



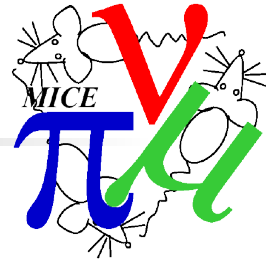
MAUS Overview



Chris Rogers,
ASTeC,
Rutherford Appleton Laboratory



Responsibility/Block Diagram



Project management
Rogers

Build system
Tunnell => Rogers?

QA
Rogers

Documentation
Rogers

Geometry + fields
Rogers/Littlefield

Geant4 Simulation
Rogers

Data flow/API
Rogers/Richards/Jackson

TOF
Karadzhov Rajaram

Tracker
Dobbs/Santos et al

Ckov
Cremaldi/Kafka

KL
Bogomilov?

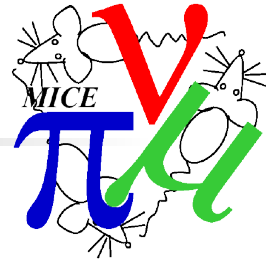
Data Unpacking
Karadzhov

EMR
Karadzhov/Ruslan

Detector Integration
Rogers/Lane

Accelerator physics
analysis
Rogers/Lane

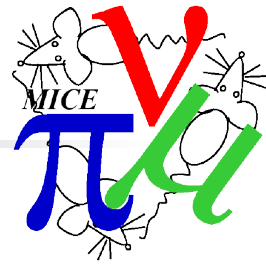
Python Test Coverage



Name	Cover	Responsible
ReducePyTOFPlot	4	Rajaram
MapPyTOFPlot	12	Rajaram
ReducePyROOTHistogram	18	Jackson
CouchDBDocumentStore	28	Jackson
mausloader	29	Jackson
MongoDBDocumentStore	42	Jackson
MapPyScalersDump	45	Karadzhov
Go	69	Tunnell -> Rogers
InMemoryDocumentStore	100	
InputPyEmptyDocument	100	
InputPyJSON	100	
InputPySpillGenerator	100	
MapPyDoNothing	100	
MapPyFakeTestSimulation	100	
MapPyGroup	100	
MapPyPrint	100	
MapPyRemoveTracks	100	
OutputPyDoNothing	100	
OutputPyImage	100	
ReducePyDoNothing	100	
ReducePyHistogramTDCAD	100	
ReducePyMatplotlibHistogram	100	
ReducePyRootHistogram	100	

Needs distributed
computing libraries
on test server

Cpp Test Coverage



Directory	Line Coverage ↕		
src/common_cpp/DetModel		91.7 %	11 / 12
src/common_cpp/DetModel/SciFi		34.4 %	45 / 131
src/common_cpp/Simulation		90.8 %	824 / 907
src/common_cpp/Utils		65.7 %	463 / 705
src/input/InputCppDAQData		60.9 %	337 / 553
src/map/MapCppPrint		90.5 %	19 / 21
src/map/MapCppSimulation		96.1 %	49 / 51
src/map/MapCppTOFDigits		79.7 %	145 / 182
src/map/MapCppTOFSlabHits		93.3 %	111 / 119
src/map/MapCppTOFSpacePoints		90.1 %	146 / 162
src/map/MapCppTrackerDigitization		88.0 %	169 / 192

Dobbs

See below
Karadzhev

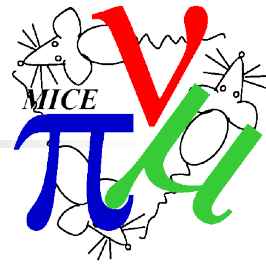
Filename	Line Coverage ↕		
CppErrorHandler.cc		78.6 %	33 / 42
CppErrorHandler.hh		100.0 %	8 / 8
DAQChannelMap.cc		50.5 %	46 / 91
DAQChannelMap.hh		83.3 %	10 / 12
JsonWrapper.cc		89.6 %	60 / 67
MAUSEvaluator.cc		77.1 %	64 / 83
PyMausCpp.cc		93.8 %	15 / 16
TOFCalibrationMap.cc		59.4 %	129 / 217
TOFCalibrationMap.hh		100.0 %	8 / 8
TOFChannelMap.cc		50.7 %	73 / 144
TOFChannelMap.hh		100.0 %	17 / 17

Karadzhev

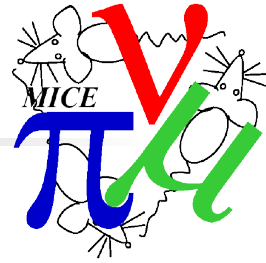
Rajaram

Rajaram

Infrastructure



Geometry + Fields (Littlefield)



CAD import now in MAUS trunk

Needs work on TOF, fields

Discussion with Jason Tarrant on CAD model implementation

South Mezzanine design is finished and MAUS geometry is next priority

Get a “best effort” by March running

Rolling issue – it will be a lot of effort to get a decent geometry model for every iteration/survey of MICE

An issue I have been struggling with for 2 years!

Verification of geometry implementation

Easy in CAD to accidentally get material wrong/whatever

“Survey is accurate to microns, errors $O(\text{cm})$ ”

Detector experts are responsible for verifying their detector

Beamline expert is responsible for verifying beamline geometry

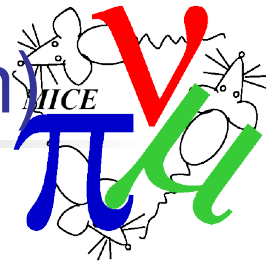
Analysis group is responsible for verifying cooling channel geometry

First attempt at a full iteration in March...

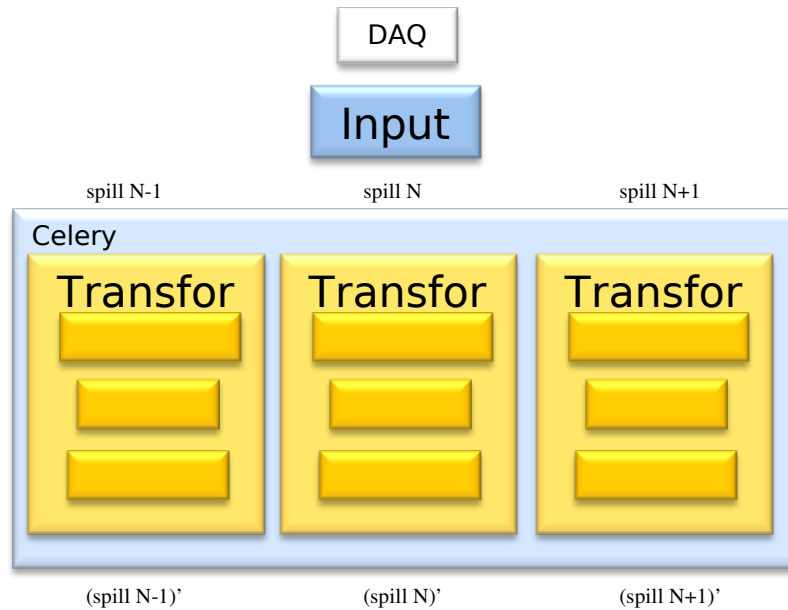
Future Work

- Once Jason Tarrant's CAD is complete upload it to database (first official geometry)
- Finalise details of management (Fastrad translation and internal detector geometries)
- Write code which downloads the geometry directly to the simulation (doesn't download a local copy)
- Collect internal geometries from each detector group (Oleg from the tracker and Gene from the CKOV have already begun) and then extend the GDML_MICE schema
 - As always TESTS! TESTS! TESTS!
- I currently have a CAD of the EMR and will be starting tests with this geometry.

Dataflow/API - online model (Jackson)



**parallel
transform
execution**



**document-oriented
database**



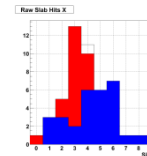
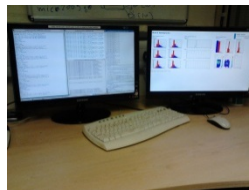
Merge

Output

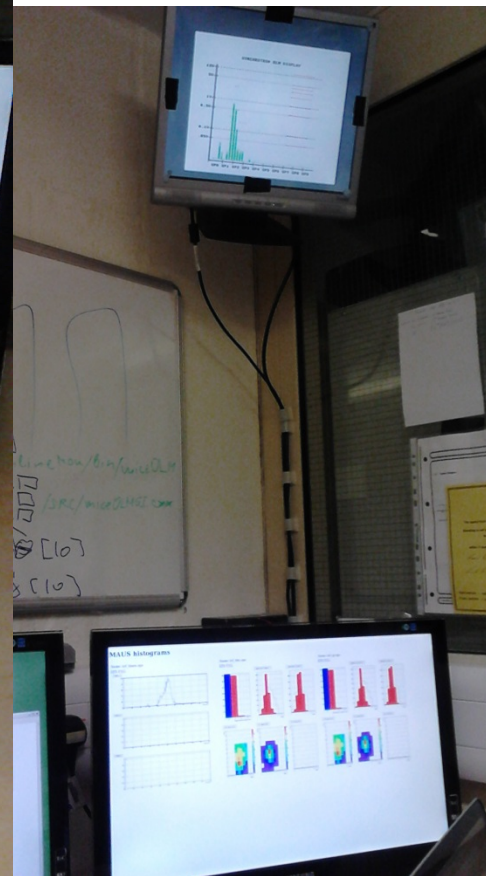
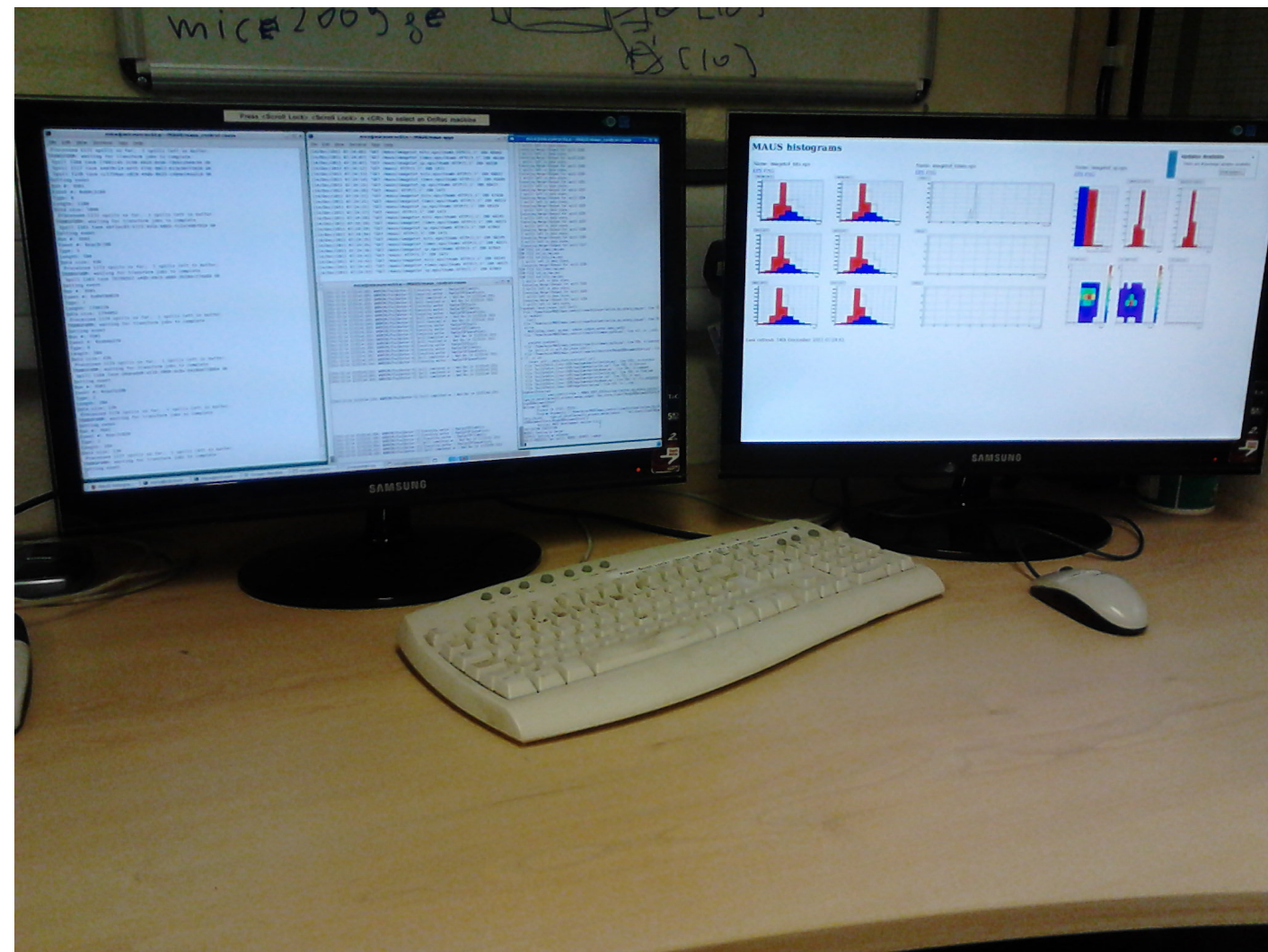
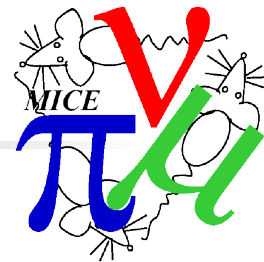
Web front-end

**histogram
mergers**

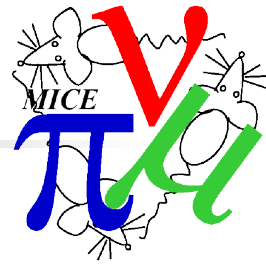
**web
front-
end**



Current state (Jackson)



Data flow / API - data representation (Richards/Rogers)



Implementation of ROOT IO + cpp class structure

- Framework is in place

- In testing phase

- Need to implement the actual data structure

- Analysis group request explicitly documentation on datastructure

Future plans

- Abstraction layer for the module structure

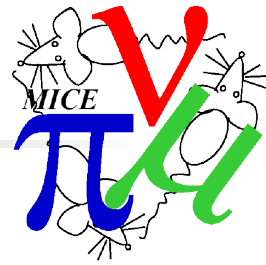
- Generic converter between data representations

- Python logging + interface to Cpp logging (Squeak)

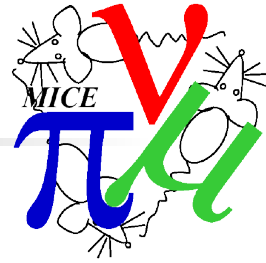
- Module control at runtime

 - We include this recon or that recon at runtime

Detectors



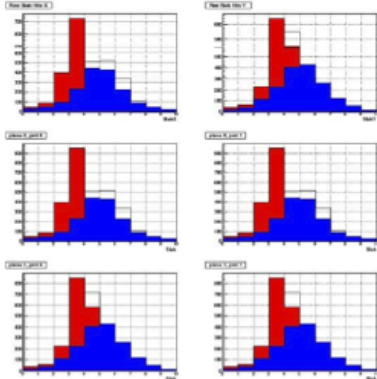
TOF - Status (Rajaram)



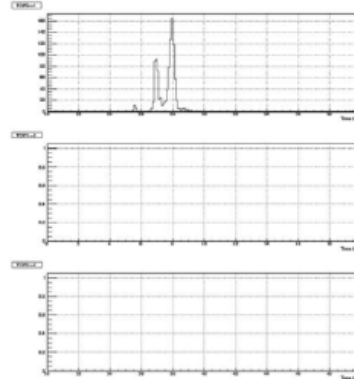
- TOF online monitoring was ported from G4MICE & were working in the December run

MAUS histograms

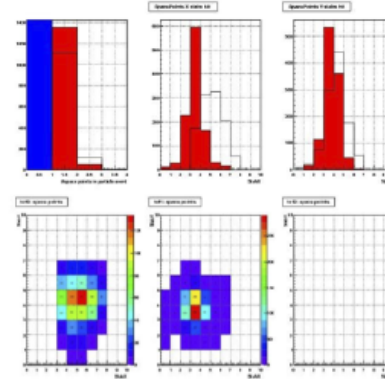
Name: imagetof_hits.eps
EPS PNG



Name: imagetof_times.eps
EPS PNG

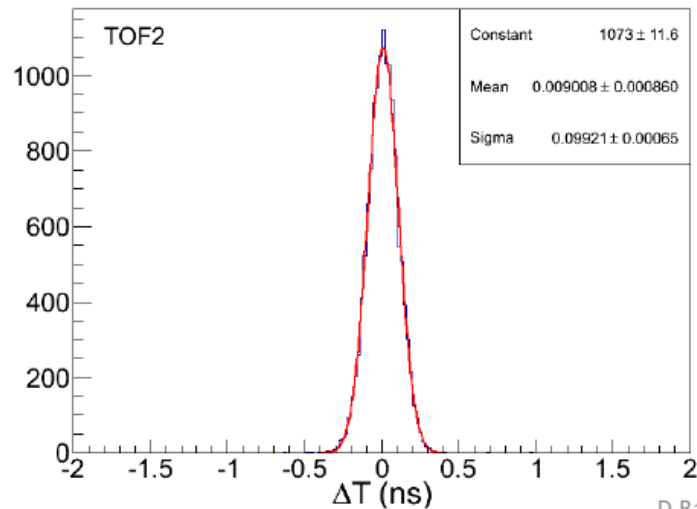
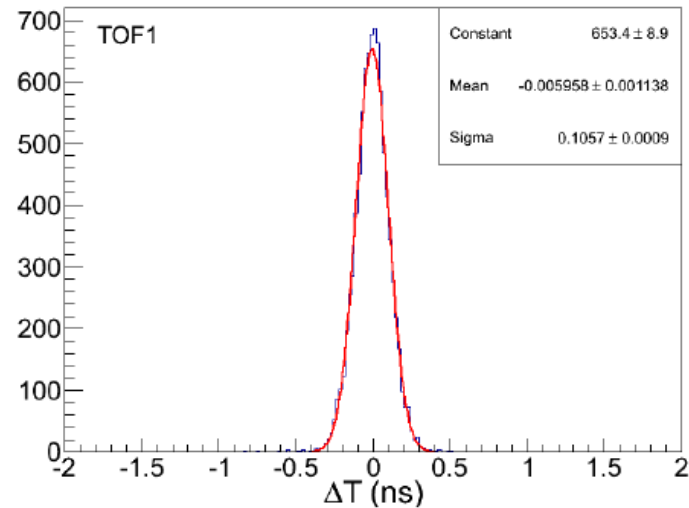
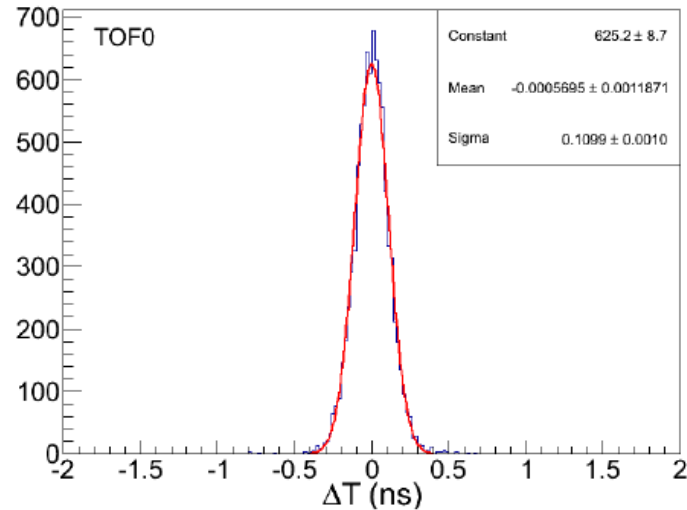
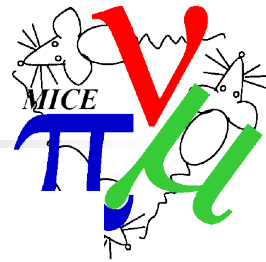


Name: imagetof_sp.eps
EPS PNG



- Calibration with December data is done

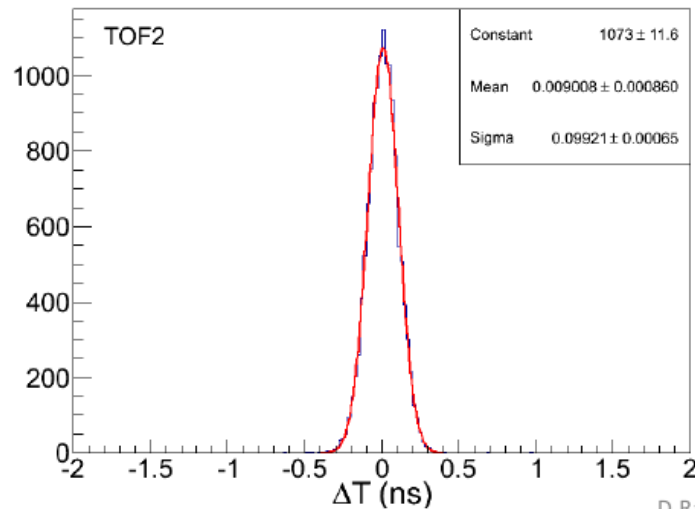
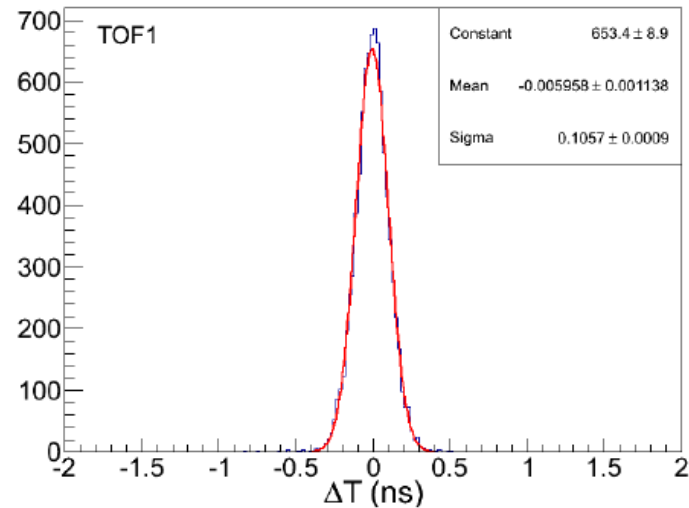
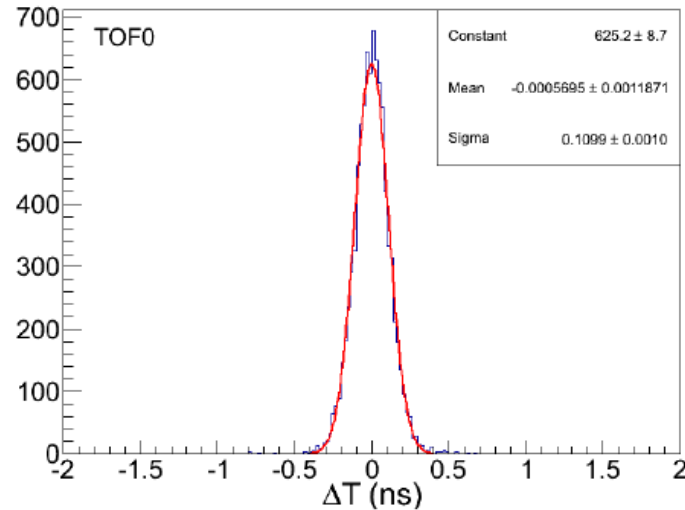
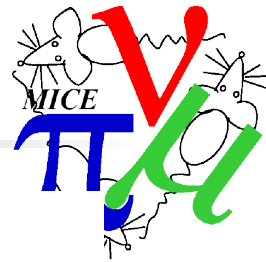
TOF - resolution (Rajaram)



Resolutions

- TOF0: 55 ps
- TOF1: 53 ps (improved)
- TOF2: 50 ps

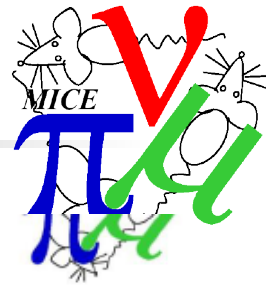
TOF - resolution (Rajaram)



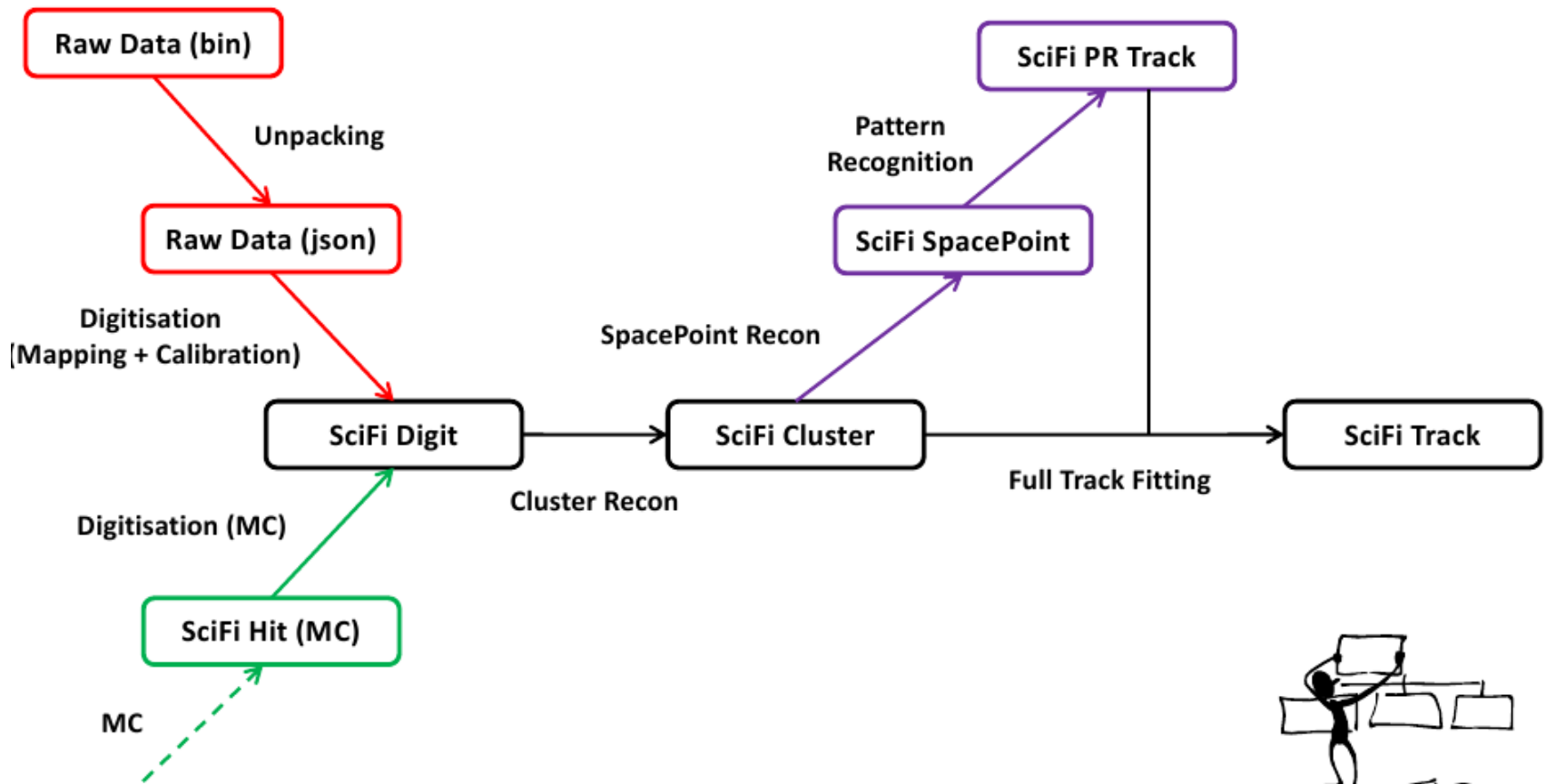
Resolutions

- TOF0: 55 ps
- TOF1: 53 ps (improved)
- TOF2: 50 ps

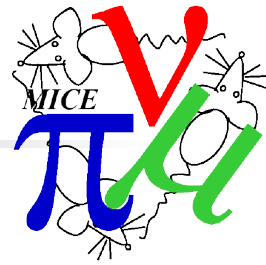
Tracker (Dobbs et al)



Data flow



Tracker (Dobbs)



Tracker software is proceeding

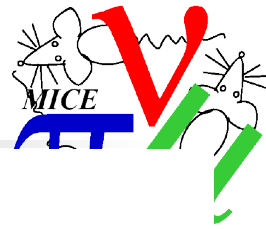
The deadline is tight

So far reconstruction up to digitisation is in the trunk

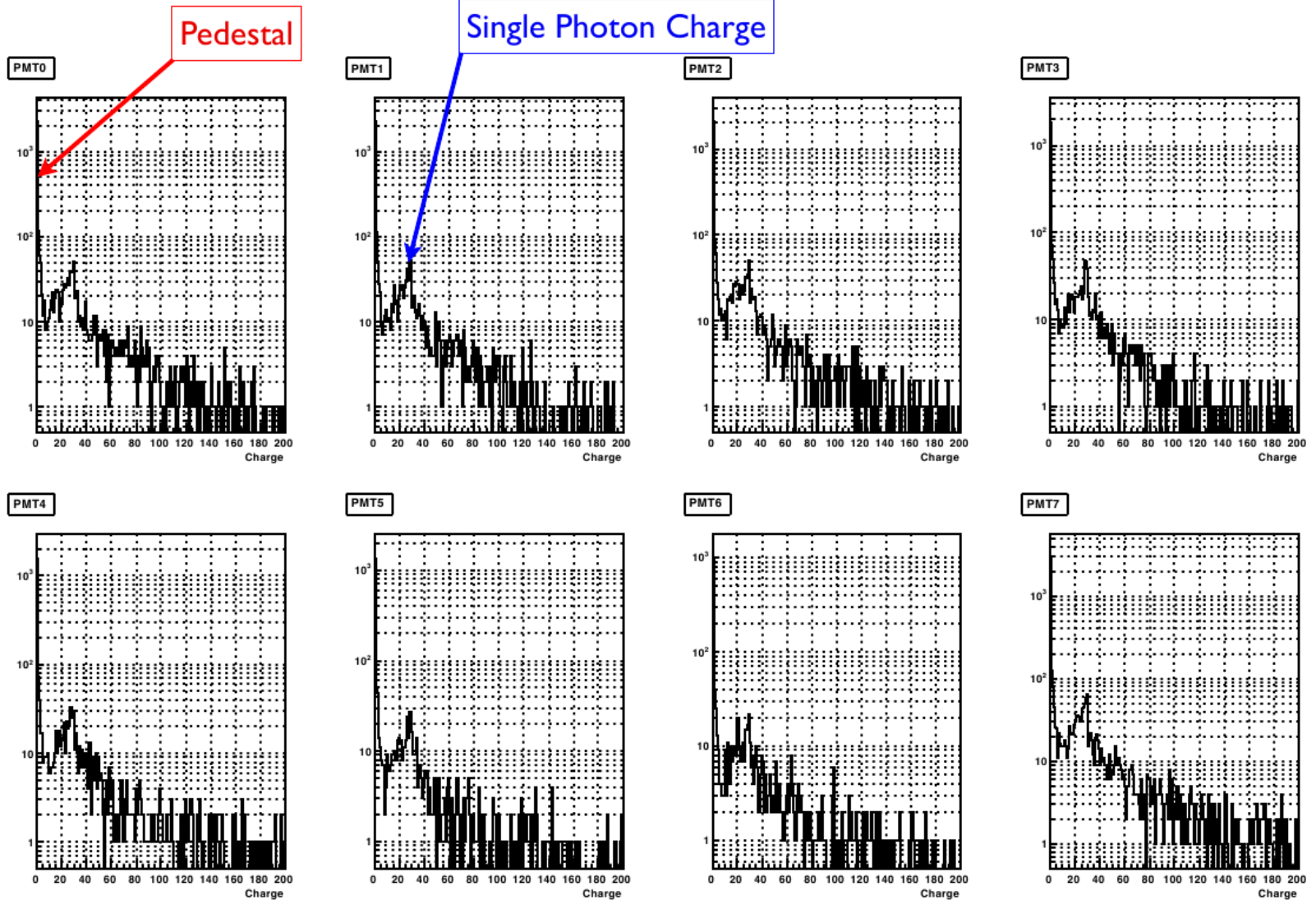
Spacepoints is close

Pattern recognition and track fit follows

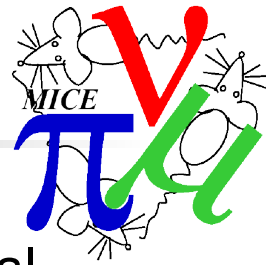
Cerenkov (Kafka)



Charge Distributions in Each PMT



Detector Integration + Accelerator physics (Rogers/Lane)



Some work on transfer matrix routines - clean up, and general introduction to the code

This also forms basis for some accelerator routines that we want

Plan for global track fitting is in progress

Discussion on implementation details still needed

Hope to have a first draft plan for this by end of next week

Plan for beam envelope evolver is in progress

GUI frontend specified

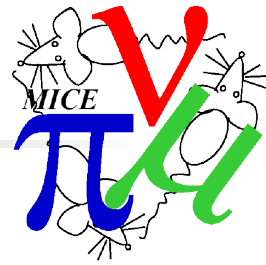
Need to understand the backend

Code is already in place, needs some cleaning

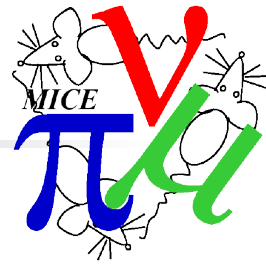
Some other infrastructure-type jobs on the list

Configuration management

Other things



Software Workshop



Software workshop next week

Try to merge some outstanding code

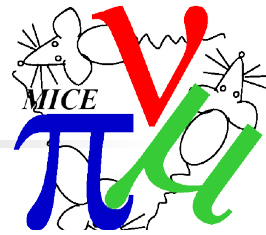
Code reviews

Tutorials/user support for those who want it

Coffee for those who want that

CR13, start 9 am on Monday

March Running



Aim	Responsible	Time
Online monitoring visible through micewww	Chris Rogers	4 hours
Data mover per run	Henry Nebrensky	4 hours
State machines and run control sequence	Pierrick Hanlet	4 hours
GRID/Batch job	Chris Rogers	4 hours
Cerenkov online monitoring	Gene Kafka	2 hours

- Cerenkov online monitoring
 - we would like to check that we can pull plots from the Cerenkov and into the MAUS online code.
 - can be checked offline before running.
- Online monitoring visible through micewww
 - would like to check that we can push online monitoring plots through to mice website
 - can be checked offline before running
- Data mover per run
 - change the data mover script so that it executes every run rather than nightly
- State machines and run control sequence
 - Pierrick to comment
- GRID/Batch job
 - Automatically run reconstruction + MC offline every time a new file is pushed to the data store
- Need to include run up/run down in this also

the end...

