MC update to Geant4-v.11

Simulation meeting

Melba D'Astolfo, 28/04/2025

CYGNO'S

MONTE CARLO

GEANT4
A SIMULATION TOOLKIT

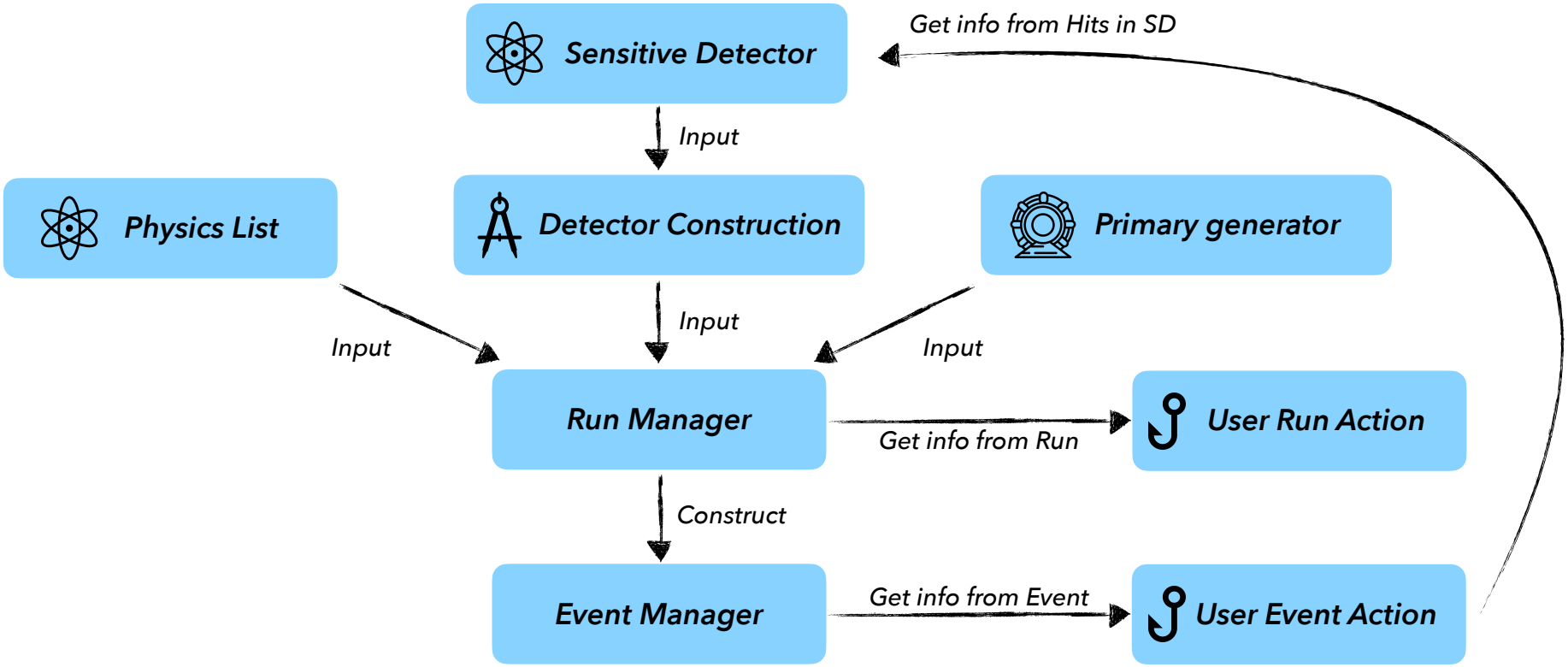
CADMesh

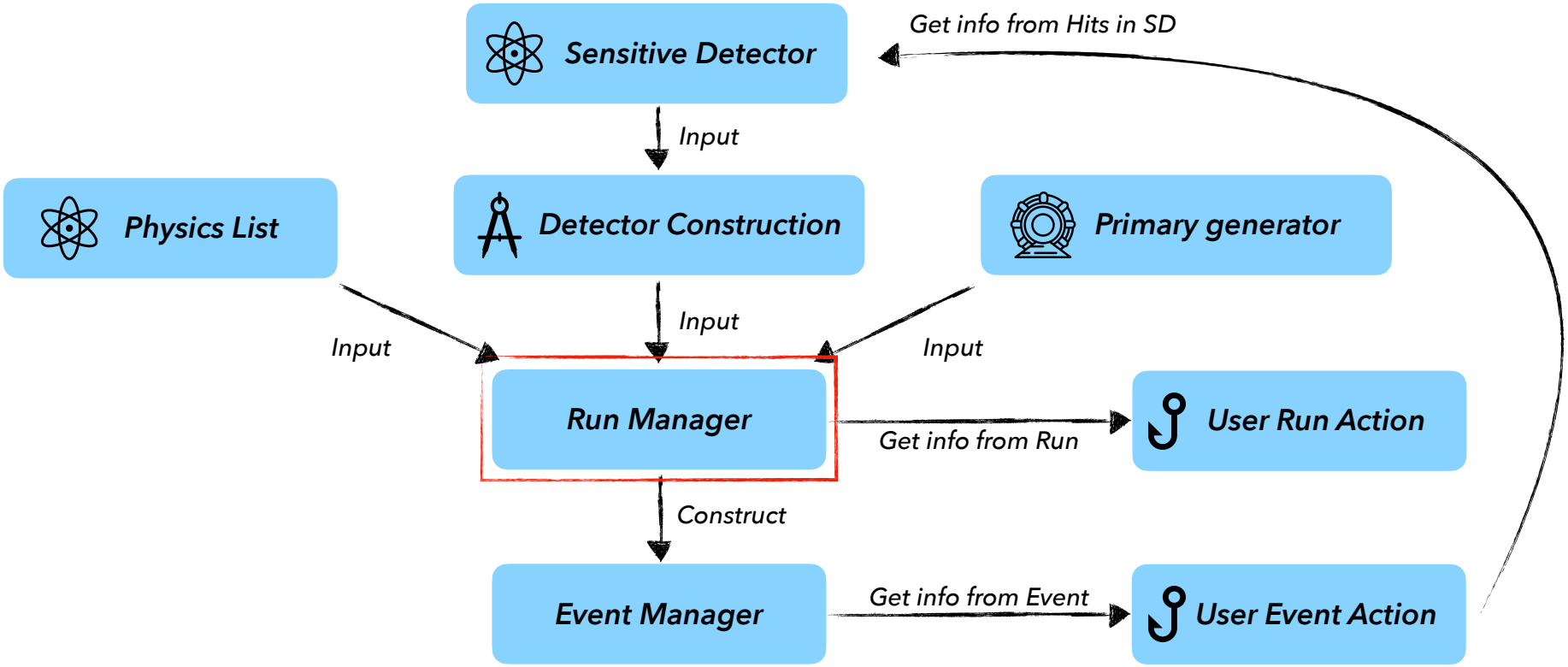
CYGNO'S

MONTE CARLO

GEANT4
A SIMULATION TOOLKIT

CADMesh


Sensitive Detector
Get info from Hits in SD
Input
Physics List
Detector Construction
Primary generator
Input
Input
Input
Run Manager
Get info from Run
User Run Action
Construct
Event Manager
Get info from Event
User Event Action


Sensitive Detector
Get info from Hits in SD
Input
Physics List
Detector Construction
Primary generator
Input
Input
Input
Run Manager
Get info from Run
User Run Action
Construct
Event Manager
Get info from Event
User Event Action

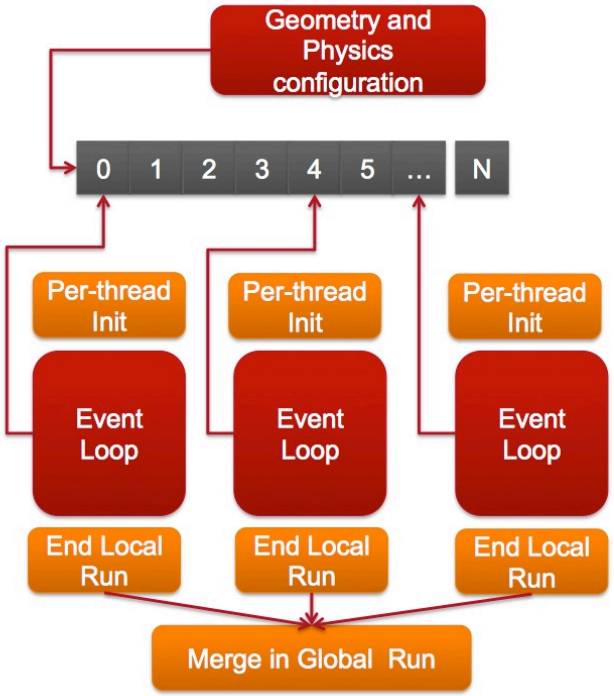
CYGNO.cc

```
// Run manager
G4MTRunManager* runManager = new G4MTRunManager;
runManager->SetNumberOfThreads(G4Threading::G4GetNumberOfCores());
```

+

CYGNOActionInitialization.cc

```
void CYGNOActionInitialization::BuildForMaster() const
{
  CYGNORunAction* run_action = new CYGNORunAction(event_action, fDetector);
  SetUserAction(run_action);
}
```

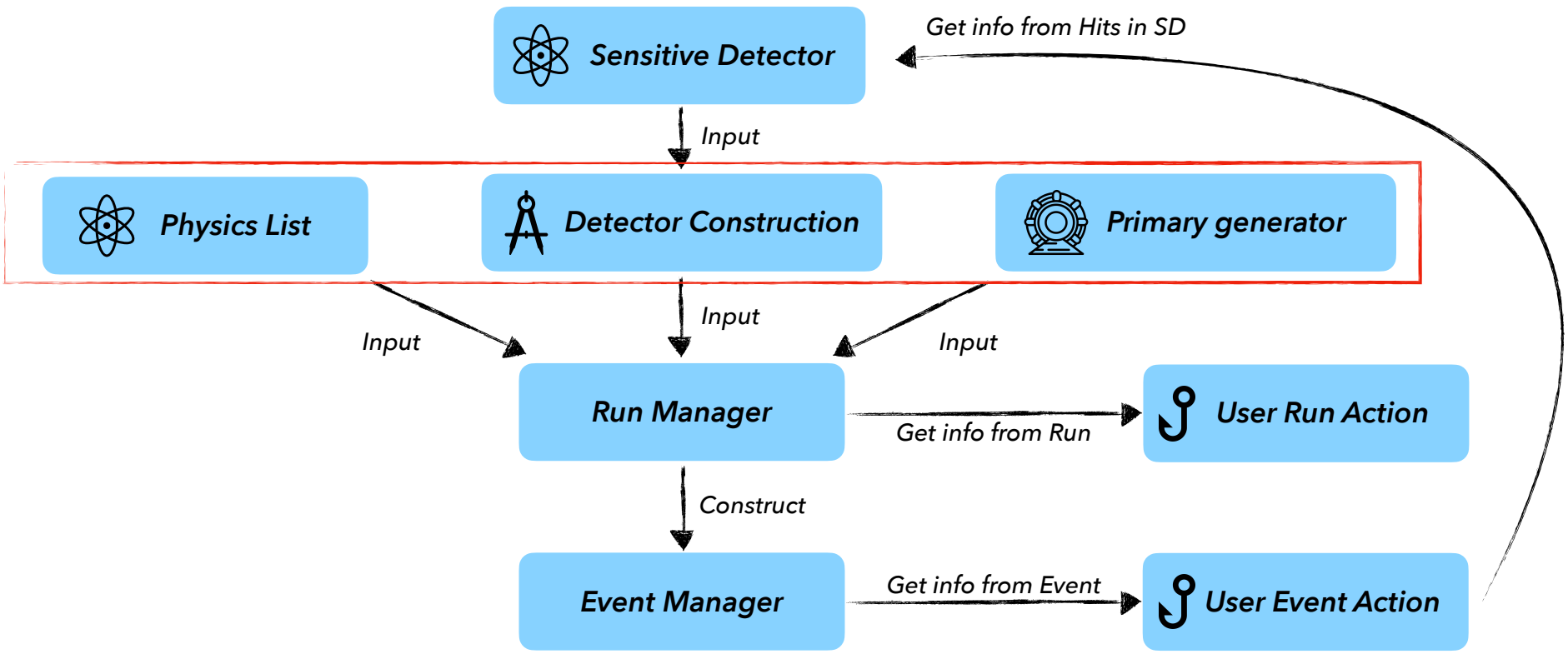


Per-event seeds pre-prepared in a "queue"

Threads compete for next event to be processes (new in ref-08)

Command line scoring and G4tools automatically merge results from threads

MUCH FASTER!


Sensitive Detector
Get info from Hits in SD
Input
Physics List
Detector Construction
Primary generator
Input
Input
Input
Run Manager
Get info from Run
User Run Action
Construct
Event Manager
Get info from Event
User Event Action

```
// Em options
G4EmProcessOptions emOptions;
emOptions.SetBuildCSDARange(true); //not really fundamental
emOptions.SetDEDXBinningForCSDARange(10*10); //not really fundamental
emOptions.SetFluo(true);
emOptions.SetAuger(true);
emOptions.SetPIXE(true);
```

BEFORE

```
// Em options
G4EmParameters* emParameters = G4EmParameters::Instance();
emParameters->SetFluo(true);
emParameters->SetPixe(true);
emParameters->SetAuger(true);
```

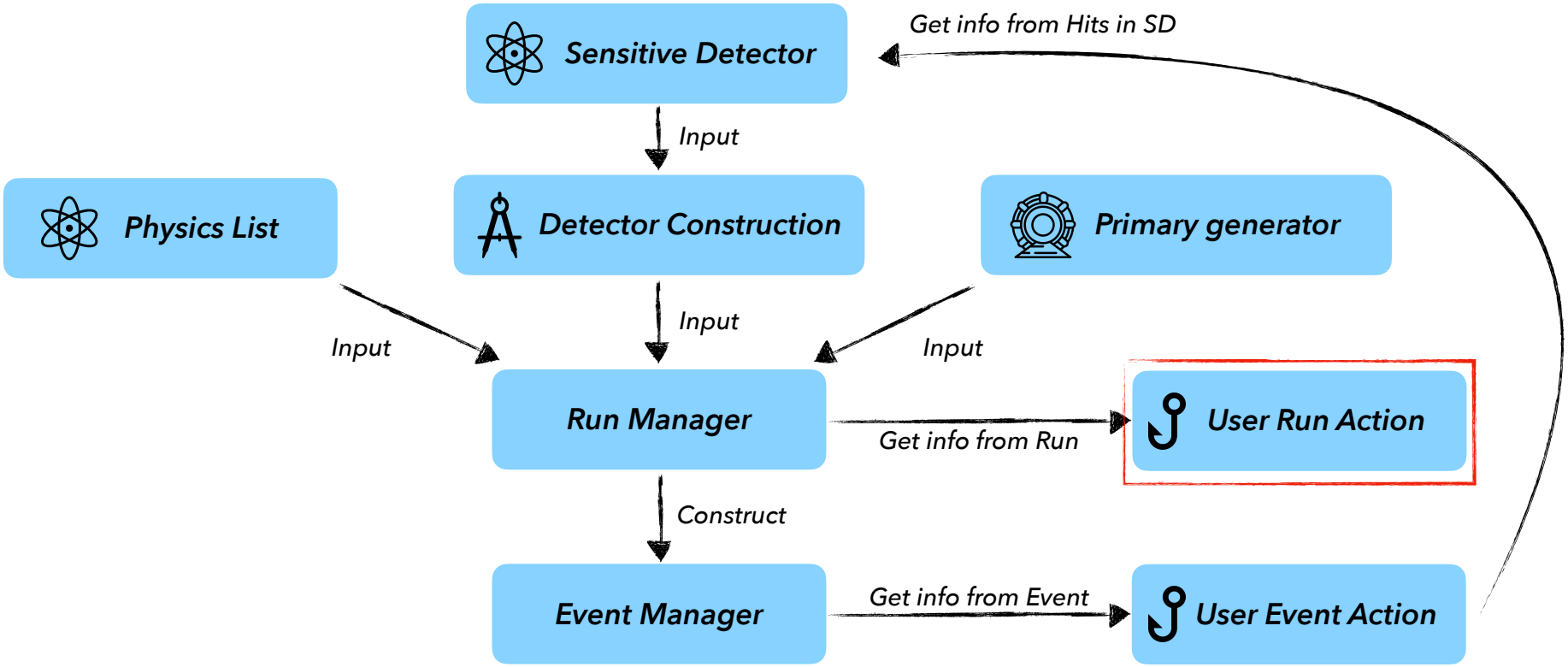
AFTER

CYGNODetectorConstruction.cc

```
void CYGNODetectorConstruction::ConstructSDandField()
{
  //
  // Sensitive detectors
  //
  auto cygnoSD
    = new CYGNOSensitiveDetector("cygnoSD");
  G4SDManager::GetSDMpointer()->AddNewDetector(cygnoSD);
  SetSensitiveDetector("CYGNO_log",cygnoSD);

}
```

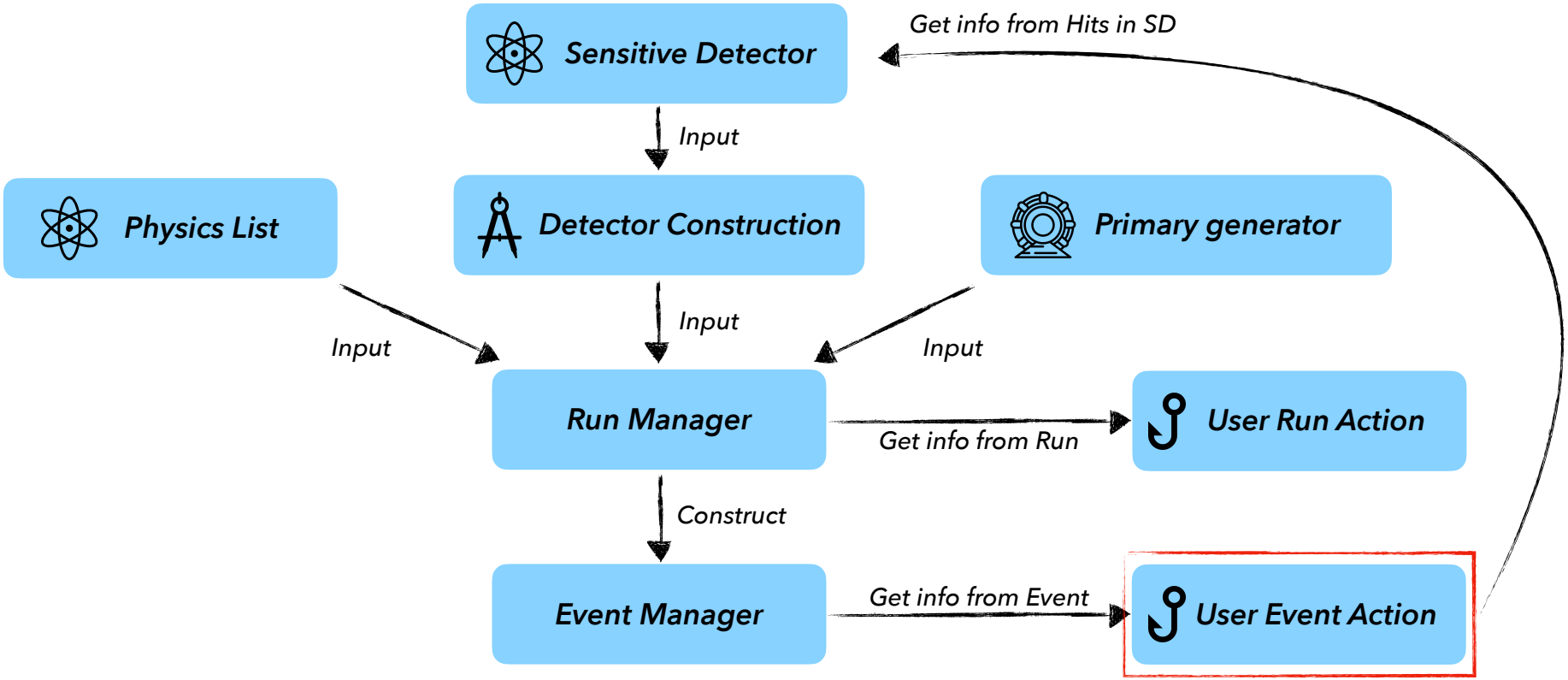
```
void CYGNOPrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent)
{
  // Generate primary vertex
  particleGun->GeneratePrimaryVertex(anEvent);

  if (fFillNtuple) {
    energy_pri = 0.;

    // Access particle energy
    energy_pri = particleGun->GetParticleEnergy();

    // Access particle position
    G4ThreeVector particlePosition = particleGun->GetParticlePosition();

    //Fill ntuple #1
    G4AnalysisManager* man = G4AnalysisManager::Instance();
    man->FillNtupleDColumn(1,0,energy_pri);
    man->FillNtupleDColumn(1,1,particlePosition[0]);
    man->FillNtupleDColumn(1,2,particlePosition[1]);
    man->FillNtupleDColumn(1,3,particlePosition[2]);
    man->AddNtupleRow(1);
  }
}
```

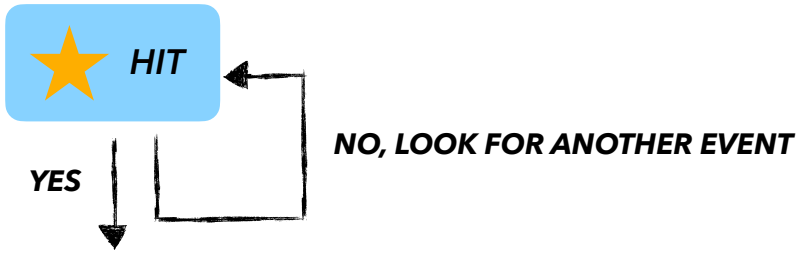
STORE PRIMARY PARTICLES INFO


Sensitive Detector
Get info from Hits in SD
Input
Physics List
Detector Construction
Primary generator
Input
Input
Input
Run Manager
Get info from Run
User Run Action
Construct
Event Manager
Get info from Event
User Event Action

```
void CYGNORunAction::Book()
{
  // Get/create analysis manager
  G4AnalysisManager* man = G4AnalysisManager::Instance();
  man->SetDefaultFileType("root");

  man->SetNtupleMerging(true);

  man->SetFirstNtupleId(1);


  // ---- primary ntuple ------
  // id==1
  man->CreateNtuple("tree1", "Particle Source Info");
  man->CreateNtupleDColumn("Energy");
  man->CreateNtupleDColumn("xpos_vertex");
  man->CreateNtupleDColumn("ypos_vertex");
  man->CreateNtupleDColumn("zpos_vertex");
  man->FinishNtuple();

  // ---- secondary ntuple ------
  //id==2
  man->CreateNtuple("nTuple", "Hits Info");
  man->CreateNtupleIColumn("eventnumber");
  man->CreateNtupleIColumn("numhits");
  man->CreateNtupleDColumn("ekin_particle");
  man->CreateNtupleIColumn("particle_type");
  man->CreateNtupleDColumn("energyDep");
  man->CreateNtupleDColumn("energyDep_NR");
  man->CreateNtupleIColumn("pdgID_hits", fEventAction -> Get_pdgID_hits());
  man->CreateNtupleDColumn("tracklen_hits", fEventAction -> Get_tracklen_hits());
  man->CreateNtupleDColumn("px_particle", fEventAction -> Get_px_particle());
  man->CreateNtupleDColumn("py_particle", fEventAction -> Get_py_particle());
  man->CreateNtupleDColumn("pz_particle", fEventAction -> Get_pz_particle());
  man->CreateNtupleDColumn("energyDep_hits", fEventAction -> Get_energyDep_hits());
  man->CreateNtupleDColumn("energyDep_ion_hits", fEventAction -> Get_energyDep_ion_hits());
  man->CreateNtupleDColumn("x_hits", fEventAction -> Get_x_hits());
  man->CreateNtupleDColumn("y_hits", fEventAction -> Get_y_hits());
  man->CreateNtupleDColumn("z_hits", fEventAction -> Get_z_hits());
  man->FinishNtuple();

  // Open an output file
  man->OpenFile(FileName);

  return;
}
```

ADD A FUNCTION TO CREATE NTUPLES TO STORE DATA


Sensitive Detector
Get info from Hits in SD
Input
Physics List
Detector Construction
Primary generator
Input
Input
Input
Run Manager
Get info from Run
User Run Action
Construct
Event Manager
Get info from Event
User Event Action




```
if (CYGNO_hits > 0) {

      man->FillNtupleIColumn(2,0,event_id);
      man->FillNtupleIColumn(2,1,CYGNO_hits);
      man->FillNtupleDColumn(2,2,(*CYGNOHC)[0]->GetKineticEne());
      man->FillNtupleIColumn(2,3,(*CYGNOHC)[0]->GetParticleID());
      G4cout << "firstParticleE =" << (*CYGNOHC)[0]->GetKineticEne() << "keV" << G4endl;

      for (G4int i=0; i<CYGNO_hits; i++) {

        hitEnergy         = (*CYGNOHC)[i]->GetEdep();
        hitEnergy_ion     = (*CYGNOHC)[i]->GetEdep_ion();
        totEnergy        += hitEnergy;

        v_pdgID_hits.push_back((*CYGNOHC)[i]->GetParticleID());
        v_tracklen_hits.push_back((*CYGNOHC)[i]->GetLength());
        v_px_particle.push_back((*CYGNOHC)[i]->GetMom().x());
        v_py_particle.push_back((*CYGNOHC)[i]->GetMom().y());
        v_pz_particle.push_back((*CYGNOHC)[i]->GetMom().z());
        v_energyDep_hits.push_back(hitEnergy);
        v_energyDep_ion_hits.push_back(hitEnergy_ion);
        v_x_hits.push_back((*CYGNOHC)[i]->GetPos().x());
        v_y_hits.push_back((*CYGNOHC)[i]->GetPos().y());
        v_z_hits.push_back((*CYGNOHC)[i]->GetPos().z());

        if ((int)(*CYGNOHC)[i]->GetParticleID()>1000000000){
          totEnergyNR += hitEnergy;
        }
      }
      man->FillNtupleDColumn(2,4,totEnergy);
      man->FillNtupleDColumn(2,5,totEnergyNR);
      man->AddNtupleRow(2);
    }
```

FILL NTUPLES WITH DATA

CYGNO'S

MONTE CARLO

GEANT4
A SIMULATION TOOLKIT

CADMesh

```
char namestl[50];
sprintf(namestl,"%s/LIMEbody-ShortCone.stl",CYGNOGeomPath.c_str());
G4cout << namestl << G4endl;
ifstream infile(CYGNOGeomPath.c_str());
if (infile.good())
    mesh_LIMEDetectorBody = new CADMesh(namestl);
sprintf(namestl,"%s/LIMEinternalStructure.stl",CYGNOGeomPath.c_str());
G4cout << namestl << G4endl;
```

BEFORE

NO NEED TO INSTALL CADMESH ANYMORE!

```
char namestl[70];

    snprintf(namestl, sizeof(namestl), "%s/LIMEbody-ShortCone.stl", CYGNOGeomPath.c_str());
    G4cout << namestl << G4endl;
    ifstream infile(CYGNOGeomPath.c_str());
    if (infile.good())
      mesh_LIMEDetectorBody = CADMesh::TessellatedMesh::FromSTL(namestl);
```

AFTER