



# Structure of Reconstruction Program

Towards BRUNEL



# Some definitions

---

- ◆ Reconstruction program transforms RAW data into Event Summary Data (ESD)
  
- ◆ RAW data
  - IS: digitised detector data as produced by the DAQ system:
    - ADC counts, TDC counts...
    - Integer data
  - IS NOT: SICB "raw" data:
    - Output of GEANT 3 transport step - entrance and exit points in detectors, energy loss...
  - IS NOT: SICB "digitisings"
    - Output of SICB local tracking and "digitisation" step - wire numbers, signed drift distance...



# More definitions

---

## ◆ Coordinates

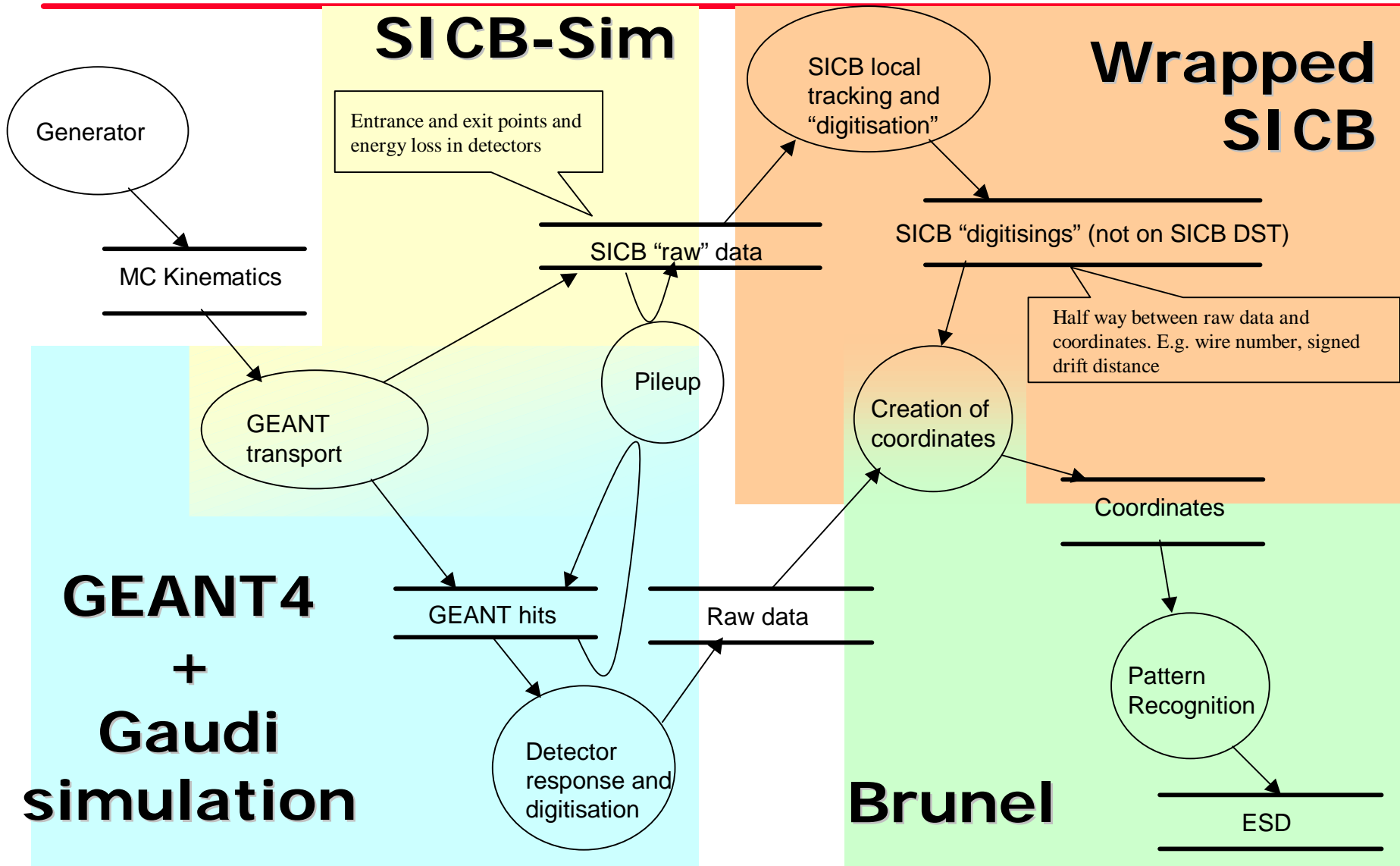
- Results from decoding of RAW data
  - after applying channel maps, calibration, drift-time relation...
  - 2D or 3D points in space, pulse heights, energy per cell...
- Floating point data
- Determined locally to each subdetector
- Intermediate data (usually not saved)

## ◆ ESD

- Results from pattern recognition and reconstruction
  - Tracks, Energy flow objects, PID
- Determined globally
  - May require input from other subdetectors (e.g. tracks for RICH)
  - May combine results of several subdetectors (e.g. tracking, PID)
- Implies sequencing of subdetectors, several passes



# Old and New





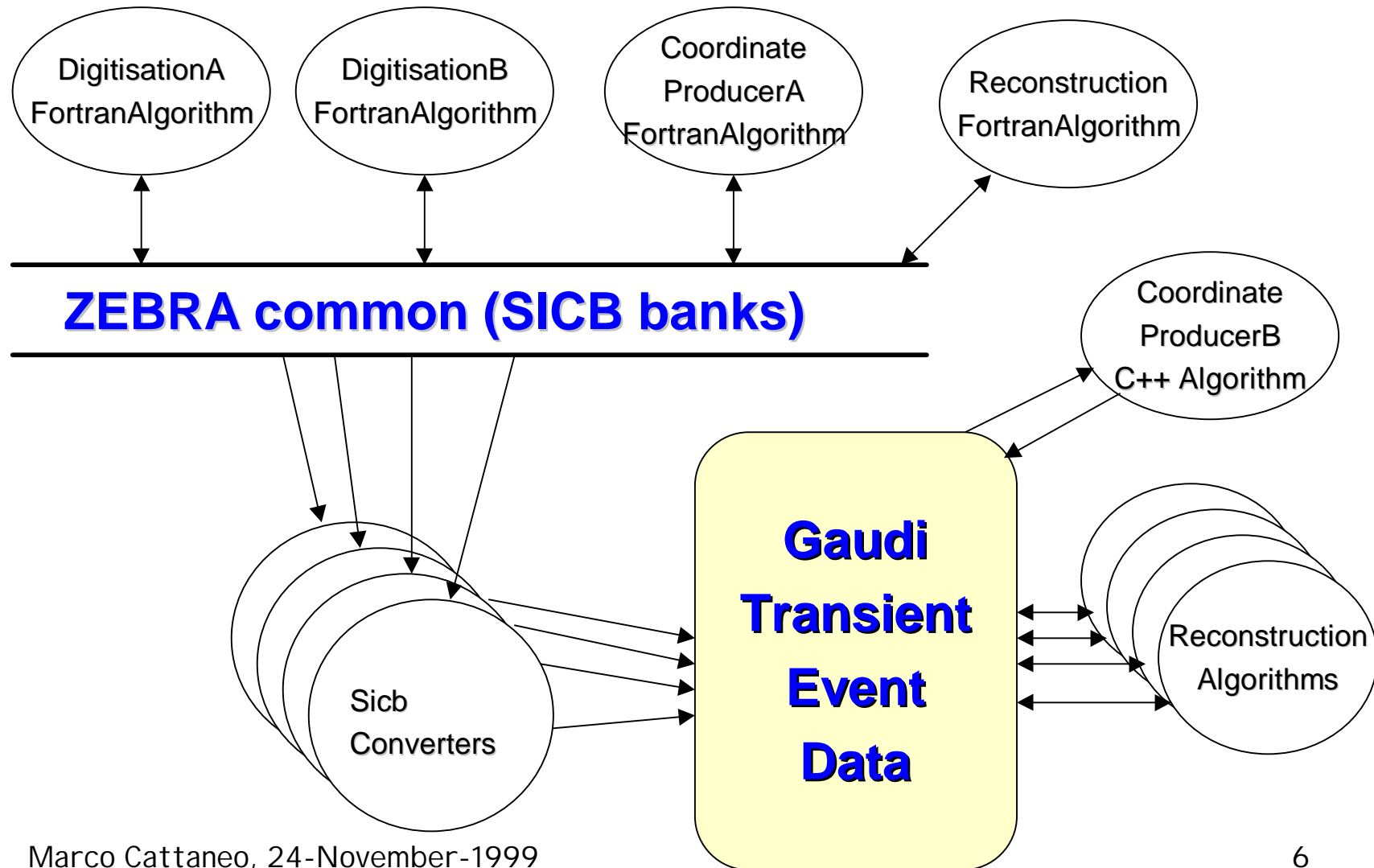
# Towards BRUNEL

---

- ◆ **Input: SICB “raw” data**
- ◆ **Produce SICB “digitisings” in GAUDI FortranAlgorithm**
  - i.e. wrapped FORTRAN. Can be several Algorithms, e.g. one per subdetector
- ◆ **Produce coordinates**
  - Wrapped FORTRAN to SICB coordinates
  - GAUDI SicbConverters to Transient Event Data Model
- ◆ **Pattern recognition, reconstruction**
  - Ideally all C++
  - FortranAlgorithm possible if very modular



# Structure





# Caveat

---

- ◆ In previous slide, SICB converters are unidirectional
  - FortranAlgorithms can produce data for later use in Fortran or C++
  - C++ algorithms can only produce data for later use in C++
  
- ◆ Choice:
  - Write also converters in opposite direction OR
  - Re-implement in C++ ALL reconstruction algorithms requiring input from upstream C++ algorithms



# More Caveats

---

- ◆ **Ultimately, BRUNEL will require RAW data as input**
  - From GEANT4 based simulation, from DAQ
  
- ◆ **Existing SICB “digitisation” will then be obsolete**
  - No point to re-implement now in C++ for BRUNEL
  - Consider converters from SICB “digitisings” to transient raw data, followed by OO co-ordinate creation.
  
- ◆ **Existing wrapped co-ordinate creation code to be re-implemented with RAW data as input**
  - No point to re-implement now in C++ with SICB “digitisings” as input



