university of groningen

kvi - center for advanced radiation technology

FEE and DAQ Protocols

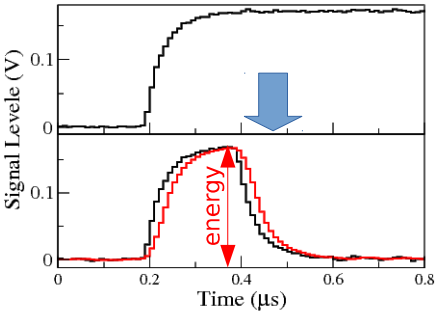
M. Kavatsyuk

KVI-CART, University of Groningen

for the PANDA collaboration

Readout Approach for PANDA

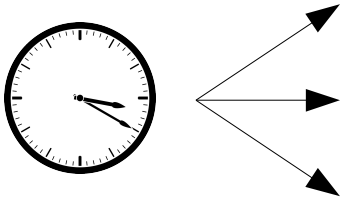
The PANDA readout consist of:



- **Intelligent self-triggered front-end**:
  autonomous hit detection and data pre-processing (e.g. based on **S**ampling **A**nalogue to **D**igital **C**onverter)

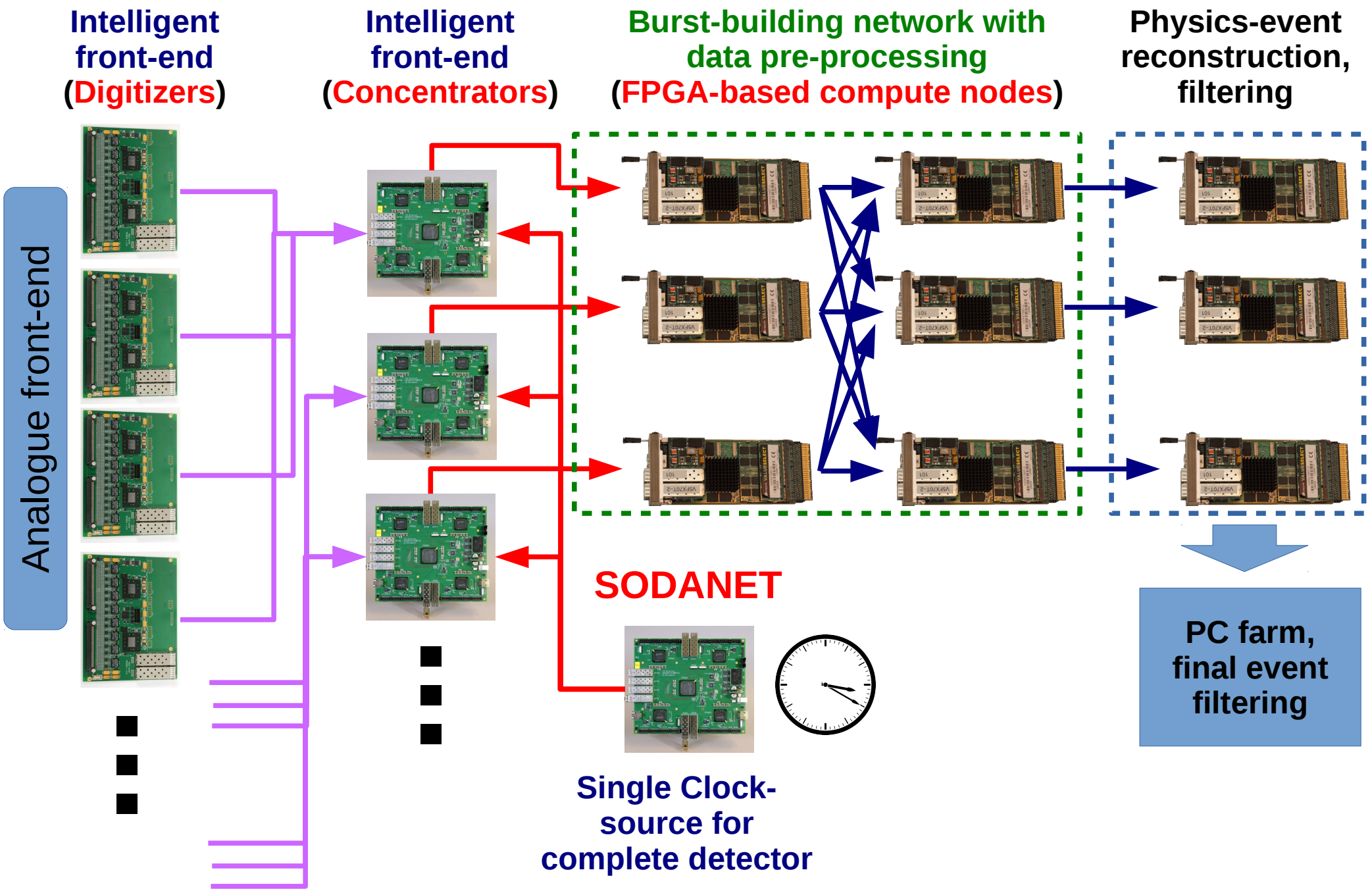
100101101

- **a very precise time distribution system (SODANET)**:
  single clock-source for PANDA (event correlation)

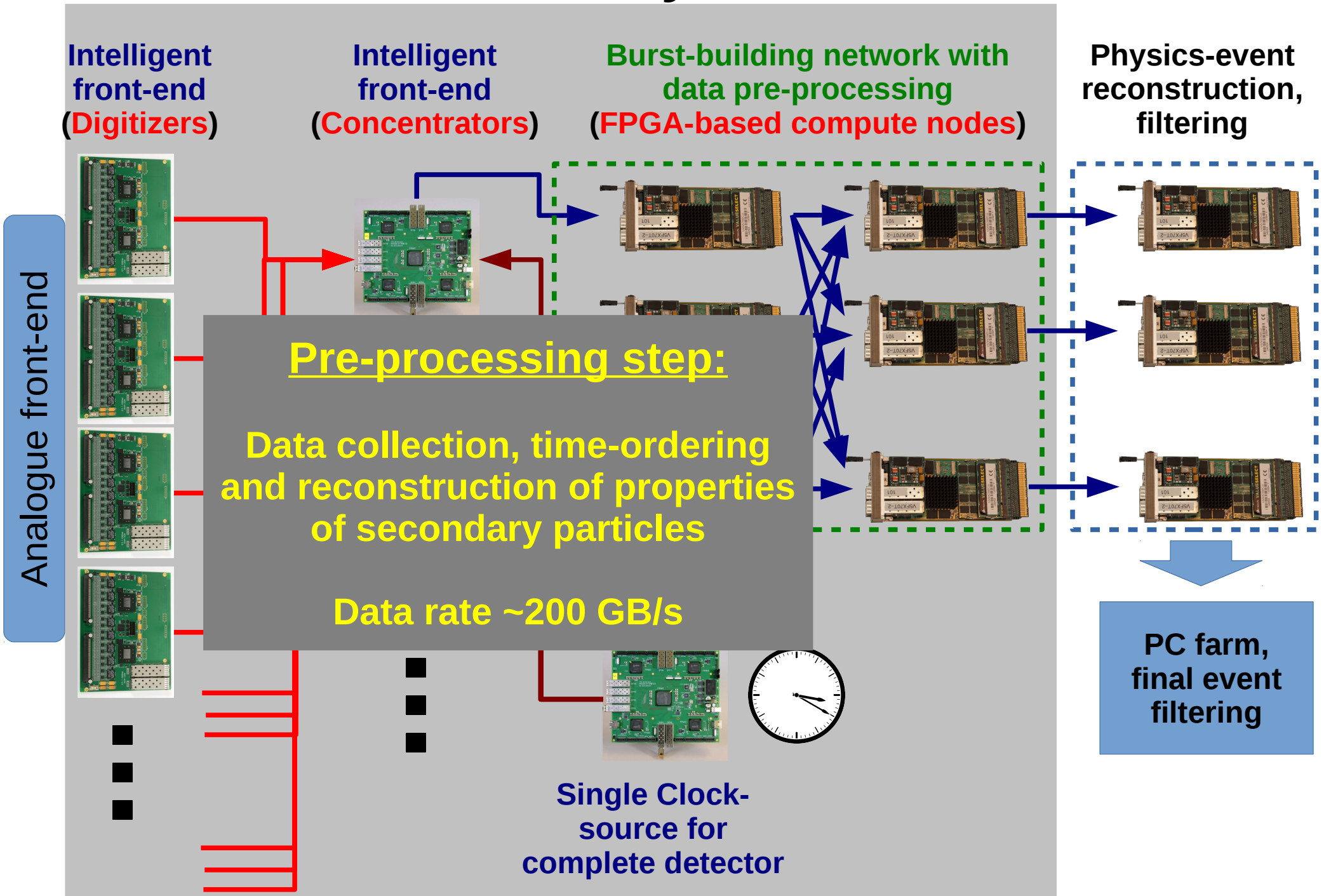


- **time-sorting and processing data in real-time**:
  processing in FPGA (**F**ield-**P**rogrammable **G**ate **A**rray)

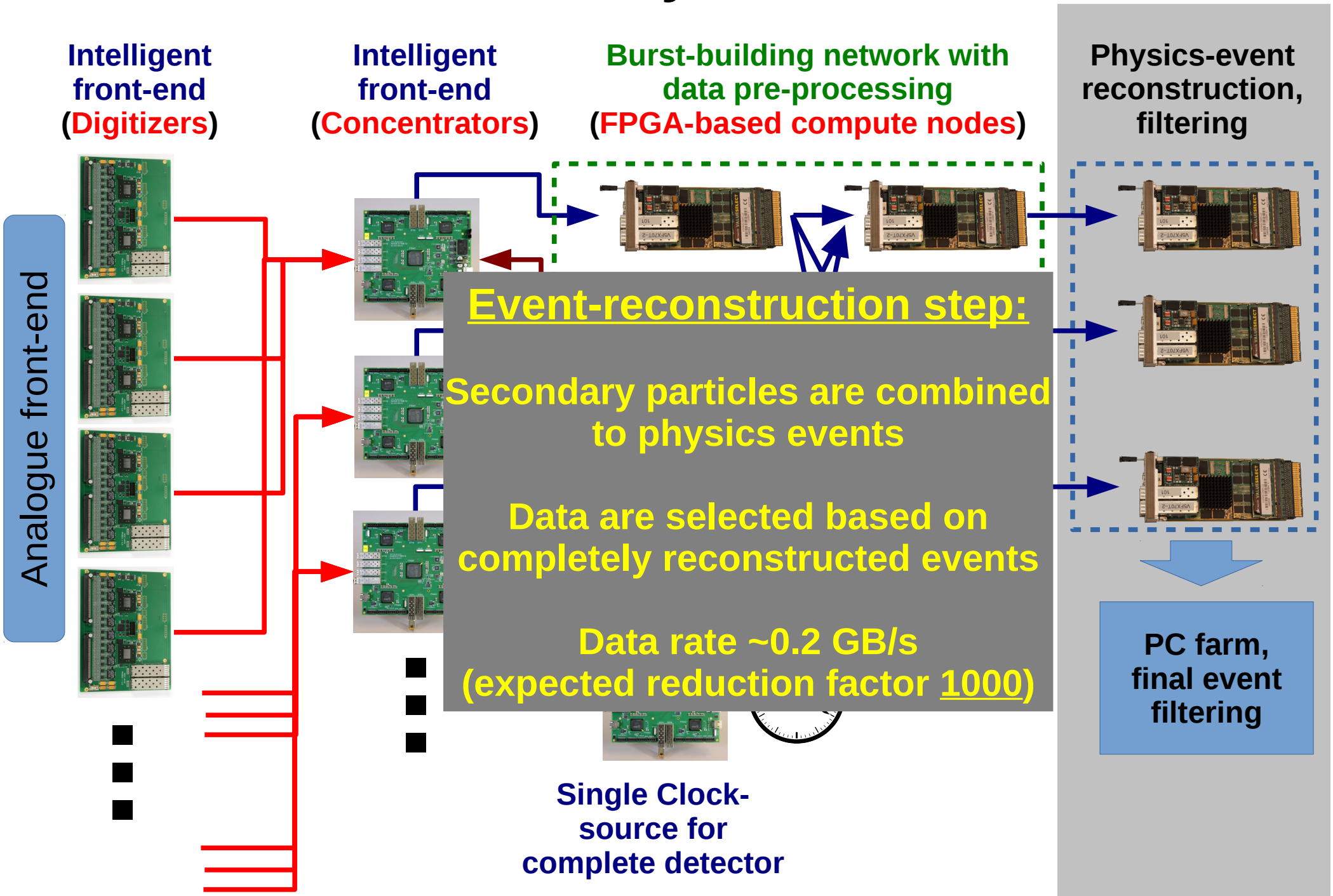
Push-Only Readout



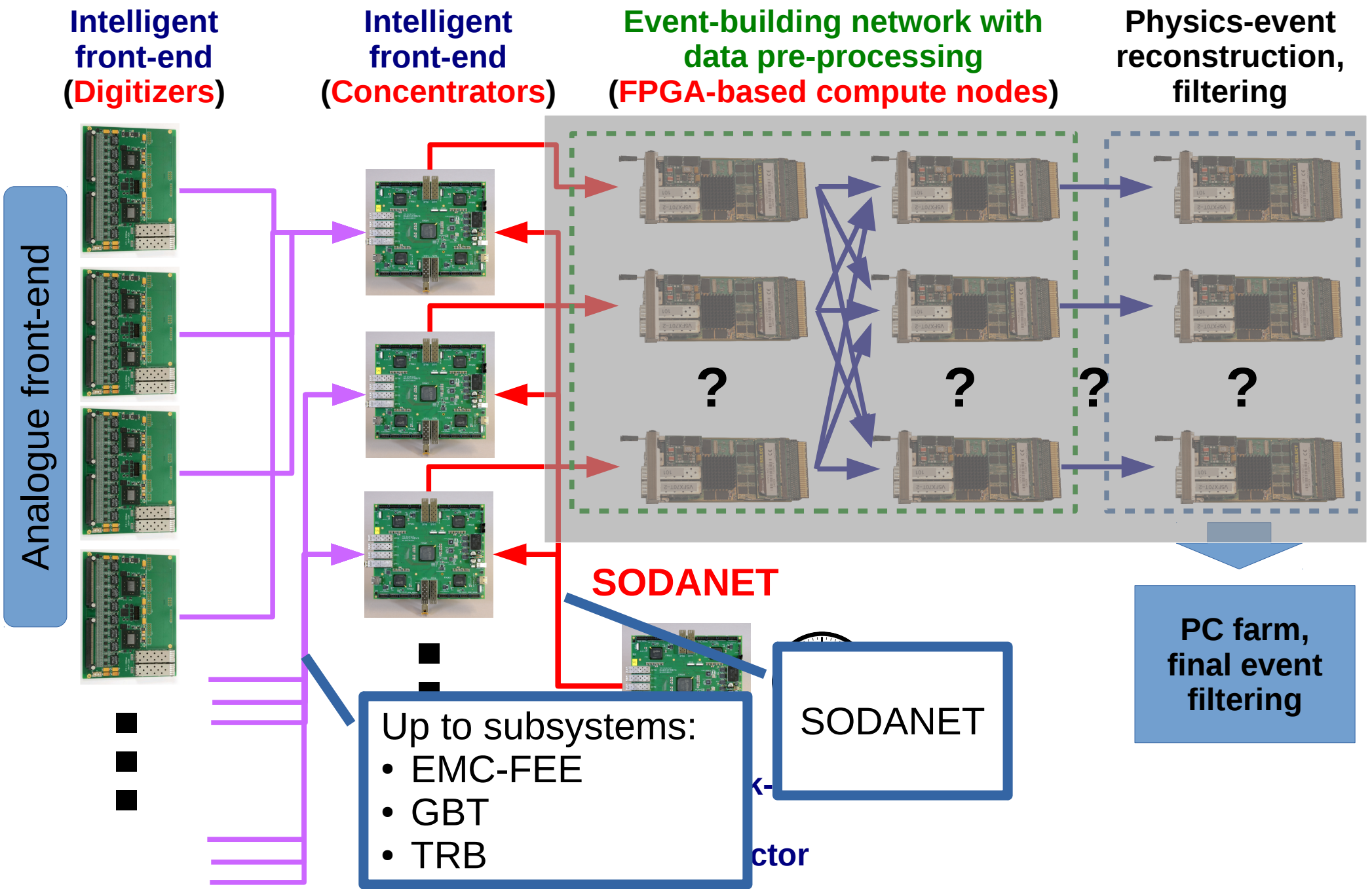
Push-Only Readout



Push-Only Readout



Protocols

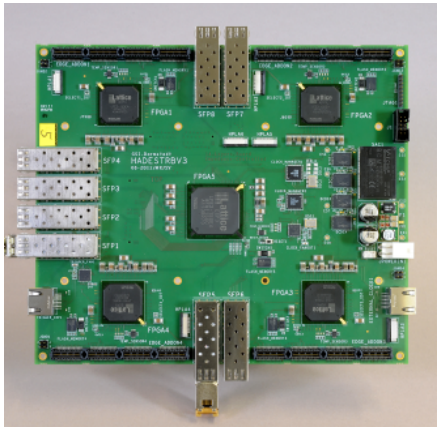

Intelligent front-end (Digitizers)
Intelligent front-end (Concentrators)
Event-building network with data pre-processing (FPGA-based compute nodes)
Physics-event reconstruction, filtering
Analogue front-end
?
?
?
?
SODANET
PC farm, final event filtering
Up to subsystems:
• EMC-FEE
• GBT
• TRB
SODANET

SODANET Protocol

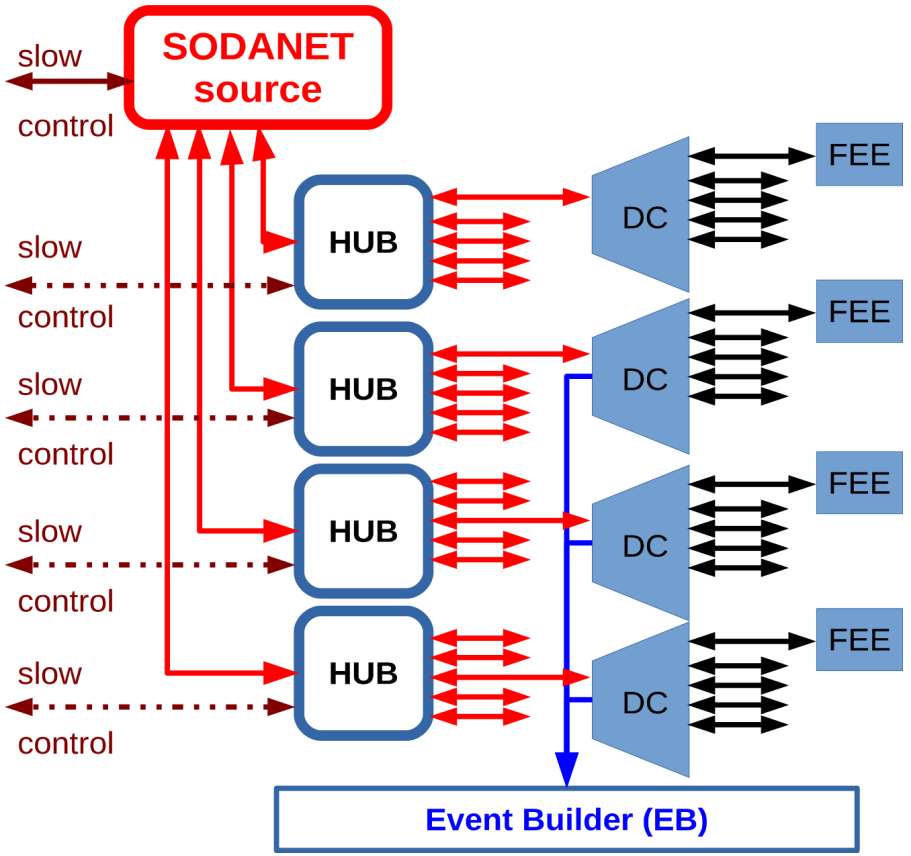
All data and clock are transferred using optical link (speed multiple to 40 MHz clock, e.g. 2 Gbs, 2.4 Gbs, 4.8 Gbs...):

- All SERializes-DE-Serializes (SerDeS) are working in synchronous mode: parallel clock has defined phase with respect to bit #1 of a serial clock
- Time-synchronization commands can interrupt low priority transmission of a slow-control package
- Synchronous commands are identified by the receiving side by special K-characters (FB)
- Synchronization commands are regular and define periods which are named **“Super-bursts”** (related to the timing of accelerator)

- TRBNET protocol [J. Michel, PhD thesis, University of Frankfurt, 2012] is used to transfer slow-control data and TRB v3 hardware is used for the development of the SODANET protocol



The SODANET



Implementation

The SODANET protocol implemented on several FPGA platforms:

- Lattice ECP3
- Xilinx Kintex 7
- Xilinx Virtex 6

Verification

- Stability of the clock phase after reset/power cycle of the optical link
- Synchronisation of several FEE modules (using “SB start” command)
- Long-term stability of the system

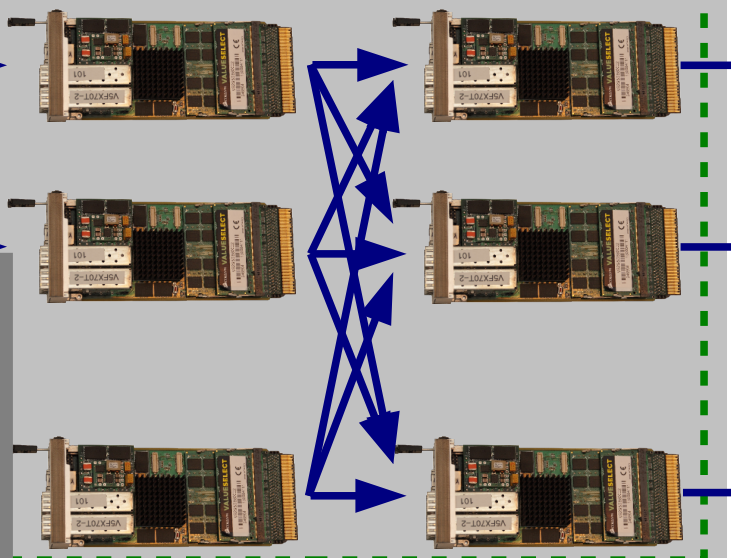
Push-Only Readout

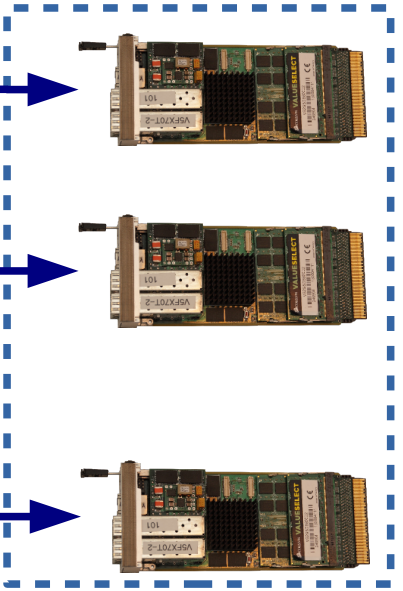
Intelligent front-end (Digitizers)

Intelligent front-end (Concentrators)

Burst-building network with data pre-processing (FPGA-based compute nodes)

Physics-event reconstruction, filtering





Burst building:

Collecting data with the same burst number in the same CN:

- **Make sure that all data is collected**
- **Monitor functionality of the readout**

Necessary step at any rate!
It is not event building!

gle Clock-
ource for
lete detector

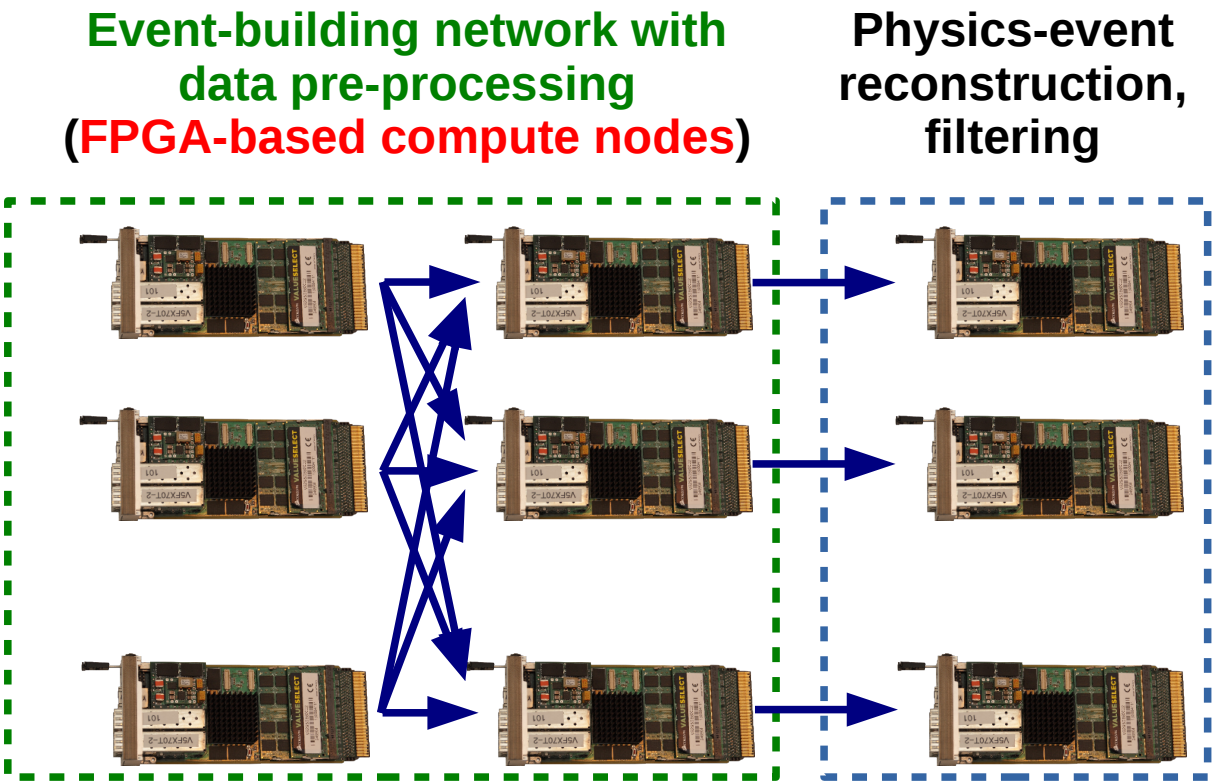
PC farm, final event filtering

Burst and Event building

Burst-building communication protocols:

- Defined:
  - Input stream
- Not Defined:
  - Communication between nodes
  - Network topology
  - Slow control
  - Output protocol

We should define all protocols and IOs for data-processing IP-cores which might run on the nodes (e.g. pre-clustering, time ordering)



Event building is performed after particle reconstruction (CN stage or PC farm)

Next steps

Burst-building network:

- Define protocols (data, control)
- Define interfaces for the standard data-processing IP-cores for the burst-building network:
  - Acquire requests from all sub-systems information on required data processing (e.g. pre-clustering, at least time-ordered merging of streams)

Event building:

- Define protocol between the burst-building network and compute nodes (where final particle reconstruction will take place)
- Define network topology (static, dynamic with load-based distribution)
- Define interface to the PC farm