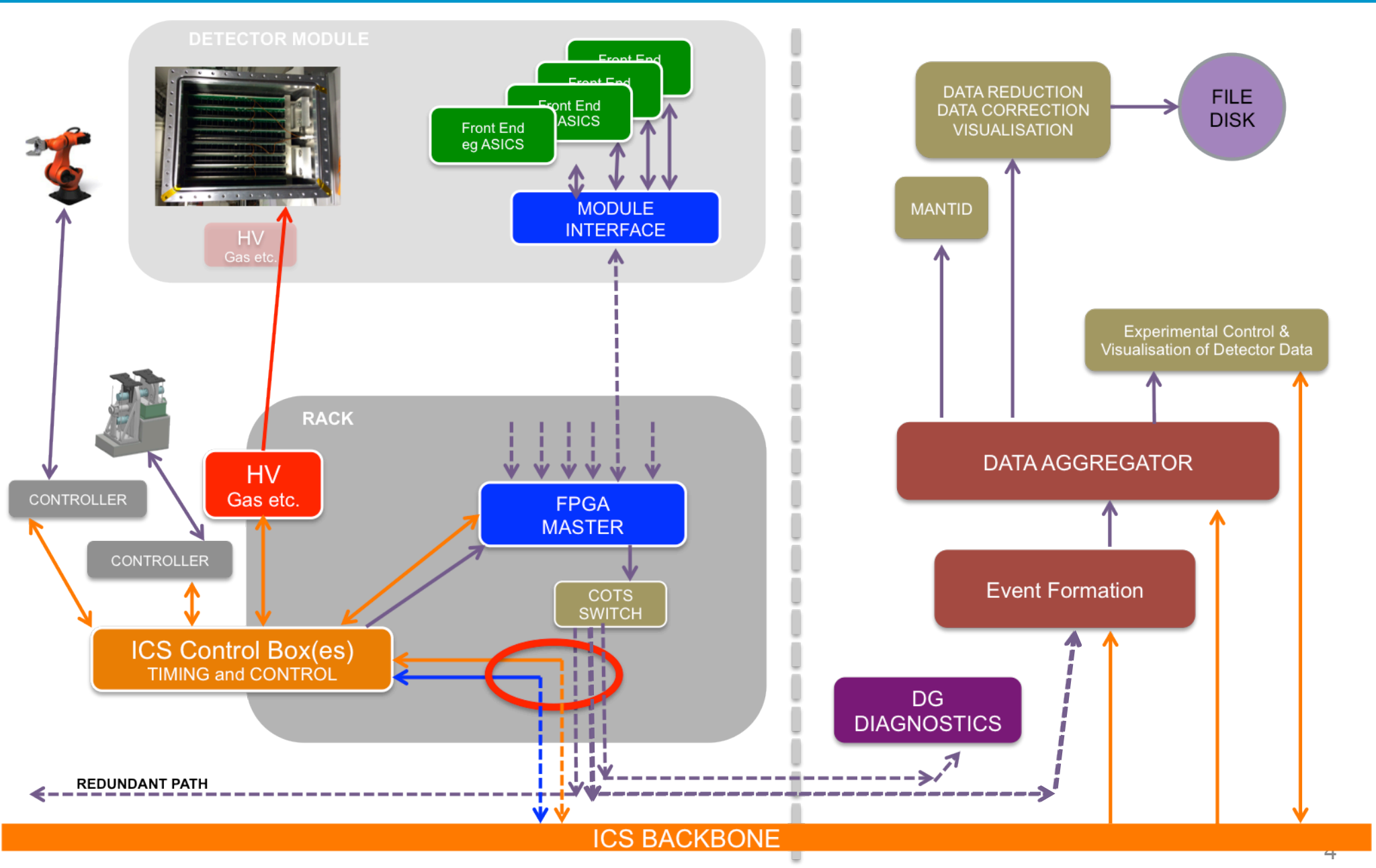
Must ensure relevant data stays together, eg anode/cathode for 2D match.

Ancillary data?

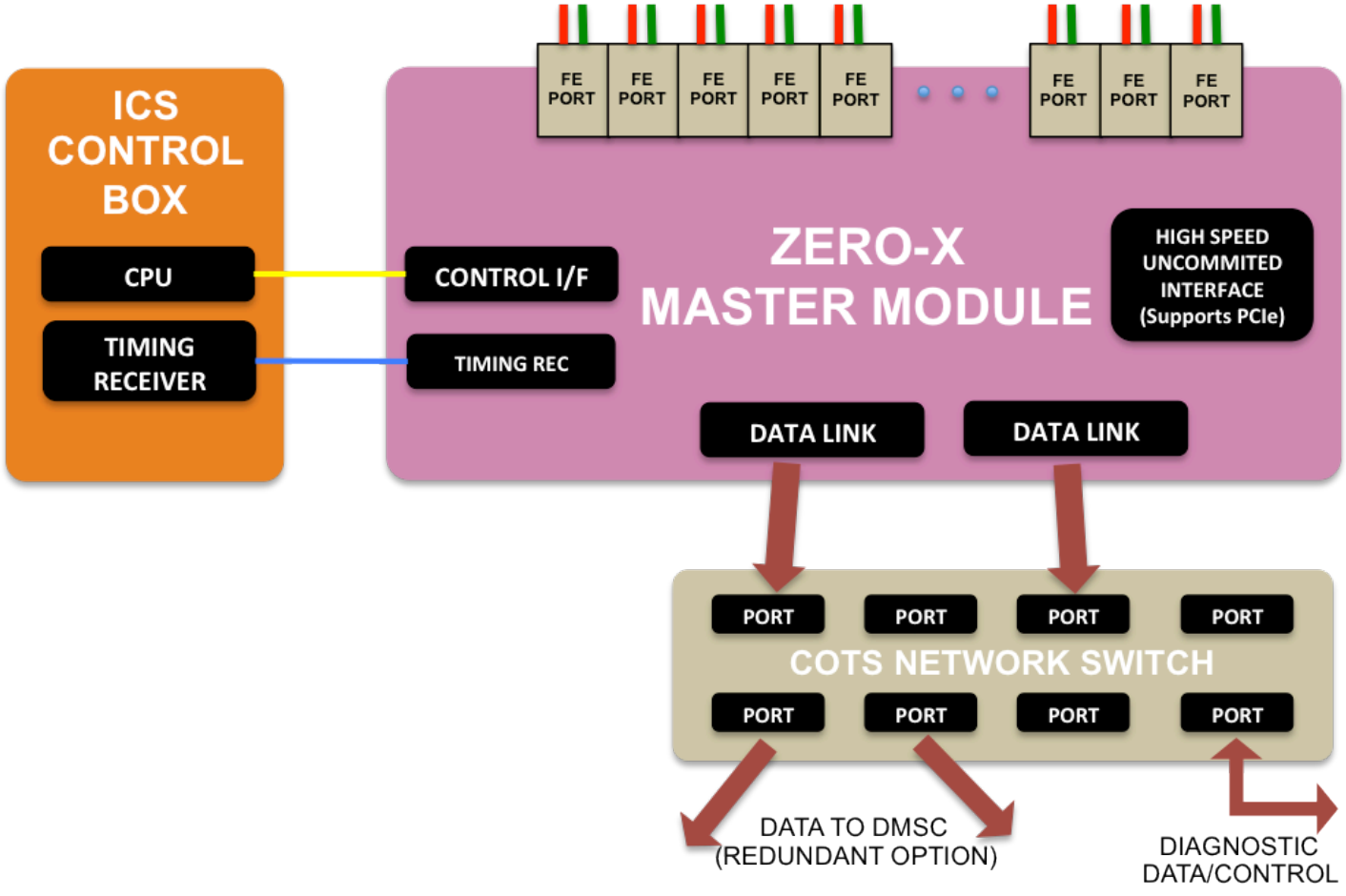
Some fast computation may be possible in the master, eg to expand a data format to help the software. Balance of network bandwidth used (bandwidth inside EF) vs. compute requirements. Should data fall naturally onto words or long-word boundaries? Should certain bits be extracted into byte fields?

We would be happy to investigate the feasibility of any beneficial processing.

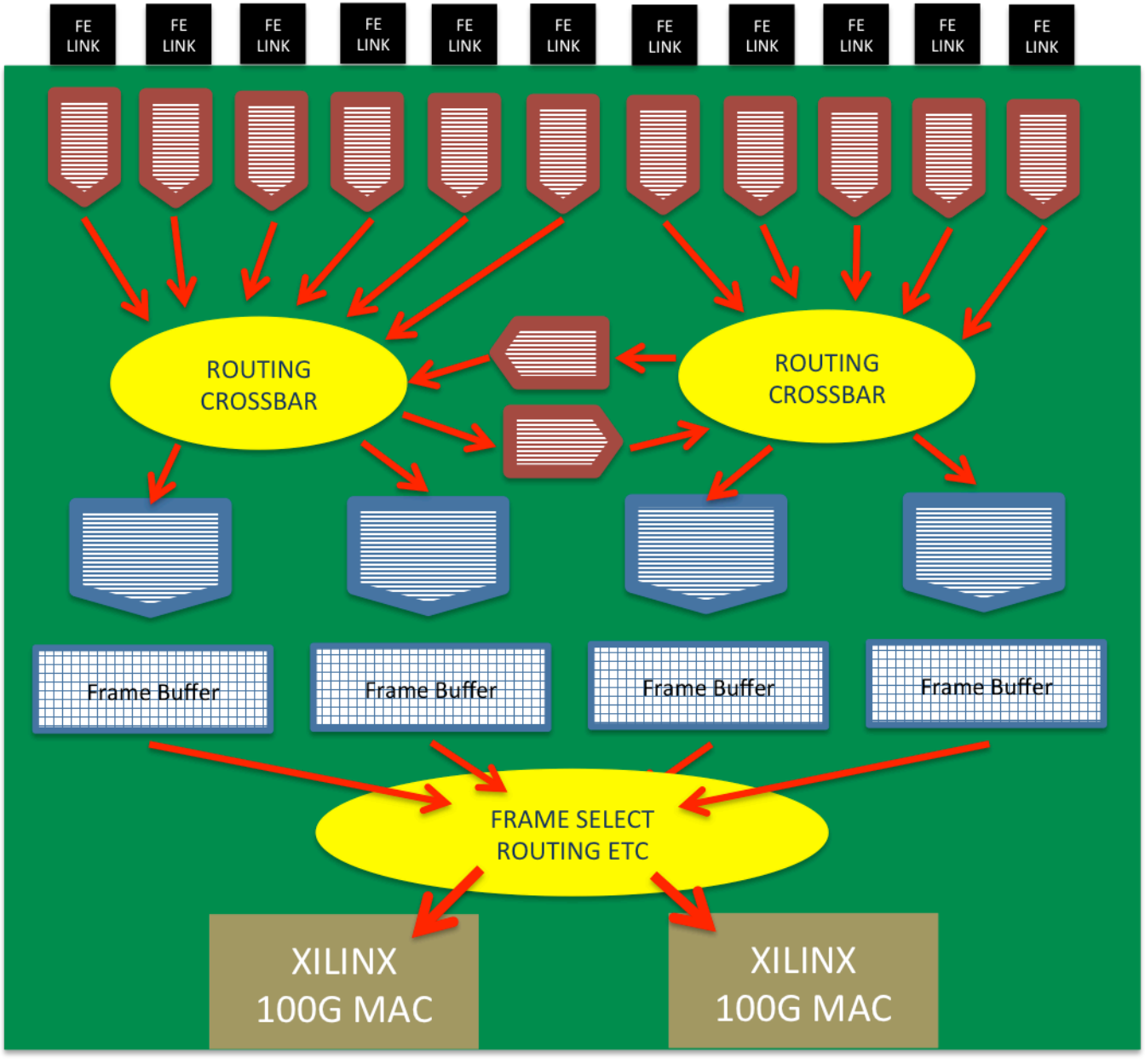
Data Paths in a Typical Detector Installation



Master transmits only....



Backend operation



Data from front end links (24 max speed 10G/25G)

Packets from front end received into FIFO.

Complete data packets routed to output FIFO for that type of data. Typically use one engine per output type (engines may 'overlap', eg one engine might have several distinct input FIFO buffers)

When there is enough data a full frame can be constructed in the Frame Buffer. Destination MAC can be different for each buffer. We can set a maximum latency, transmit partially full frame.

Load sharing multiple destinations possible.

When at least one Frame Buffer is ready we can fire a packet as soon as we are ready.

Output packets need to be scheduled so that they will not overload the destination link speed. A programmable delay will be introduced. This can also reflect any limitations in the receiving workstation

It will be possible to reduce the overall rate of transmission by increasing this delay, but note that transmission will in any case only reflect the data rate (very low rate of status packets)

It will be possible to assign priority for transmission, and to mark some data as disposable (eg monitoring data could be dropped if data rate becomes too high)

DMSC Questions from Tobias....

To get the discussion ahead at the readout meeting, here are some questions we have. I'll make no attempt to make them very specific. They just serve as a pointer to areas we think we could use more clarity on. The list could be longer or shorter.

- what timing data can we get from the ICS timing system through the "master"? is that configurable?
  Precise details depend on the timing resolution from the front end. See Suggestions elsewhere in this presentation

- while the accelerator is running, receiving a pulse packet at 14 (or 7) Hz is probably sufficient, do we also get some idle packet when the machine is off?
  The timing system runs continuously, and we can generate control packets at any time. Depends on handling of Metadata

- for load balancing: what are the options? Can we map channel groups to IPs/Ports? How would we configure that (protocol and what could the parameters available)? Could we send the same channels to two destinations?
  See slide on this

- when are the next batches or hardware ready?
  We have 13 more boards arriving imminently, and would plan to box up at least 6 of those. We would like to identify any revisions to the data generator box etc. at this meeting. It would be good to have a plan for usage.

- what are the things you think we can/should test with the packet generator?
  There are implementation issues associated with high bandwidth which you can only really test with a high bandwidth data source. We would expect you to test packet reception, data unpacking (first stage) and whatever load balancing scheme you expect to deploy.

Once we discuss that at bit at the meeting, we will find more detailed or other problems.

Load Balancing

Load balancing is performed by sending data from a single master to more than one EF unit.

- **Geographical**
  - Different regions of the detector go to different engines, that each potentially send to different EF nodes.
  - Need to be careful that all necessary data is included (eg regions complete, timing metadata etc.)
  - System is essentially decomposed into a number of parallel subsystems.
  - The configuration is static.
- **Throttle**
  - Programmable delay between packets
- **Round Robin**
  - Each engine can be programmed to send packets to a number (up to 8) EF units in turn (IP, UDP Port?)
  - This function is implemented independently in each engine.
- **Remote Admin**
  - You may want to implement a more sophisticated load balancing scheme, where you have a ‘supervisor’ node that tells ‘worker’ nodes what to process.
  - This needs to be implemented entirely within your system.
  - Having all data at all nodes is potentially useful in this case. The nodes are then told what data to process, and what to ignore.
  - We don’t think we’d need to use a formal broadcast, just retransmit to multiple nodes. Eight node limit.
  - This could easily overload link bandwidth.

Ancillary Data / Meta Data

It makes sense to feed all information needed by the EF units through the fast data link.
It might also makes sense to include all ancillary data with the main data stream (even if it is logged elsewhere).

This will (may) include:

- Timing higher order bits to allow reconstruction of full timestamp for front end data.
- Accelerator/Target data
- Configuration data for readout (eg thresholds (once per run?))
- High voltage values and currents (once every second?)
- Environmental values, temperatures, pressures, gas flows, water flow, etc.

How is this achieved?

The master has a control interface connected to the EPICS system, and this can write data into a memory mapped buffer in the master. This is how configuration data is delivered to the system. We can define a block of buffer space that is updated by EPICS and is forwarded by the master. This is transparent to the master, the content of the data could be anything. Could be a regular transmission (eg 1 per second), could be transmit on update.

Fast timing receiver data will probably not be available promptly (unless you really really have to have it).
If you don’t know what this is, you don’t need it.

Timing

ESS uses 64 bit timestamps.

Detector systems typically need timestamp resolutions of 10-100nsec and very much shorter counters are actually implemented in the readout hardware, eg 32 bits and less.

The assister front end supports a 32 bit timestamp counter running with resolution down to 12nsec (88Mhz), however, front end systems may run high speed clocks synchronised to that. Care must be taken with absolute synchronisation in that case.

Within one detector system (one master) running ESS generic readout (type A,B and typically type C) all the readout electronics is running in a synchronised way.

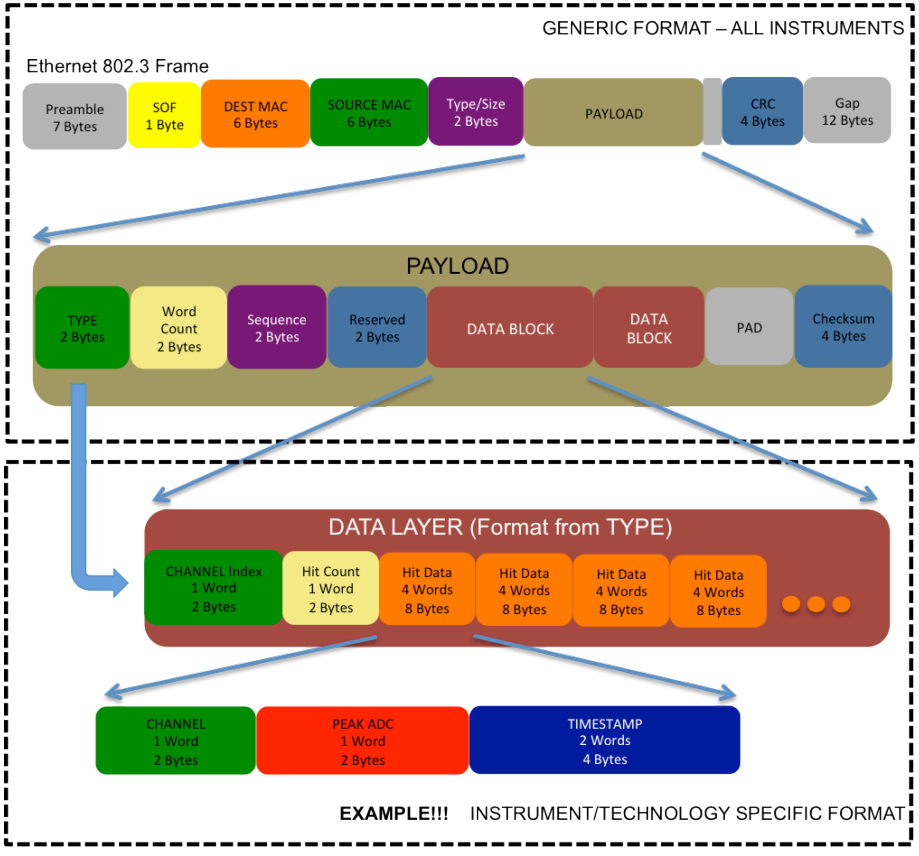
For efficient packing, we would expect to use timestamps of 32 bits or less. This means some additional information is needed to convert this data back to 64 bit ESS timestamps. This could be provided in a number of ways....

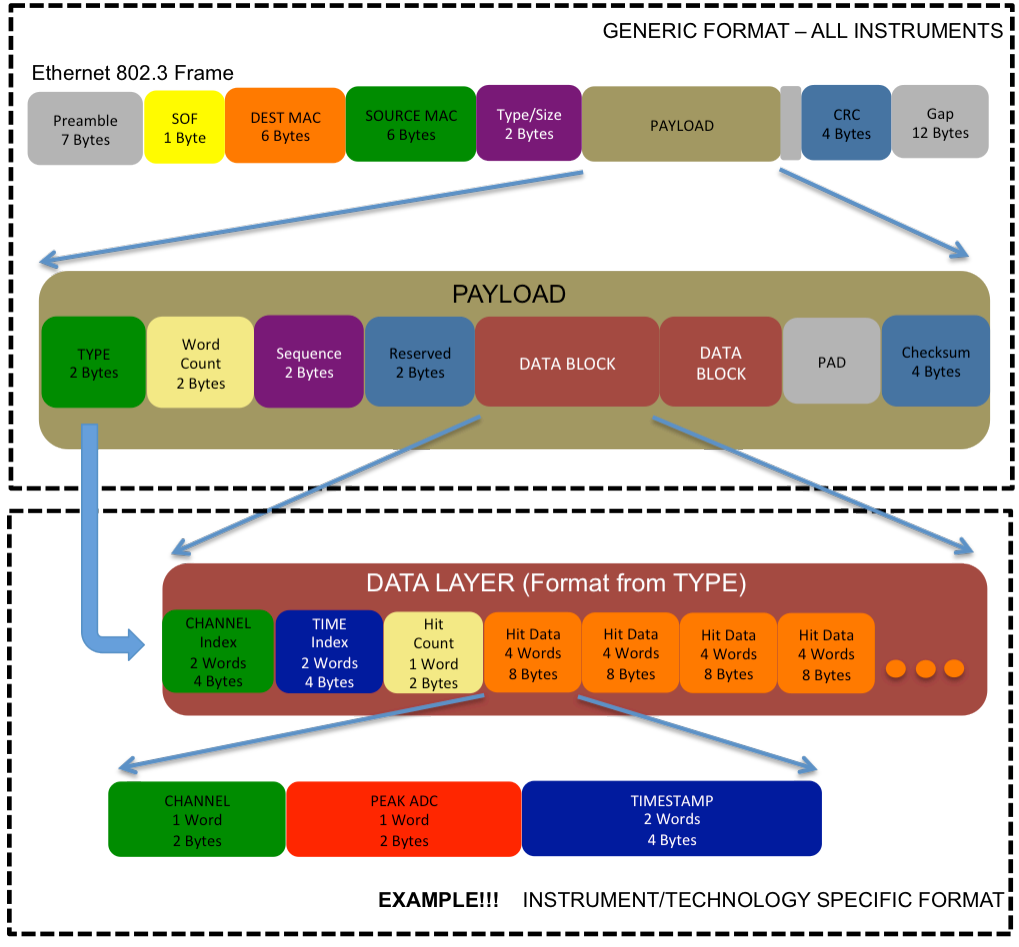
Timestamp packets – eg rollover packets.

Timestamp indexes – embedded in the data itself.

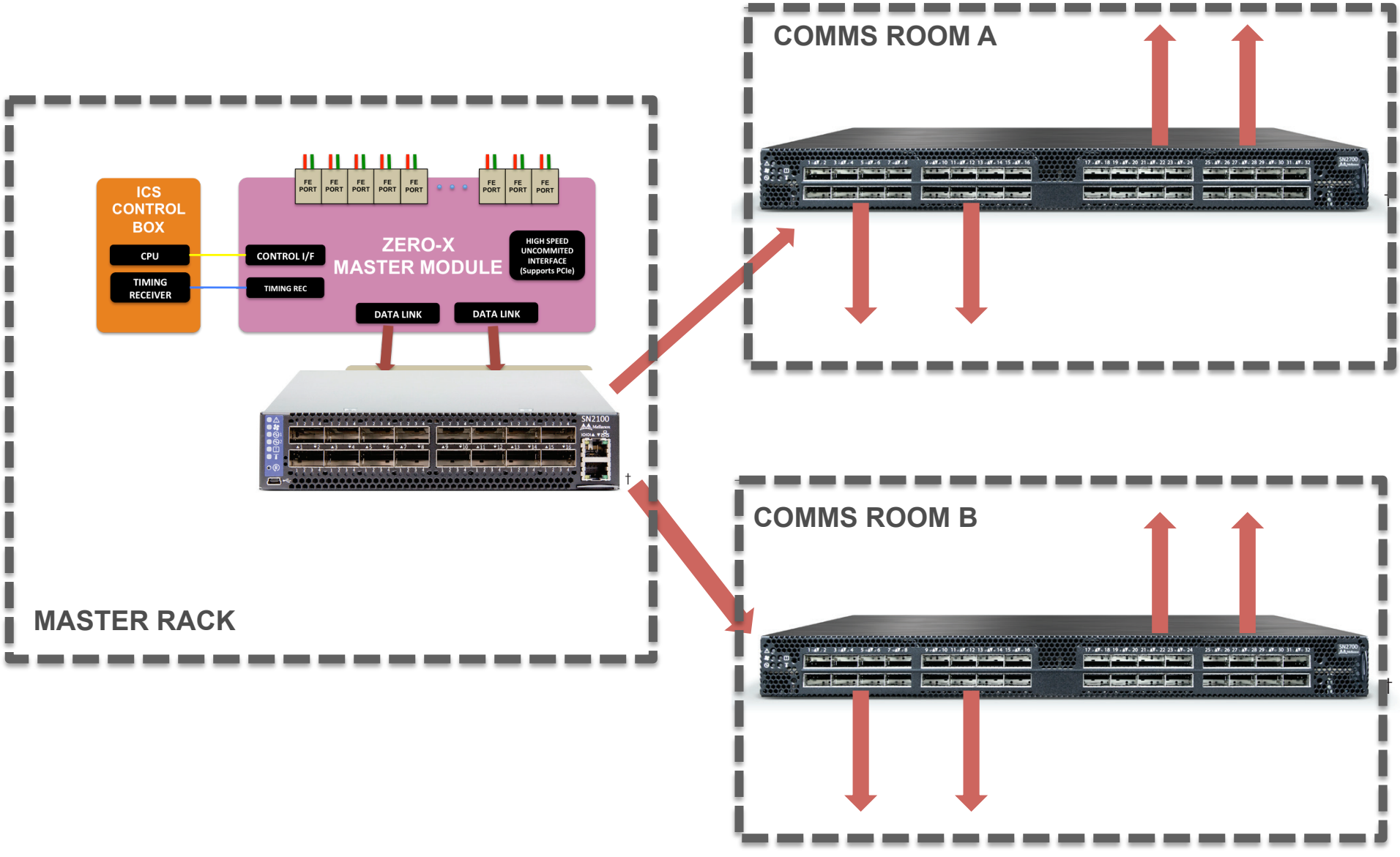
Something else??

Can embed full timing information in data layer





A quick word on network topology...



Future plans...

EUROPEAN SPALLATION SOURCE

Evolve master with simplistic front end network, real data formats.
The data generator really is the prototype master hardware!!!

What systems might we include?

- VMM3A By end of 2018
- Discrete ADC By end of 2018 (time/effort permitting)
- IDEAS 2019 (use ESS Assister)
- BandGem GEMMA/GEMINI (use ESS assister or Trenz testbed)

Front end hardware platforms

- Trenz testbed (ESS/STFC)
- Assister (Commercial)
- CDT/Jalousie (CDT/Julich)

Tobias has rightly suggested we need a fully worked solution to be able to really address the implementation issues. Steven and I agree with this.

The ADC existing ADC readout demonstrator is of limited utility. Since we have it and it works we will build more, but it won't feature significantly in our future development.

We hope you will be able to use the data generators for infrastructure testing of whatever prototype farm you have. We lose technician effort by July, so we need to press ahead with production.