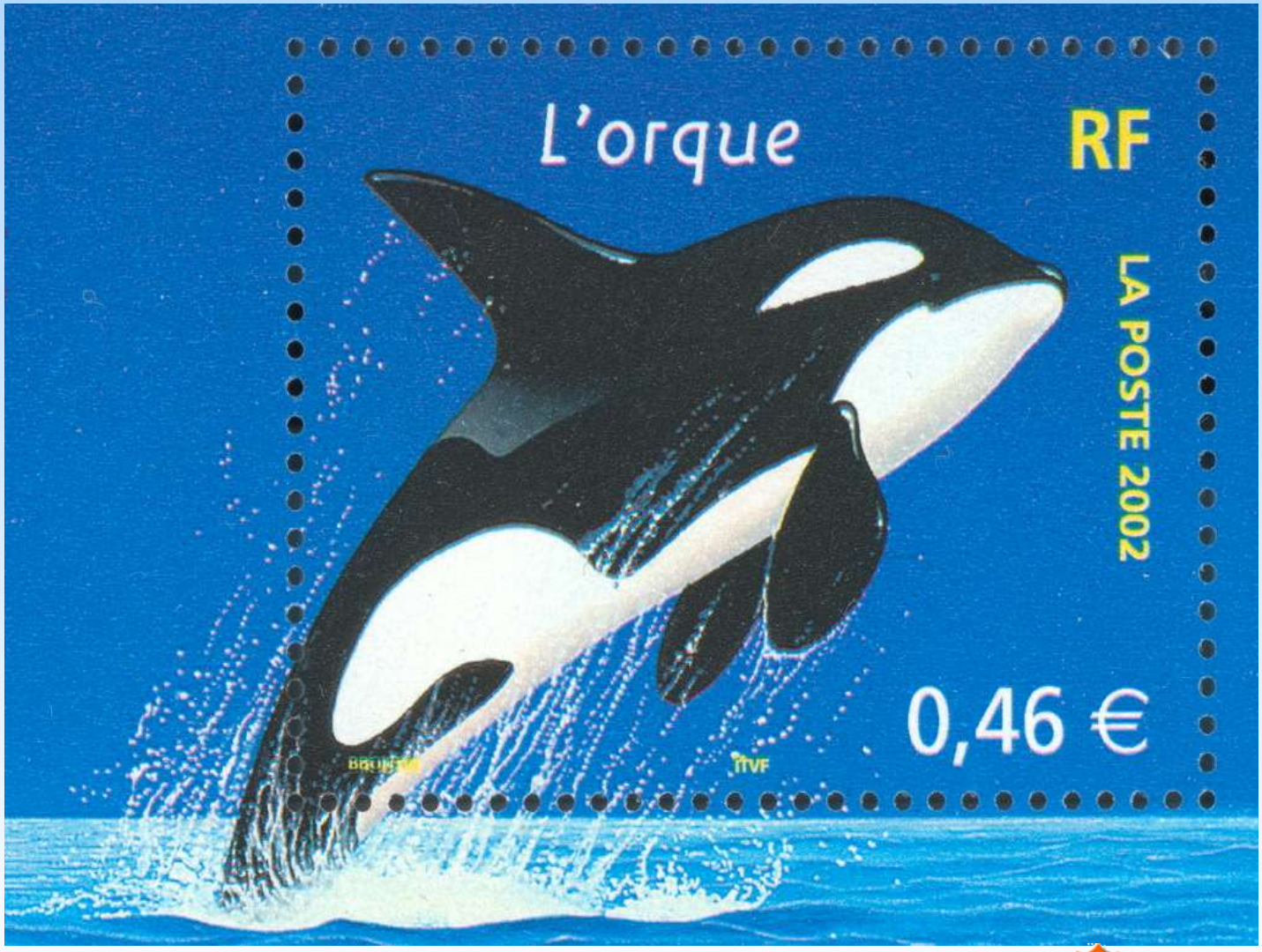
Using the Reconstruction Software, ORCA, in the CMS Data Challenge



Stephan Wynhoff, Princeton University

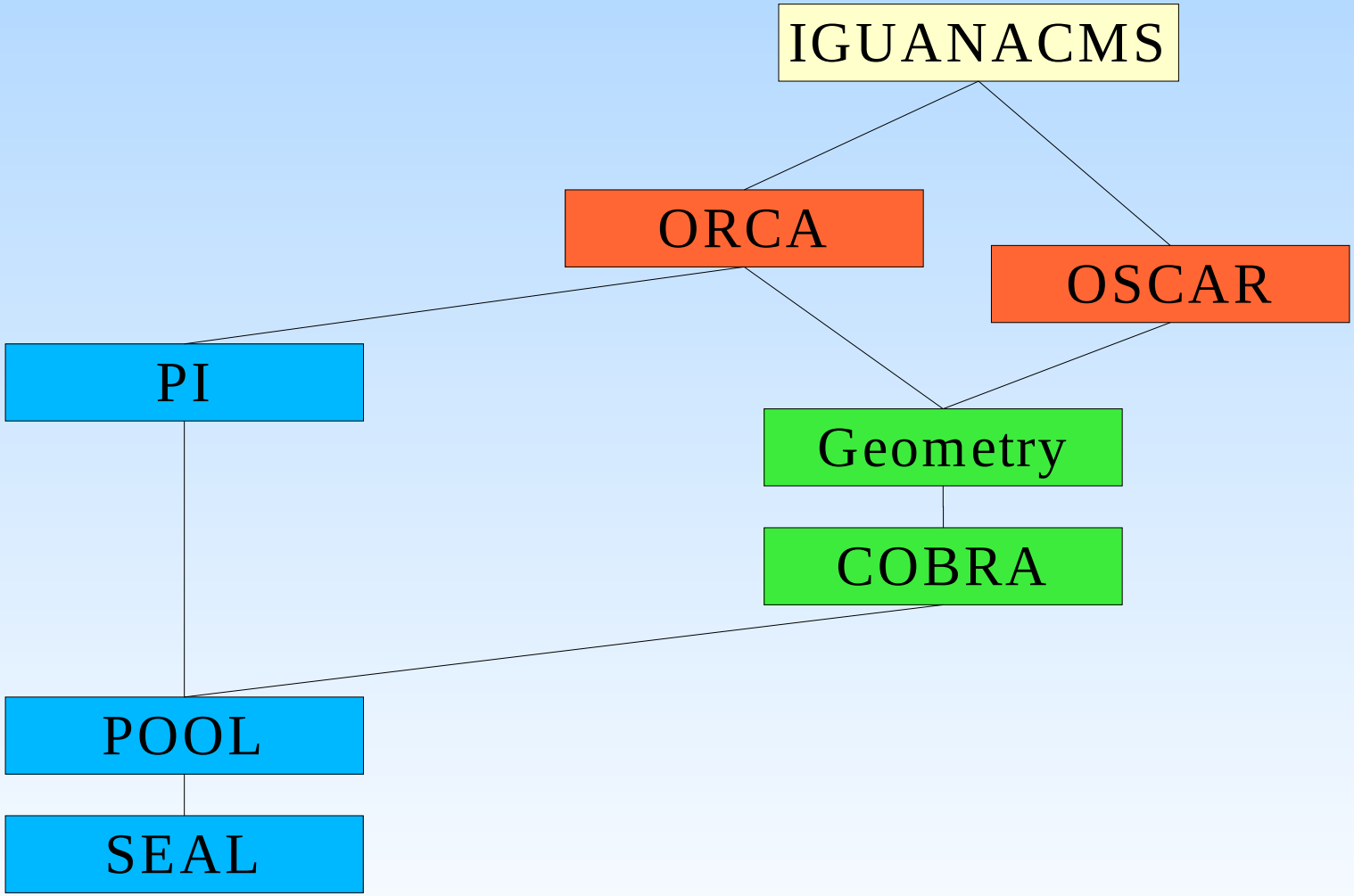
Overview

- The ORCA project
- Implementation of DSTs to store
- Spring 2004: DataChallenge '04, i.e. DST Production

The ORCA Project

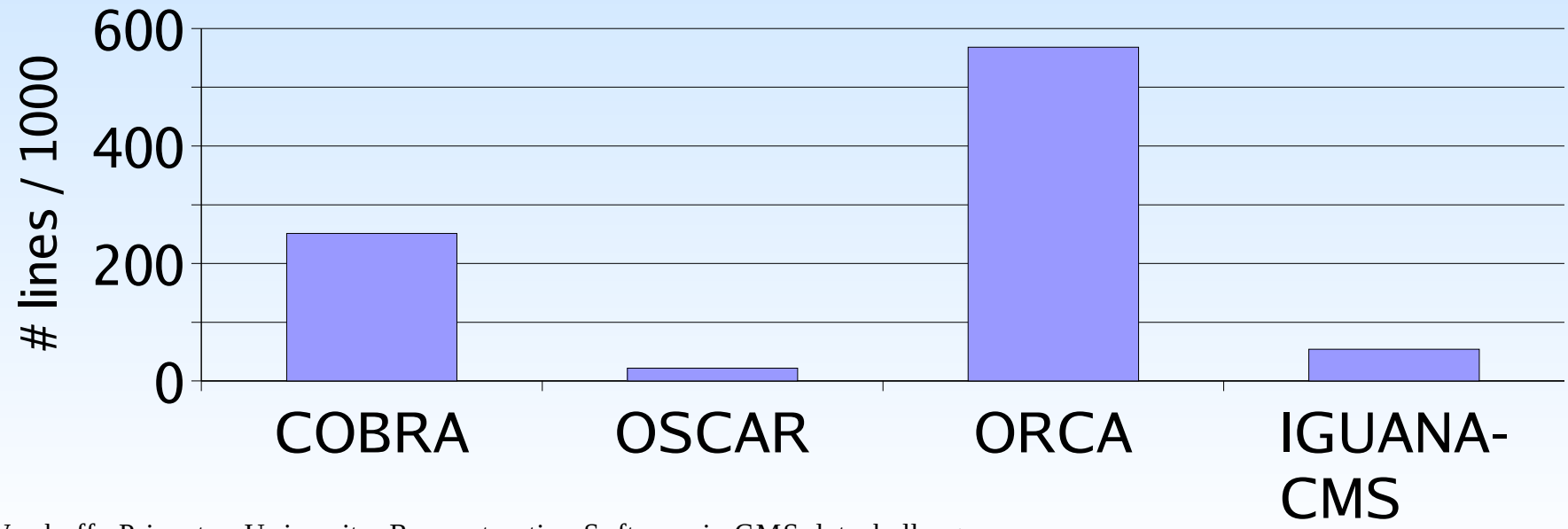
- The ORCA software project is managed by RPROM group (Reconstruction PROject Management).
  Ensures quality for:
  - Detector digitization, i.e. Combination of signal with pileup events and electronics simulation
  - Simulation and eventual validation of Trigger decisions
  - Reconstruction of detector objects:
    - Tracks, Clusters, Vertices...
    - The algorithmic code for use in the HighLevelTrigger
  - "Offline" reconstruction of Physics Objects:
    - Jets, Electrons, Photons, missing Et...

Tools & Services Required

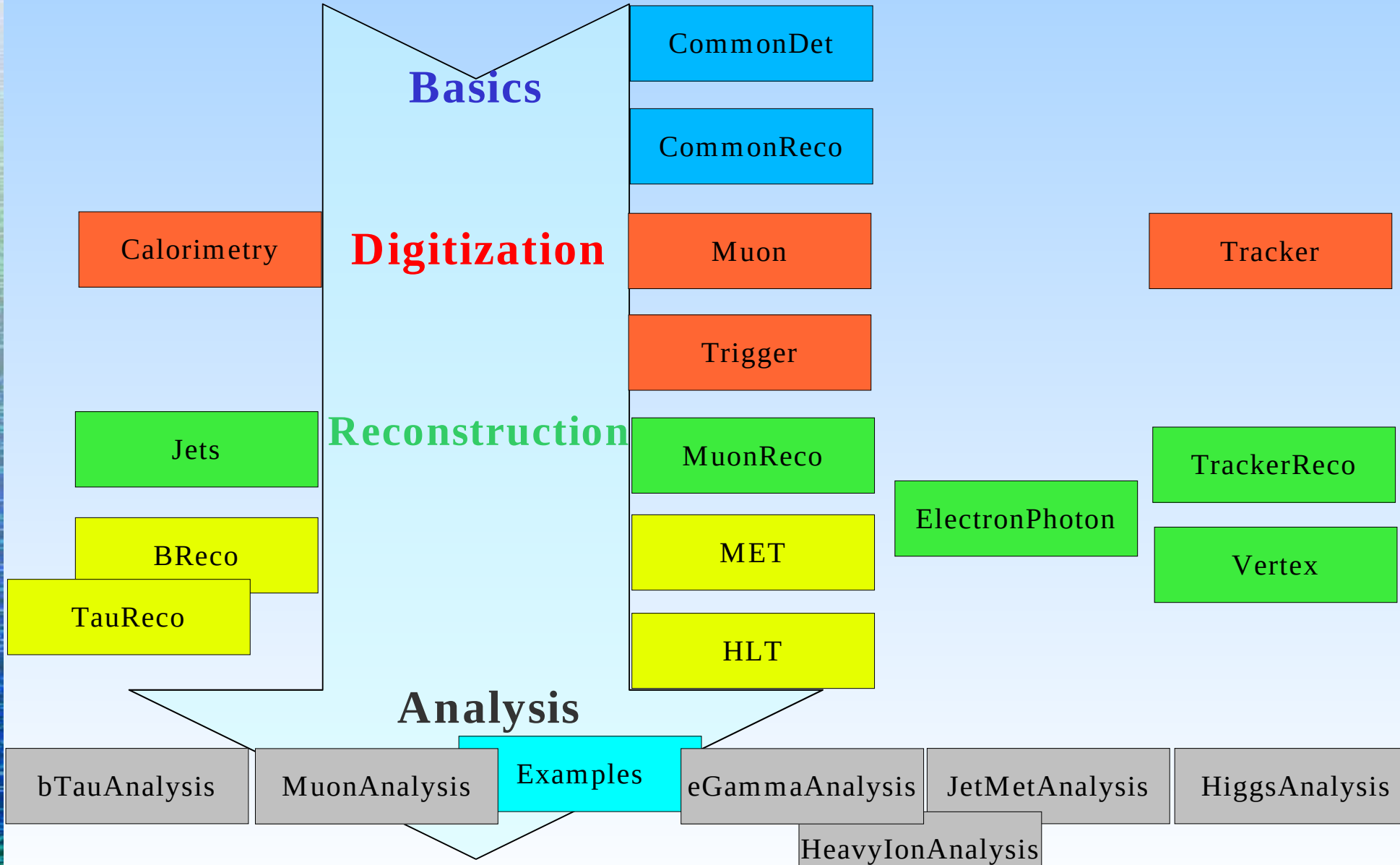


ORCA Project Size

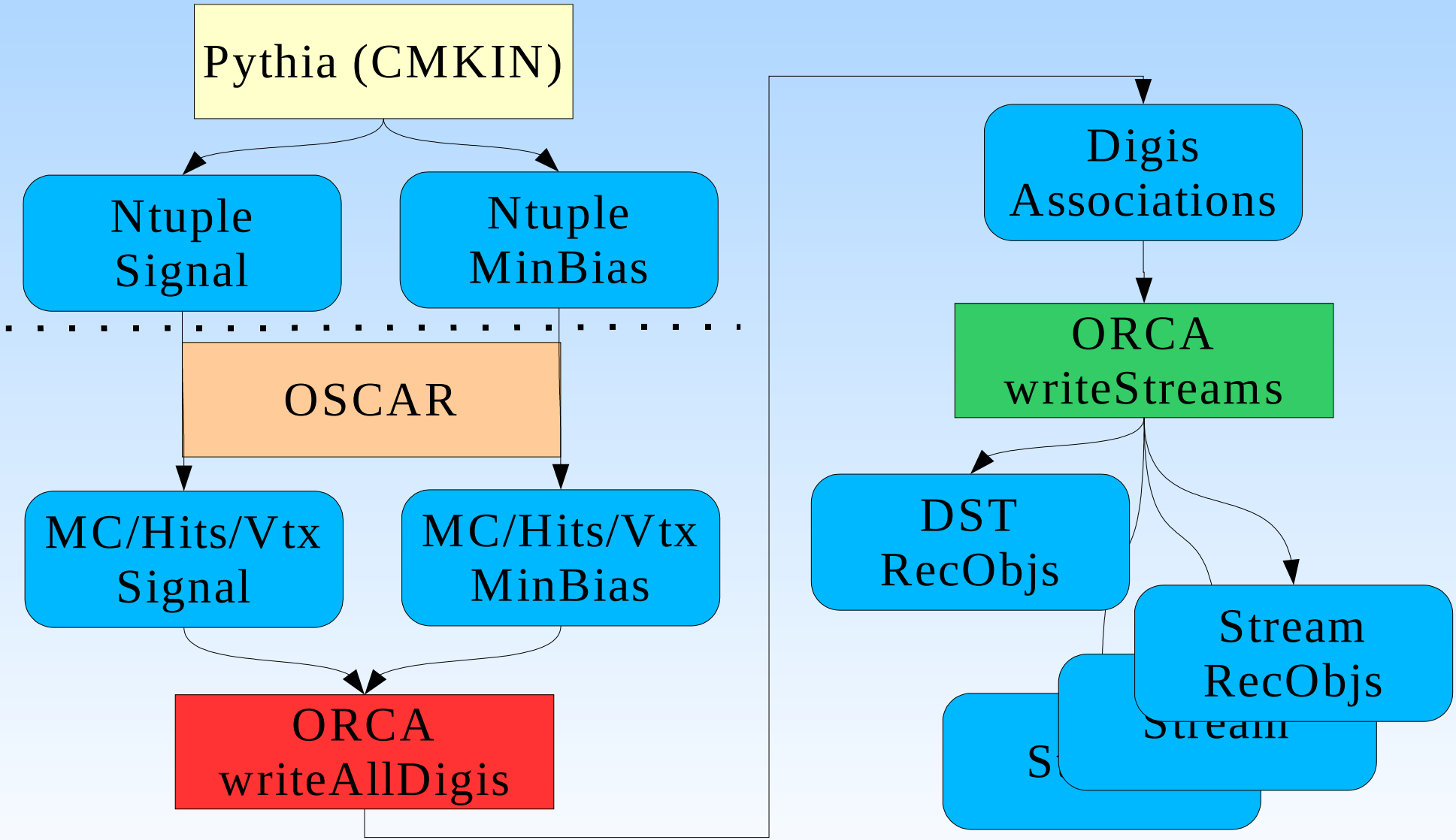
- Release ORCA 8.1.3
  - 22 subsystems
  - 373 packages
  - 5893 files
- 200 developers



ORCA Project Structure



From Simulation to Reconstruction



Implementation of DSTs

- Digitization working since a long time
- Analysis from reading Digis (i.e. Running the reconstruction) takes very long
- The size of the reconstructed information is smaller than the “raw” Digis

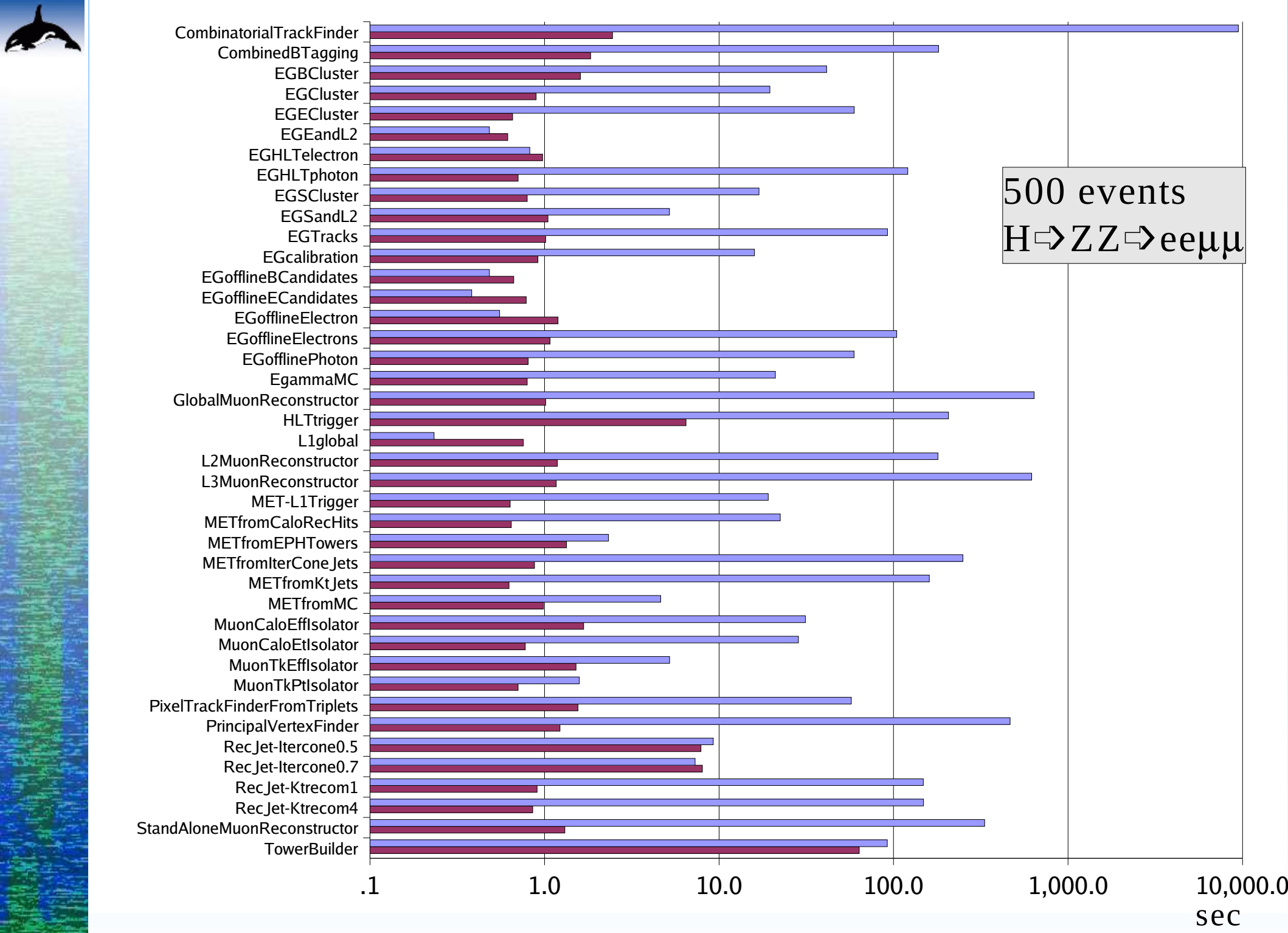
- Store all possible reconstructed info in DST
- Implemented in 2 months for DataChallenge '04
  - Very fast – a lot of work from many ORCA (and COBRA) developers
  - Not perfect but usable

DST Contents

- We stored already (in Digis and below)
  - MCinfo: generator, SimTrack, SimVertex
  - SimHits: subdetector specific
  - Digis & Associations (to MC)
- DST contains collections of reconstructed objects
  - Tracker-tracks (combinatorial track finder)
  - in total about 40 different RecCollections
  - add L1Trigger (if not in Digis sample) and CaloRecHits
- Selection of what to store via configuration file

RecCollections

```
RecCollection<RecoVertex>*                theVtxCollection;
RecCollection<TTrack>*                    theTkCollection, *theTkPxlCollection;
RecCollection<EcalPlusHcalTower>*         theCaloCollection;
RecCollection<EgammaBasicCluster>*        theBasicClusterCollection;
RecCollection<EgammaCluster>*             theClusterCollection;
RecCollection<EgammaSuperCluster>*        theSuperClusterCollection;
RecCollection<EgammaEndcapCluster>*       theEndcapClusterCollection;
RecCollection<EgammaCandidate>*           theL2BarrelCandidateCollection, *theL2EndcapCandidateCollection;
RecCollection<EgammaCandidate>*           theOfflineBarrelCandidateCollection,
RecCollection<EgammaCandidate>*           theOfflineEndcapCandidateCollection;
RecCollection<TTrack>*                    theHLTElectronTracksCollection, *theOfflineElectronTracksCollection;
RecCollection<PhotonCandidate>*           theHLTPhotonCandidateCollection;
RecCollection<PhotonCandidate>*           theOfflinePhotonCandidateCollection;
RecCollection<ElectronCandidate>*         theHLTElectronCandidateCollection,
RecCollection<ElectronCandidate>*         theOfflineElectronCandidateCollection;
RecCollection<KineHBlock>*                theMCCollection;
RecCollection<EgammaCalibObject>*         theEGCalibrationCollection;
RecCollection<RecMuon>*                   theMuCollection, *theMuL2Collection, *theMuL3Collection;
RecCollection<RecMuon>*                   theMuGlobalCollection;
RecCollection<RecMuon>*                   theMuCaloEtIsoCollection, *theMuCaloEffIsoCollection;
RecCollection<RecMuon>*                   theMuTkPtIsoCollection, *theMuTkEffIsoCollection;
RecCollection<RecJet>*                    theKtJetCollection1, *theKtJetCollection2;
RecCollection<RecJet>*                    theIterConeJetCollection1, *theIterConeJetCollection2;
RecCollection<BTaggedJet>*                theBJetCollection;
RecCollection<RecMET>*                    theMETephtCollection, *theMETcrhCollection;
RecCollection<RecMET>*                    theMETicJetCollection, *theMETktJetCollection;
RecCollection<RecMET>*                    theMETparticleCollection;
RecCollection<L1TriggerMET>*              theMETL1triggerCollection;
RecCollection<HighLevelTriggerResult>*    theHLTCollection;
```


CombinatorialTrackFinder
CombinedBTagging
EGBCluster
EGCluster
EGECluster
EGEandL2
EGHLTelectron
EGHLTphoton
EGSCluster
EGSandL2
EGTracks
EGcalibration
EGofflineBCandidates
EGofflineECandidates
EGofflineElectron
EGofflineElectrons
EGofflinePhoton
EgammaMC
GlobalMuonReconstructor
HLTtrigger
L1global
L2MuonReconstructor
L3MuonReconstructor
MET-L1Trigger
METfromCaloRecHits
METfromEPHTowers
METfromIterCone Jets
METfromKt Jets
METfromMC
MuonCaloEffIsolator
MuonCaloEtIsolator
MuonTkEffIsolator
MuonTkPtIsolator
PixelTrackFinderFromTriplets
PrincipalVertexFinder
RecJet-Itercone0.5
RecJet-Itercone0.7
RecJet-Ktrecom1
RecJet-Ktrecom4
StandAloneMuonReconstructor
TowerBuilder
500 events
H⇨ZZ⇨eeμμ
.1
1.0
10.0
100.0
1,000.0
10,000.0
sec

Using the DSTs

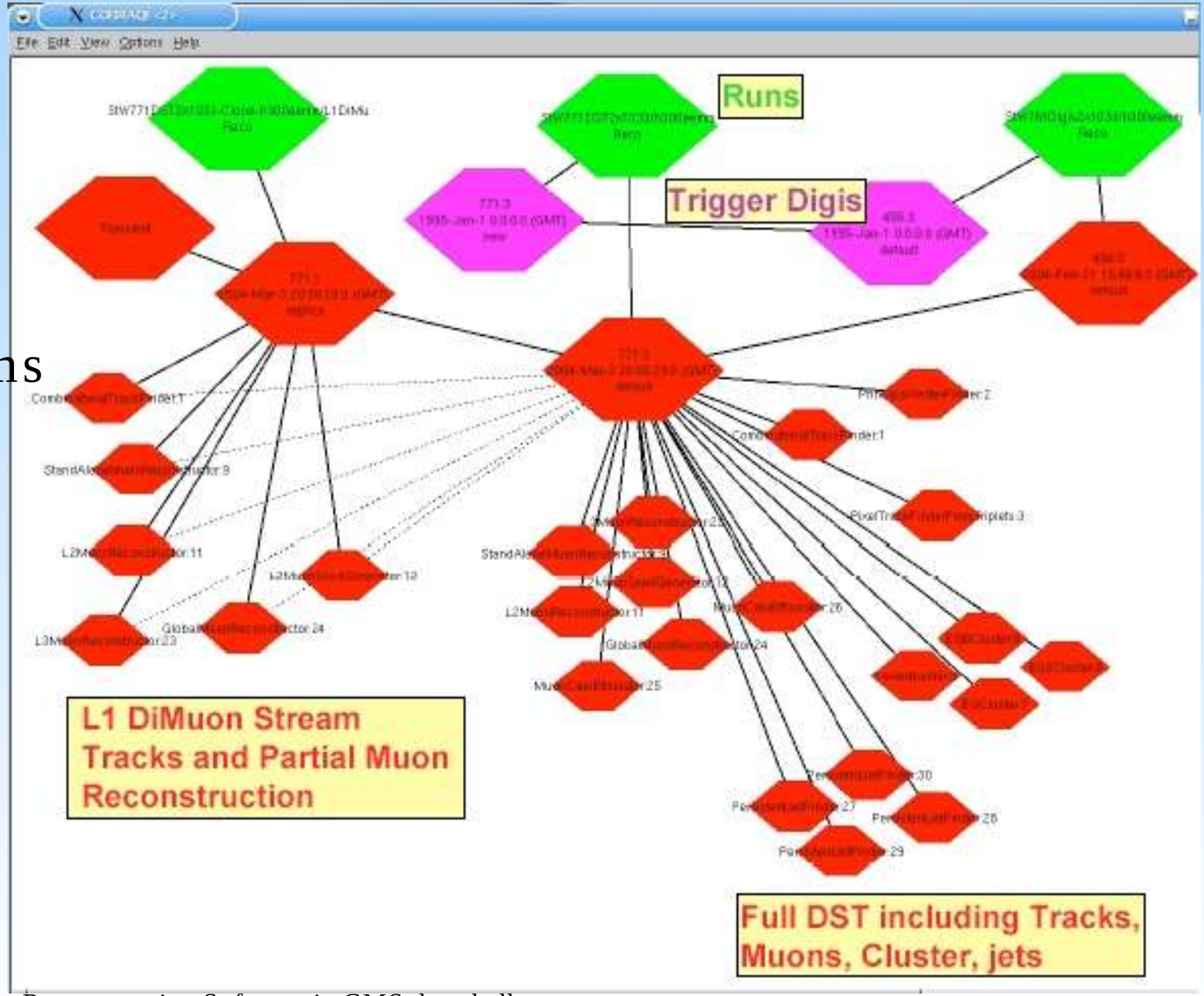
- Simple access to the stored objects

```
if (theTkCollection == 0) {
      theTkCollection = new RecCollection<TTrack>( RecQuery
("CombinatorialTrackFinder"));
}
cout << "[OVAL] Found " << theTkCollection->size()
     << " Tracker tracks." << endl;
GlobalVector p;
for(RecCollection<TTrack>::const_iterator it=theTkCollection->begin();
    it!=theTkCollection->end(); it++) {
  p = (*it)->momentumAtVertex();
  cout << "p=" << setw(8) << setprecision(3) << p.mag()
       << " GeV, theta=" << setw(8) << setprecision(4) << p.theta()
       << " rad, phi=" << setw(8) << setprecision(4) << p.phi()
       << " rad =" << setw(8) << setprecision(4) << p.phi()*180/3.1416
       << " deg, eta=" << setw(8) << setprecision(4) << p.eta()
       << endl;
}
```

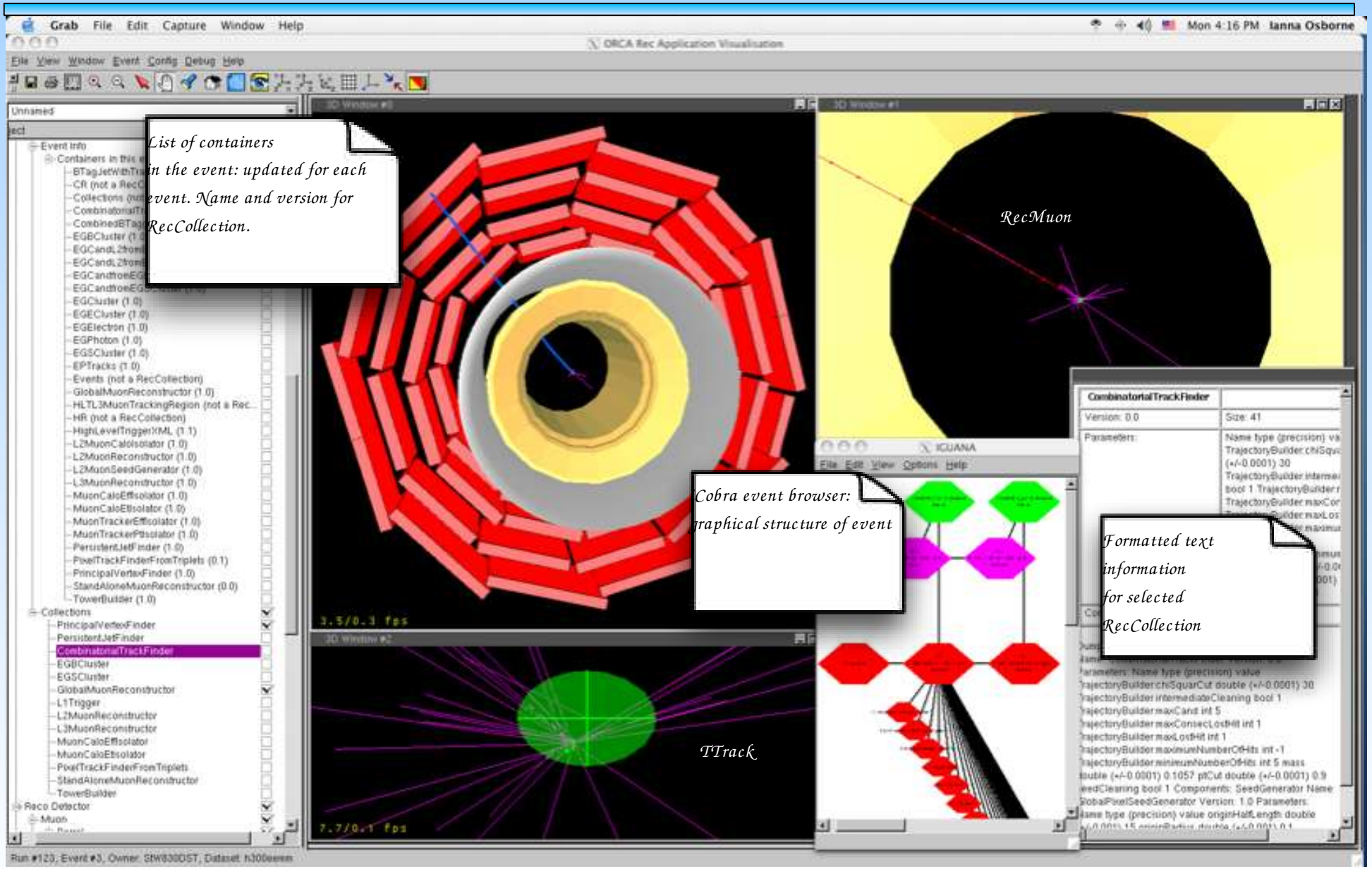
- Reconstruction from the “assigned” Digi event samples by the specified algorithm (`CombinatorialTrackFinder`)

DST Streams

- Streams get complex
- Links back to Digis etc
- Writing streams creates many files
- Reading fast and furious!
- Tested with Calorimetry calibration

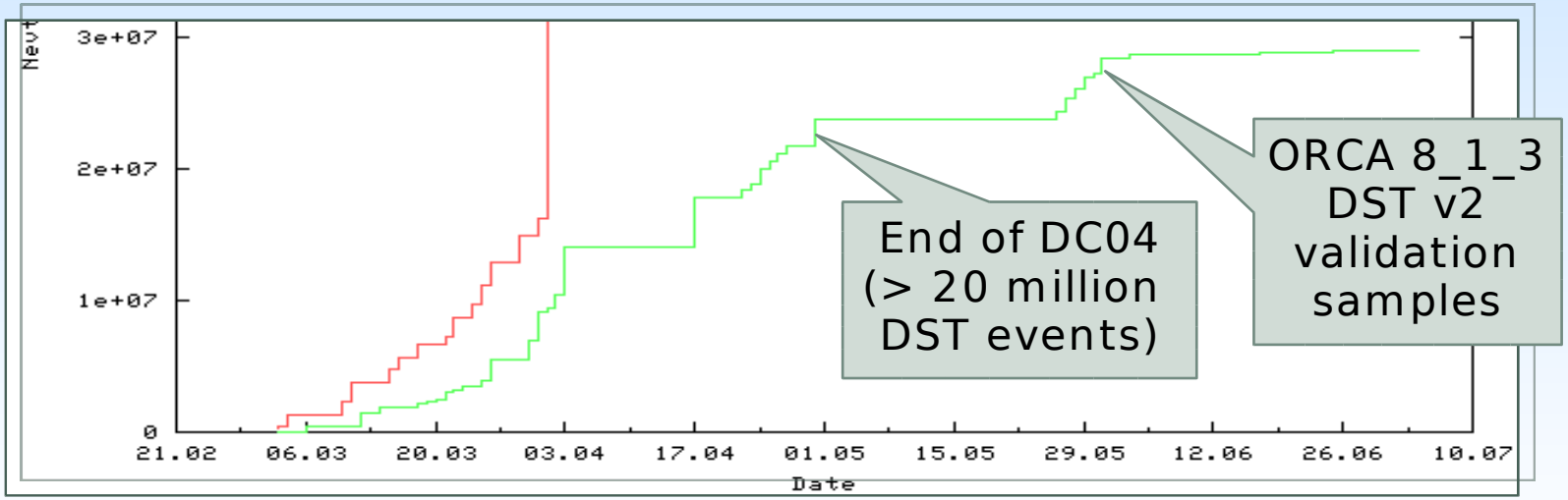


IGUANACMS on DSTs



Data Challenge '04

- ORCA code running at high reliability
  - After hunting memory leaks; Started with 8.0.1
- More than 20 million events (some repeated)
- Time dominated by Tracker-Track reconstruction (90%, all Tracks)
- Some improved ORCA releases since then (8.4.0)



Summary

- ORCA is big reconstruction software project of CMS
  - Up to 2003 focussed on Digitization
- DST allows to make Reconstructed Objects persistent
  - All RecObjs asked for storable in main collection
  - Different (sub-)sets in parallel streams
  - Analysis fast with reading DSTs
- Data challenge 2004 was first main test
  - ORCA executable running stable
  - > 20 million events processed and stored in main collection and parallel streams