NUSTAR DAQ

Status of Technical Design Report

Alexandre Charpy

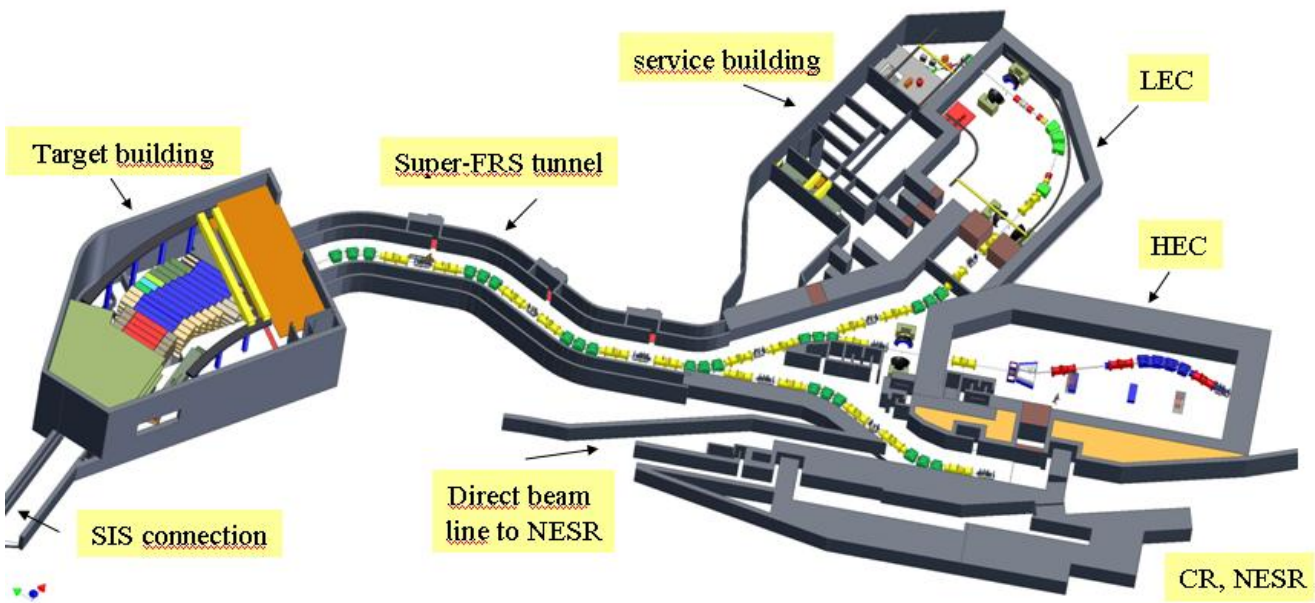
Chalmers University of Technology

NUSTAR Week 2013

Environment

From the Super-FRS to NUSTAR experiments

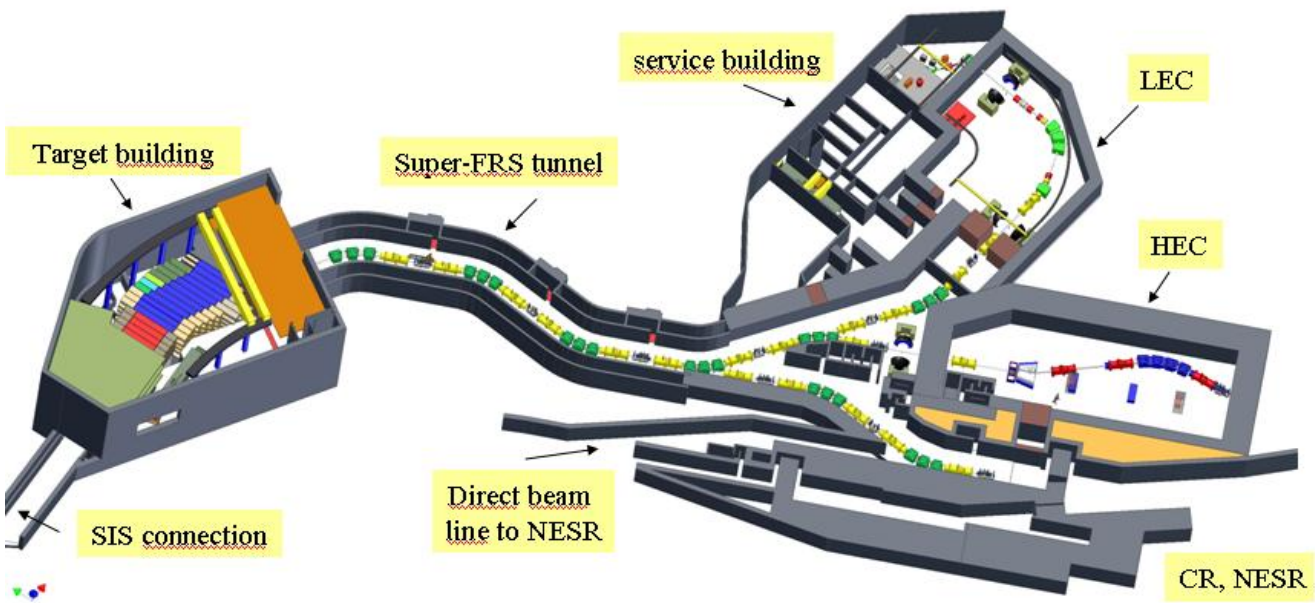
- experiments rely on the Super-FRS detectors for beam identification
- correlate data streams of the Super-FRS and the experiments



NUSTAR DAQ Requirements

Requirements

- fixed NDAQ infrastructure: flexible and low maintenance
- maximum of interoperability

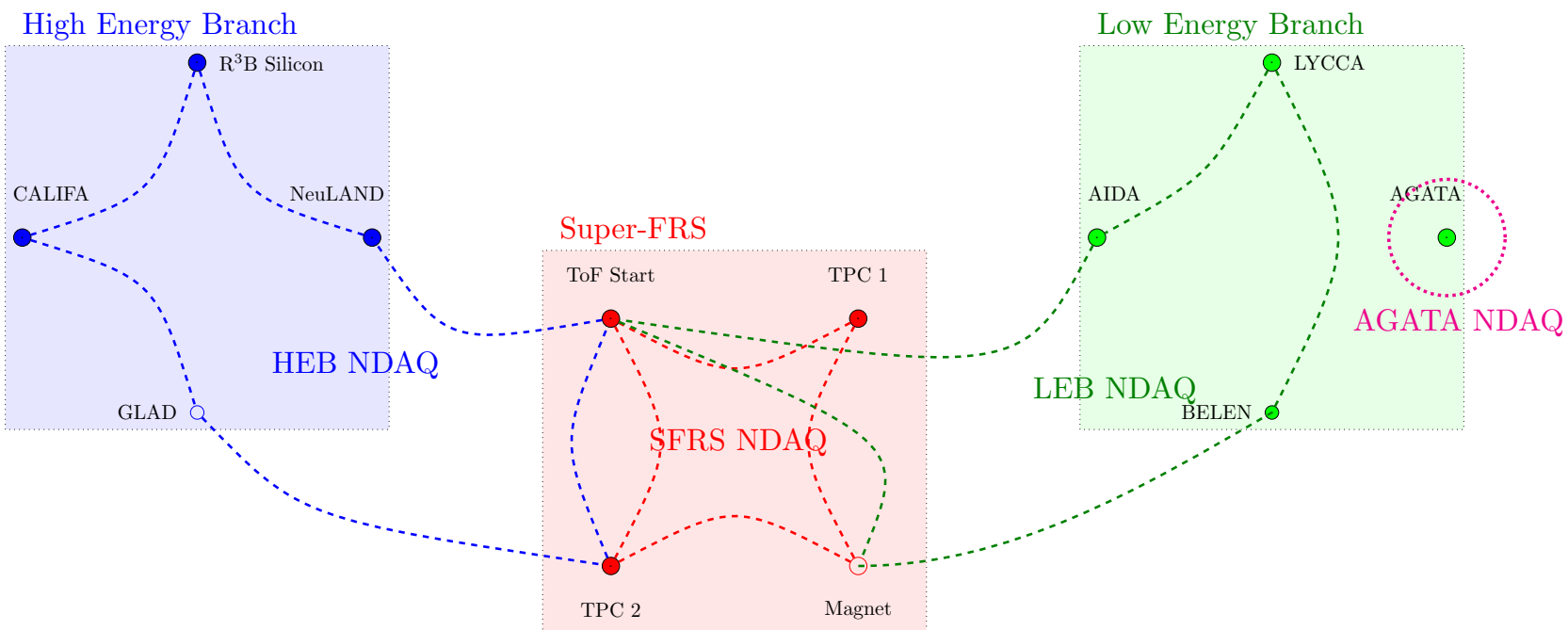


What is NUSTAR DAQ ?

Outlines

Specification based on the DAQ operator and collaborator feedback

- keep the sub-systems independent $\rightarrow$ distributed DAQ
  NDAQ infrastructure = n $\times$ NDAQs
- each NDAQ is related to an experimental data flow

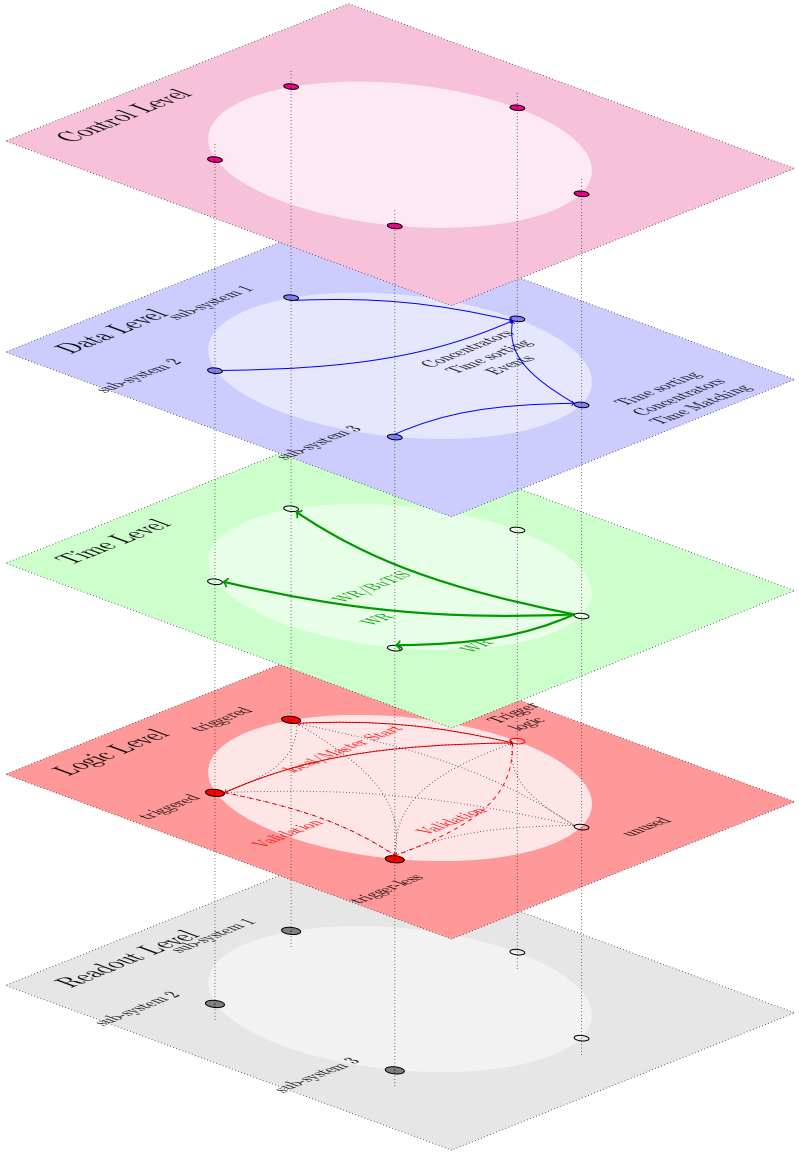


How does a NDAQ operate ?

- an experimental component is connected as a NDAQ node
- every NDAQ consists of a group of nodes

Each NDAQ node interacts along 5 optional interaction levels:

- readout
- logic
- time
- data flow
- control/command

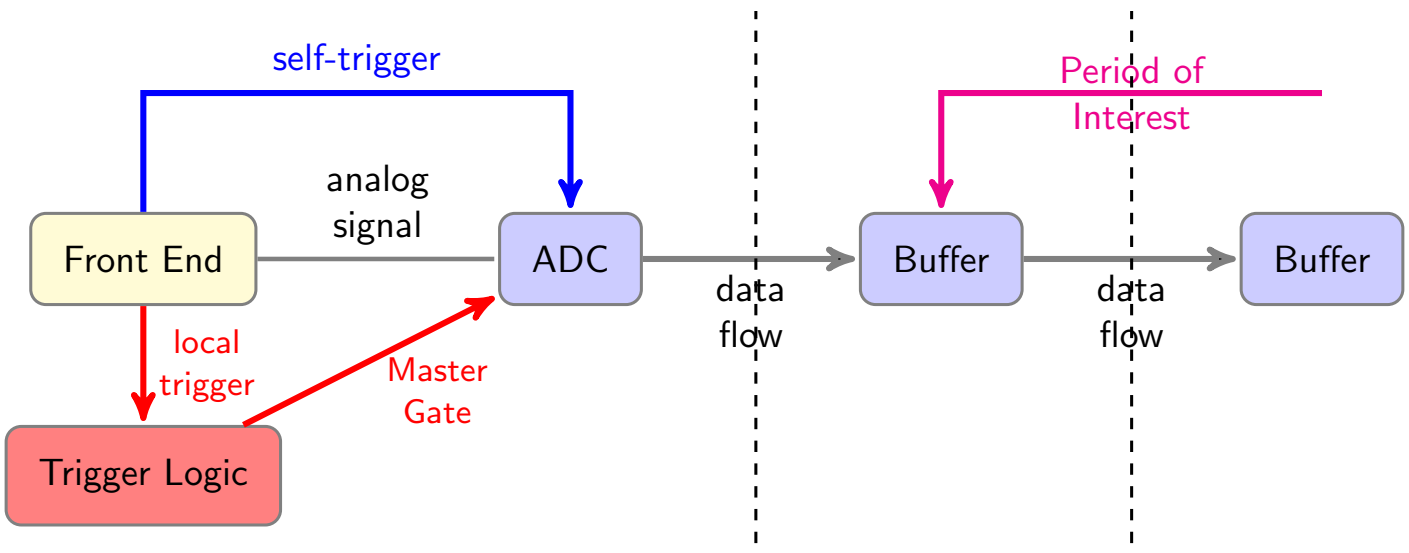


Operating mode

Operating modes

Integrates existing and new detector generations with different operating modes:

- triggered (single or multi-event)
- trigger-less
- hybrid mode (triggered and trigger-less)



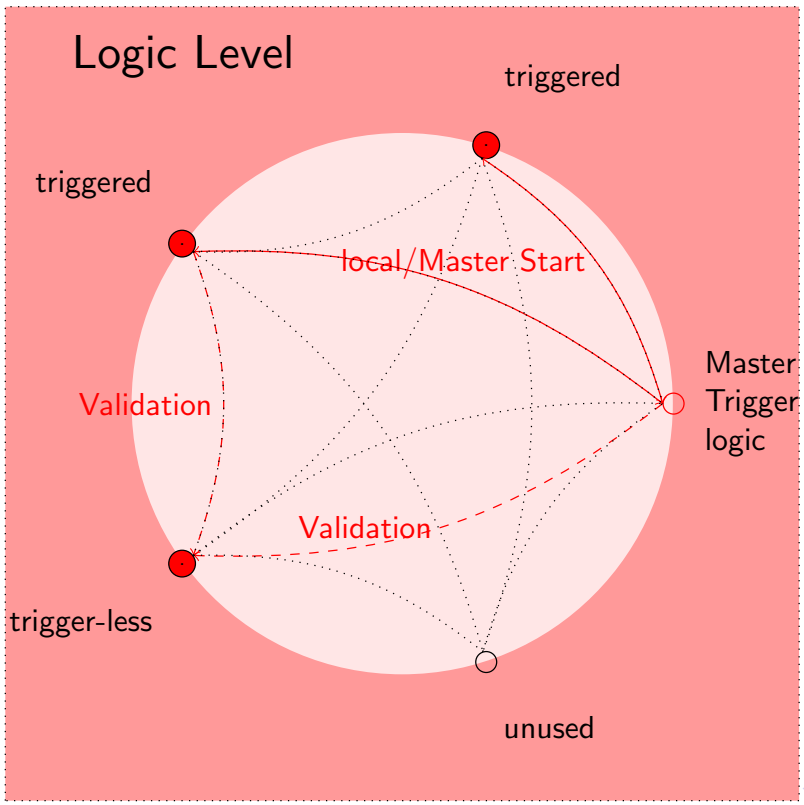
Operating mode

Trigger Logic Topology

A NDAQ needs to be robust and flexible:

- maximum of **permanent** optical inter-connection between NDAQ nodes
- **topology switch requires software commands only**

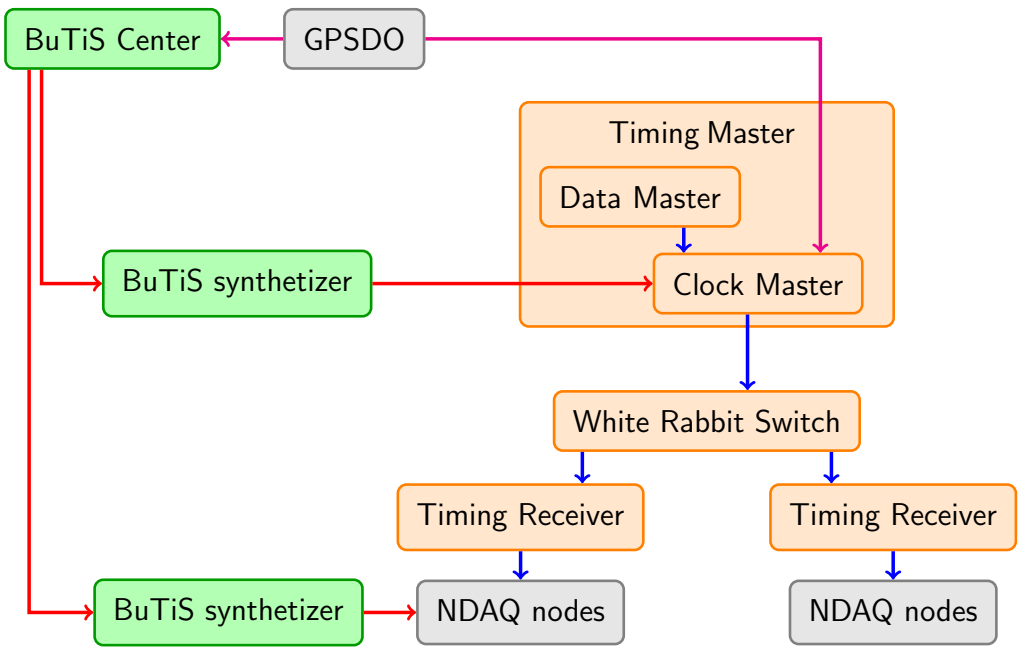
Logic topology customised via FPGAs

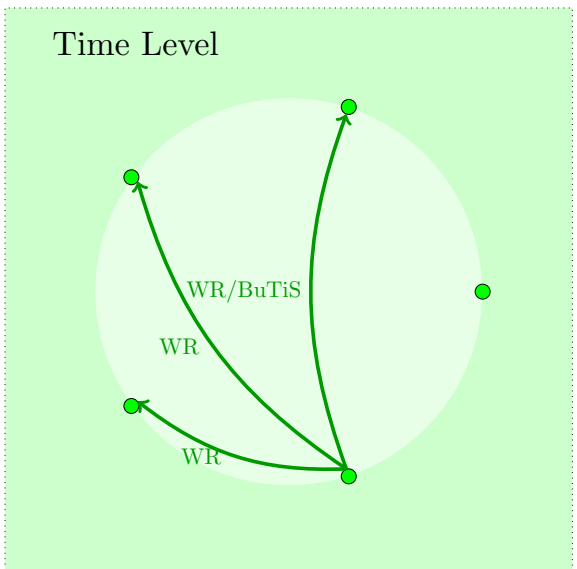


Time Stamping

Time reference system and the NDAQ infrastructure

1. White Rabbit (WR) for an absolute time reference between components:
   - time-stamping events and trigger signals
   - time sorting and merging all meta-data
2. BuTiS for high-accuracy time measurements
   - generation and distribution of a common clock (jitters $\sim$ 100 ps.km^{-1})



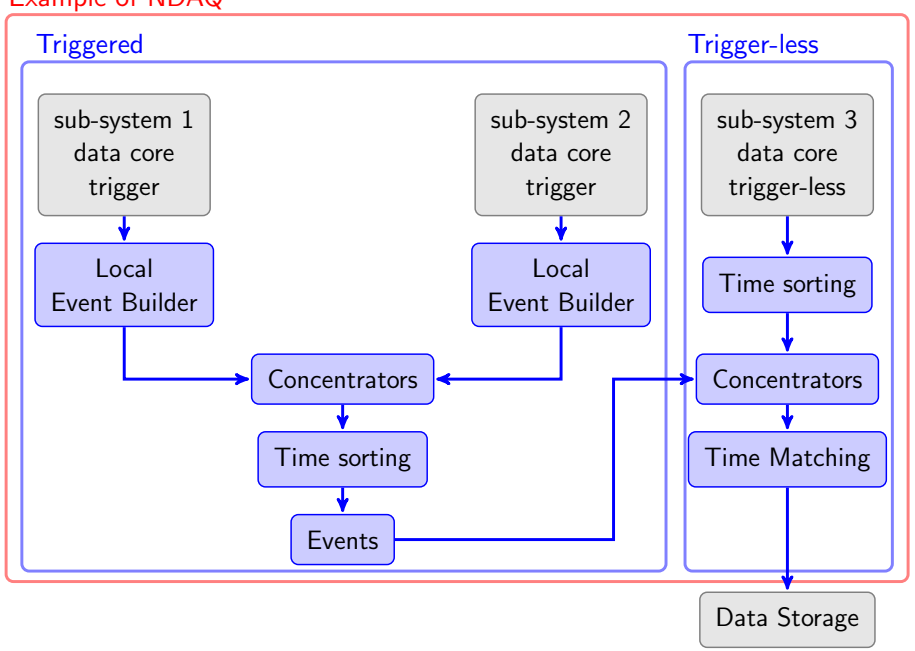


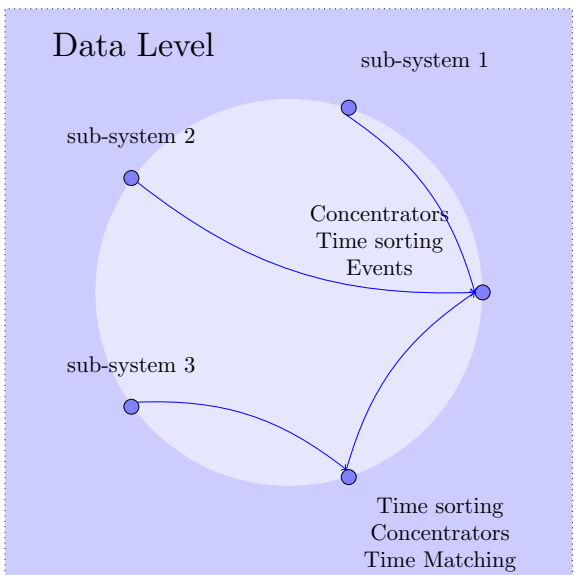
Data Flow ...

Toward list-mode data

- every NDAQ consists of different tasks over the network (*time sorting*, *event matching*, *data concentration/dispatch* ...)
- data flow correlated to the logic layer topology
- dynamic redirection (a NDAQ is running continuously)
- data integrity check based on check-sums performed at every stage

Example of NDAQ

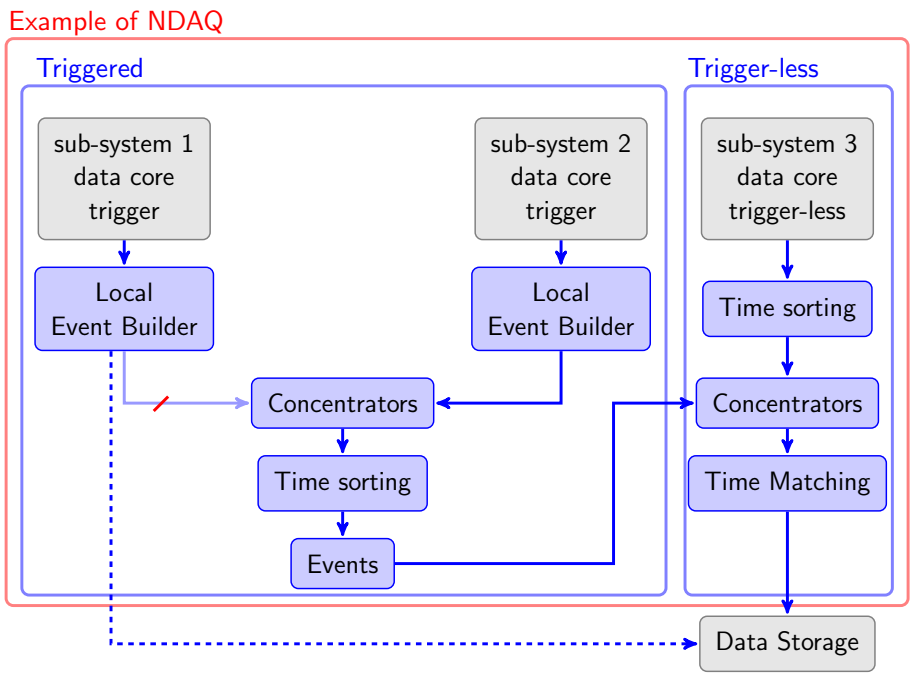


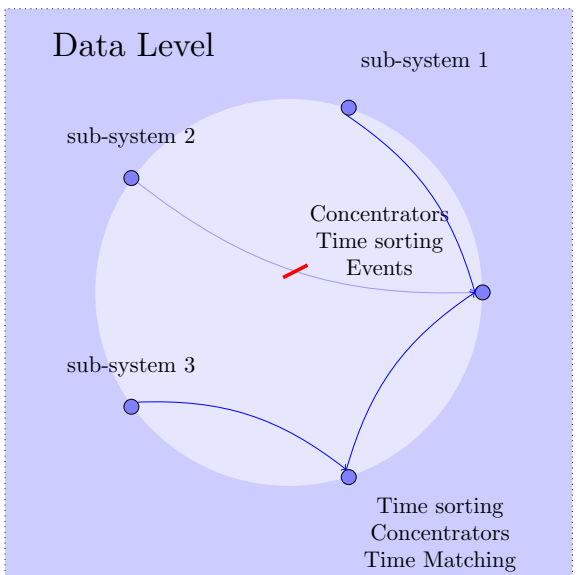


Data Flow ...

Toward list-mode data

- every NDAQ consists of different tasks over the network (*time sorting*, *event matching*, *data concentration/dispatch* ...)
- data flow correlated to the logic layer topology
- dynamic redirection (a NDAQ is running continuously)
- data integrity check based on check-sums performed at every stage





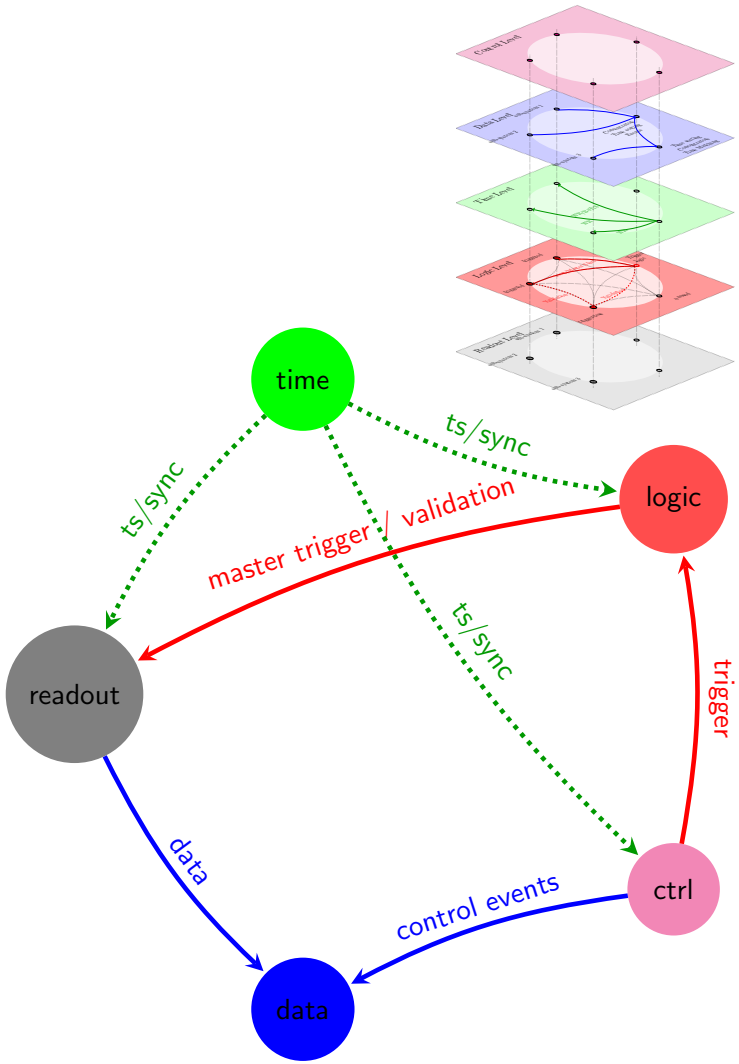
.. to meta-data flow

Control Events

- *events* affecting analysis (e.g. accelerator settings, data flow topology)
- time-ordered and *sticky*
- allow for calibration routines (e.g. ramping a magnet)

Requirements

- Control Event data use the same stream as raw data
- raw data format to build a coherent file structure

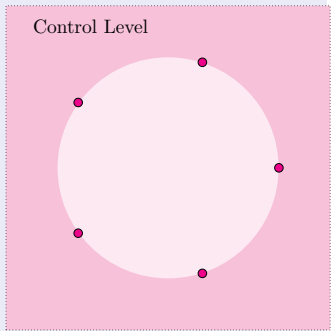


Slow Control

General

Slow Control is detector specific

- monitor and adjust crucial detector/component parameters
- prevent fatal hardware failures
- restore parameter values



NDAQ properties

- EPICS will support the NDAQ slow control infrastructure
- open system $\rightarrow$ user can provide open access
- propose advanced features (e.g. automatic modification as function as time)

Monitoring

Monitoring

- allow to visualise the logic and data flow topology with their monitored parameters
- local and global dead-time of the connected logic cores
- back pressure at every stage (buffer occupancy, CPU usage, bandwidth, ...)
- distribution/notification of Control & Command messages
- data integrity before event building
- display the sub-systems status (warnings, errors ...)
- on/off-line analysis is responsibility of experimentalists
- web-application

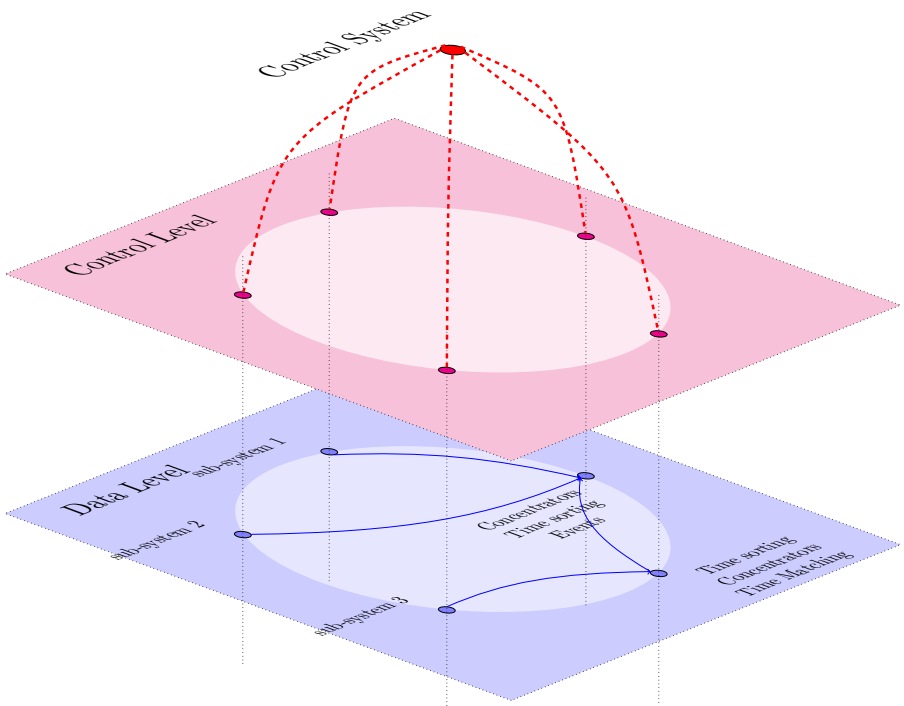
Run Control

Control & Command

- NDAQ nodes are configured individually
- NDAQ nodes are configured via **setup files**
- generate "control events"

NUSTAR Run Control

- client related to one NDAQ
- unify global control & command



Collaborative work

Contacts (so far)

- A. Charpy, A. Chatillon, R. Gernhaeuser, A. Heinz, H.T. Johansson, N. Kurz, I. Lazarus, T. Le Bleis, B. Loeher, C. Nociforo, S. Pietri, C. Pucknell, H. Schaffner, H. Simon

Other active parties are welcome!!!

Conclusion

1. NDAQ infrastructure provides a unified DAQ solution
2. specification defined according to the DAQ operators and to the needs of experimentalists
3. new features (DAQ always running, setup changes in-flight, control event...)
4. documentation in progress $\rightarrow$ input and suggestions are very appreciated