Hardware Acceleration
LAB

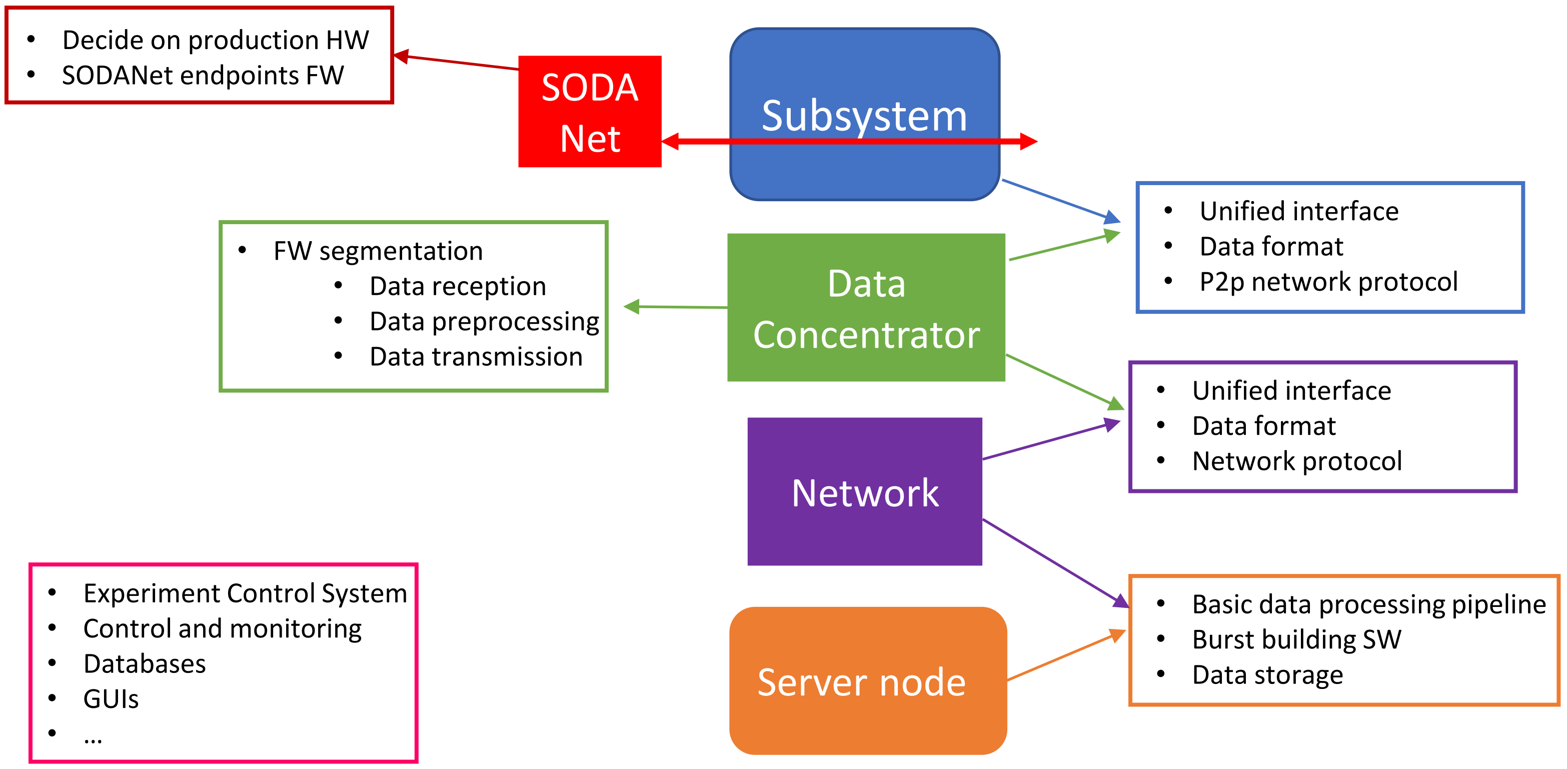
PANDA
FEE/DAQ Workshop 2021
Summary

Grzegorz Korcyl

Panda Collaboration Meeting 21/3

27.10.2021

Towards DAQ-0



- Decide on production HW
- SODANet endpoints FW

- FW segmentation
  - Data reception
  - Data preprocessing
  - Data transmission

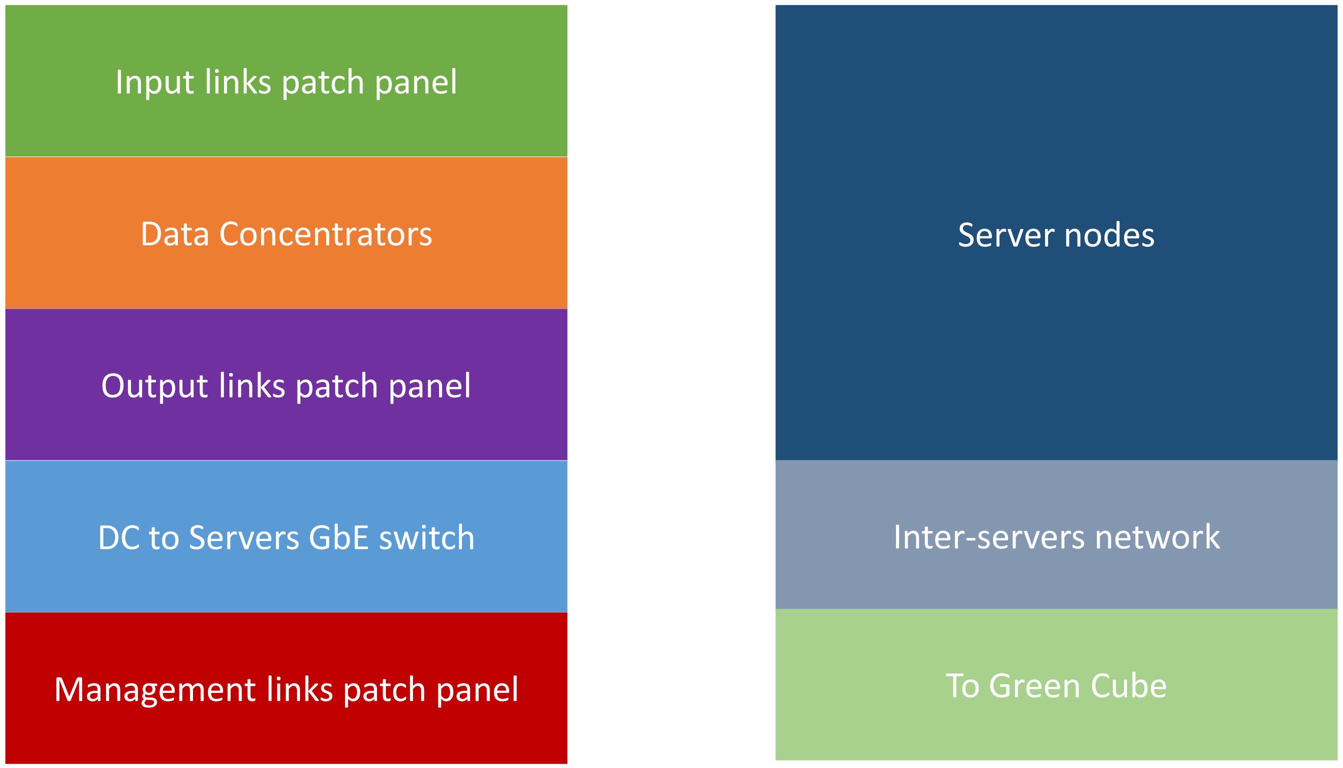
- Unified interface
- Data format
- P2p network protocol

- Unified interface
- Data format
- Network protocol

- Basic data processing pipeline
- Burst building SW
- Data storage

- Experiment Control System
- Control and monitoring
- Databases
- GUIs
- …

System composition



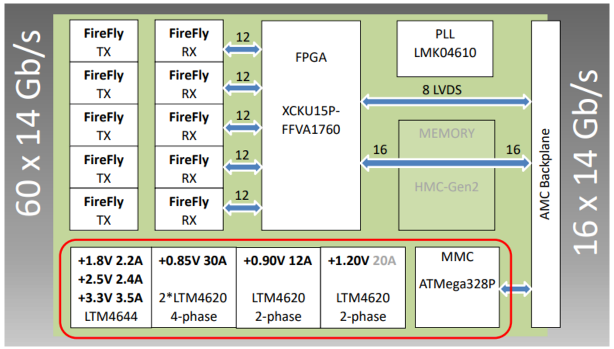
Subsystems Overview

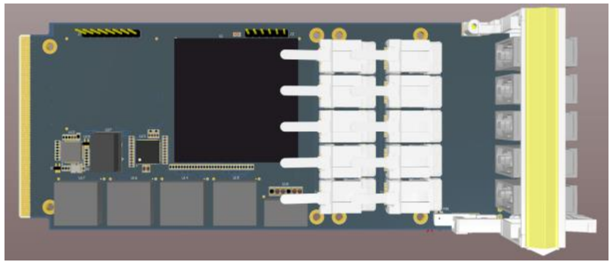
FT			
PASTTREC (352)	TRB5SC (88)	Crate Master (10)	DC (1-?)
STT			
PASTTREC (270)	TRB5SC (68)	Crate Master (8)	DC (1-?)
MVD			
TOAST ()	MDC (296)		DC (5-?)
Endcap Disc DIRC			
TOFPET2	RDB (96)		DC (2-?)
Barrel DIRC			
DIRICH	TRB3/TRB5SC		DC (?)
EMC			
SADC	Crate Controller (41)		DC (1-?)
HDA	LVDS DC (40)		DC (1-?)
Lumi			
MuPIX	LVDS DC ()		DC (?)

Data Concentrators

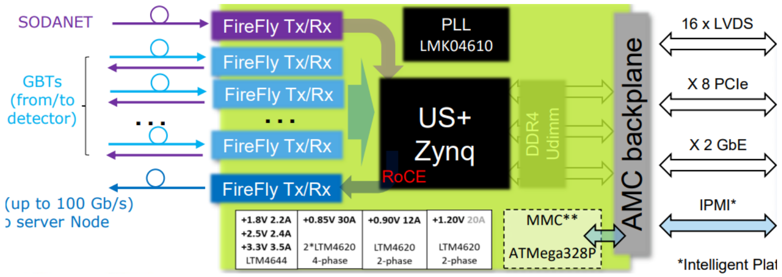
- ATCA – AMC platform
  - 60x bidirectional optical links (FireFly)
  - Backplane I/O for management and board-board communication

  - Processing unit:
    - **Xilinx Kintex: FPGA-DC**
      - Regular FPGA resources
    - **Xilinx Zynq MPSoC: SoC-DC**
      - Regular FPGA resources
      - Quad-core ARM Cortex A53
      - Dual-core ARM Cortex R5F
      - Mali 400MP2 GPU



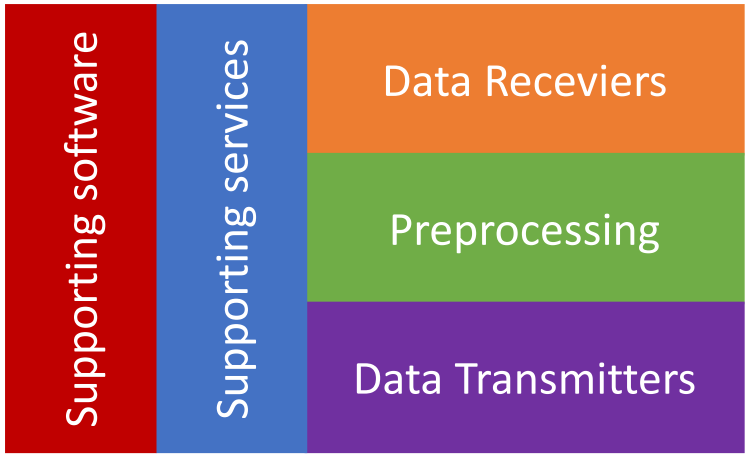


P. Marciniewski



M. Caselle

DC Firmware



Particular DC firmware has to be tailored to a given subsystem:

- In/Out link configurations (resource balance)
- Preprocessing algorithms
- Supporting services

Data receivers region:

- Logic common to all DCs (configurable num. of links)
- Common link type and protocol
- Unified interface to Preprocessing region

Preprocessing region:

- Fixed, unified data in and out interfaces
- Region available to Subsystem devs.

Data transmitters region:

- Logic common to all DCs (configurable num. of links)
- Unified interface to Preprocessing region
- Commercial network interface

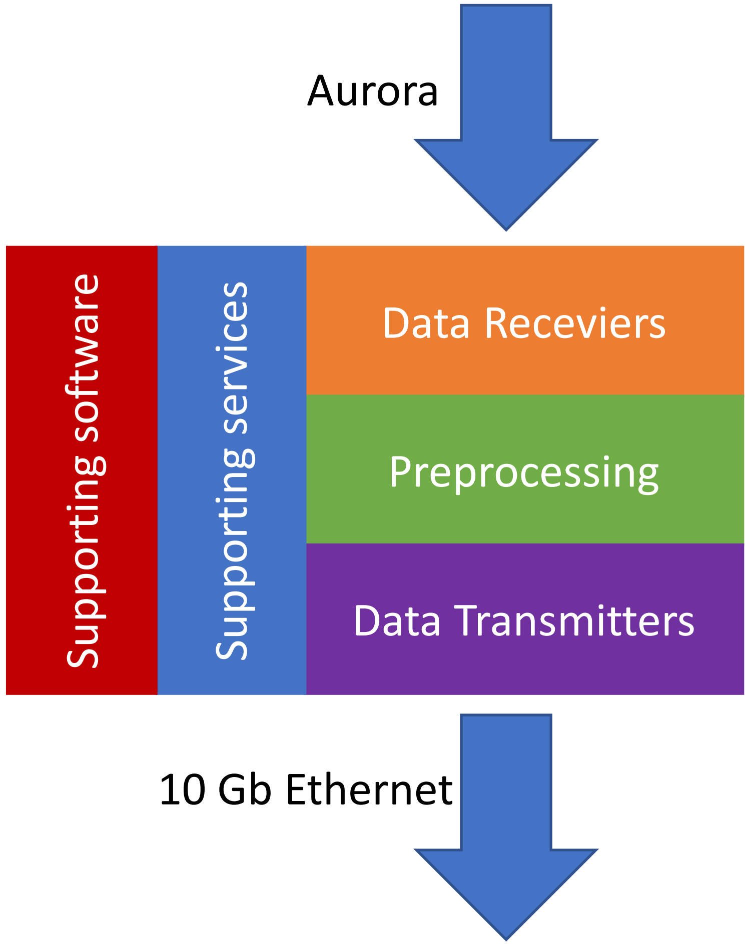
Supporting services:

- Control and monitoring
- SODANet
- ATCA management
- …

Supporting software:

- Board management
- Data QA
- …

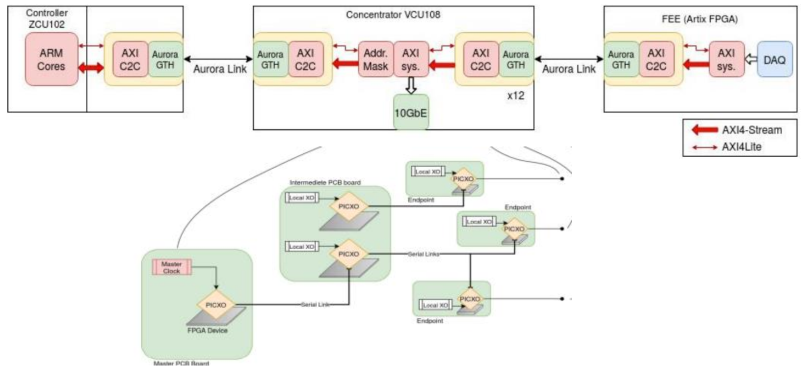
DC Protocols



Data Receivers:

- All Subsys. end up with Xilinx FPGAs
- Wide range of link speeds
- Aurora IPs are well tested and easy to use
- Lightweigth
- Can transport AXI transactions
- Can be synchronous

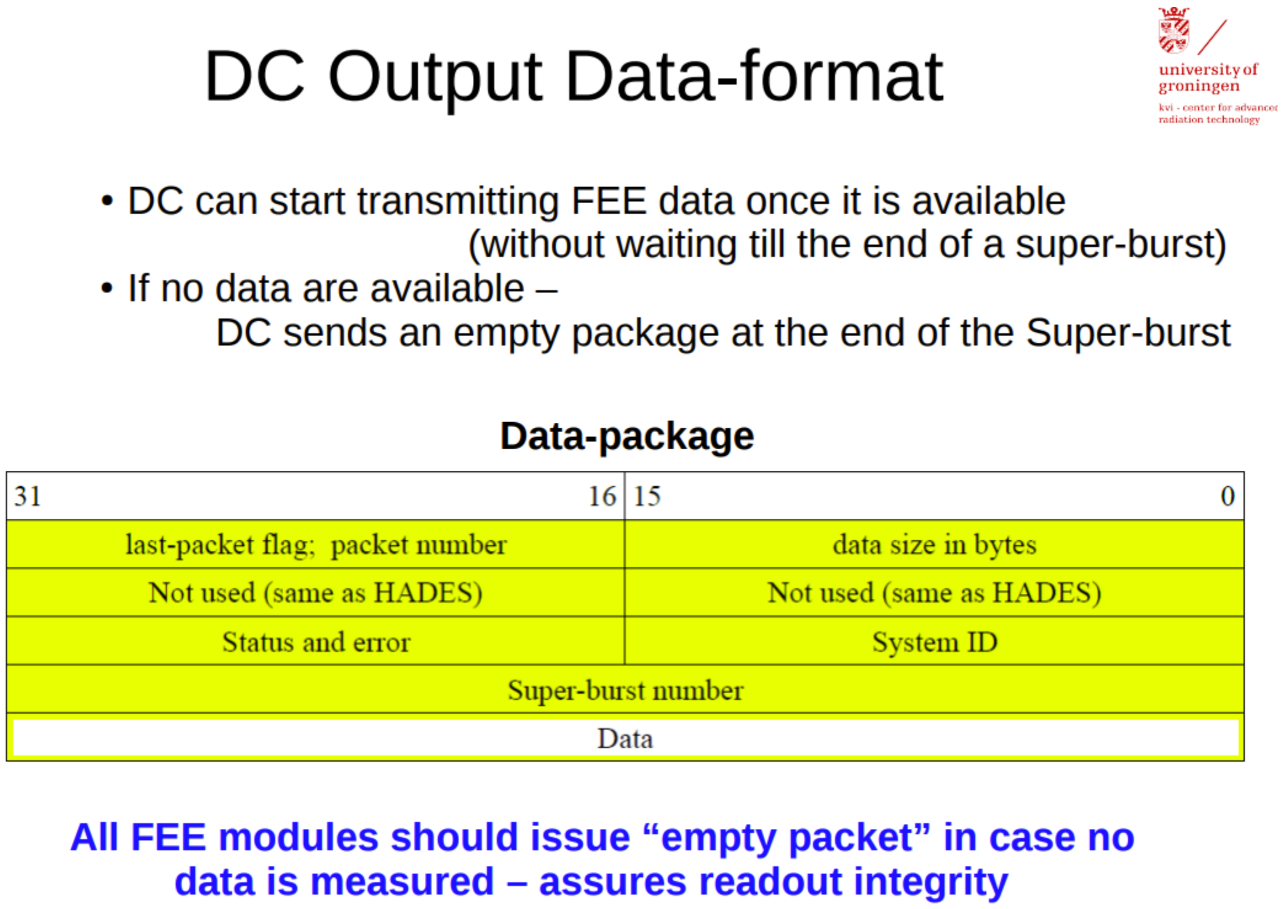
M. Bakalarek talk



Data Transmitters:

- Common networking standard
- Easy to set up a simple UDP blaster for basic operation and evaluation of the readout chains

Data Formats



DC Output Data-format

- DC can start transmitting FEE data once it is available (without waiting till the end of a super-burst)
- If no data are available – DC sends an empty package at the end of the Super-burst

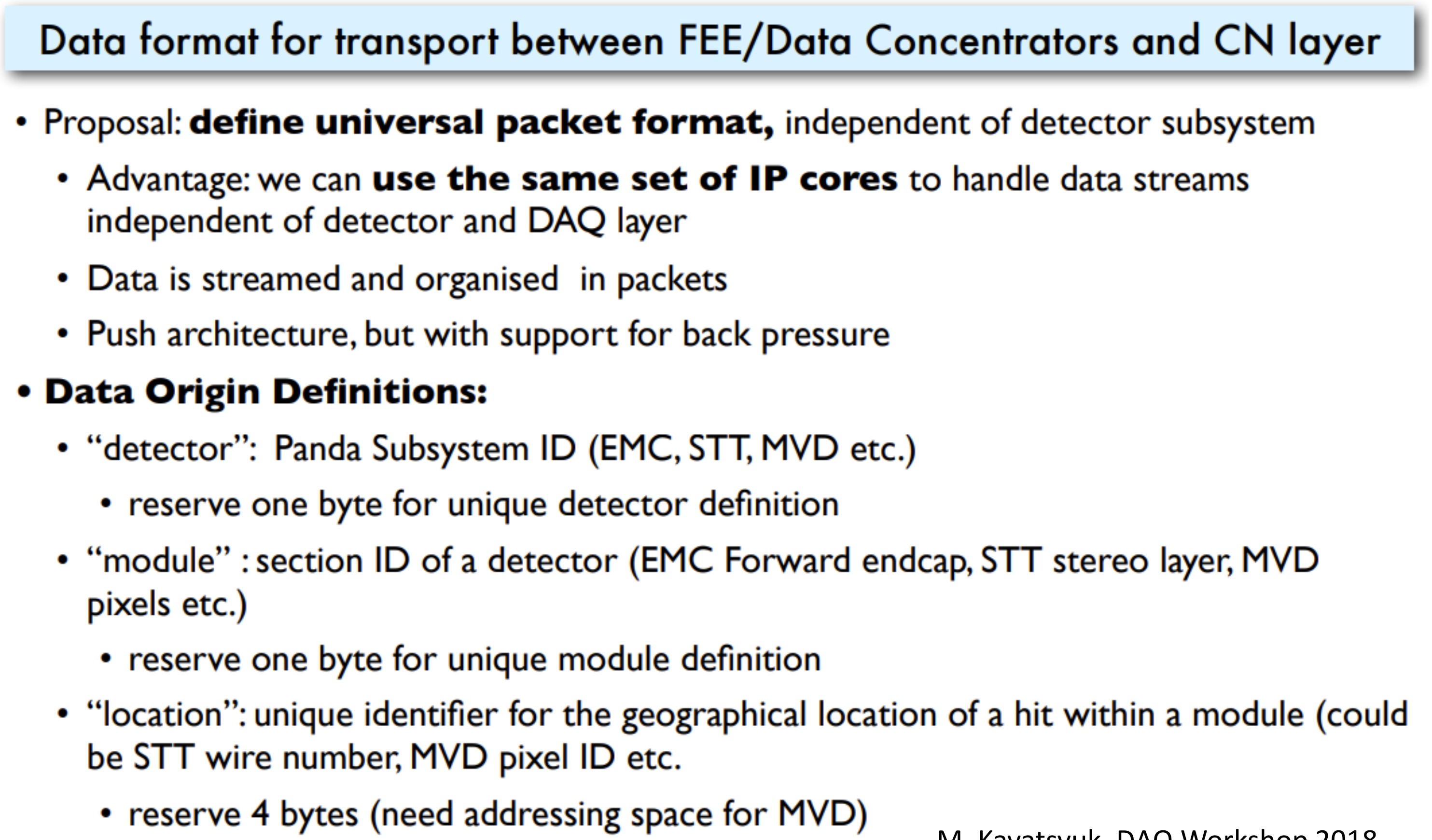
Data-package

31 … 16	15 … 0
last-packet flag; packet number	data size in bytes
Not used (same as HADES)	Not used (same as HADES)
Status and error	System ID
Super-burst number	
Data	

All FEE modules should issue "empty packet" in case no data is measured – assures readout integrity

M. Kavatsyuk, DAQ Workshop 2018

Data formats

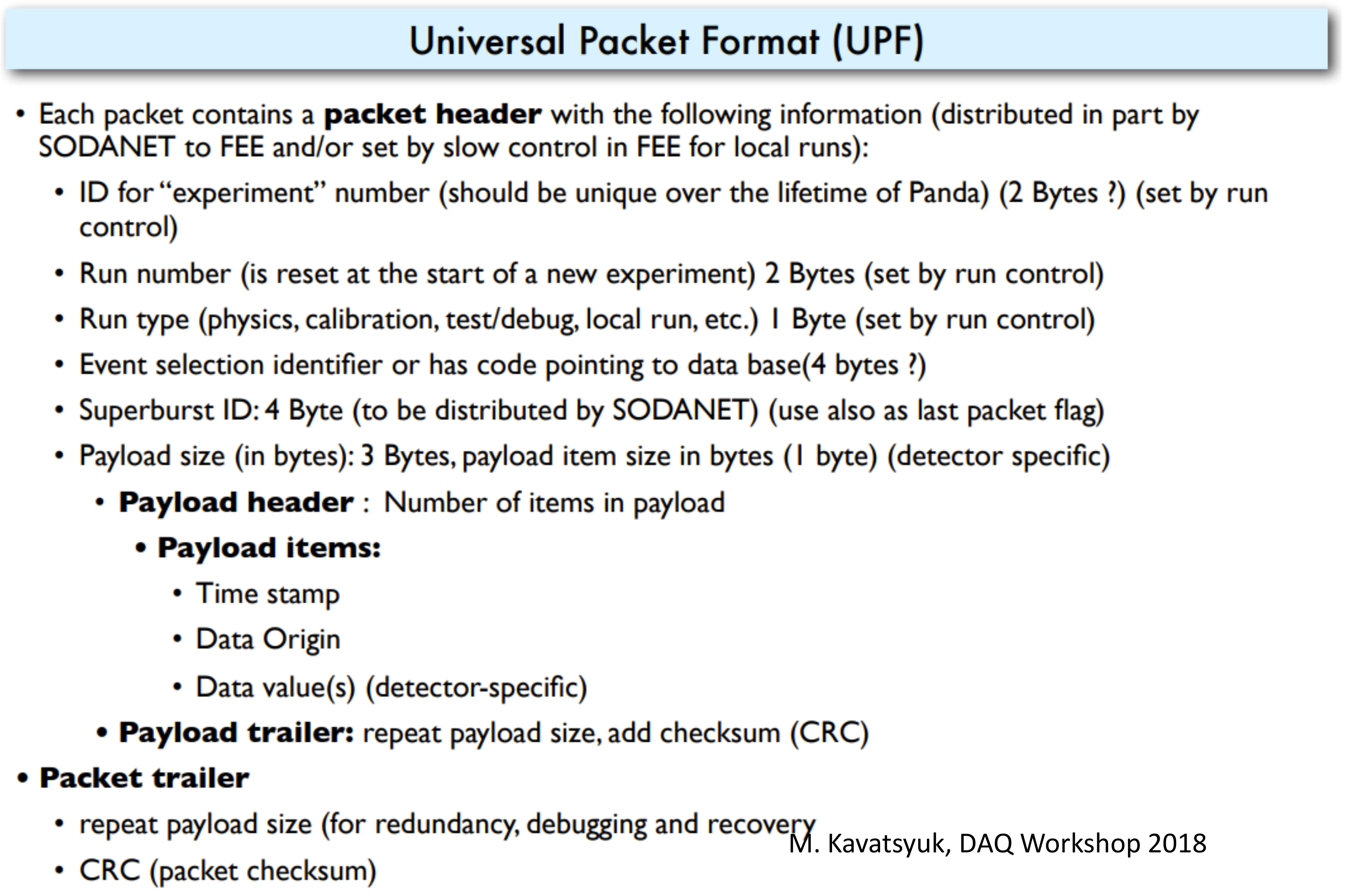


Data format for transport between FEE/Data Concentrators and CN layer

- Proposal: **define universal packet format,** independent of detector subsystem
  - Advantage: we can **use the same set of IP cores** to handle data streams independent of detector and DAQ layer
  - Data is streamed and organised in packets
  - Push architecture, but with support for back pressure
- **Data Origin Definitions:**
  - "detector": Panda Subsystem ID (EMC, STT, MVD etc.)
    - reserve one byte for unique detector definition
  - "module" : section ID of a detector (EMC Forward endcap, STT stereo layer, MVD pixels etc.)
    - reserve one byte for unique module definition
  - "location": unique identifier for the geographical location of a hit within a module (could be STT wire number, MVD pixel ID etc.
    - reserve 4 bytes (need addressing space for MVD)

M. Kavatsyuk, DAQ Workshop 2018

Data formats



Universal Packet Format (UPF)

- Each packet contains a **packet header** with the following information (distributed in part by SODANET to FEE and/or set by slow control in FEE for local runs):
  - ID for "experiment" number (should be unique over the lifetime of Panda) (2 Bytes ?) (set by run control)
  - Run number (is reset at the start of a new experiment) 2 Bytes (set by run control)
  - Run type (physics, calibration, test/debug, local run, etc.) 1 Byte (set by run control)
  - Event selection identifier or has code pointing to data base(4 bytes ?)
  - Superburst ID: 4 Byte (to be distributed by SODANET) (use also as last packet flag)
  - Payload size (in bytes): 3 Bytes, payload item size in bytes (1 byte) (detector specific)
    - **Payload header** : Number of items in payload
      - **Payload items:**
        - Time stamp
        - Data Origin
        - Data value(s) (detector-specific)
    - **Payload trailer:** repeat payload size, add checksum (CRC)
- **Packet trailer**
  - repeat payload size (for redundancy, debugging and recovery
  - CRC (packet checksum)

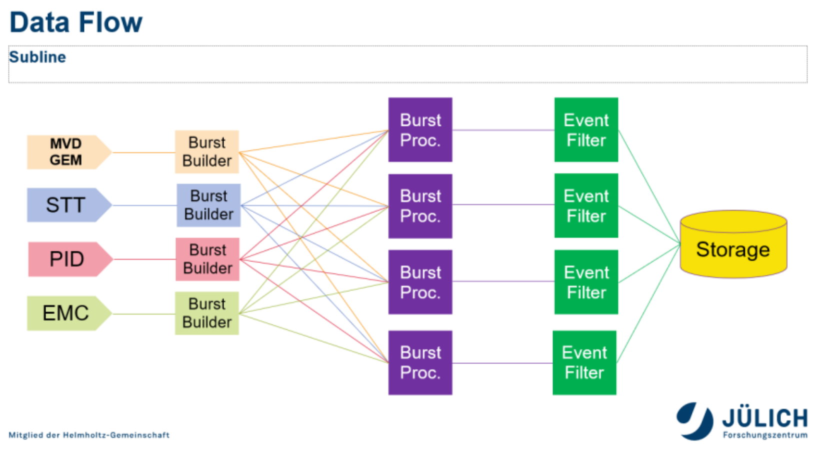
M. Kavatsyuk, DAQ Workshop 2018

DC to Servers network

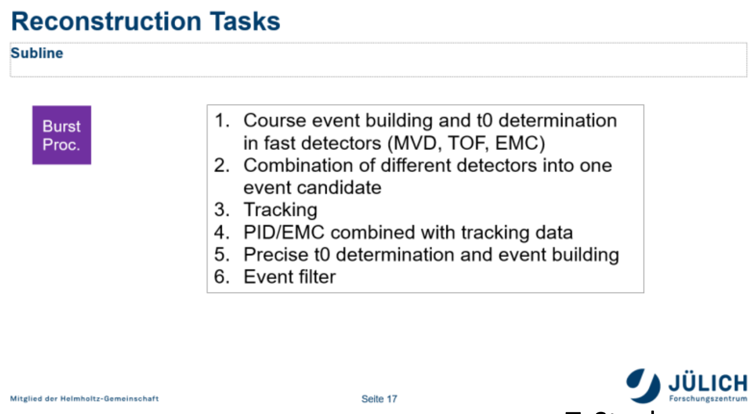
- **Maintain subsystems isolation**
  - Subsystem -> DC -> dedicated server
  - Server instances optimized for particular subsystem data characteristics
  - Superburst/burst building done manually
  - Communication between nodes required
  - Distributed processing

- **Superburst round-robin**
  - Complete superburst -> particular server
  - Superburst building done by network „for free”
  - No communication between nodes required
  - Possible bottlenecks (all DCs -> one server at a time)

Processing pipeline



T. Stockmanns


Reconstruction Tasks
Subline
Burst Proc.
JÜLICH Forschungszentrum
Mitglied der Helmholtz-Gemeinschaft
Seite 17

1. Course event building and t0 determination in fast detectors (MVD, TOF, EMC)
2. Combination of different detectors into one event candidate
3. Tracking
4. PID/EMC combined with tracking data
5. Precise t0 determination and event building
6. Event filter

T. Stockmanns

1. **Subsystem independet processing**
   - Coarse event building / clustering
   - Calibrations, mappings, …
   - FT, STT tracking with coarse T0
   - PID on EMC
2. **Burst building**
   - Combination of tracks and PID
   - Precise T0 determination
   - Complete event building
3. **Event filter**
   - Software trigger
   - Event tagging
   - Event rejection

Processing pipeline

1. Subsystem independent processing

- Subsystem independent
- Processing of time-ordered stream
- Course event building – clustering
- Calibrations, mappings, …

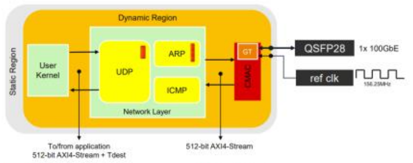
- Clustering efficient on FPGA
  - Preprocessing region on DC
  - FPGA accelerator in server node
    - Direct network connection

- Coarse tracking on GPU
  - Intermediate storage host DDR, CPU interaction required
  - RoCE from Data Concentrators

- Product subevents stored in host DDRs

1. **Subsystem independet processing**
   - Coarse event building / clustering
   - Calibrations, mappings, …
   - FT, STT tracking with coarse T0
   - PID on EMC
2. **Burst building**
   - Combination of tracks and PID
   - Precise T0 determination
   - Complete event building
3. **Event filter**
   - Software trigger
   - Event tagging
   - Event rejection

VNx (XUP)



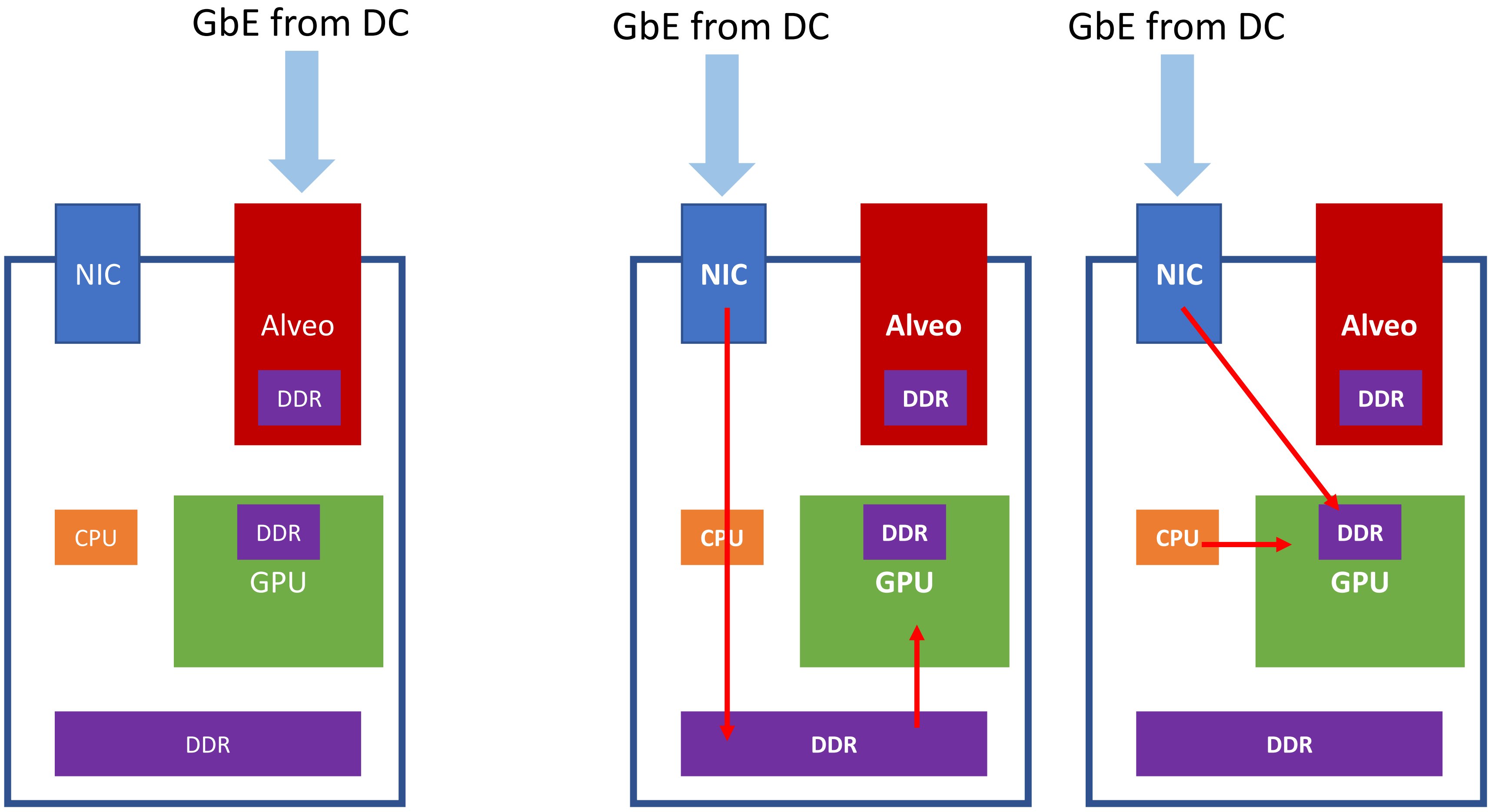
- CMAC kernel
  - Card and interface specific configuration
- Network Layer
  - UDP as transport protocol
  - Single channel application interface
- Built in layers
  - Easily adaptable
- Support for multiple interfaces and cards
  - Currently U50, U250 and U280

https://github.com/Xilinx/xup_vitis_network_example

XILINX

M. Ruiz

Processing pipeline



Through Alveo

To GPU regular

To GPU with RoCE

Processing pipeline

2. Burst Building

- Combination of tracks and PID
- Precise T0 determination
- Complete event building

- Mostly CPU intensive memory scans and data macthing
- Inter-server communication required
- Fast interconnect each-each

- Burst building processes on subsystems processing nodes
- Round-robin job distribution

1. **Subsystem independet processing**
   - Coarse event building / clustering
   - Calibrations, mappings, …
   - FT, STT tracking with coarse T0
   - PID on EMC
2. **Burst building**
   - Combination of tracks and PID
   - Precise T0 determination
   - Complete event building
3. **Event filter**
   - Software trigger
   - Event tagging
   - Event rejection

Processing pipeline

3. Event filter

- Software trigger for event tagging or filtering
- Promissing implementation as a set of Neural Networks (P. Jiang)
- Very fast inference ~1.5M candidates per second on a single RTX3090
- Transmission to Green Cube for storage or further processing

1. **Subsystem independet processing**
   - Coarse event building / clustering
   - Calibrations, mappings, …
   - FT, STT tracking with coarse T0
   - PID on EMC
2. **Burst building**
   - Combination of tracks and PID
   - Precise T0 determination
   - Complete event building
3. **Event filter**
   - Software trigger
   - Event tagging
   - Event rejection

Towards DAQ-0

- Dummy boards with data from simulations
- Data concentrator:
  - Target DC board or any development board with Xilinx and transceivers
- Server node:
  - Any modern PC
- Development work groups:
  - Data Concentrator firmware
  - Software processing pipeline

Summary

- Basic concepts and elements are defined, require to be more and more detailed

- All input is appreciated:
  - Updates on subsystems data rates
  - Updates on numbers of modules and links
  - Pottential preprocessing methods/algorithm on levels: DC and Servers