Data Acquisition Backbone Core

J.Adamczewski, H.G.Essel, N.Kurz, S.Linev

Work supported by EU RP6 project JRA1 FutureDAQ RII3-CT-2004-506078

Driving forces

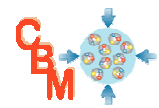
2004 → EU RP6 project JRA1 FutureDAQ*
2004 → CBM FutureDAQ for FAIR
2005 → FOPI DAQ upgrade (skipped)

→ Intermediate demonstrator

1996 → MBS future
50 installations at GSI,
50 external
http://daq.gsi.de

- Detector tests
- FE equipment tests
- Data transport
- Time distribution
- Switched event building
- Software evaluation
- MBS event builder
- General purpose DAQ

→ **D**ata
Acquisition
Backbone
Core

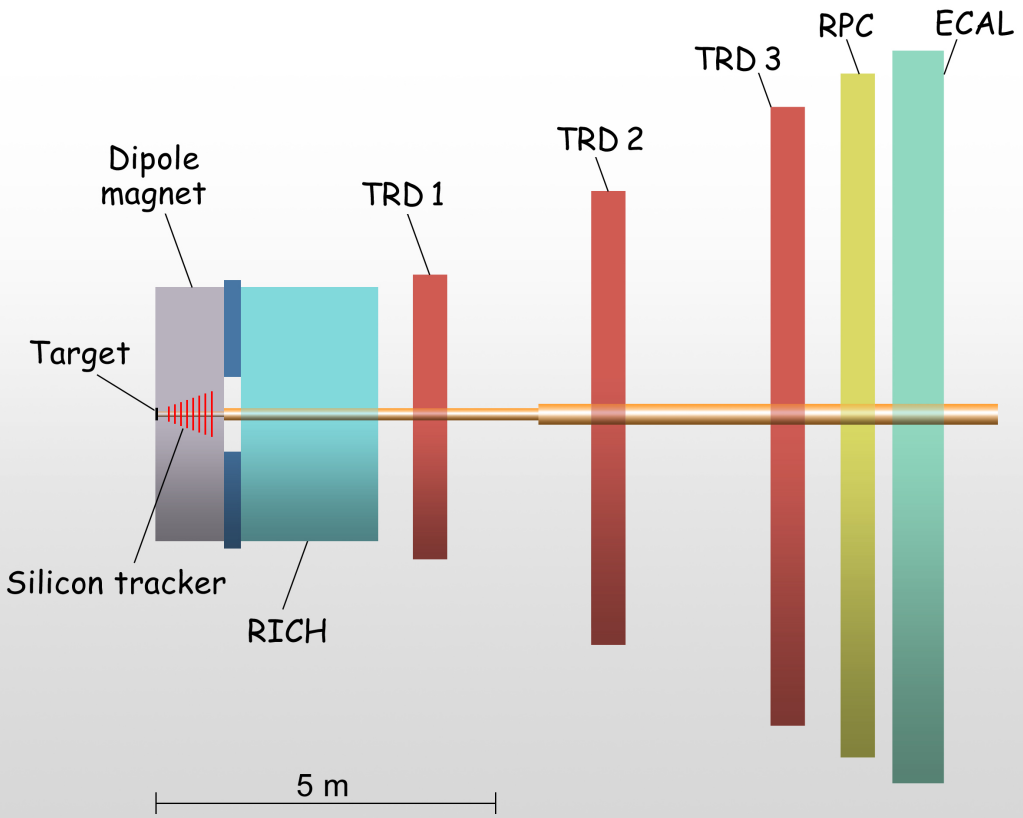


* RII3-CT-2004-506078

The CBM detectors

P.Senger, 2003

At 10^7 interactions per second!



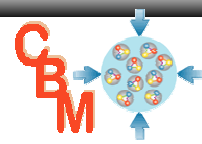
- Radiation hard Silicon (pixel/strip) tracker in a magnetic dipole field
- Electron detectors: RICH & TRD & ECAL pion suppression up **to 10^5**
- Hadron identification: RICH, RPC
- Measurement of photons, π^0,η and muons electromagn. calorimeter ECAL

Central multiplicities:

160	p
400	π^-
400	π^+
44	K^+
13	K
800	γ

average 500 at 10 MHz

CBM data acquisition



New paradigm: switch full data stream into event selector farms

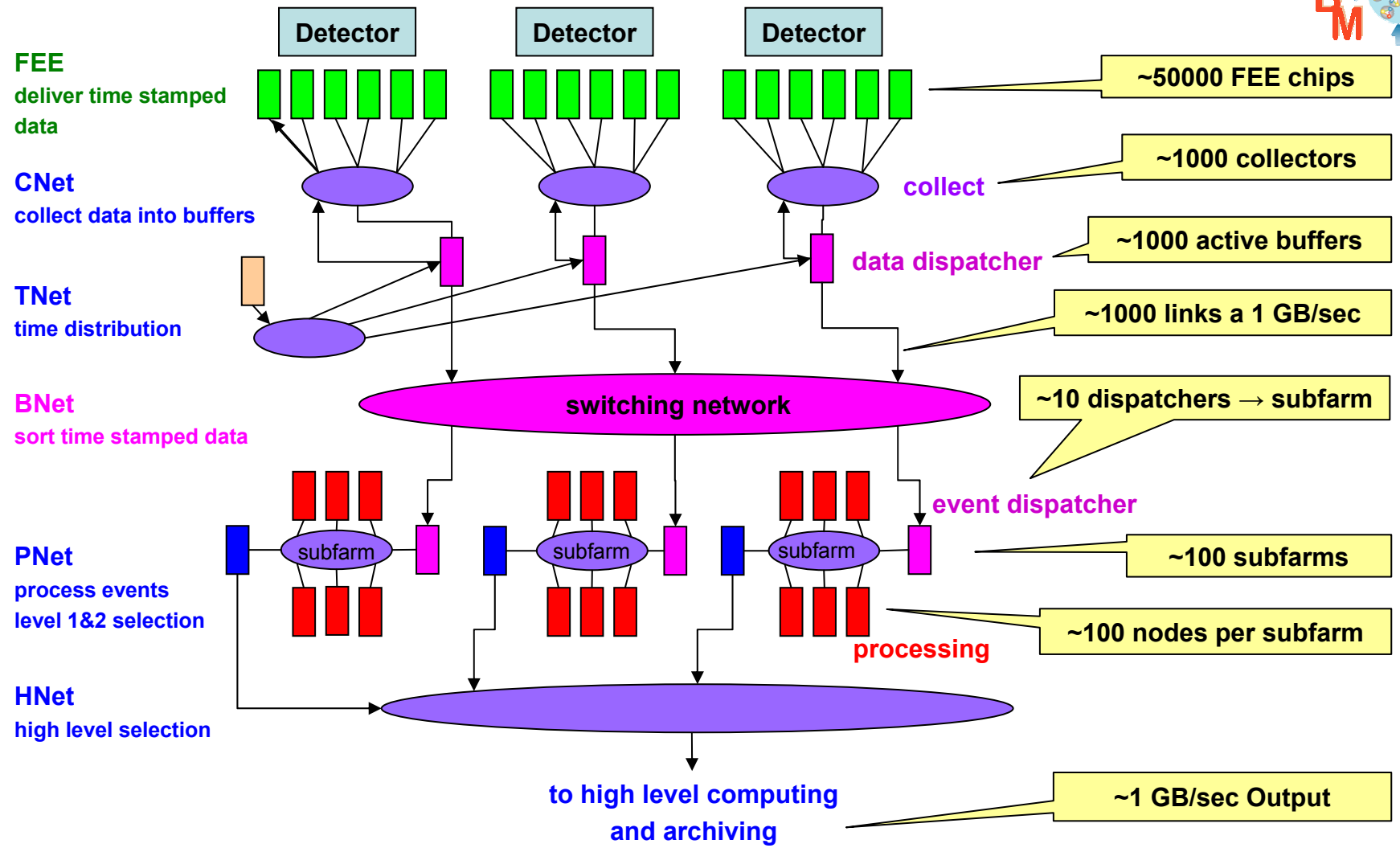
1. **A conventional LVL1 trigger would imply full displaced vertex reconstruction within fixed latency.**
2. **Strongly varying complex event filter decisions needed on almost full event data**

➡ **No common trigger! Self triggered channels with time stamps! Event filters**

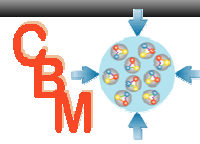
- **10 MHz** interaction rate expected
- Time stamps in all data channels, typical resolution **~ns**) required
- **1 TByte/s** primary data rate (Panda < 100 GByte/s) expected
- **1 GByte/s** maximum archive rate (Panda < 100 MByte/s) required
- Event definition (time correlation: multiplicity over time histograms) required
- Event filter to **20 KHz** (1 GByte/s archive with compression) required
- On-line track & (displaced) vertex reconstruction required
- Data flow driven, no problem with latency expected
- **Less complex** communication, but **high data rate** to sort

CBM data acquisition

W.F.J.Müller, 2004



CBM DAQ features summary



- Triggerless data acquisition and transport until filter farm
- Event building on full data rate ~1TB/s
- BNet: ~1000 nodes, high-speed interconnections
- Linux may run on most DAQ nodes (even FPGAs)
- Test cluster with InfiniBand: small „demonstrator“ set-up → **DABC**

Extensive simulations of BNet using SystemC

DABC requirements

Data
Acquisition
Backbone
Core

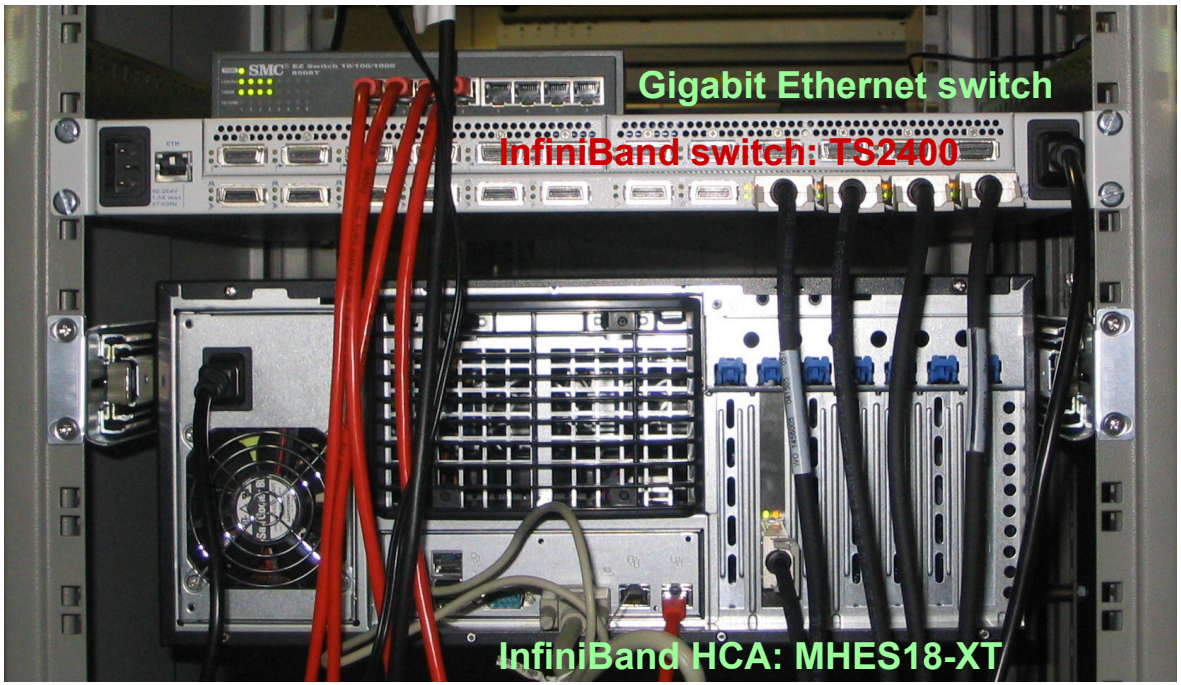
Requirements

- build events over fast networks
- handle triggered or triggerless front-ends
- process time stamped data streams
- provide data flow control (to front-ends)
- connect any front-ends
- connect MBS readout or collector nodes
- provide interfaces to plug in application codes
- be controllable by several controls frameworks

 InfiniBand

InfiniBand test cluster at GSI

- 4 nodes:
  - Double Opteron 2.2 GHz, 2GB RAM
  - Mellanox MHES18-XT host adapter (PCIe)
  - 2x Gigabit Ethernet host adapters
  - SuSE Linux 9.3, x64bit version
- Mellanox MTS2400 24X InfiniBand switch



InfiniBand software

- Mellanox IB Gold 1.8.0
  - IPoIB: IP over InfiniBand driver
  - uDAPL: User Direct Access Programming Layer
  - MPI: Message Passing Interface
- OpenFabric Enterprise Distribution (OFED) 1.1
  - IB Verbs (similar to uDAPL, plus Multicast)
- MVAPICH2: MPI 2 implementation for IB
- IBAdmin
- OpenSM (subnet manager) (for Multicast)

Very extensive testing of all packages.

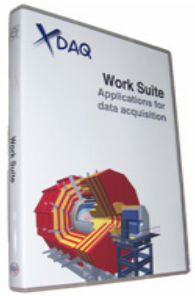
Scheduled all to all data transfer (basic pattern for switched event building):
500-800 MByte/s (2K-64K buffer) per node.

Cooperation with Forschungszentrum Karlsruhe established to scale the tests to 32-64 InfiniBand nodes.

The CMS xDAQ framework

Standard DAQ framework for LHC CMS experiment *

- C++ libraries on Linux, modular packages (SourceForge)
- Distributed xDAQ applications
- Configuration: XML
- Data transport: I_2O protocol (Intelligent IO)
- Communication: http, cgi; SOAP messages
- InfoSpace: Global parameter access (subscription)
- State machines (sync/async FSM)
- Message logger, error handler
- Monitoring tool
- Hardware access library (HAL)
- Job Control (task handler for node control)
- others: exceptions, threads, infospace, data (de)serializers...



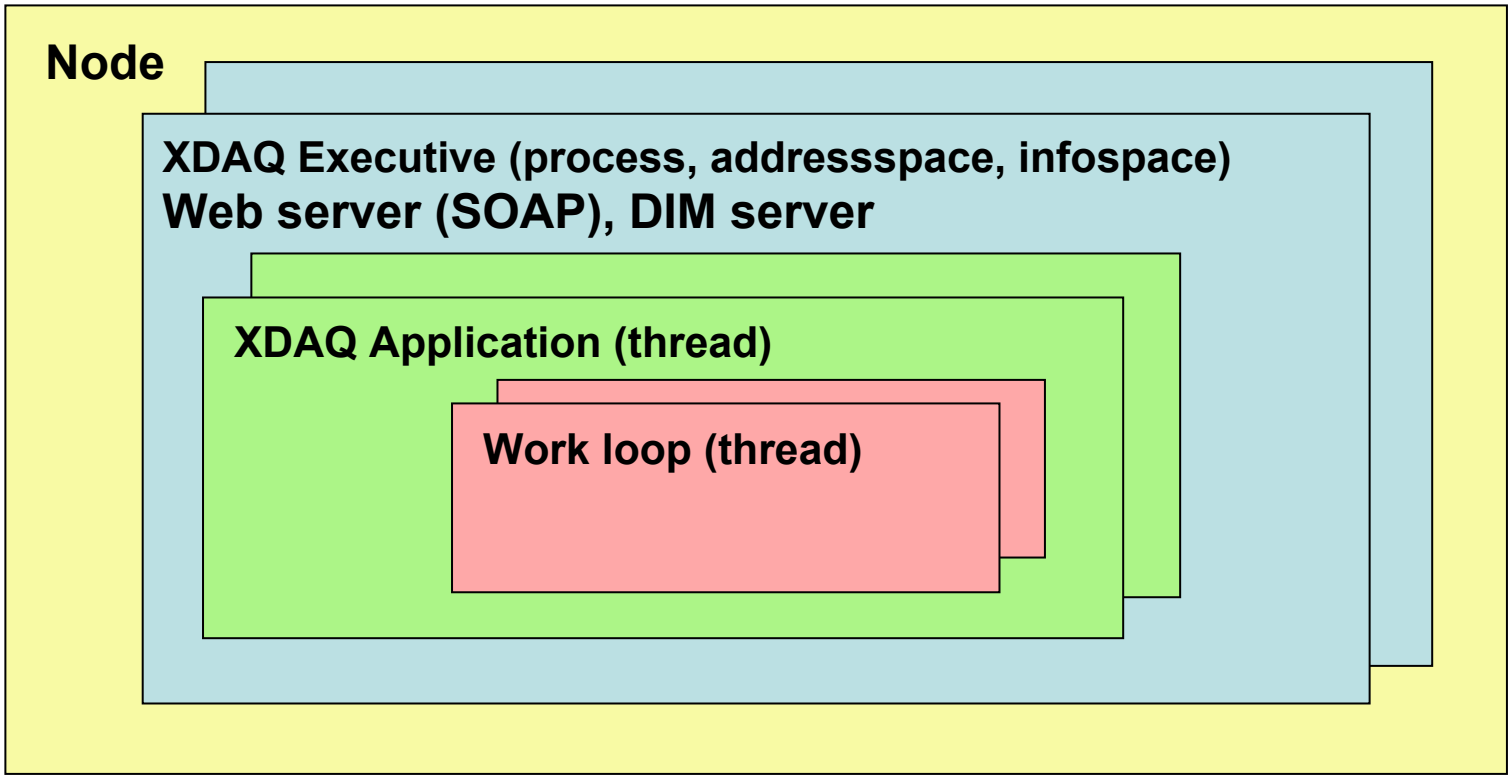
* Orsini, Gutleber http://xdaqwiki.cern.ch

Decision to build DABC on xDAQ

Developments at GSI for DABC

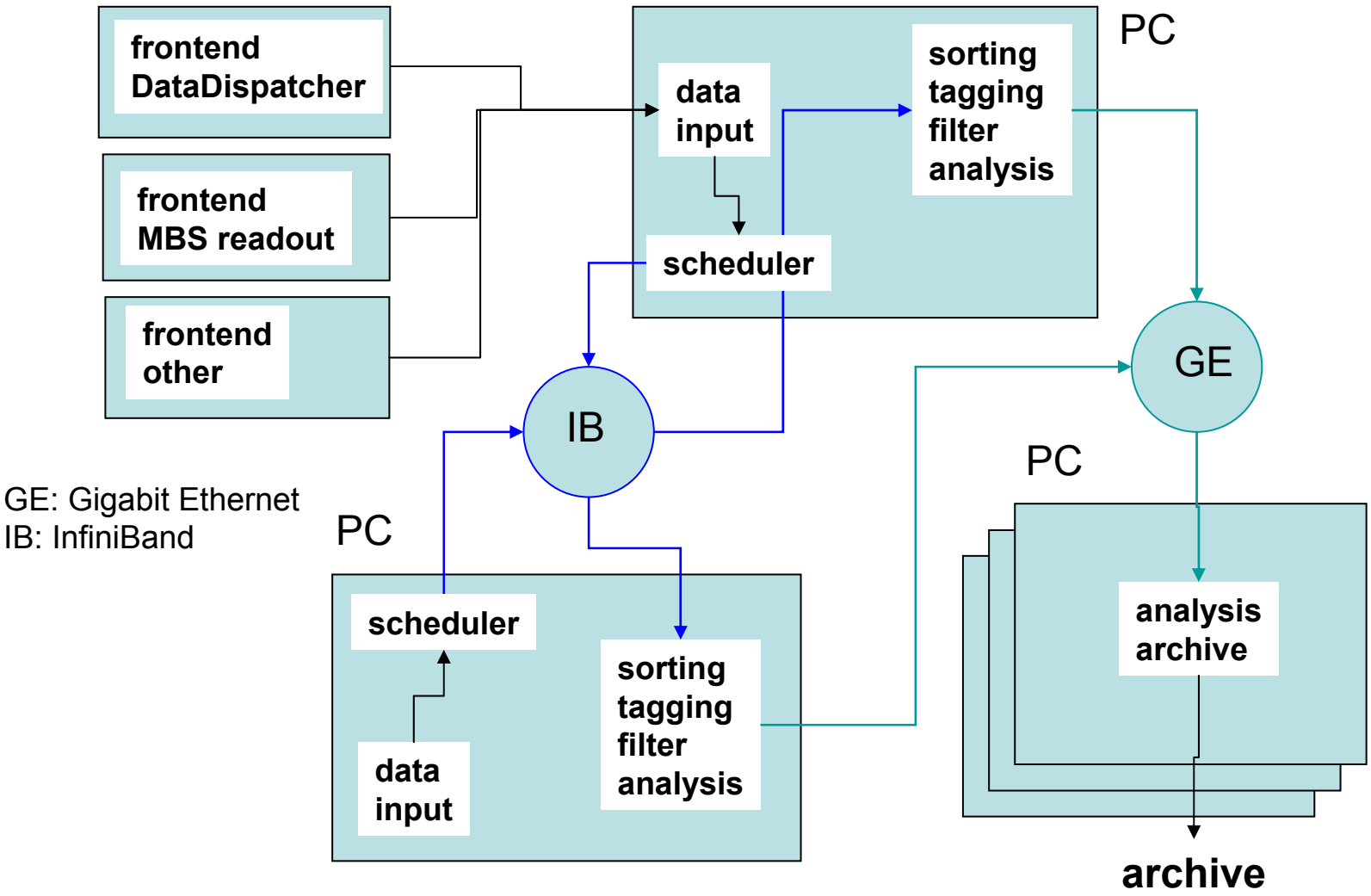
- xDAQ peer transport for InfiniBand (uDAPL)
  - uDAPL buffers managed within XDAQ memory pool
  - avoids memcopy and new buffer allocation for each send package:
  - lookup if posted memory reference is known as send buffer
  - user code can write directly into uDAPL send buffer
- multiple threads for sending, releasing, and receiving buffer
- PCI/PCIe drivers integrated in Hardware Access Library HAL
- Test GUI in Java speaking SOAP with xDAQ executives
- DIM server to export xDAQ InfoSpace and control applications

xDAQ entities



DABC

DABC overview

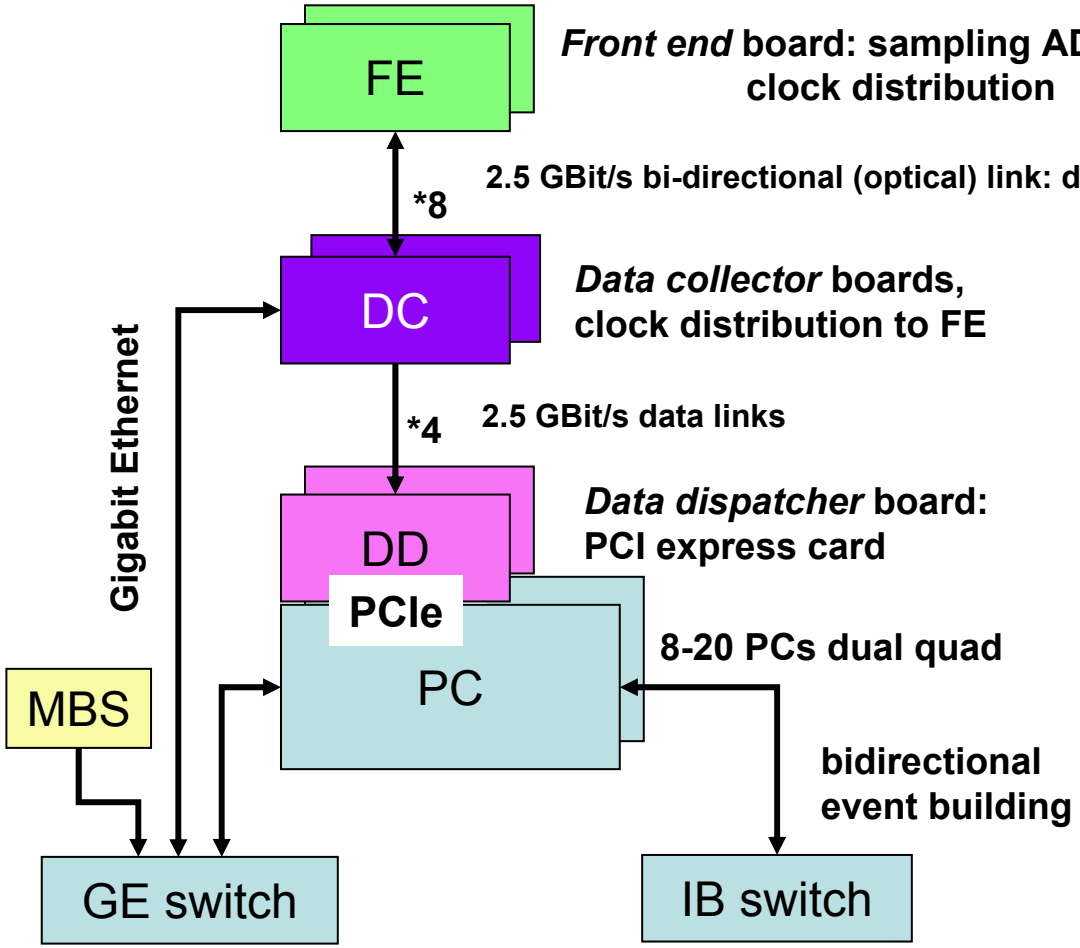


GE: Gigabit Ethernet
IB: InfiniBand

Controls

- State machines, Infospace, message/error loggers, monitoring
- Communication: Webserver, SOAP, DIM
- Connectivity through DIM to: LabView, EPICS, any DIM client/server
- GUI not yet evaluated
- Front-end controls?
- Mix of cooperating control systems

Frontend components test

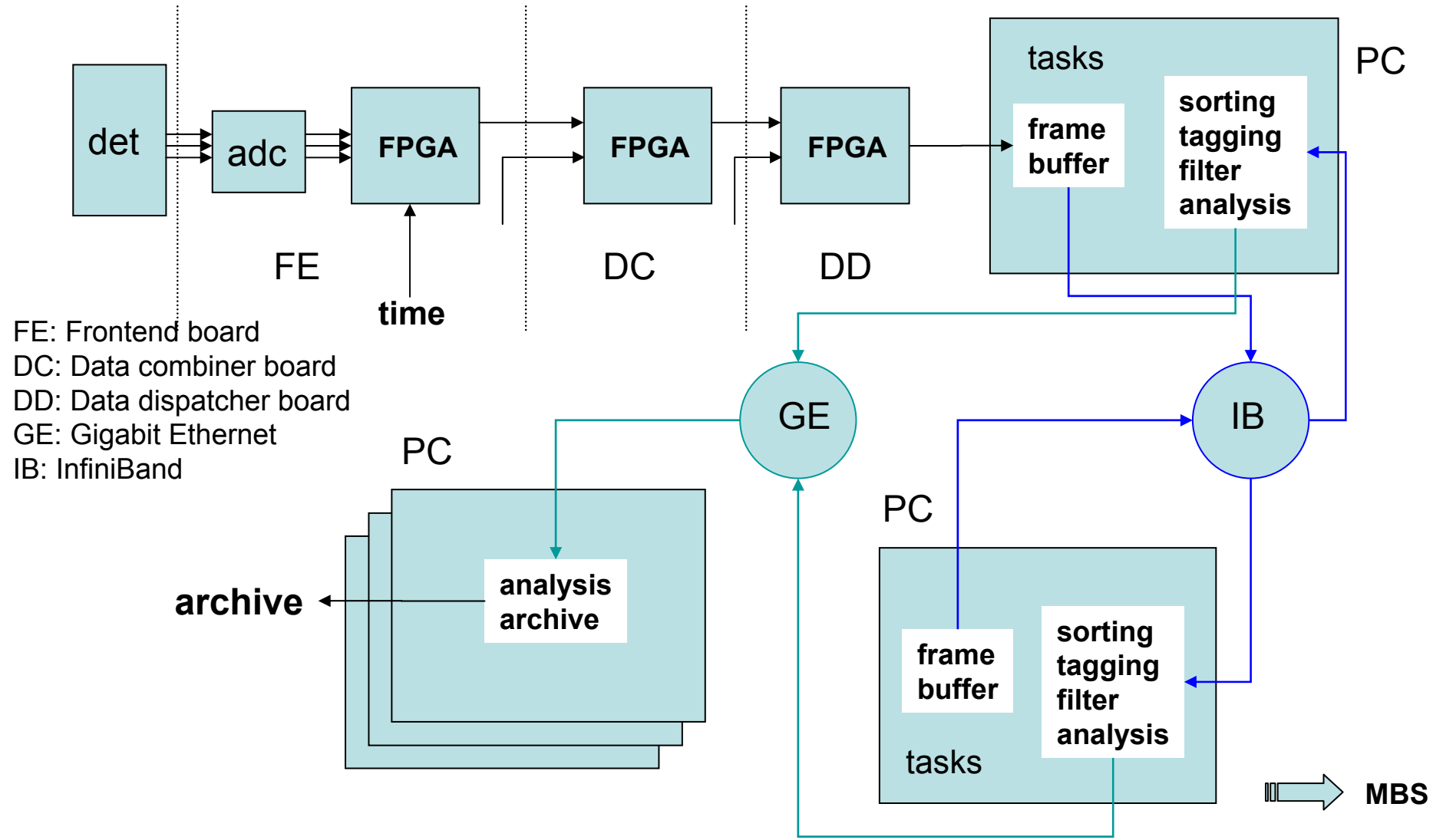


FE: Frontend board
DC: Data combiner board
DD: Data dispatcher board
GE: Gigabit Ethernet
IB: InfiniBand

The goal:
- Investigate critical technology
- CBM detector tests
- Replace existing DAQ

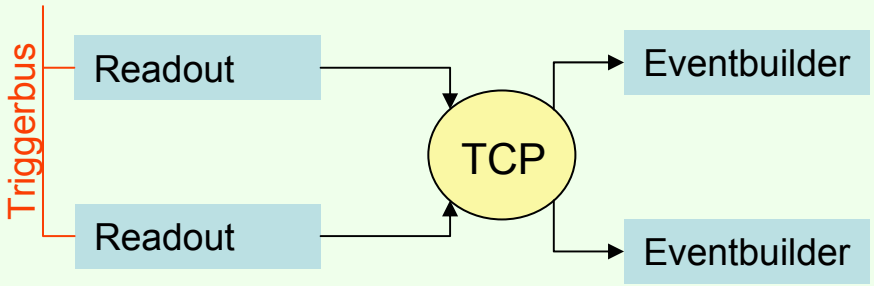
Scales up to 10k channels, 160 CPUs (FOPI upgrade)

Data flow

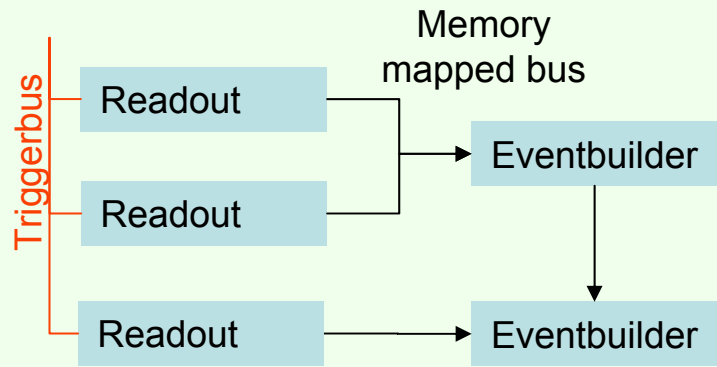

det
adc
FPGA
FPGA
FPGA
tasks
PC
frame buffer
sorting tagging filter analysis
FE
time
DC
DD
FE: Frontend board
DC: Data combiner board
DD: Data dispatcher board
GE: Gigabit Ethernet
IB: InfiniBand
GE
IB
PC
archive
analysis archive
PC
frame buffer
sorting tagging filter analysis
tasks
MBS

MBS topologies

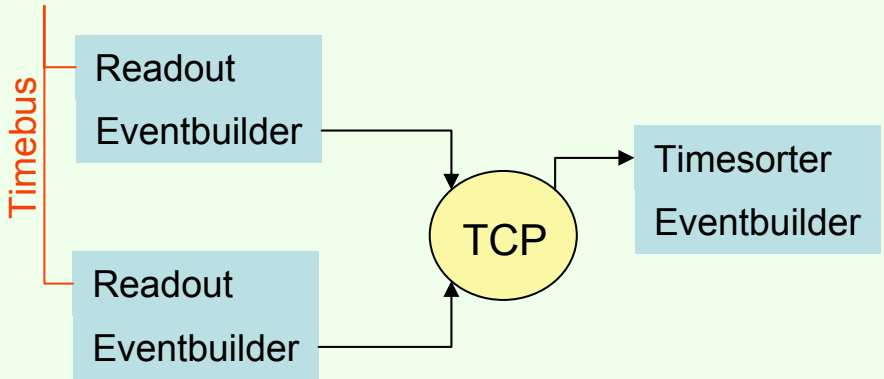
n x m multiple EBs



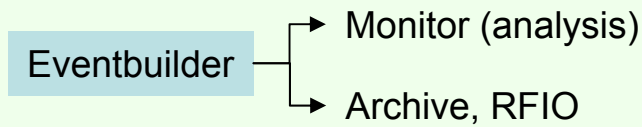
EB hierarchy



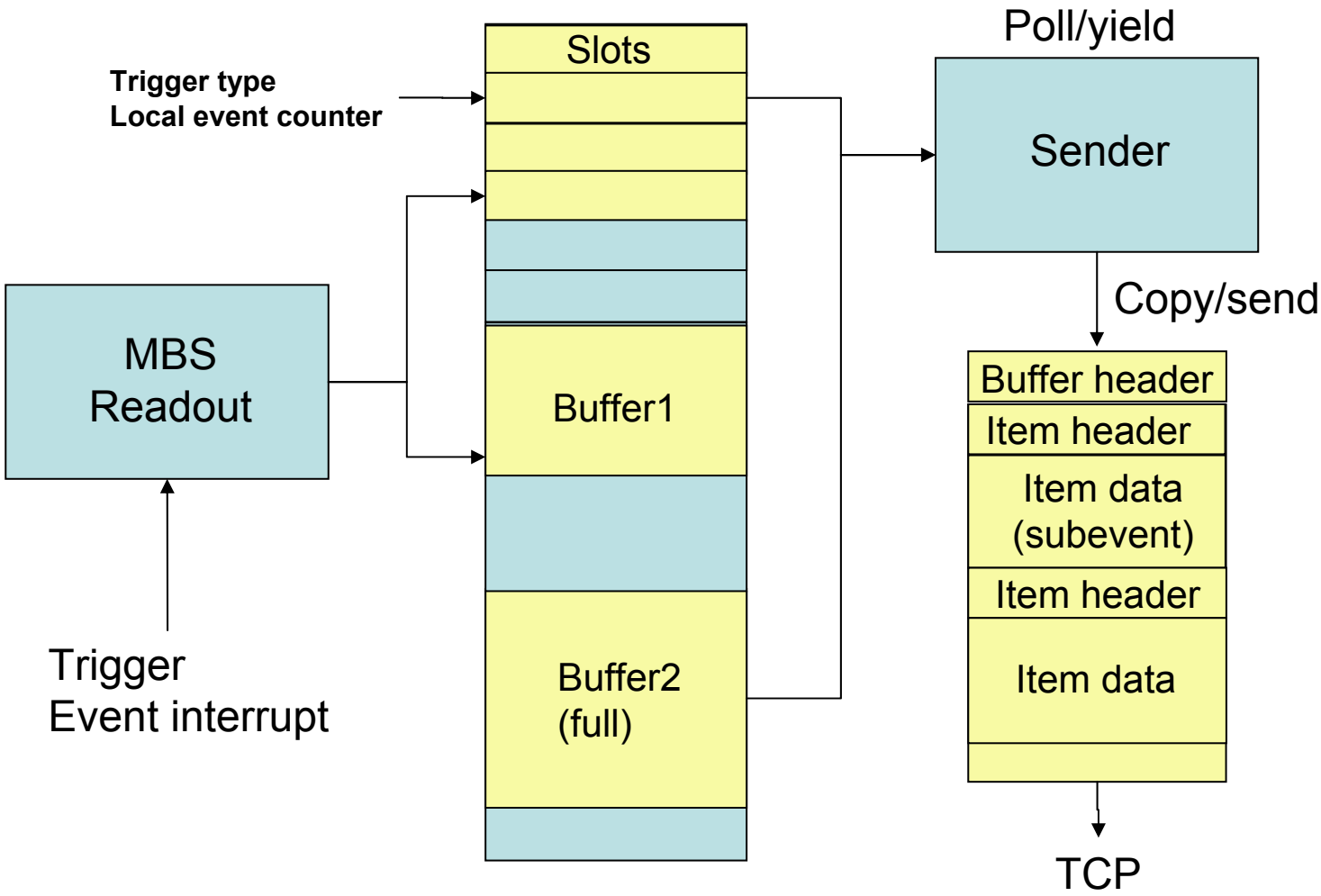
Time stamped



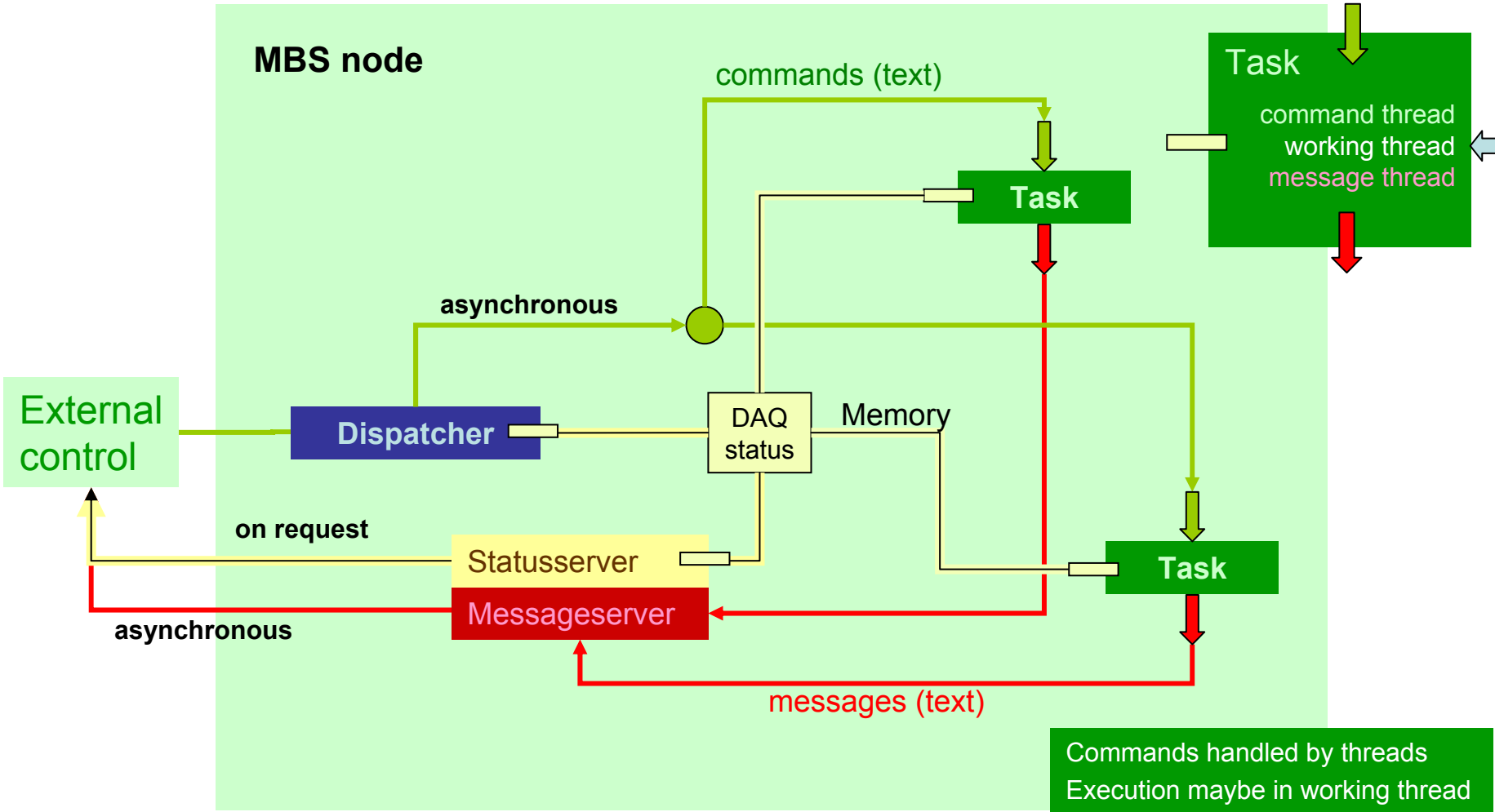
EB data transport



MBS data channel to DABC

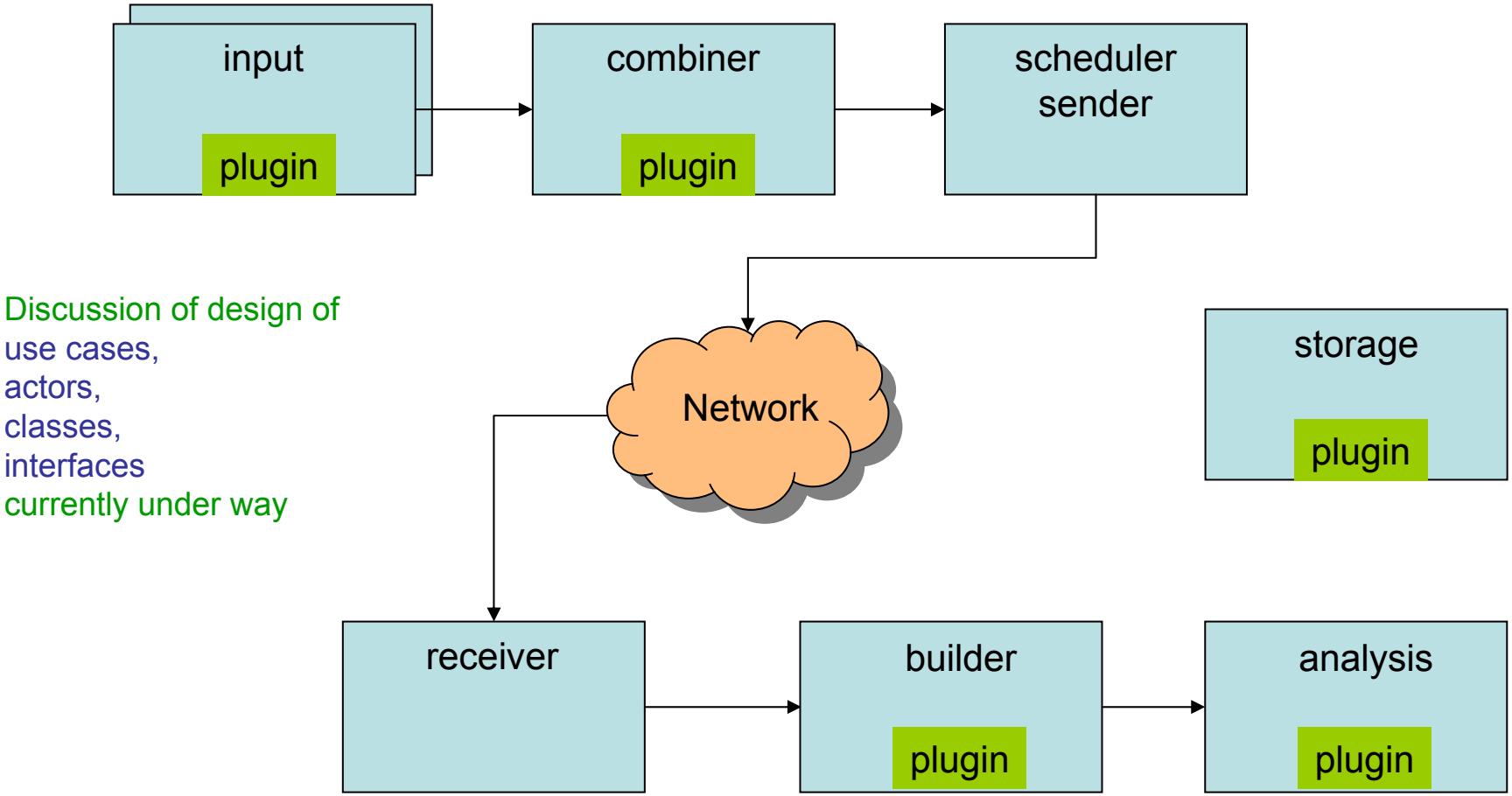


MBS node control and monitor



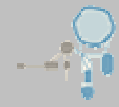
Commands handled by threads
Execution maybe in working thread
Message thread maybe not needed

DABC: schematic view of functional components



Discussion of design of
use cases,
actors,
classes,
interfaces
currently under way

Resources and time



- People of data processing group
  H.G.Essel
  J.Adamczewski
  N.Kurz
  S.Linev
- People of controls group
  maybe one FTE
- People from CBM
  hopefully

- CBM requires in 2007 a data taking system
  Preliminary controls