Status of

ATLFAST integration into Gaudi

P.Clarke
H.Phillips
E. Richter-Was

+R.Steward
+P.Sherwood

Evolution of task

Original:

- Re-engineer ATLFAST++ into OO design
- Remove ROOT dependencies
- Should be "standalone" capable
- Later: integrate into Gaudi

Short project (3 months ?)

Decoupled from Gaudi

Do full proper design without constraints

Actual task as it has evolved:

- Integrate fully into Gaudi framework imediately
  - not standalone
  - not ROOT
- Implement current ATLFAST++ functionality as STARTING point

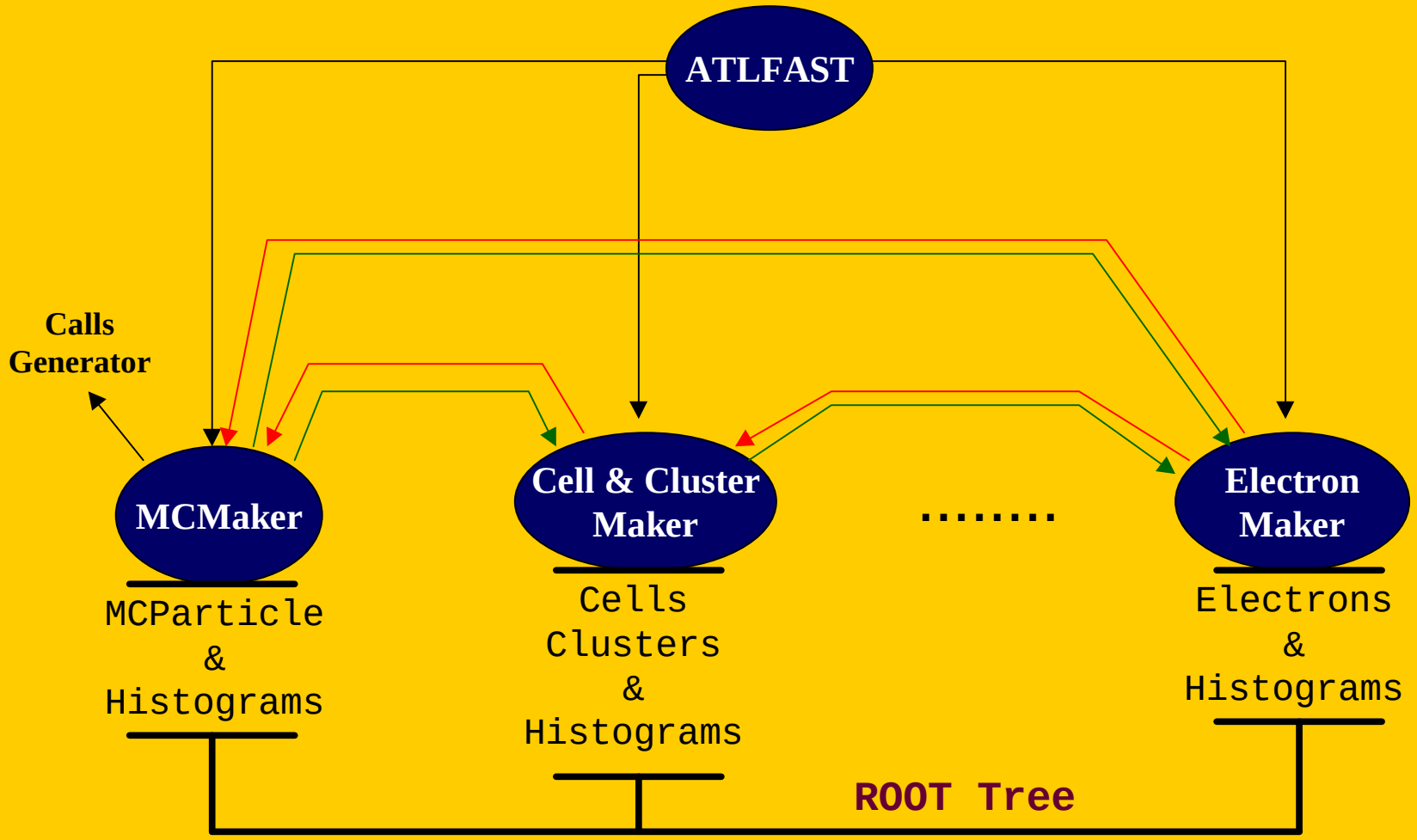
Target September

Completely coupled to Gaudi

Very little design left as Gaudi specifies how you run algorithms

Starting point

Starting point : ATLFAST++



Working decisions

Following task appraisal, several discussions, and outcome of last A-shop:

- **Keep basic concept of "Makers" Þ each to become a "Gaudi Algorithm"**
- **Re-design each Maker into OO**
- **De-couple Detector simulation Û Reconstruction Û Histogramming**
- **Disentangle Makers from having to know of each other, and of ATLFAST structure**

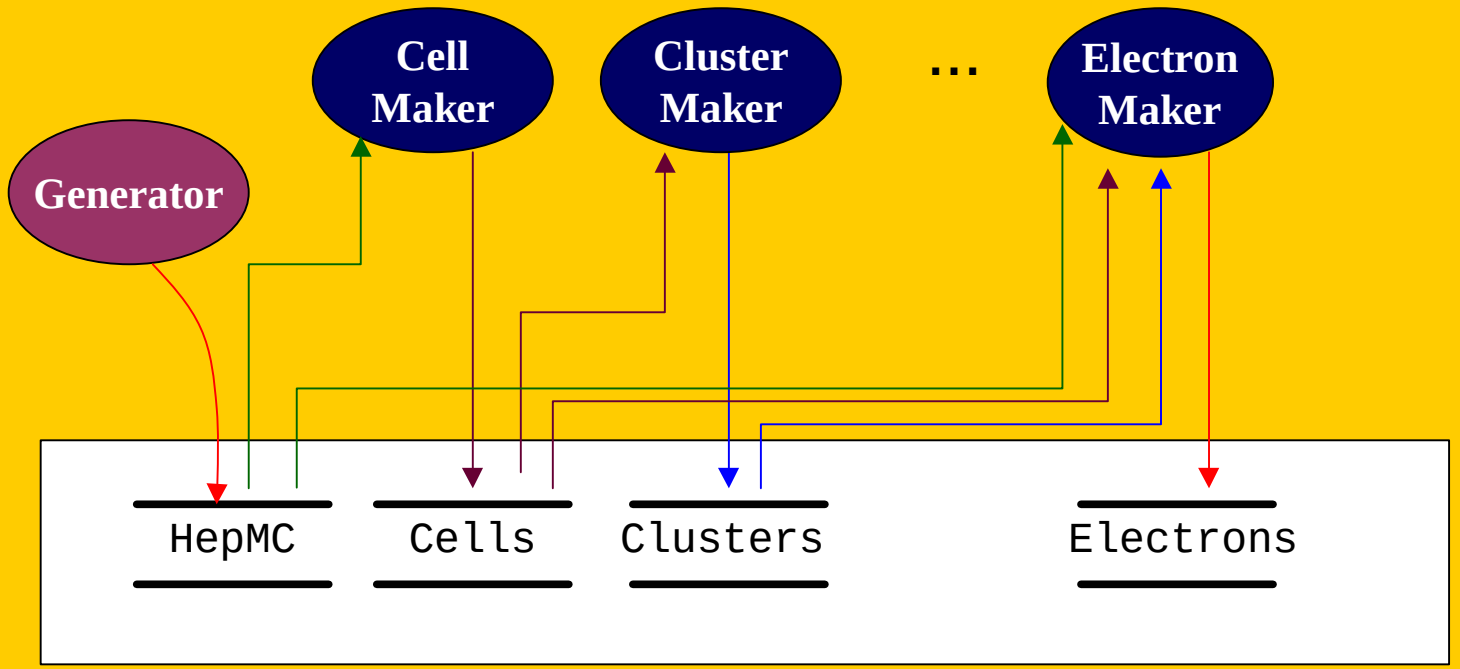
  Use an "event bag" too communicate between Makers Þ Transient event store
- **Keep output entities very similar to current ATLFAST++**
- **Ensure output available in same form as current ATLFAST for comparison**

  (common ntuples used for TDR)

New Structure

Each "Maker" becomes a Gaudi Algorithm

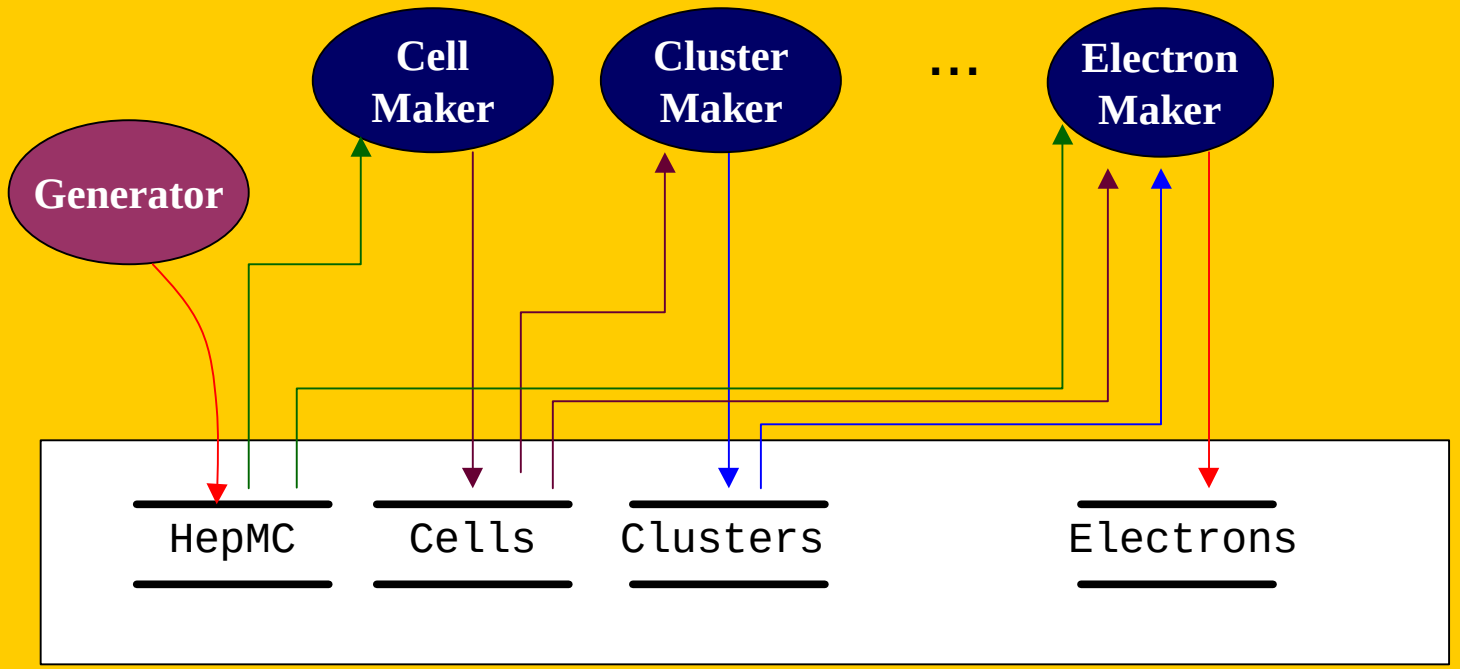
- **Scheduled by Gaudi**
- **Structure is almost completely dictated by Gaudi rules.**
- **Each Algorithm takes from T.E.S and returns its products to T.E.S**

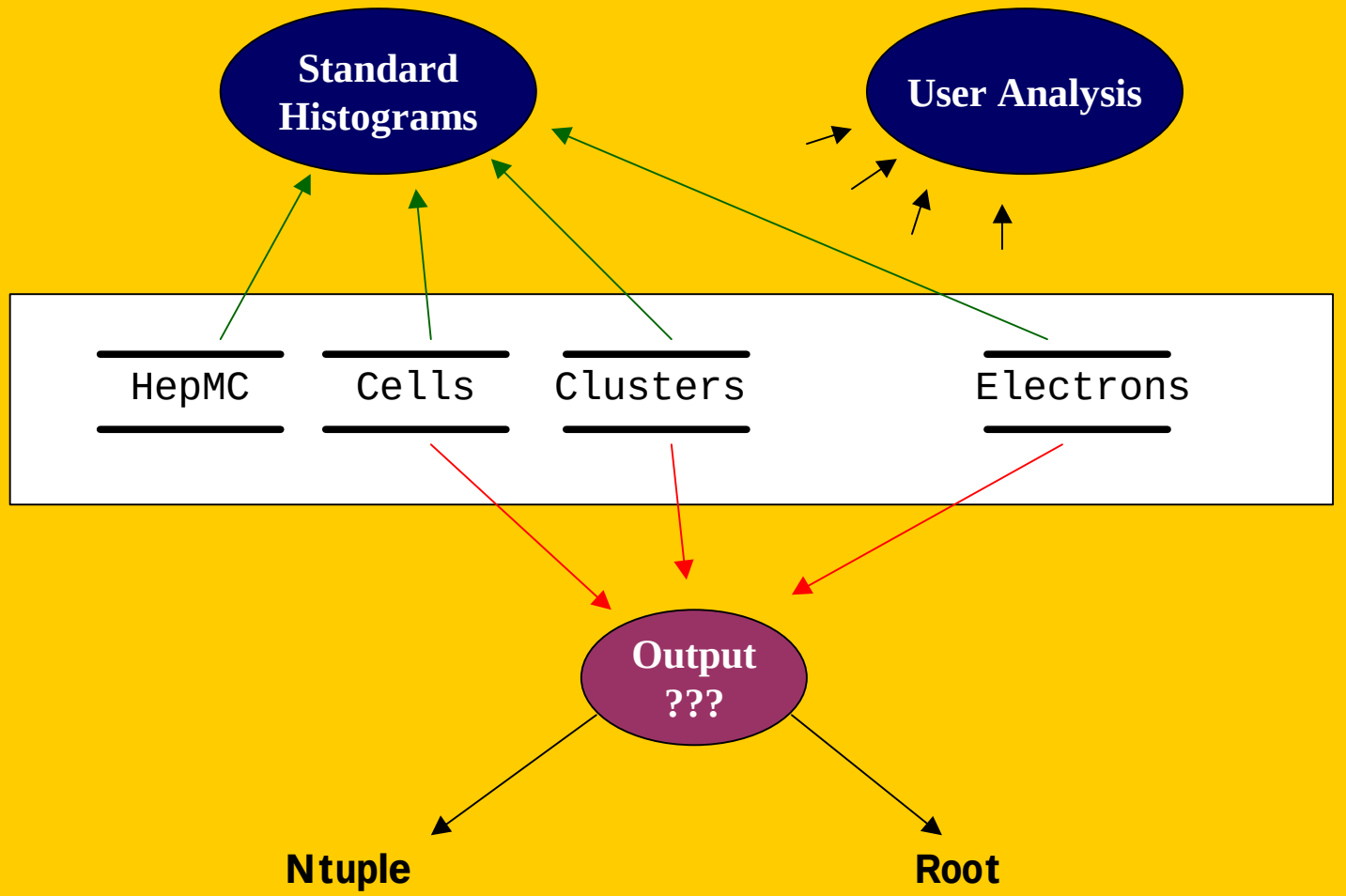


Note to potential users:

- **Straightforward to replace an algorithm with your own favourite (eg different Jet finders)**
- **Straightforward to add new algorithms, following template**

You dont need to know anything about ATLFAST structure, only entities in T.E.S




Standard Histograms
User Analysis
HepMC
Cells
Clusters
Electrons
Output ???
Ntuple
Root

Status of work today (September target)

Task is now defined

Example Algorithm (ElectronMaker) written

- **to act as template for other ATLFAST algorithm providers (eg different JetMakers)**

  Þ **Immediate next task is to get it running in Gaudi**

CellMaker, ClusterMaker, MuonMaker underway following example

Technical points:

- **All parameters set by job-options service**
- **Products (eg Electrons) go into store in collection: ObjectVector<Electron>**
- **Collections will be typedefined : ElectronCollection**
- **Iterator to collection typedfined : ElectronIterator**

Work to do in parallel (perhaps, but not necessarily for September)

1. More complete long term USER REQUIREMENTS definition

- E.Richter - Was organising a trawl of users and meeting in June at CERN
- Will lead to written defintion of user requirements

2. For each Maker

- Upgrade to include same sophistication as ATLFAST-FORTRAN version

- Add knowledge from other users / providers who have modified ATLFAST

  Þ preferably by enabling other users / providers to write ATLFAST algorithms themselves