Global Reconstruction Data Structure

Ian Taylor
University of Warwick

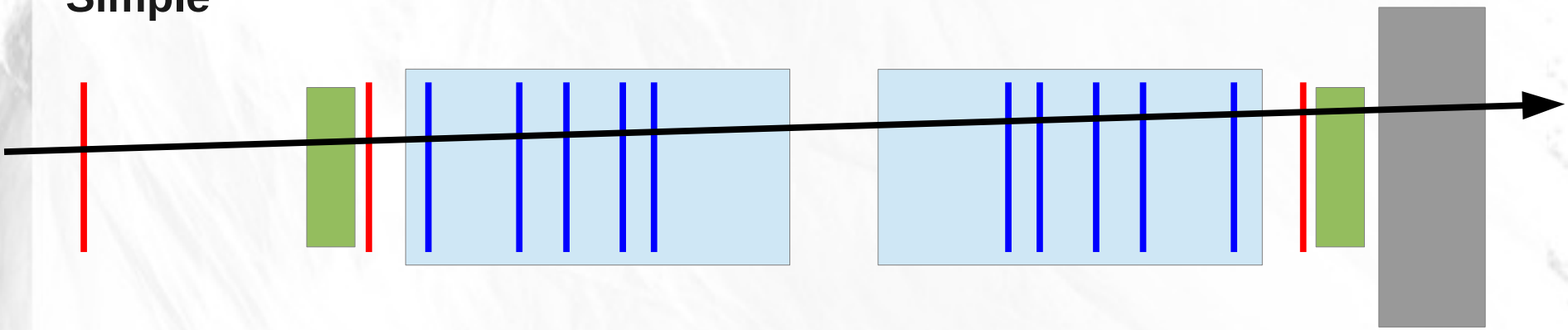
MICE CM35
15/02/2013

Proposed Data Structure

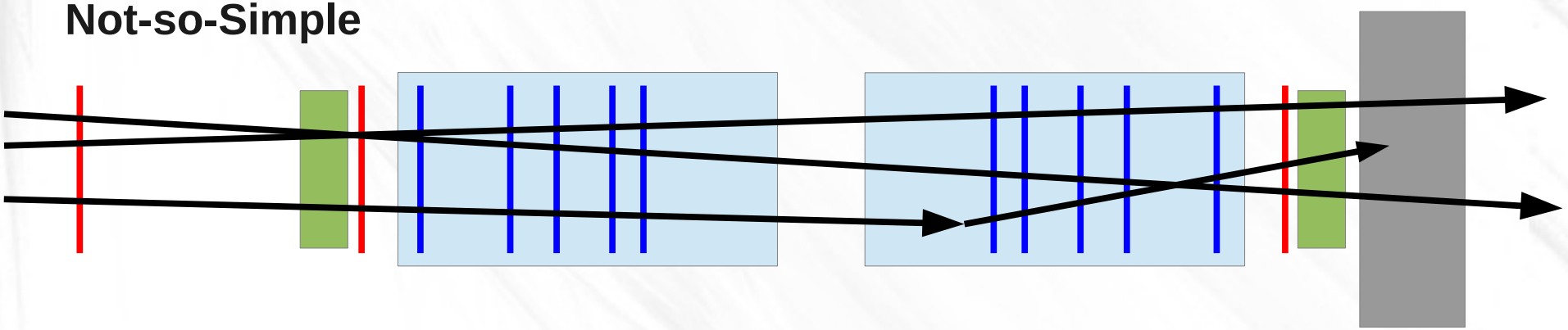
- Proposed data structure for the global reconstruction was discussed at the last MICE Software Workshop:
  - Chris Rogers, Peter Lane and Ian Taylor
- Goal was for a flexible structure, simple to use but capable of storing complicated triggers and event topologies if required.
  - Multiple primary particles in trigger
  - Decay in flight of particles

Examples

Simple

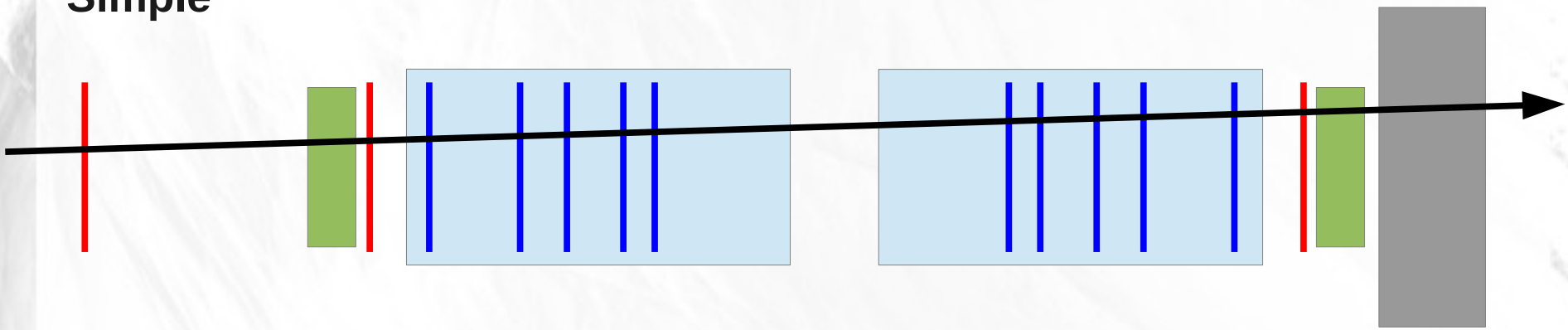


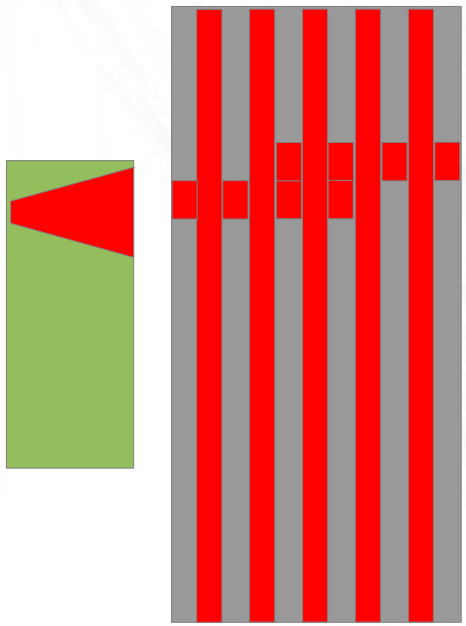
Not-so-Simple

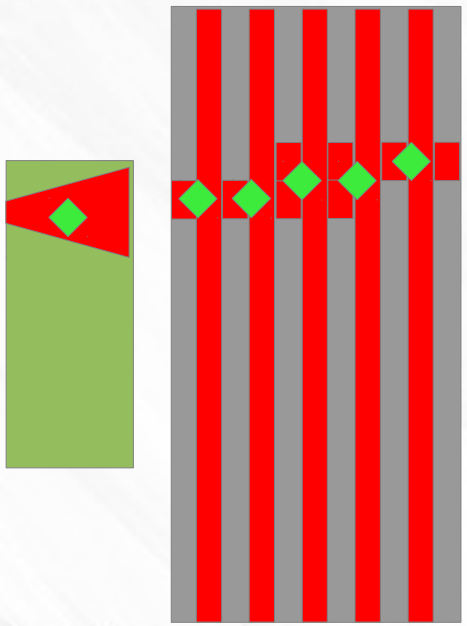


Simple Example: SpacePoints

Simple





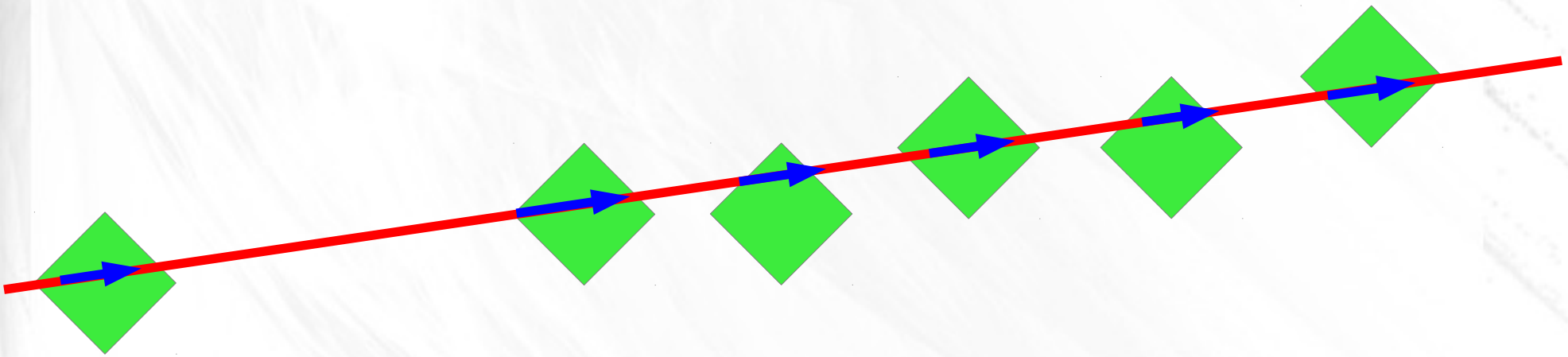


Space Points

```
double          Charge; //energy deposit
TLorentzVector  Position;
TlorentzVector  PositionError;
Enum            Detector;
std::string     GeometryPath;
```

- Represent a detector measurement:
  - This is the closest global reconstruction will come to the detectors, so position and error should be defined accurately.
  - Will trust detector reconstruction, e.g. 3 planes of tracker station -> 1 SpacePoint.

Simple Example: Tracks & TrackPoints



Not shown here, the possibility of scattering: i.e. Track not necessarily straight.

Track Point

```
std::string     MapperName;
TLorentzVector  Position;
TlorentzVector  PositionError;
TLorentzVector  Momentum;
TlorentzVector  MomentumError;
Enum            Detector;
std::string     GeometryPath;
TRef            SpacePoint; // Reference
```

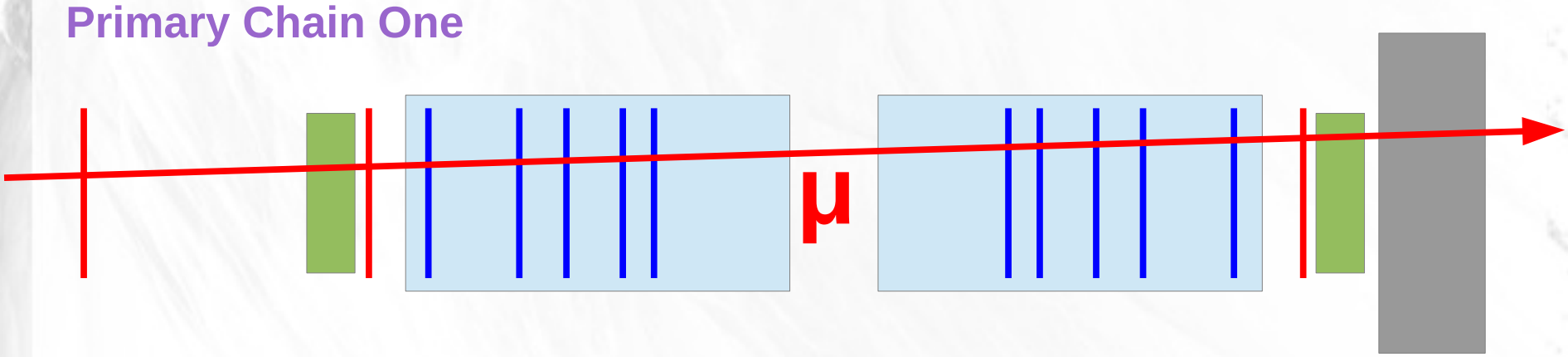
- Reconstructed location, according to a track fit, with an associated Space Point.

Track

```
std::string     MapperName;
Enum            PID;
Enum[]          DetectorPoints;
std::string[]   GeometryPaths;
double          Charge;          // +/-
std::string     GeometryPath;
TRefArray       TrackPoints; // References
TRefArray       ConstituentTracks;
double          GoodnessOfFit;
```

- A collection of track points, which the reconstruction believes came from the same particle.
  - Tracks can be combined (SciFi + TOF,CKOV)

Simple Example: PrimaryChain



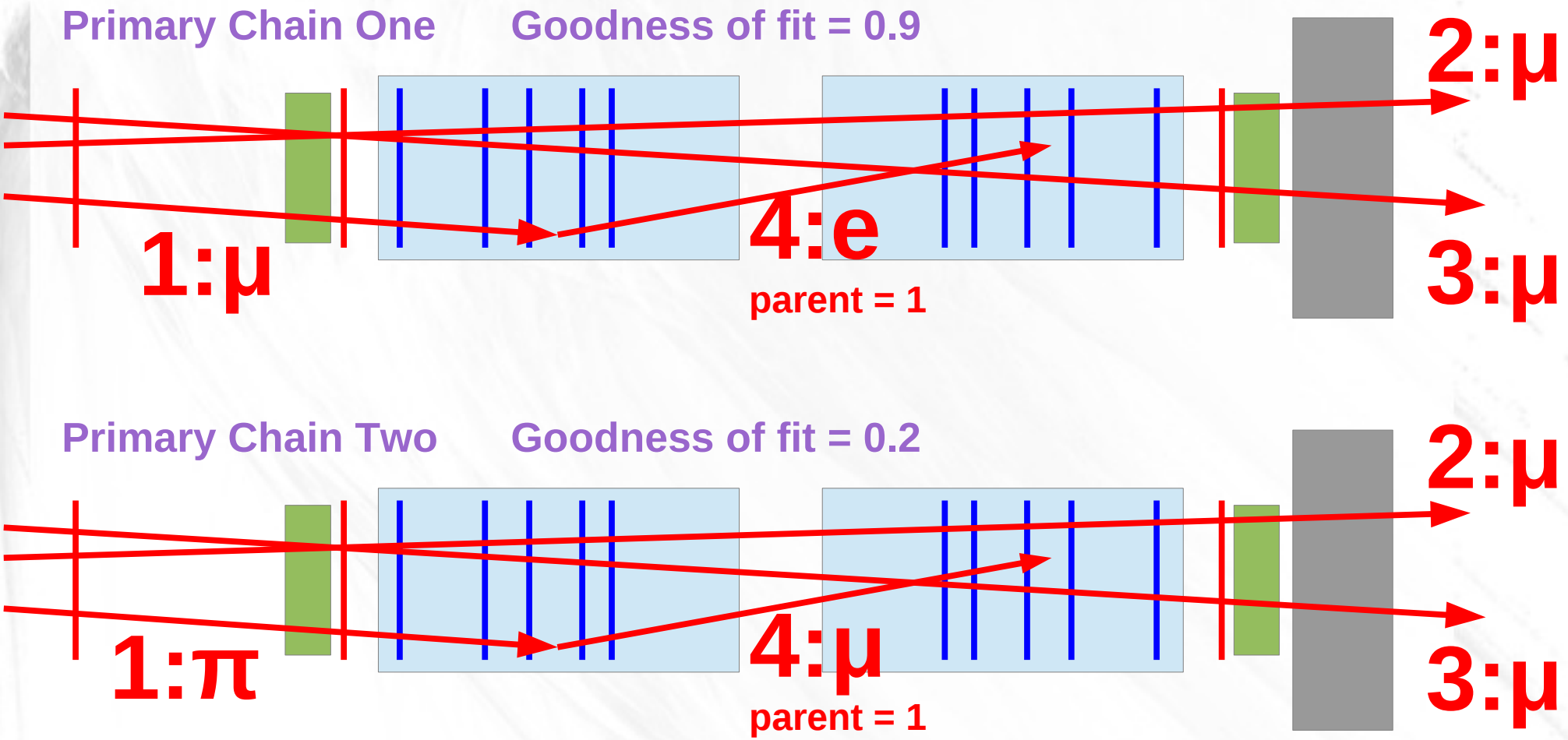
- One Primary Chain
  - One Track
    - Track Point @ each detector plane
    - Space Point @ each detector plane

PrimaryChain (Hypothesis)

```
std::string     MapperName;
TRefArray       PreceedingPrimaryChains;
TRefTrackPair[]TracksAndParents;
double          GoodnessOfFit;
```

- A collection of Tracks, which the reconstruction thinks accounts for all particles in event.
- This is the top-level analysis object, for your analysis to consider and accept / reject...

Example – Not so Simple



Request

- Comments on data structure welcome.
- Specifically from an analysis point of view:
  - Would this give sufficient information for an analysis?
  - Can provide 'virtual' Track/SpacePoints, e.g.:
    - Entering / Exiting AFC, Upstream end (D2), etc.
    - Other requests?
- Input from detector groups:
  - Can you detector's input conform to a set of SpacePoints?

Data Structure: Conclusions

- The code as described is implemented.
- It has documentation and tests.
- It has been through code review:
  - Finalizing corrections / comments now.
- Will be included in next MAUS release.
  - Baring requests / arguments now...

Global Reconstruction Algorithm Update

Peter Lane
Illinois Institute of Technology

Status Summary

- Updated code to use space points instead of Monte Carlo
- Using smeared MC tracker momentum until tracker group finishes their reconstruction code
- Split work flow component (mapper) into two:
    - 1) Take raw data and create all possible track hypotheses (“raw tracks”)
    - 2) Fit raw tracks and return goodness of fit

MapCppGlobalRawTracks

- Takes data in several different input “formats”
  - MC, space points, mock test data, random
- Guess at PID using TOF0 & TOF1.
- Ambiguous PID generates multiple hypotheses
- Generate raw tracks for each hypothesis
- Save as a ROOT-compatible data structure for handing off to MapCppGlobalTrackReconstructor

MapCppGlobalTrackReconstructor

- Gathers raw track hypotheses from MapCppGlobalRawTracks
- Uses configured optics model and fitting algorithm to fit the track hypotheses
- Only the Minuit-based fitting algorithm is currently available, and we're still working on improving it
- Planning to incorporate the Kalman filter into this framework—stalled.