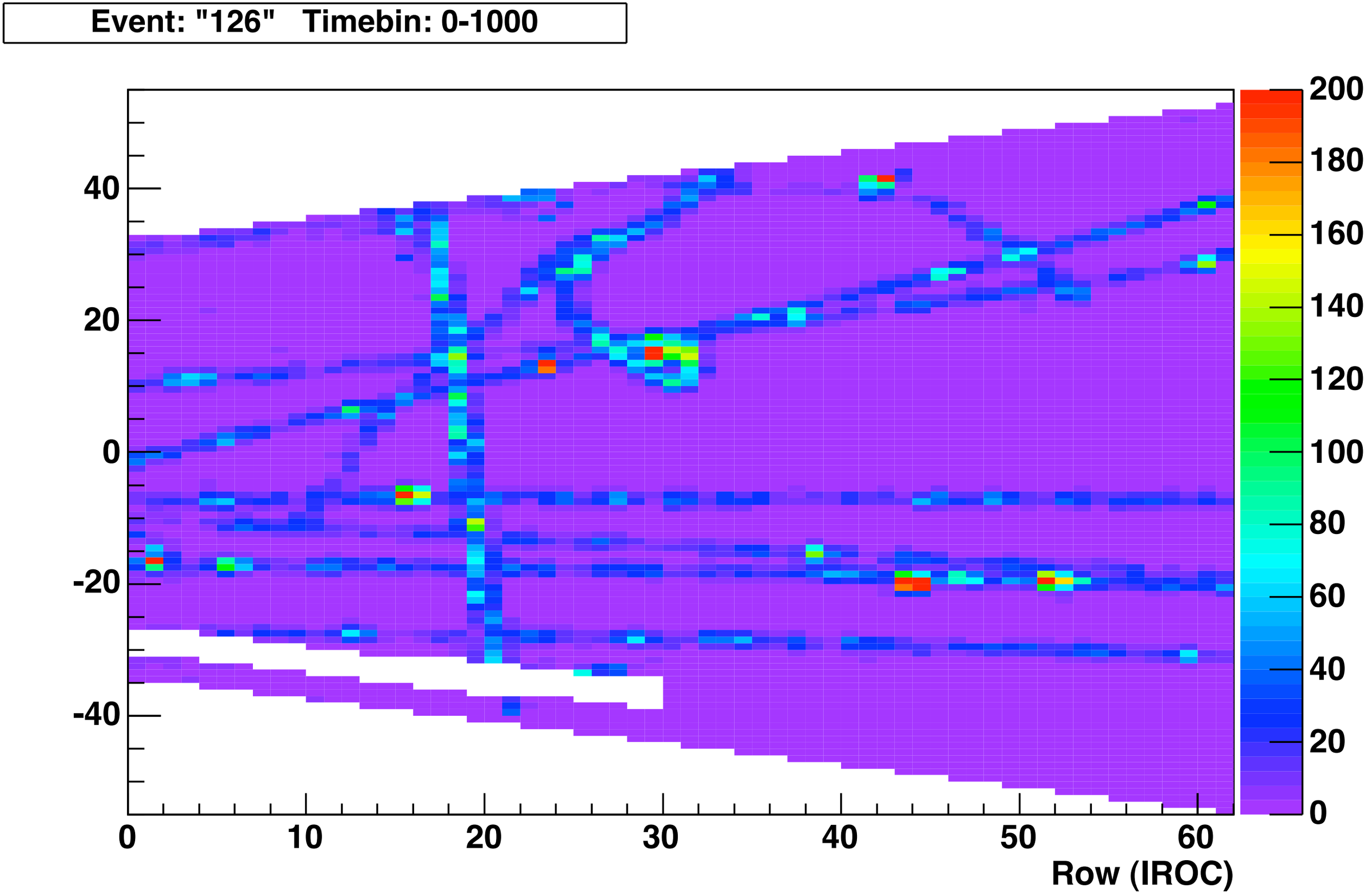
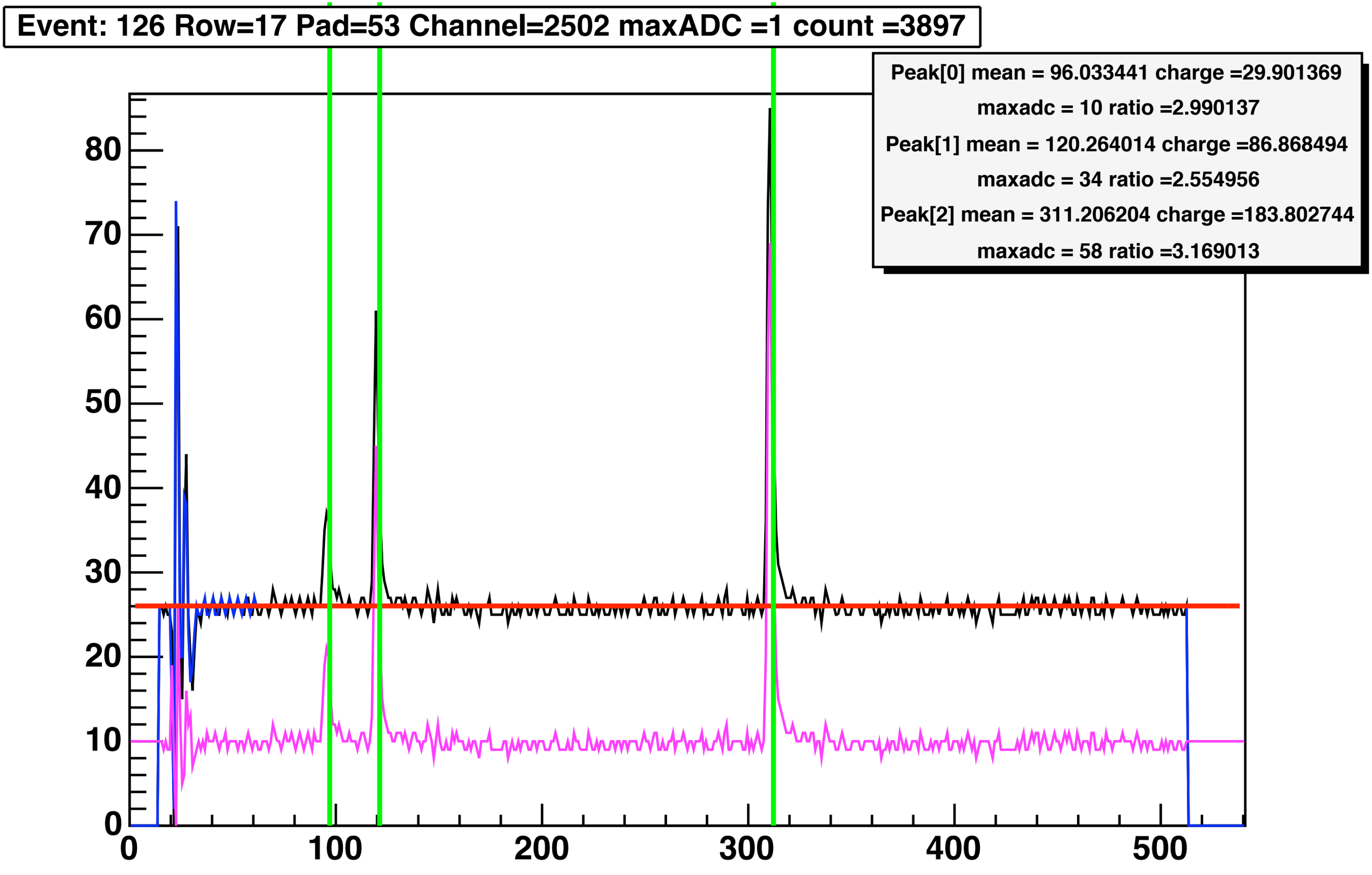
TPC Pad Monitor
(a Raw Data Display)

I. User's View
II. System
III. Future

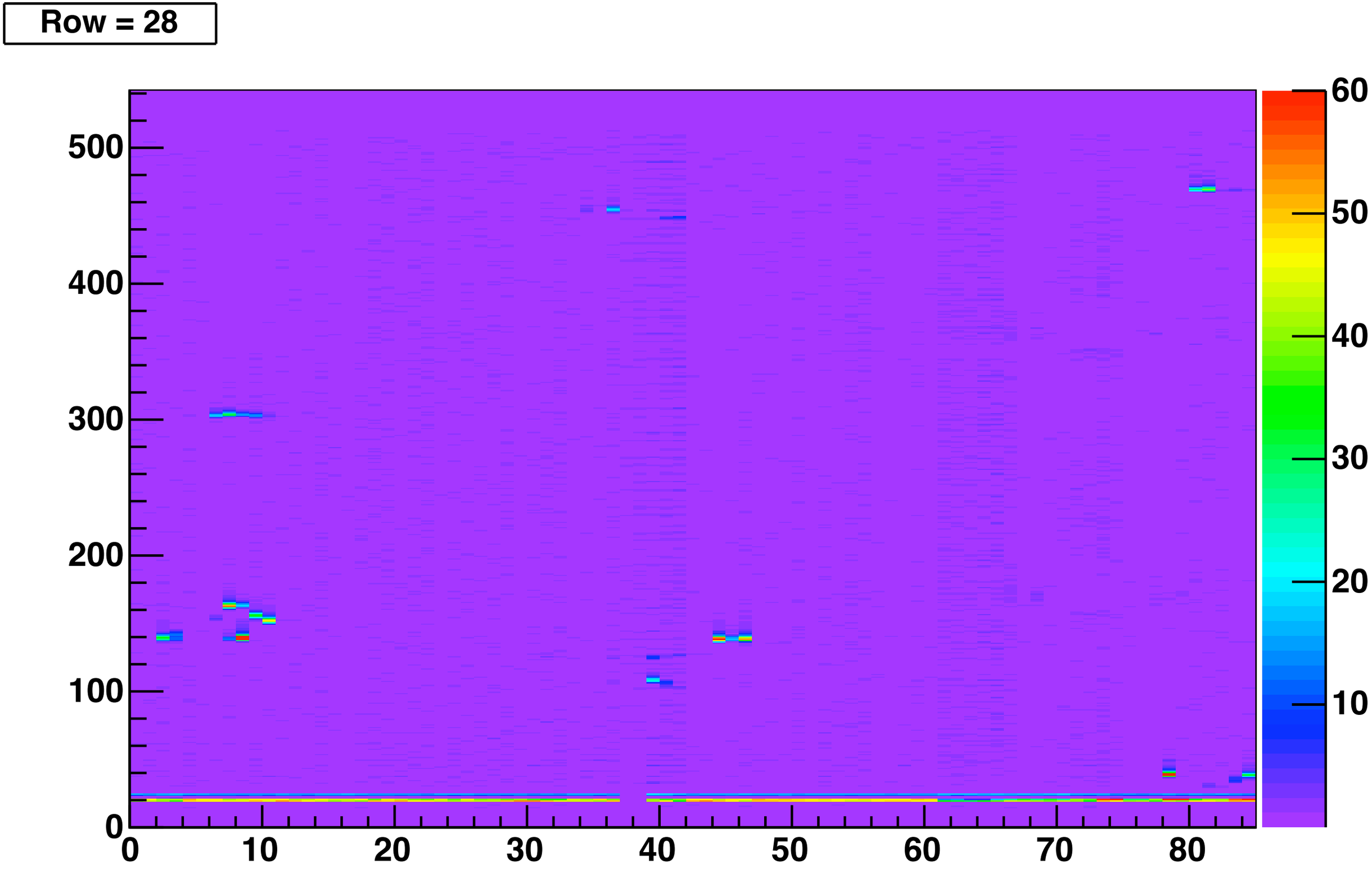
Topview


Event: "126" Timebin: 0-1000
40
20
0
-20
-40
0
10
20
30
40
50
60
Row (IROC)
200
180
160
140
120
100
80
60
40
20
0

Padview


Event: 126 Row=17 Pad=53 Channel=2502 maxADC =1 count =3897
Peak[0] mean = 96.033441 charge =29.901369
maxadc = 10 ratio =2.990137
Peak[1] mean = 120.264014 charge =86.868494
maxadc = 34 ratio =2.554956
Peak[2] mean = 311.206204 charge =183.802744
maxadc = 58 ratio =3.169013
80
70
60
50
40
30
20
10
0
0
100
200
300
400
500

Rowview


Row = 28
500
400
300
200
100
0
0
10
20
30
40
50
60
70
80
60
50
40
30
20
10

What's Displayed

- Dynamic Baseline Calculation and Display
- Static Display of Pedestal Memory
- Static Display of Baseline
- Dynamic Moving Average Calculation and Display
- Simple Pulse & Clusterfinder
- ALTRO++

User Interface

- "Mouse over Pad" updates Padview
- "Mouse & Hold Button" freezes Padview
- "Mouse over Row + Double Click"
  opens Rowview
- Zoom Axis, Edit Plot, Save Plot, ...
  (everything offered in ROOT's interactive mode)

CINT Interface

- ROOT's Functions via CINT
- Help available "help()" or help("topic")
  (has to be updated)
- NextEvent(...), PrevEvent(...),
  OMon(...) == "lastEvent",
  AutoNextEvent(time), AutoLastEvent(time)

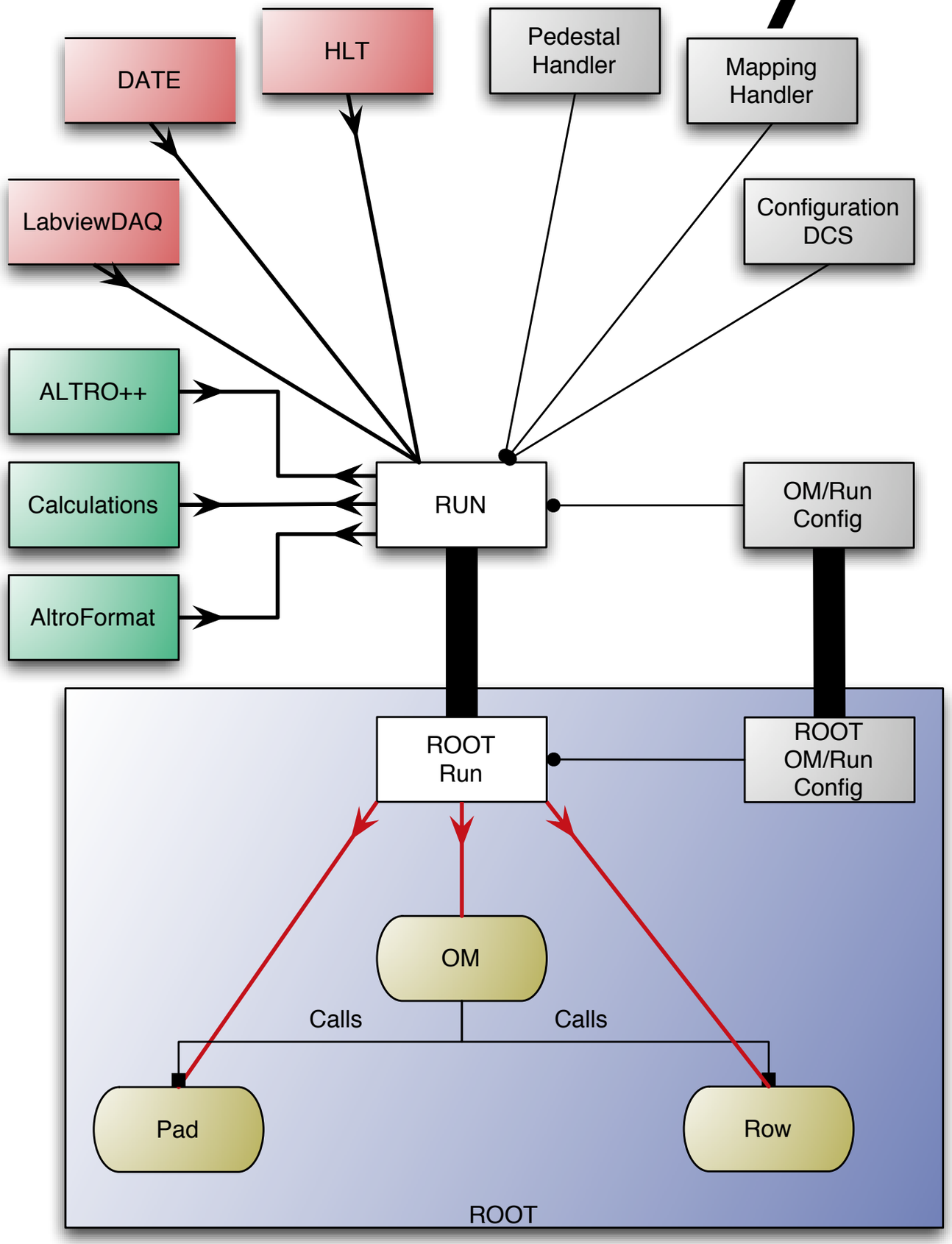
Configuration

```
<OM.Cv1.4>
    <General>
        fixedBaseline        0
        topviewMode          maxadc
        screenresx           1600
        screenresy           1200
        autosaveas           eps
        ...
</General>
<MovingAverage>
        on                   0
        normAndzoomed        0
        ...
```

Speed

- Decoding of 2 patches < 0.2s
  Testbeam Setup, 5500 channels
  (Opteron 246, Power 970 1.8GHz)
- Display < 1s
- Padview > 10 Hz
- Rowview < 0.5 Hz

System



- All in C++
- Mainly ROOT Independent
- Data Sources
- Processing Units
- ROOT Class
- ROOT Macro
- Configuration

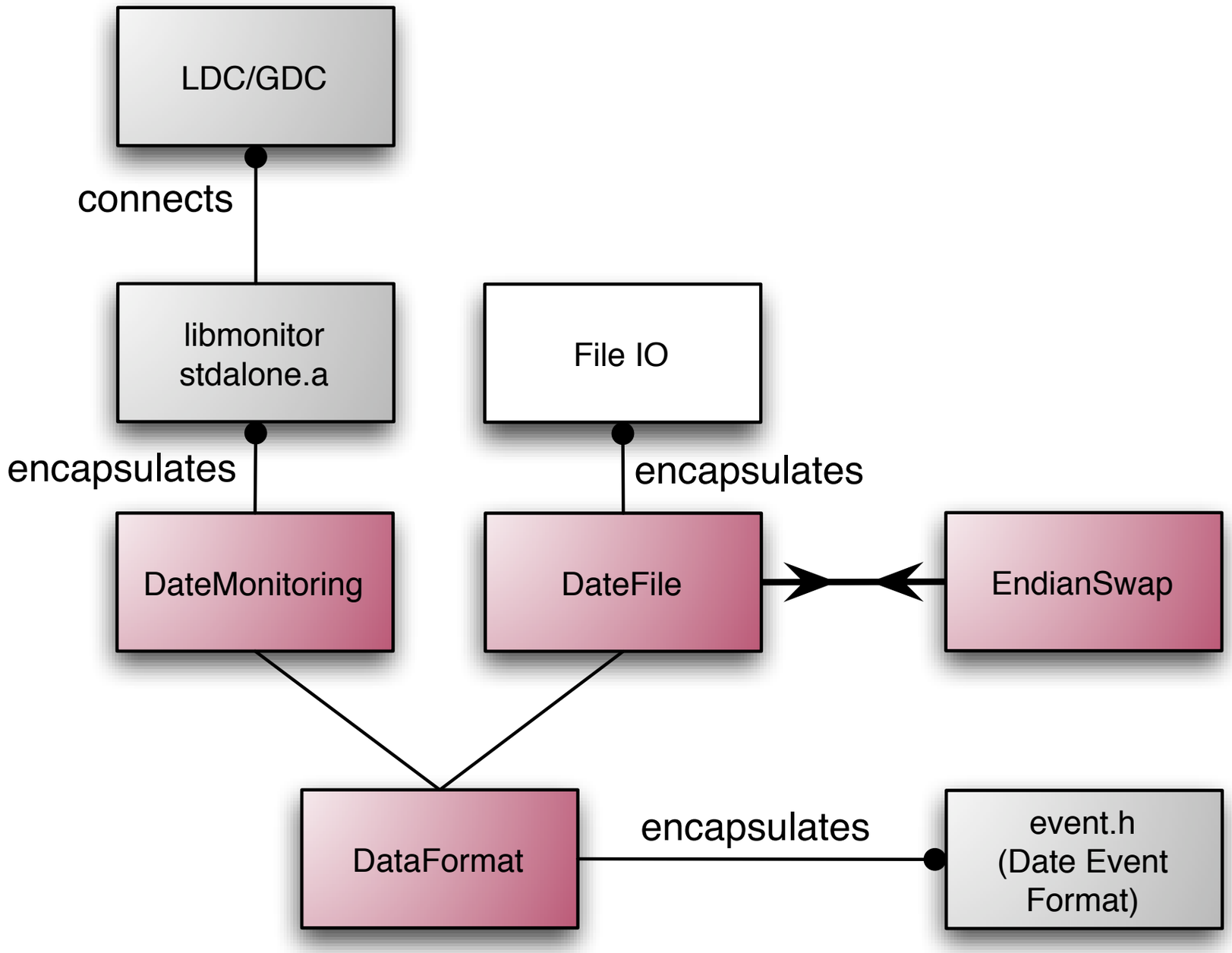
Requironments & Supporting

- ROOT 3 (> 3.05)
- ROOT 4 (> 4.02) (older v4 never tested)
- gcc 3.x (no gcc 2.96!) or e.g. cc
- Runs on Bigendian and Littlendian Machines
- Tested on: Linux, MacOSX, BSD, Solaris
  (Should run on any Real OS)

Labview DAQ

- “Online” Interface was done reading the latest Eventfile
- “Offline” reads selected Eventfile
- Already decompressed Data with small Header
- “mapping” done via Readout Order
- 1 Event == 1 File

Date



- DATE environment
- Encapsulation of DATE Environment in C++
- Implementation of DATE Data Structure
  - For File IO
  - For Online IO
  - Added (automatic) Endian Swap

HLT

- Temporary Interface defined
  (data exchange via shared memory)
- But only Partially Implemented
- Decoding done as Publisher Subscriber Module

AltroFormat

- Decodes ALTRO Data
  (with no Zero Suppression)
  (will be added ...)
- Implements Additional (more stable) Decoding Functions
- Quite Fast but still Room to Improve
  (e.g. big memory overhead)
- Used by HLT

Calculations

- Collection Class for several Calculations
  - Pulse Finder
  - Moving Average Calculation
  - RMS Calculation
  - Cluster Finder

ALTRO++

- Software emulation of ALTRO Digital Chain Implements:
  - BCS1: without AUTOCAL emulation
  - TCF: Bit exact Emulation
  - BCSII: exact Emulation, pre-knowledge missing
  - ZSU: exact Emulation, but Data Formatter not implemented

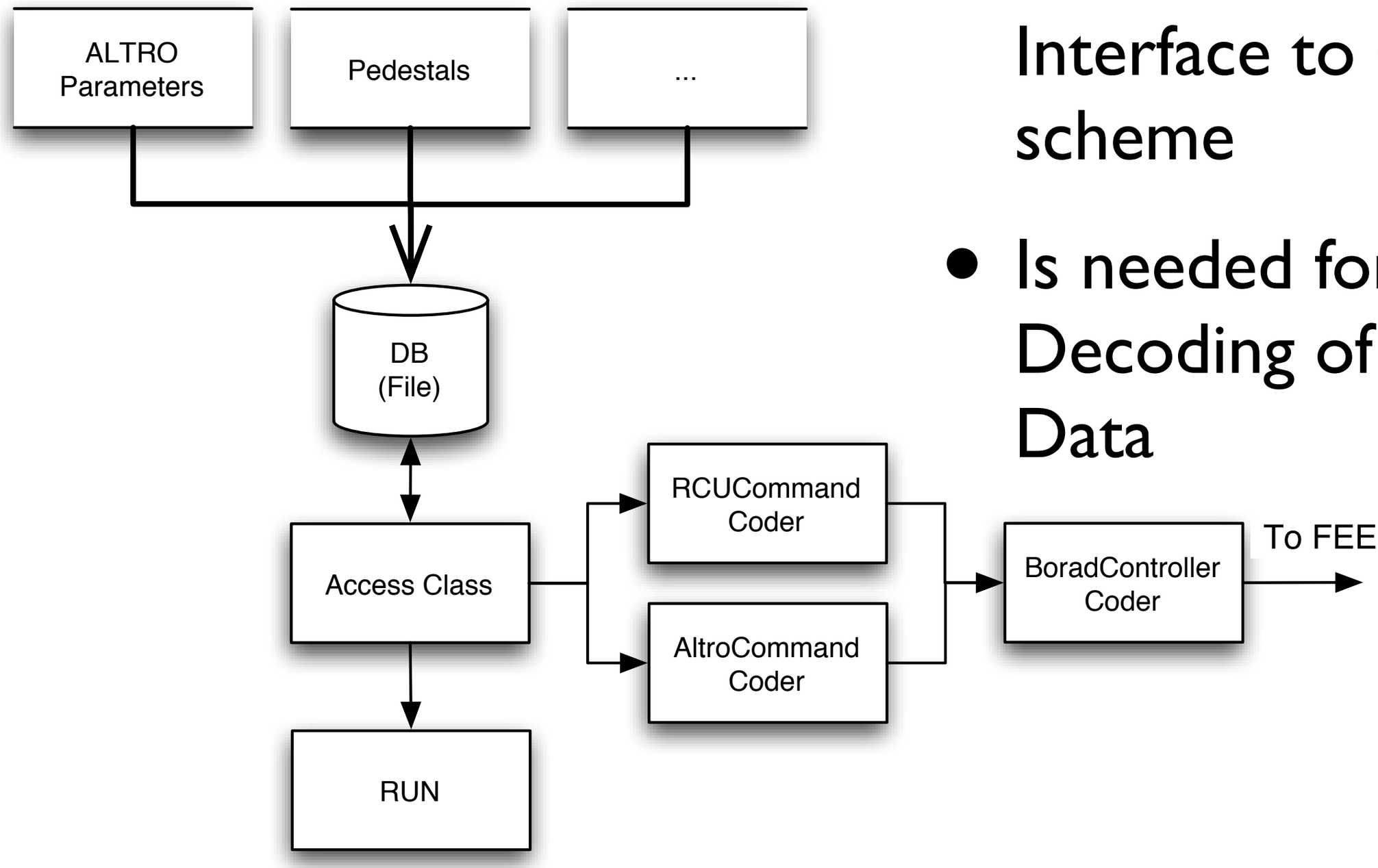
Mapping Handler

- Created unique Address of each Channel
  - Added Branch Information (later done by RCU)
  - Added Patch Information
  - Future ? (unique address for whole TPC?)
- Mapping done from “unique Address” to Pad, Row, etc.
- LabviewDAQ done via Readout Order

Pedestal Handler

- Handles Data for:
  - Pedestal Memory (ascii, bin, compressed bin)
    - Pedestal Value for DATE based & Labview based Data
- Bigendian & LittleEndian safe
- Will move to the FEE Configuration Scheme

Config/DCS



- Started to Implement Interface to Config-scheme
- Is needed for Decoding of ALTRO Data

Run / ROOTRun

- Is Interface Class to “everything”:
  - Handler Classes
  - Data Sources
    - Calculation Classes
- Exists as C++ Class
  (for processing without root)
- Exists as ROOT Class
  - Data Preparation for the OM Macros

Config / ROOTConfig

- Is Interface Class for OM/Run Configuration
- Exists as ROOT Class
- Exists as C++ Class
  (for processing without root)
- XML like configuration (as shown)

OM
Pad / Row

- OM is a ROOT Macro, which uses the ROOTRun Class
- Runs in Interactive Mode
- Calls Pad and Row
- All Important Processing is done in/via Run

Future

- Extension to Whole Sector is Minimal Work
- Extension to Whole TPC possible but:
  - Needs Implementation of Etha Slices
  - Layer above shown Topview
  - Speed Increase/Partial Decode
    (20s for complete decoding)

Where to get

Code is Archived and Maintained in a Subversion Repository:

pcikf43.ikf.physik.uni-frankfurt.de:8080/svn/repos

Doxygen Code Documentation Available:

pcikf43.ikf.physik.uni-frankfurt.de/~documentation/TestTPCCode/html/

or via the IKF Alice Homepage (TPC-Test Data Page):

http://www.ikf.physik.uni-frankfurt.de/~alice/