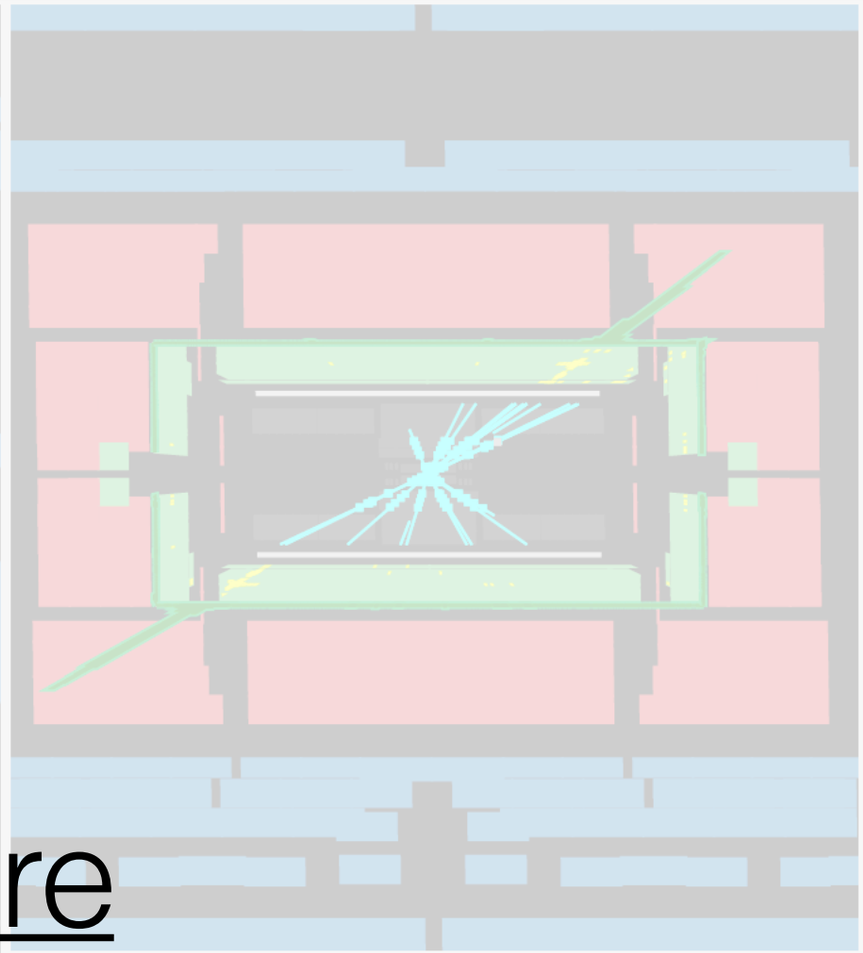
egamma software

Jovan Mitrevski
Ludovica Aperio Bella
Roger Naranjo



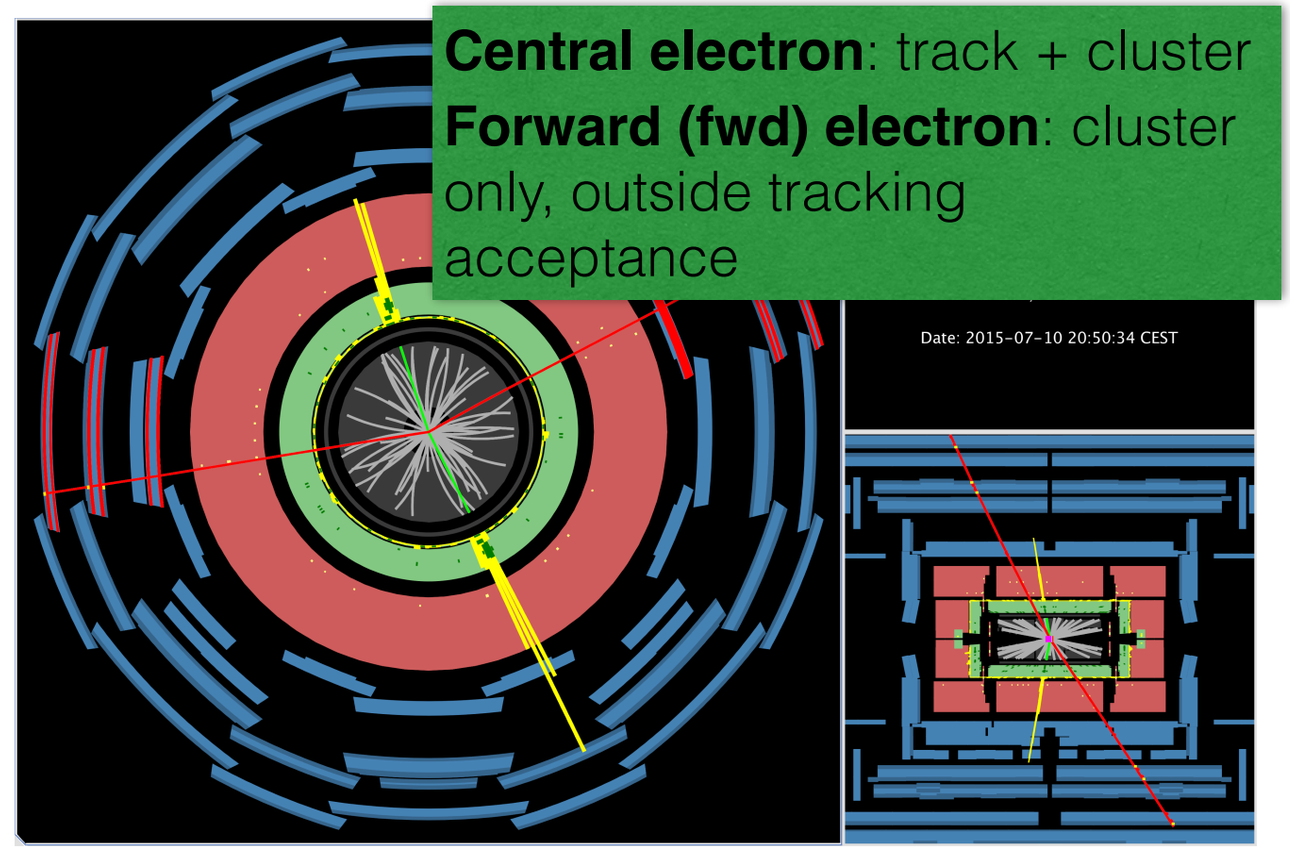
Run Number: 166786, Event Number: 77246540

Date: 2010-10-14 10:16:28 CEST

ATLAS Software Tutorial

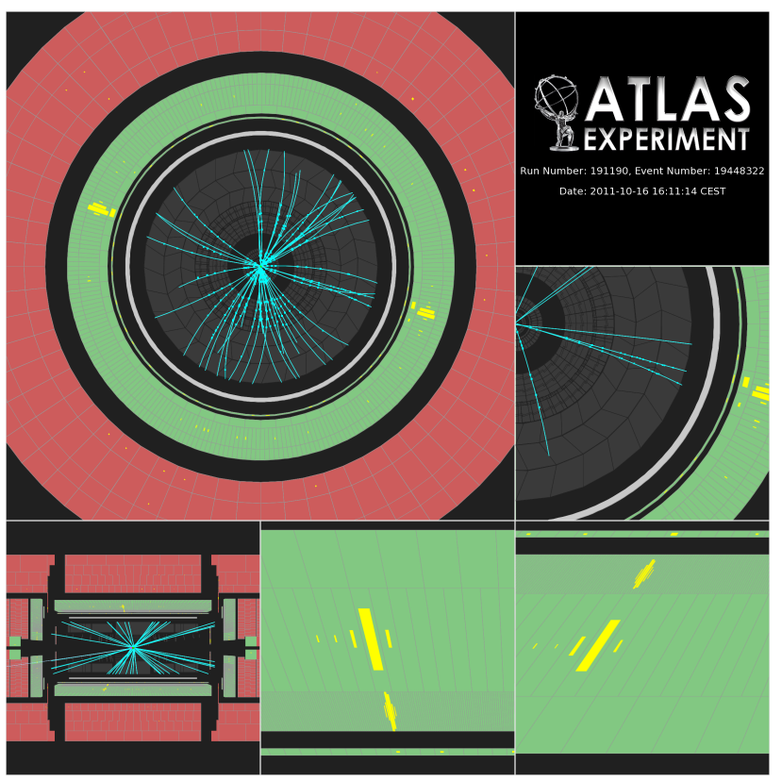
eγ reconstruction

- Electrons and photons are built from:
- **EM-Clusters**
- **Track particles**
- **(Conversion) Vertices**
  - And come in a few “flavours”



Central electron: track + cluster

Forward (fwd) electron: cluster only, outside tracking acceptance

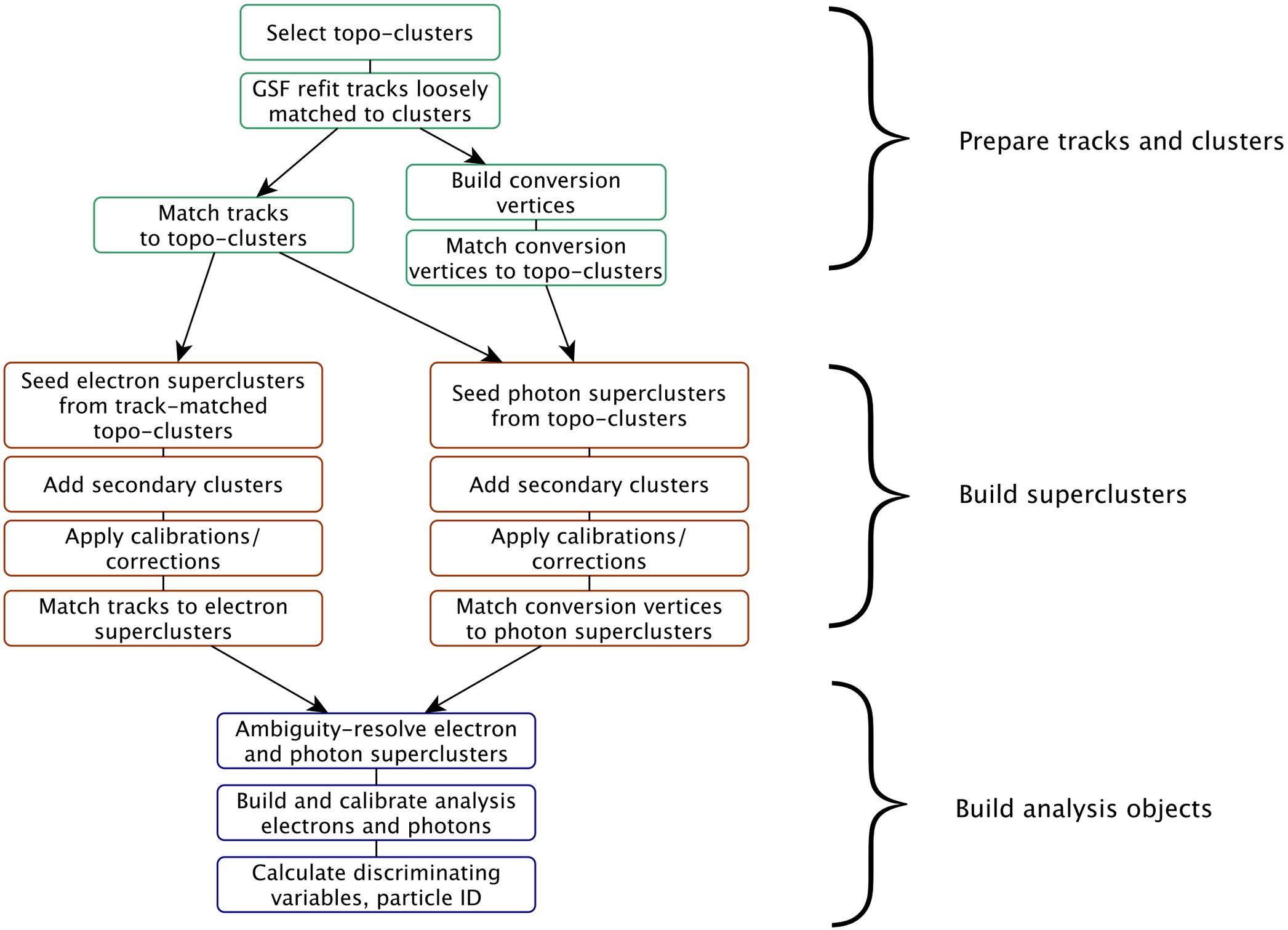


Converted photon: cluster + conversion vertex (ideally 2-tracks but also 1-track only)

Unconverted photon: cluster only

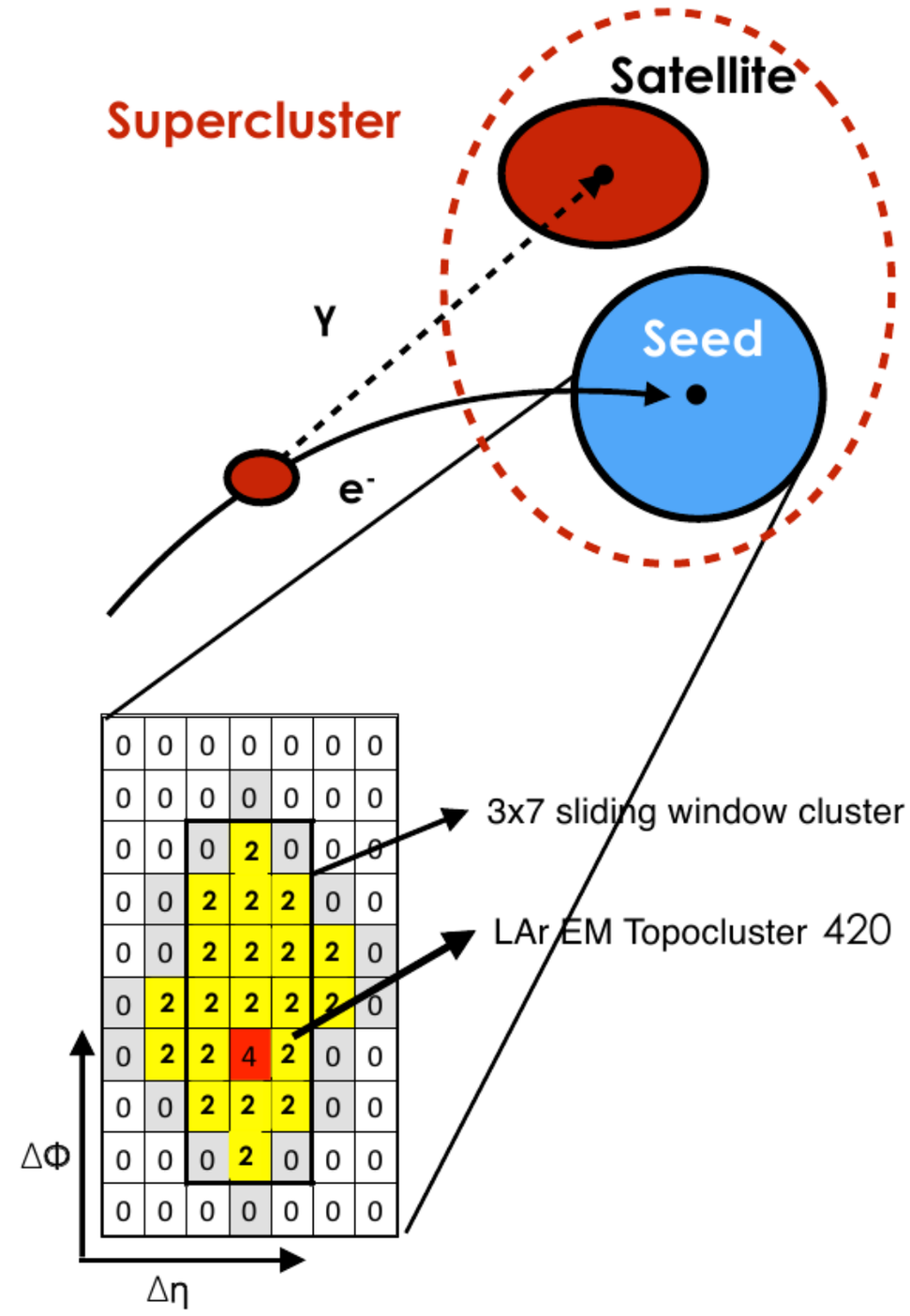
- unconverted = 0 : unconverted photon
- singleSi = 1 : one track only, with Si hits
- singleTRT = 2 : one track only, no Si hits (TRT only)
- doubleSi = 3 : two tracks, both with Si hits
- doubleTRT = 4 : two tracks, none with Si hits (TRT only)
- doubleSiTRT = 5 : two tracks, only one with Si hits

eγ reconstruction work-flow in a nutshell



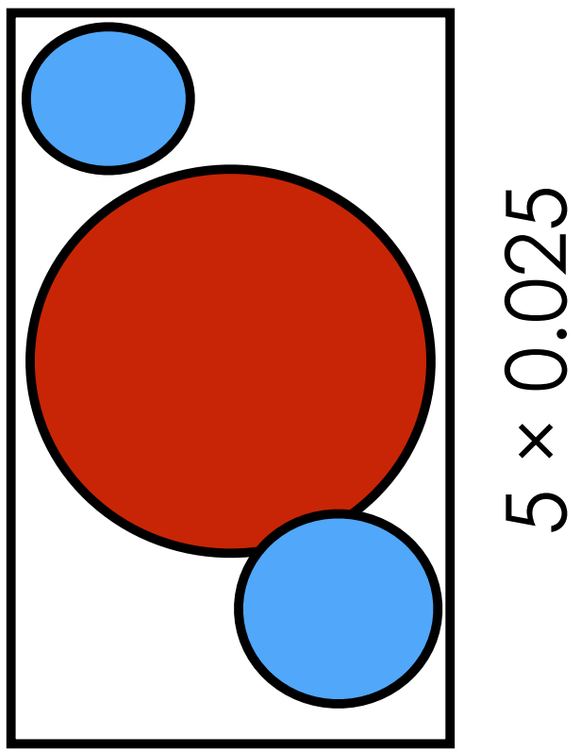
The new Super-cluster algorithm

- **Main goal:** Include radiative losses from bremsstrahlung into cluster to improve energy measurement
- "**Superclusters**" built from
  - Primary ("**seed**") electron cluster
  - Nearby, secondary ("**satellite**") clusters
- **Cluster reconstruction uses 420 topoclusters**
- Better reconstruction of lower E_T clusters useful for reconstructing clusters from soft bremsstrahlung photons
- Two-step procedure:
  - Primary topocluster
  - Search for satellite clusters



New Clustering algorithm

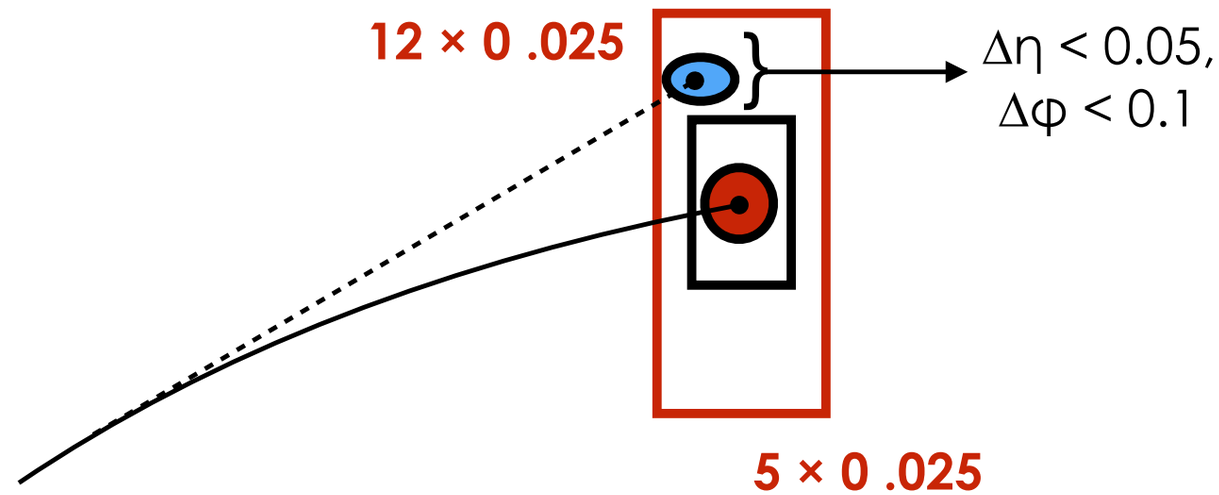
1. Add all clusters within 3 × 5 window around seed cluster.



($e^{\pm}$, all γ)

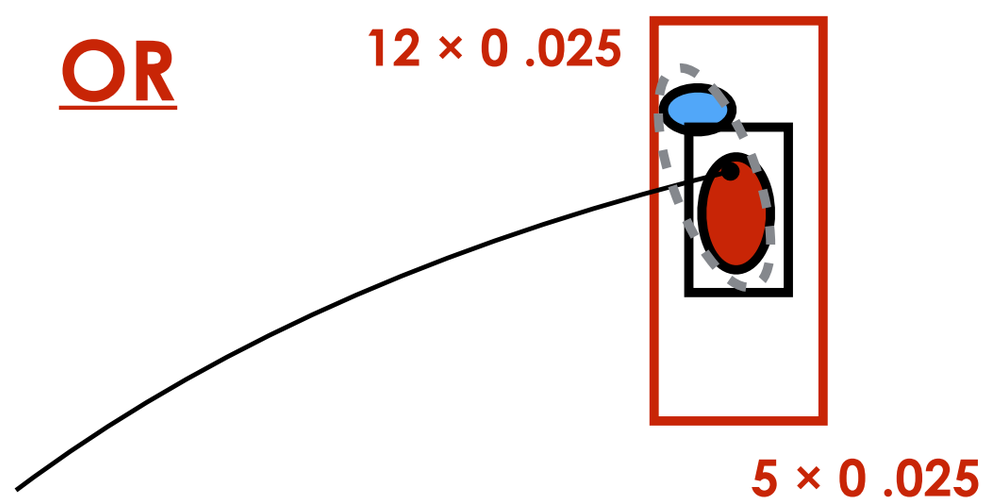
Electrons only:

2. For clusters with Δη, Δφ **outside 3 × 5 window**, but **within 5 × 12 window**, add to seed cluster if:



Δη < 0.05, **Δφ < 0.1** between **extrapolation from perigee** and **secondary cluster**.

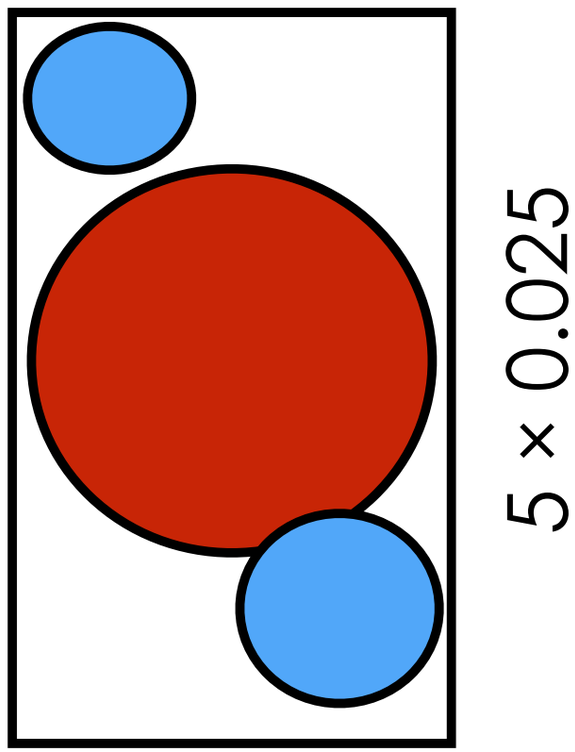
OR



Seed, secondary cluster **match the same track**.

New Clustering algorithm II

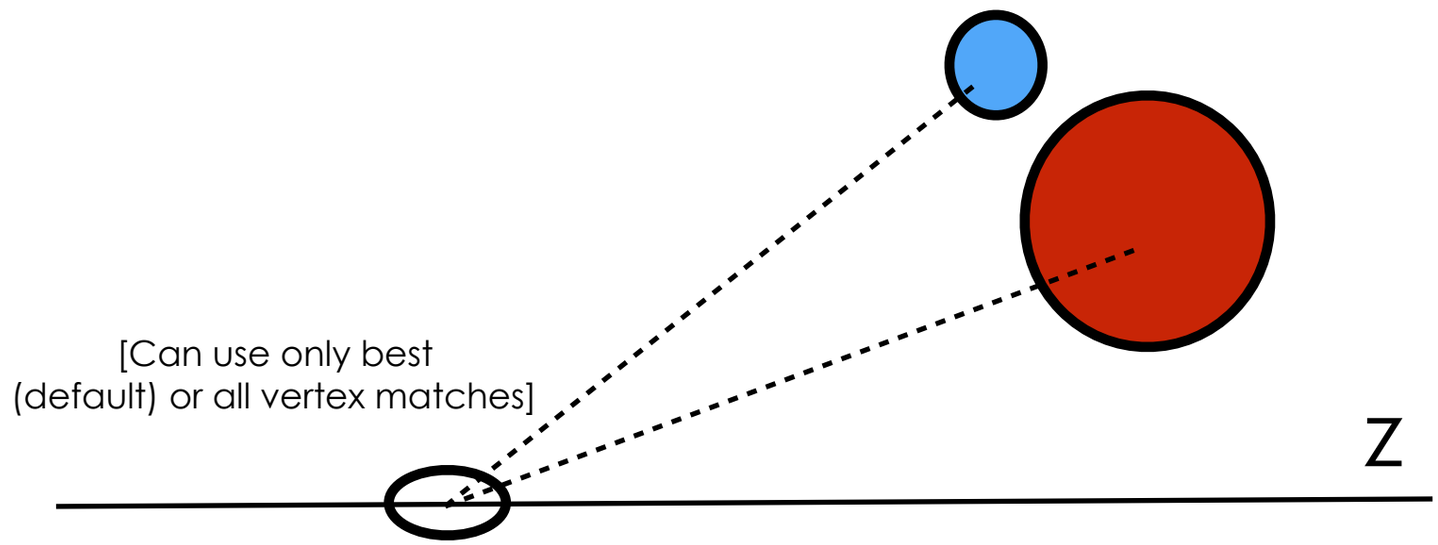
1. Add all clusters within 3 × 5 window around seed cluster.



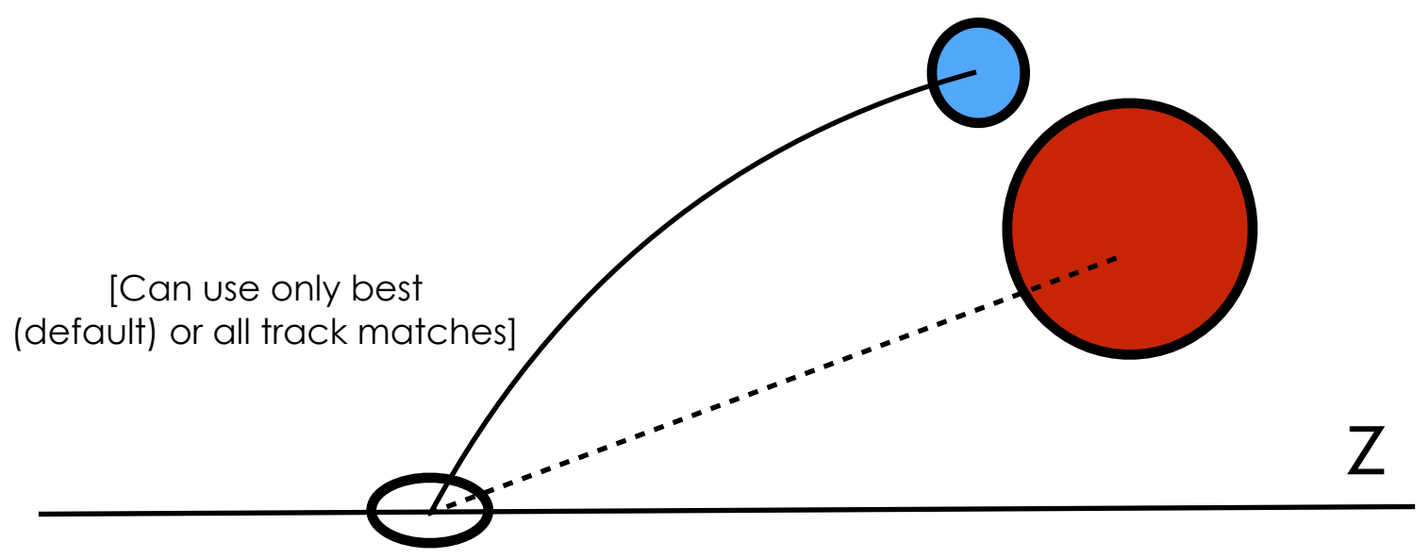
($e^{\pm}$, all γ)

Conversions only:

2. Add topoclusters that have the **same conversion vertex** matched as the seed cluster.

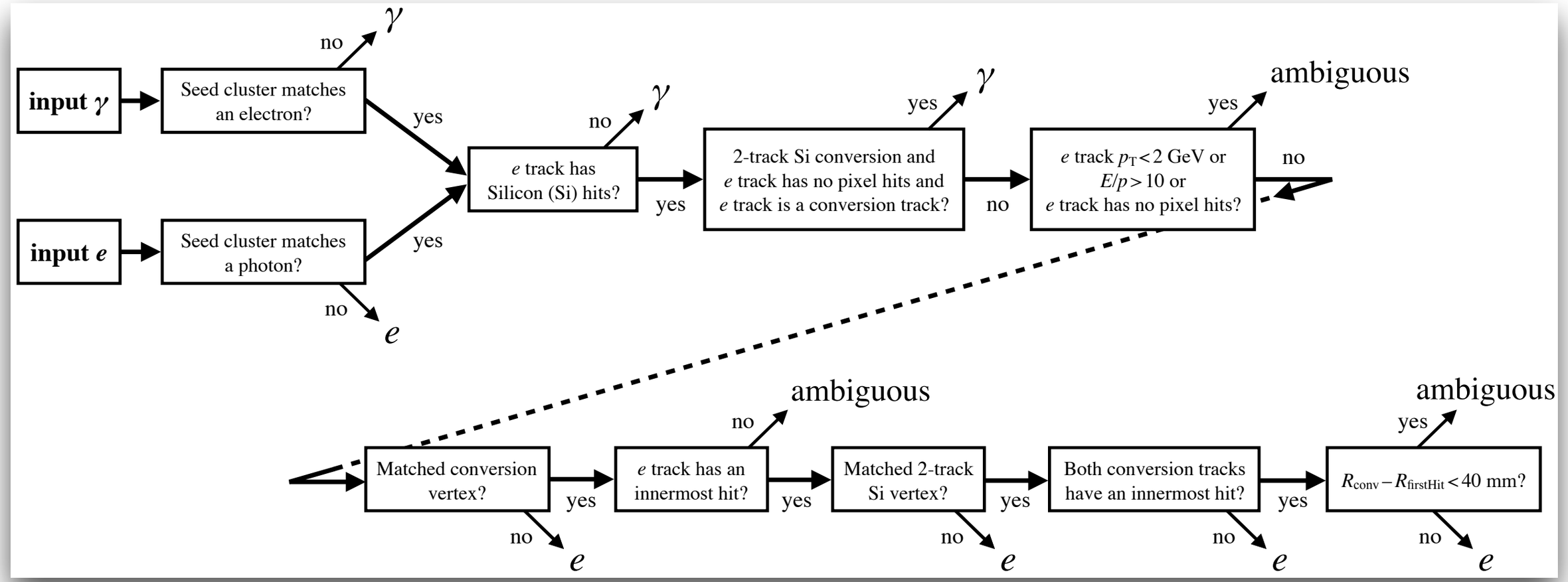


3. Add topoclusters that **have a track match** that is a **part of the conversion vertex** matched in the seed cluster.



eγ reconstruction work-flow: Built analysis object

```
ambiguity chain to choose if the candidate is an electron or a photon
```

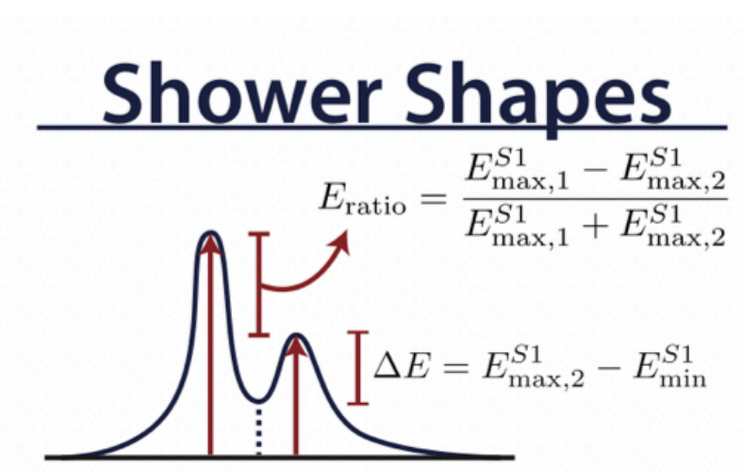


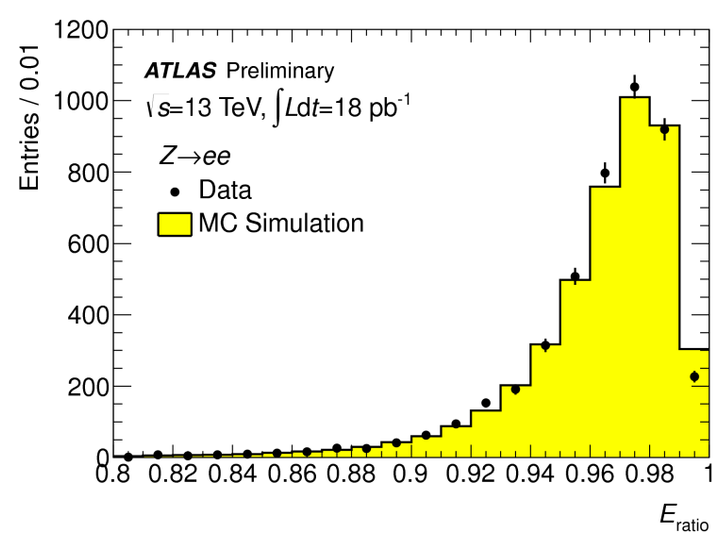
https://twiki.cern.ch/twiki/bin/view/AtlasProtected/EGammaSoftware

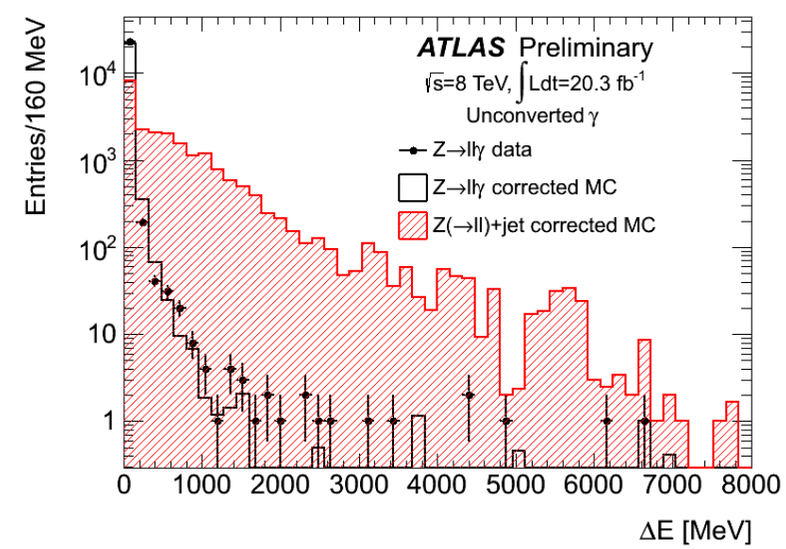
eγ reconstruction work-flow III

4. Create and decorate electrons and photons.

- Apply MVA calibration
- Fill IParticle class:
  - Energy/Momentum : energy from cluster
  - η,ϕ form tracks or cluster (for unconverted photon and TRT conversion)
- Decorate with object quality, shower-shape and identification criteria







eγ reconstruction work-flow IV

5. Truth association: links from reco to truth and vice-versa, true type and origin

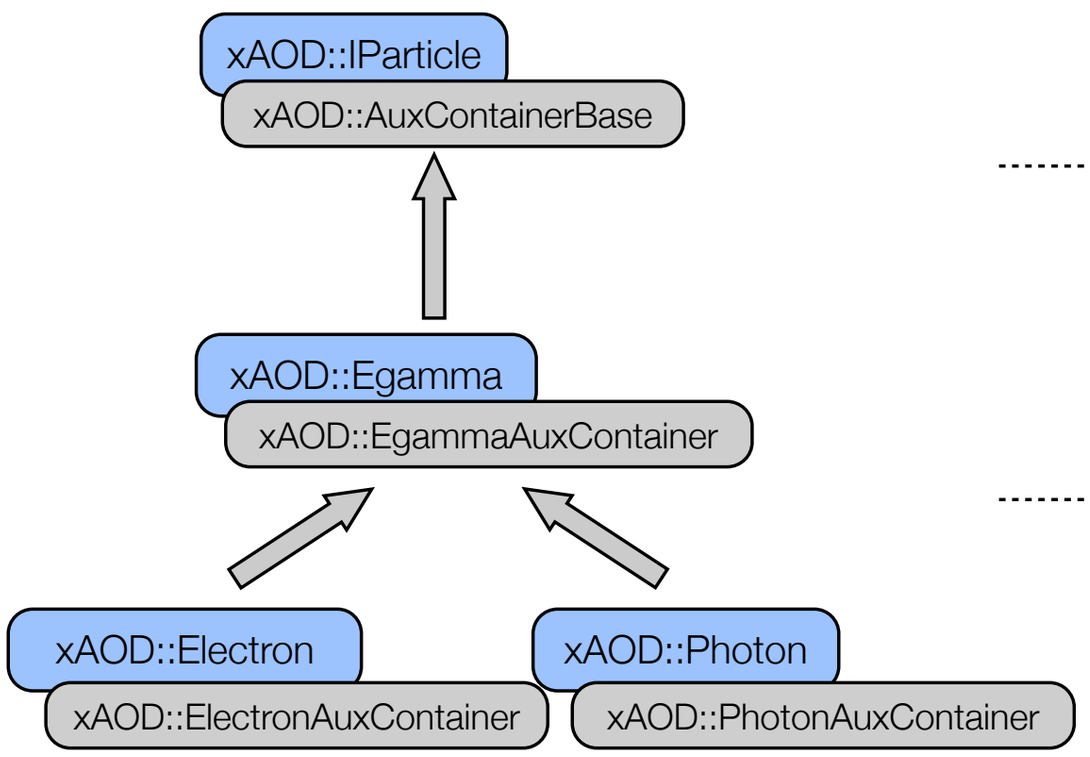
- Electrons: truth matching based on track hits
- Photons: ΔR matching at calorimeter (neutral extrap. from truth particles)
- *egammaTruthParticles* collection: copy of prompt truth electrons and photons
  - e.g. 2 photons for H →γγ (except for H → γff)
  - Links to reconstructed objects and original truth objects (in “big collection”)
  - More info can be added (e.g. truth isolation)

https://twiki.cern.ch/twiki/bin/view/AtlasProtected/EGammaTruthRun2

6. Electrons and photons decorated with isolation information

https://svnweb.cern.ch/trac/atlasoff/browser/Event/xAOD/xAODEgamma

eγ xAOD Event Data Model (EDM)



- egamma objects (IParticle class) described by 3 main classes:
  - xAOD::Egamma → base class (abstract)
  - xAOD::Electron
  - xAOD::Photon
- And stored in the following containers (+auxiliary containers):
  - Electrons (+ElectronsAux)
  - ForwardElectrons (+ ForwardElectrons Aux)
  - Photons (+PhotonsAux)

data organisation in xAOD

- All "objects", e.g. electrons, clusters, tracks, are stored in containers:
  - ElectronContainer, CaloClusterContainer, TrackParticleContainer
    - This is how things are stored in Athena, and so they will look the same now in ROOT
- The information for an electron is now split across these three containers:
  - Each electron has a "link" to its cluster and track
  - An electron holds its 4-mom, shower-shape variables, isolation variables, …
  - A cluster has a 4-mom, sampling and moment details, and links to cells, …
  - A track has a 4-mom, hits details, covariance matrix, …
- For Photons, there is a "VertexContainer" instead of the TrackParticle container

Helper functions (xAODEgamma)

- EgammaxAODHelpers.h
  - isElectron, isPhoton, isConvertedPhoton
- ElectronxAODHelpers.h
  - getOriginalTrackParticle (before GSF)
  - numberOfSiHits
- PhotonxAODHelpers.h
  - conversionType, conversionRadius, ...
- EgammaTruthxAODHelpers.h
  - getTruthParticle, getRecoElectron, getRecoPhoton (for egammaTruthParticles) , getBkgElectronMother,
  - isTrueConvertedPhoton

- We have several helpers to access relevant information for electrons and photons
- Do you want to use them?
  - Just include the headers into your code (AnalysisBase) and the xAODEgamma in your cmake file

xAOD::TruthHelpers Namespace Reference

Functions

const **xAOD::TruthParticle** *	**getTruthParticle** (const **xAOD::IParticle** &p) Return the truthParticle associated to the given **IParticle** (if any) More...
int	**getParticleTruthType** (const **xAOD::IParticle** &p)
int	**getParticleTruthOrigin** (const **xAOD::IParticle** &p)

egamma CP Tool

How-to for Run2 analyses

- Electron and photon selection and identification
- Electron efficiencies (scale factors and uncertainties)
- Photon efficiencies (scale factors and uncertainties)
- Electron and photon calibration (energy scale and resolution)
- Electron and photon four-momentum
- Electron and photon truth in xAOD
- xAOD Egamma Variables (translation to D3PD where applicable)

Documentation:
How to access e/γ related variables via the xAOD interface: EgammaD3PDtoxAOD
Electron and Photon Truth Information for Run2: EGammaTruthRun2
Electron and Photon Selection and Identification for Run2: EGammaIdentificationRun2
Summary of Recommendations: EGammaRecommendationsR21

- **Most of the information is stored in xAOD or recalculated using Derivation.**
  - User might need to (re-)compute variables, scale factors and uncertainties:
- CP-tools providing recommendation for AnalyisBase/(Ath)Athena and guidance on how use electron and photon object in analysis (twiki)

https://twiki.cern.ch/twiki/bin/viewauth/AtlasProtected/EGammaIdentificationRun2

How to select Electrons and Photons

E/gamma twiki pages: guidance on how use electron and photon object in analysis:
Few examples:

- Fiducial region and Calorimeter crack definition
- Electron and photon authors and types
  - Using the Ambiguity tool in analysis
- Object quality cut
  - Photon cleaning
- Basic kinematic selection of electrons
- Electron and photon identification
  - traking recomandation on lepton-vertex association cuts

Calorimeter Crack definition

```
if  ( fabs( eg->caloCluster()->etaBE(2) ) >1.37
      &&  fabs( eg->caloCluster()->etaBE(2) ) <1.52) {
       // This falls in the calorimeter crack
      //Do this if you want to exclude the crack region in your analysis
      //This is analysis dependent !
 }
```

Photon conversion type / flags

- unconverted = 0 : unconverted photon
- singleSi = 1 : one track only, with Si hits
- singleTRT = 2 : one track only, no Si hits (TRT only)
- doubleSi = 3 : two tracks, both with Si hits
- doubleTRT = 4 : two tracks, none with Si hits (TRT only)
- doubleSiTRT = 5 : two tracks, only one with Si hits

https://twiki.cern.ch/twiki/bin/view/AtlasProtected/EGammaIdentificationRun2#Using_the_Ambiguity_tool_in_anal

Using the Ambiguity tool in analysis

Using the Ambiguity tool in analysis (Electron --> Photon fakes)

```
// in the header file of MyClass
class IEGammaAmbiguityTool;

// ...and then
private:
  IEGammaAmbiguityTool *m_egammaAmbiguityTool;

// in the .cxx of MyClass
#include "ElectronPhotonSelectorTools/EGammaAmbiguityTool.h";

MyClass::MyClass() : m_egammaAmbiguityTool(0);

//add to StatusCode MyClass::initialize()
 m_egammaAmbiguityTool = new EGammaAmbiguityTool("EGammaAmbiguityTool");
 ATH_CHECK( m_egammaAmbiguityTool->initialize() );

// add to StatusCode MyClass::execute () (in the event loop, call for each photon)
 if (!m_egammaAmbiguityTool->accept( photon ) ) // skip this photon
```

```
namespace xAOD {
  namespace AmbiguityTool {
    /// @name Ambiguity types
    enum AmbiguityType {
      electron = 0,
      ambiguousTrackQoverPBetterThanVertexQoverP=1,
      ambiguousTrackLowPt=2,
      ambiguousTrackLowQoverP=3,
      ambiguousTrackNoPixel =4,
      ambiguousVertexQoverPBetterThanTrackQoverP=5,
      photon=6,
      unknown=7
    };
  }
}
```

type to classify why the egamma object is classified as ambiguous.

```
// add to StatusCode MyClass::execute () (in the event loop, call for each photon or electron. Remember, this is *NOT* the default or recommended usage)
 if (!m_egammaAmbiguityTool->accept( electron, false ) ) // skip this electron
```

https://twiki.cern.ch/twiki/bin/viewauth/AtlasProtected/EGammaIdentificationRun2#ID

Electron and photon identification

a) Getting the e/γ ID output from the derivation

In the derivations, electron and photon ID is rerun using the tag of ElectronPhotonSelectorTools in the cache used to produce the derivation, and electron and photons are decorated with the output (true/false) of the selector and (for cut-based selection) its integer word (isEM). The recomputed quantities are called:

- `DFCommonElectronsLHLoose`, `DFCommonElectronsLHMedium`, `DFCommonElectronsLHTight`, (true/false for electron likelihood loose, medium, tight)
- `DFCommonElectronsIsEMLoose`, `DFCommonElectronsIsEMMedium`, `DFCommonElectronsIsEMTight`, (true/false for electron cut-based loose, medium, tight)
- `DFCommonPhotonsIsEMLoose`, `DFCommonPhotonsIsEMTight` (true/false for photon cut-based loose/tight).

These are included in the smart slimming output of electrons and photons.

b) Getting the ID output running the Tool :

```
asg::AnaToolHandle<IAsgElectronLikelihoodTool> electronMediumLHSelector ("AsgElectronLikelihoodTool/electronMediumLHSelector");
ANA_CHECK(electronMediumLHSelector.setProperty("WorkingPoint", "MediumLHElectron"));
ANA_CHECK(electronMediumLHSelector.setProperty("OutputLevel", mylevel));
ANA_CHECK(electronMediumLHSelector.initialize());

for (const xAOD::Electron* el : *electrons) {
ANA_MSG_INFO("--------------------------");
ANA_MSG_INFO("Electron: " << counter);
ANA_MSG_INFO("Electron LH Medium accept result: " <<electronMediumLHSelector->accept(el));
```

Available working points:

- `Electron Likelihood ID: VeryLooseLHElectron, LooseLHElectron, LooseBLLHElectron, MediumLHElectron, TightLHElectron`
- `Forward electrons: LooseForwardElectron, MediumForwardElectron, TightForwardElectron`
- `photon ID: Tight, Loose`

https://twiki.cern.ch/twiki/bin/viewauth/AtlasProtected/EGammaIdentificationRun2#Electron_d0_and_z0_cut_definitio

Lepton-vertex association cuts

- **d0 and z0 definition**: https://twiki.cern.ch/twiki/bin/view/AtlasProtected/InDetTrackingDC14#Impact_parameters_z0_d0_definiti

- $|d_0^{BL}$ significance $| < 5$
- $|\Delta z_0^{BL} \sin\theta| < 0.5$ mm

- A helper function to calculate the **transverse impact parameter significance** corrected by the beam-spot spread.

- The usage of **delta_z0 * sin(trk.theta())** instead of delta_z0, can be a better choice to avoid rejecting tracks with an expected larger error in the forward region.

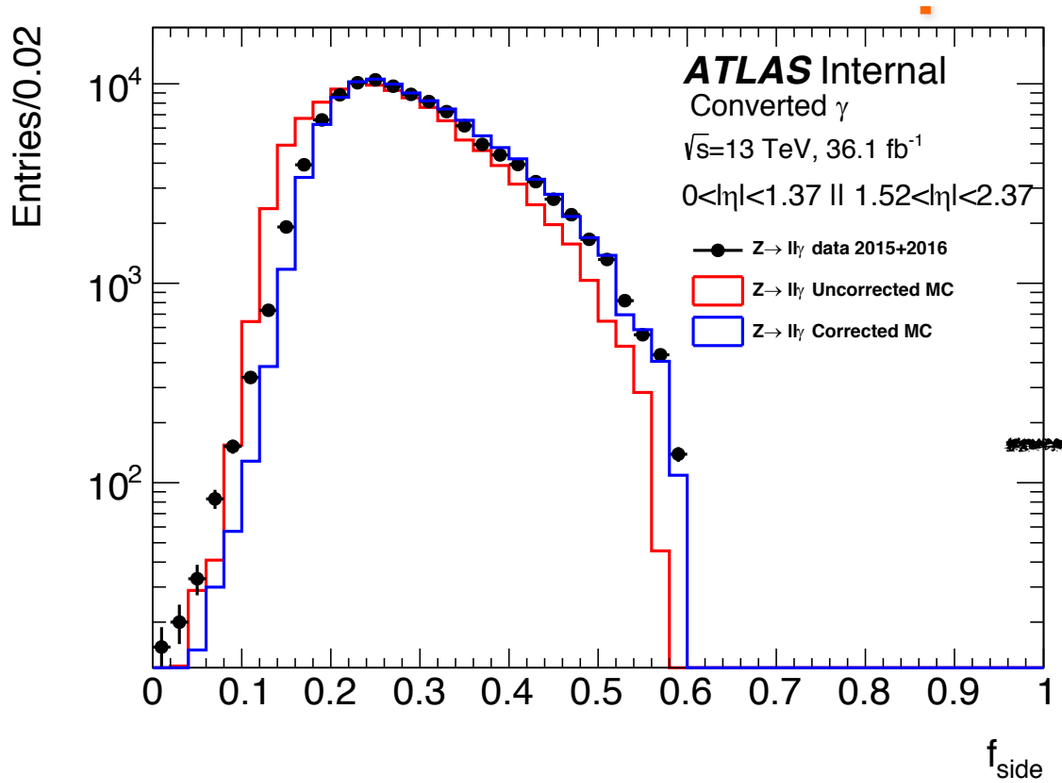
```
#include "xAODTracking/TrackParticlexAODHelpers.h"
xAOD::EventInfo *evt = ...; // the EventInfo xAOD object
xAOD::TrackParticle *tp = el->trackParticle() ; //your input track particle from the electron
double d0sig = xAOD::TrackingHelpers::d0significance( tp, evt->beamPosSigmaX(), evt->beamPosSigmaY(), evt->beamPosSigmaXY() );
```

```
//loop over vertices and look for good primary vertex
for (xAOD::VertexContainer::const_iterator vxIter = vxContainer->begin(); vxIter != vxContainer->end(); ++vxIter) {
   // Select good primary vertex
     if ((*vxIter)->vertexType() == xAOD::VxType::PriVtx) {
     //This is the primary vertex
   }
}
```

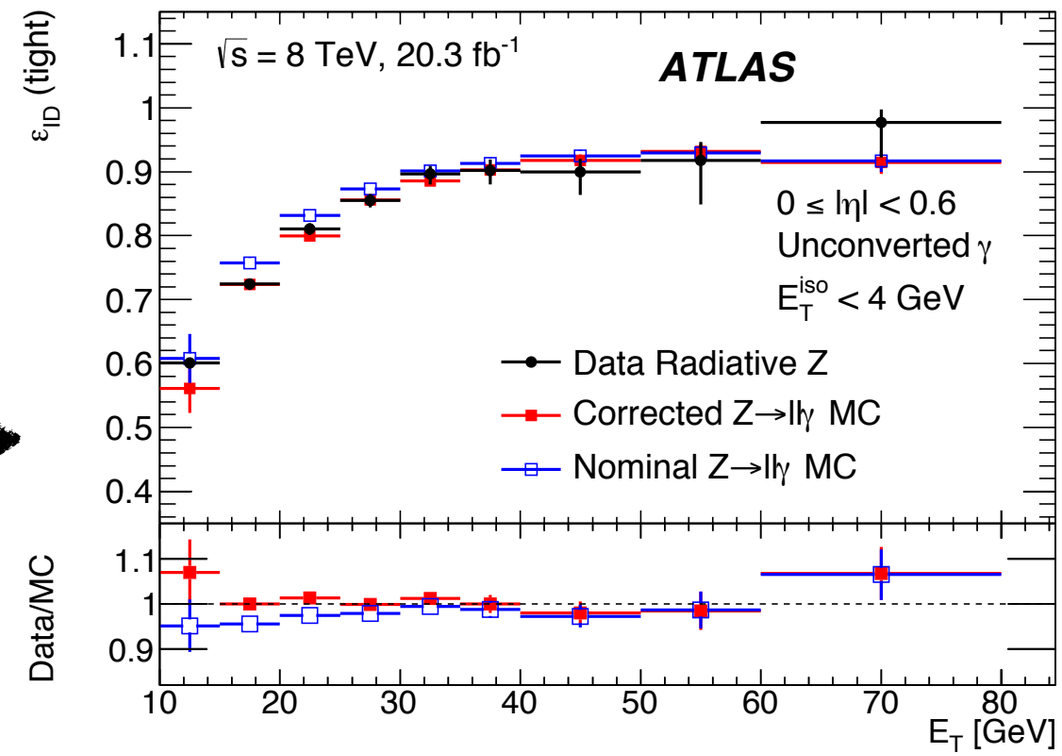
```
//Given a track particle and a vertex
xAOD::TrackParticle trk
xAOD::Vertex vtx;
float delta_z0 = fabs(trk.z0() + trk.vz() - vtx->z());
```

https://twiki.cern.ch/twiki/bin/view/AtlasProtected/EGammaIdentificationRun2#Shifting_the_photon_shower_shape

Fudge factor



effect on data/MC Eff




```
#include <EgammaAnalysisInterfaces/IElectronPhotonShowerShapeFudgeTool.h>

//create the selector tool instance
asg::AnaToolHandle<IElectronPhotonShowerShapeFudgeTool> m_fudgeMCTool;

 // Initialize Fudge Tool
    ATH_CHECK (ASG_MAKE_ANA_TOOL (m_fudgeMCTool, ElectronPhotonShowerShapeFudgeTool));
    ATH_CHECK (m_fudgeMCTool.setProperty ("Preselection", 22)); // Rel 21
    ATH_CHECK (m_fudgeMCTool.setProperty("FFCalibFile", ElectronPhotonShowerShapeFudgeTool/
v2/PhotonFudgeFactors.root"); // Rel 21
    ATH_CHECK (m_fudgeMCTool.initialize());

// shifted shower shapes replace those of the initial photon object, which can then be passed
to the photon selector.
m_fudgeMCTool->applyCorrection(photon)
```

https://twiki.cern.ch/twiki/bin/view/AtlasProtected/XAODElectronEfficiencyCorrectionTool

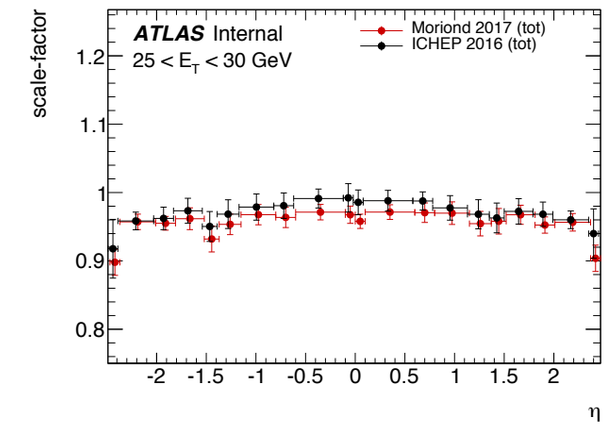
eγ SF

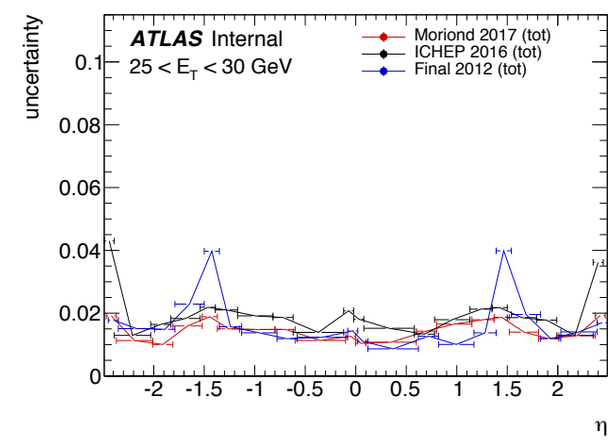
Electron efficiencies can be decomposed as

$$\epsilon_{\text{total}} = \epsilon_{\text{reconstruction}} \times \epsilon_{\text{identification}} \times \epsilon_{\textbf{iso}} \times \epsilon_{\textbf{trigger}}$$
$$= \frac{N_{\text{reconstruction}}}{N_{\text{clusters}}} \times \frac{N_{\text{identification}}}{N_{\text{reconstruction}}} \times \frac{N_{\textbf{iso}}}{N_{\text{identification}}} \times \frac{N_{\textbf{trigger}}}{N_{\textbf{iso}}}$$

Egamma group provides scale factors and uncertainties:

SF = eff(data)/eff(MC)






```
asg::AnaToolHandle<IAsgElectronEfficiencyCorrectionTool>  tool ("AsgElectronEfficiencyCorrectionTool/ElectronEffCorrection");
ANA_CHECK(tool.setProperty("CorrelationModel", "FULL" ) &&
        tool.setProperty("ForceDataType",1) &&
        tool.setProperty("IdKey", "Medium") &&
        tool.retrieve());
```

User need to specify the combination of ID + Trigger + Isolation WP to get the corresponding SF using file name using Keys:

```
//set the keys of interest for correction files
electronSF.setProperty("RecoKey", <Reconstruction Key>);
electronSF.setProperty("IdKey", <Identification Key>);
electronSF.setProperty("IsoKey", <Isolation Key>);
electronSF.setProperty("TriggerKey", <Trigger Key>);
```

https://twiki.cern.ch/twiki/bin/viewauth/AtlasProtected/ElectronPhotonFourMomentumCorrection

electron and photon calibration

Create the tool and define the correlation model for the uncertainty:

```
// Create the tool
std::unique_ptr<CP::IEgammaCalibrationAndSmearingTool> tool(new CP::EgammaCalibrationAndSmearingTool("EgammaCalibrationAndSmearingTool"));
asg::setProperty(tool.get(), "ESModel", "es2017_R21_PRE");
asg::setProperty(tool.get(), "randomRunNumber", 123456);
RETURN_CHECK(APP_NAME, tool->initialize());


//===========SYSTEMATICS
const CP::SystematicRegistry& registry = CP::SystematicRegistry::getInstance();
const CP::SystematicSet& recommendedSystematics = registry.recommendedSystematics();
std::vector<CP::SystematicSet> sysList;

std::cout << "SIZE of the systematics set:" << recommendedSystematics.size() << std::endl;

for (auto sys : recommendedSystematics) { sysList.push_back(CP::SystematicSet({sys})); }
```

check the latest recommendation

February 2018 release of recommendations for release 21

ES model : **es2017_R21_v0**

```
asg::setProperty(my_tool, "ESModel", "es2017_R21_v0")
```

Supported uncertainty schemes:

- **FULL_v1** 69 scale uncertainties + 9 resolution uncertainties
- **1NP_v1** 2 scale uncertainties, only one is non zero for full simulation, 1 resolution uncertainty)

Calibrate the shallow-copy :

```
//Iterate over the shallow copy
for (xAOD::Photon* ph : *photons_shallowCopy.first) {

CHECK(tool->applyCorrection(*ph));
std::cout << "Calibrated ph = " << ph->e() << std::endl;
```

Isolation software

- Common dual-use Tool to compute Isolation variable for: electron photon and muons
  - **IsolationSelectionTool**: common isolation working points are provided by the isolation forum for electrons, photons and muons. They can be accessed through the IsolationSelectionTool inside the IsolationSelection package.
  - There are two types of working points:
    - simple cuts of the type "topoetcone20 < XX GeV" or "topoetcone20/pT< YY"
    - targeted efficiencies (for leptons only) of the type "flat efficiency of 99% in the eta-pT plane", or "gradient efficiency of 95% at pT=20 GeV and 99% at pT=80 GeV":
  - Other feature: Leakage correction for egamma calorimetric isolation, Overlap removal for isolation

Pass Isolation wp selection:

```
const xAOD::PhotonContainer* photons(nullptr);
CHECK( event.retrieve(photons,m_sgKeyPhotons) );
for (auto x : *photons) {
  if (x->pt() > 7000.) {
    if (x->caloCluster() != nullptr) {
      if (fabs(x->caloCluster()->eta()) < 2.47) {
        if (iso_1.accept( *x )) {
          Info(APP_NAME," Photon passes Isolation");
        }else{
          Info(APP_NAME," Photon FAILS Isolation");
        }
      }
    }
  }
}
```

Retrieve the variable :

```
const xAOD::Electron *currEl;

currEl->isolation(ptvarcone,xAOD::Iso::ptvarcone20);

currEl->isolation(topoetcone,xAOD::Iso::topoetcone20);
```

https://twiki.cern.ch/twiki/bin/view/AtlasProtected/ElectronGamma

egamma twiki

ATLAS ElectronGamma

SoftwareTutorial, EGammaSoftware, Rel 20.7 recomandation , Rel 21 recomandation, summary talk

Organisation / Activites

- **Conveners**: Christos Anastopoulos, Sandrine Laplace
- Dashboard to follow egamma activities on JIRA

egamma sub-groups

- **Calibration**: Thibault Guillemin, Nansi Andari
- **ElectronID**: Philip Sommer,Sarah Heim
- **Photon ID**: Marcello Fanti, Yanping Huang
- **Reconstruction and software**: Ludovica Aperio Bella, Kerstin Tackmann
- **Isolation and Fake Forum** : Ximo Poveda Torres
- **Egamma-Trigger**: Arantxa Ruiz Martinez, Fernando Monticelli

Other responsibilities

- **MC Production**: José Ocariz
- **Upgrade** : K.Nikolopoulos and A.Bocci
- **Data Quality Monitoring**: Bertrand Laforge
- **Egamma Validation** : Mikhail Levchenko, Aleksei Ezhilov
- **xAOD derivations**: Giovanni Marchiori (requests can be made here)
- **Group disk manager**: Andrea Bocci

Results

- All Egamma Papers Glance entries
- All Egamma Conf Notes Glance entries
- All Egamma Pub Notes Glance entries

Run 2 Analyses

Egamma recommendations

- Summary of recommendations for release 21
- Summary of recommendations for release 20.7
- More details:
  - Electron and photon selection and identification
  - Electron efficiencies (scale factors and uncertainties)
  - Photon efficiencies (scale factors and uncertainties)
  - Electron and photon calibration (energy scale and resolution)
  - Isolation and Fake Forum

How-to access egamma variables

- xAOD Egamma Variables (translation to D3PD where applicable)
- Electron and photon four-momentum
- Electron and photon truth in xAOD

Open Tasks

- **Open tasks needing manpower!**
- Data/MC agreement task force in 2017

e/gamma related analysis code and ntuples

Tag and Probe framework

backup

eγ reconstruction work-flow in a nutshell

1. Select clusters, get GSF track particles

EMBremCollectionBuilder

2. Match clusters with tracks and conv. vertices

EMTrackMatchBuilder / EMVertexBuilder / EMConversionBuilder

3. Ambiguity resolution (electron and/or photon?)

EMAmbiguityTool

4. Create and decorate electrons and photons

- EMClusterTool
- EMFourMomBuilder
- egammaOQFlagsBuilder
- EMShowerBuilder
- EMPIDBuilder

Transient class (egammaRec) containing links to clusters, track particles, conversions and Δη/Δϕ info

xAOD::Electron and xAOD::Photon

5. Set links to truth and create egammaTruthParticles with links to reco objects

egammaTruthAssociationAlg

Prompt truth electrons and photons, D3PD inspired

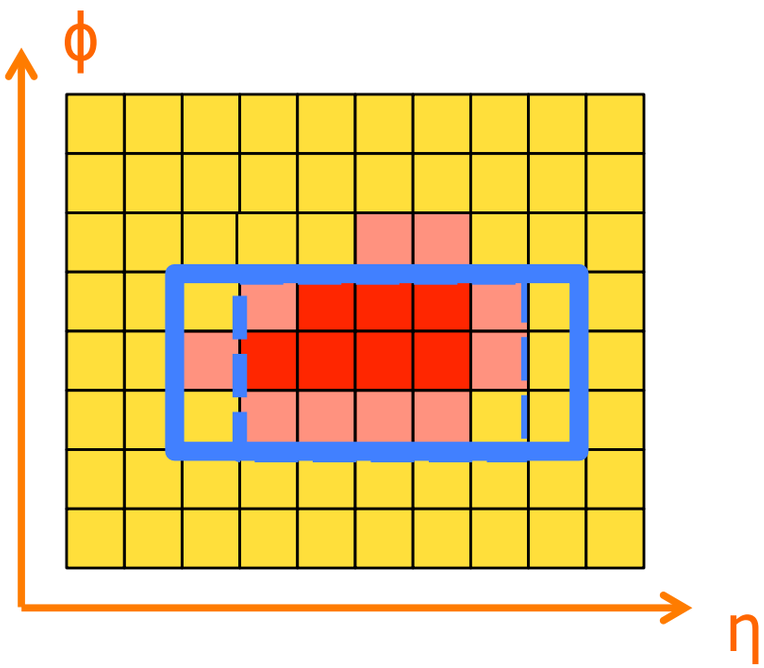
egamma reconstruction: difference Run1 RunII

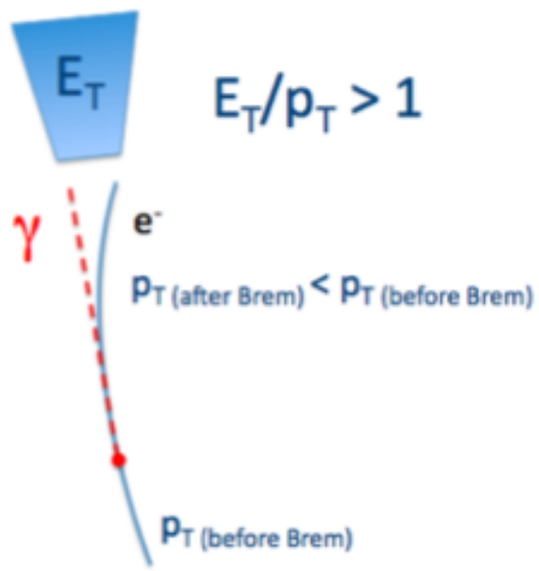
- AuthorUnknown =0x0
- AuthorElectron =0x1 (1): Object Reconstructed by standard cluster-based algorithm. Surely not Photon ==> Electron.
- AuthorSofte =0x2 (2): Electron reconstructed by the track-based algorithm, here for compatibility with old AOD format.
- AuthorPhoton =0x4 (4): Object Reconstructed by standard cluster-based algorithm. Surely not Electron ==> Photon
- AuthorFwdElectron =0x8 (8): Electron reconstructed by the Forward topological-cluster based algorithm
- AuthorAmbiguous =0x10 (16): Object Reconstructed by standard cluster-based algorithm. Ambiguous ==> can be either Electron or Photon. Goes to both collections.
- AuthorTrigElectron =0x20 (32): trigger Electrons
- AuthorTrigPhoton =0x40 (64): trigger Photons
- AuthorCaloTopo35 =0x80 (128): Photon reconstructed by SW CaloTopo35 seeded clusters
- AuthorALL = 0xFFFF : not an actual author, but used as default in some functions

- Soft electrons no longer built
- TRT tracks only used for photon conversions, not electrons
- Straight line extrapolation from conv. vertices to clusters
- No more E-p, η or ϕ combination during reconstruction
  - Energy from cluster, η,ϕ from track (central e), Si+Si vertex or cluster
- **Ambiguity tool:**
  - **Greatly simplify**

(old) eγ reconstruction work-flow

1. Select EM clusters and tracks





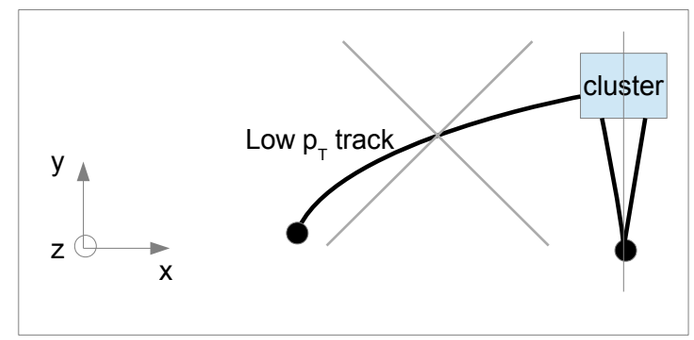
Cluster reconstructed using sliding windows algorithm:

Cells are clustered in fixed-size rectangular windows

$\Delta\eta \times \Delta\phi$(Middle)=3×5,5×5

Track particles that loosely match EM cluster refitted with GSF algorithm:

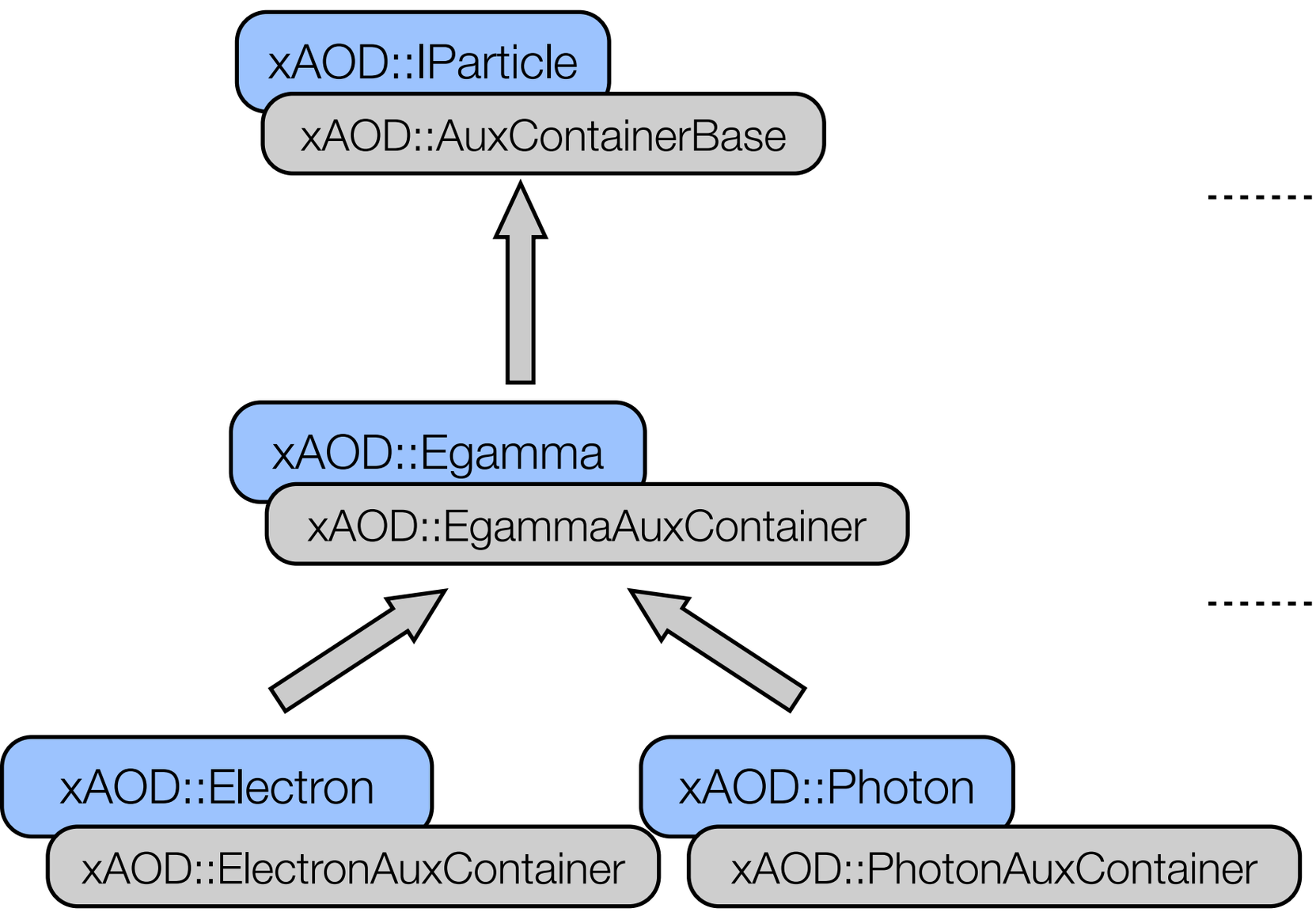
2. Match cluster with track particles and conversion vetices



- Electron: charge extrapolation using E/p
- Photon: straight line extrapolation.

In Release 21 a new clustering algorithm has been developed!

eγ xAOD Event Data Model (EDM)



N.B.: xAOD::Egamma is a link to xAOD::Egamma_v1
to allow schema evolution

- 4-mom and access to all variables in auxstore

- Link to caloCluster. Methods to access covariance matrix, author, object quality, isolation, shower shapes and ID (via enums)

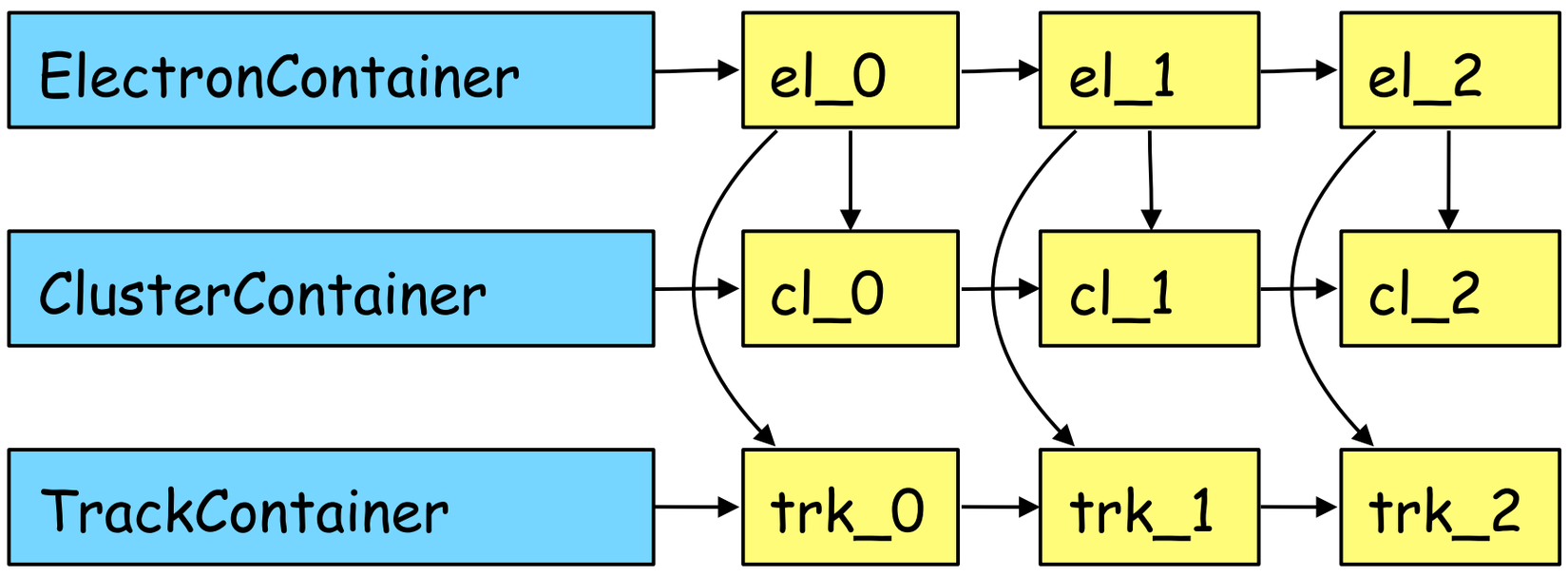
- Link to track particles, access to track-cluster matching variables

---

- Link to conversion vertices access to vertex-cluster matching variables

data organisation II

- To get to the cluster or track information, one has to get the cluster or track from the electron (e.g. methods caloCluster() or trackParticle())
  - The "element link" allows the access "behind the scenes"

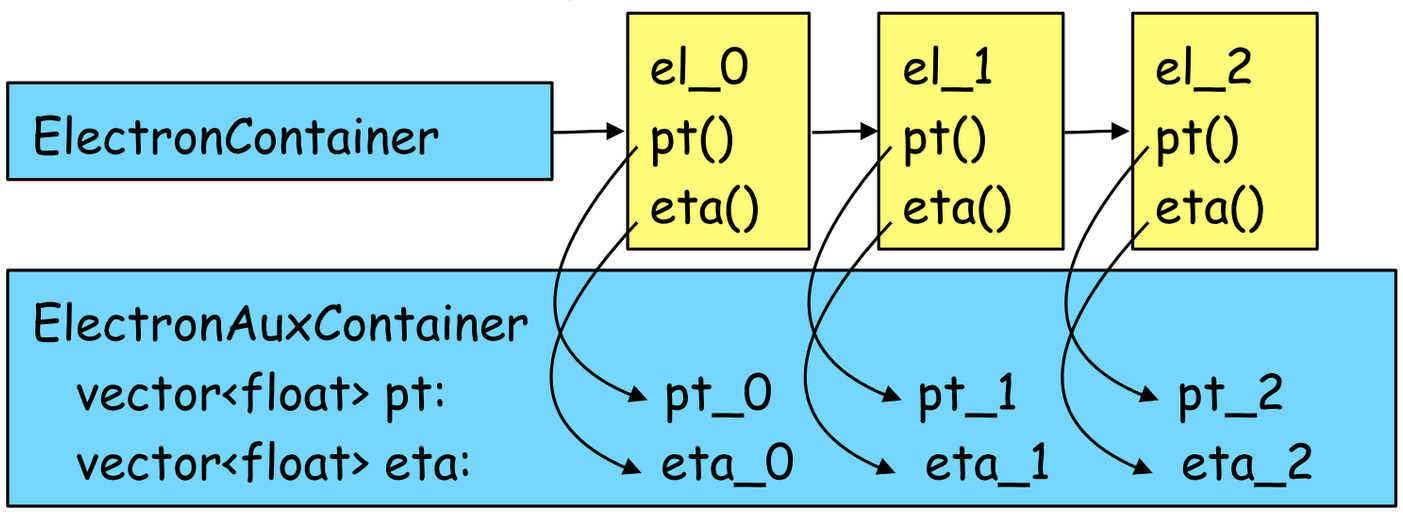


- One implication is that one needs to write out all three containers to have the full info for electrons

https://svnweb.cern.ch/trac/atlasoff/browser/Event/xAOD/xAODEgamma

egamma classes and containers II

- An ElectronContainer is just a vector of Electron objects
- However, the Electron does not actually contain any "data", it is all stored in a separate "Aux" or auxiliary container

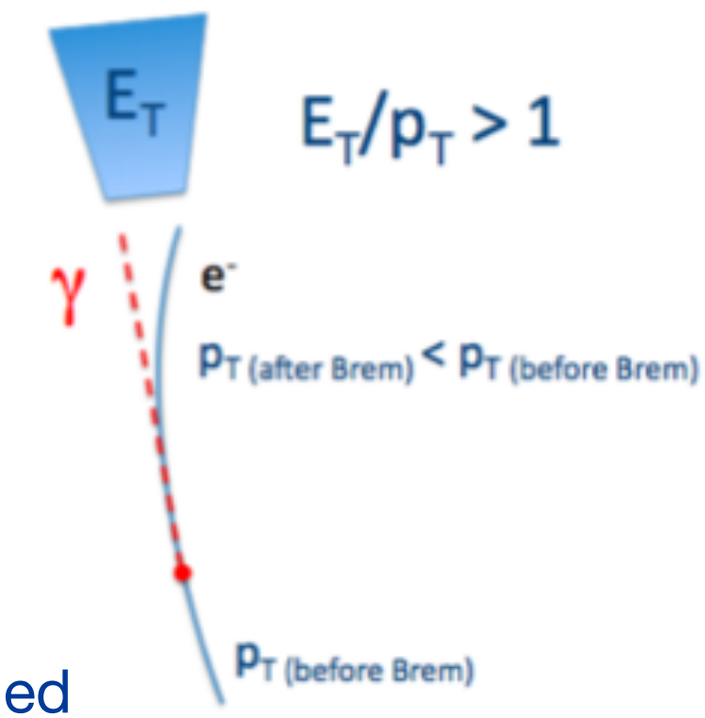


- All data is stored as simple vectors of floats or ints
- This storage is similar to D3PD, and allows "on-demand" ntuple-like access in ROOT
  - I.e. if only "pt" is accessed, only the branch with the vector<float> for pt is read

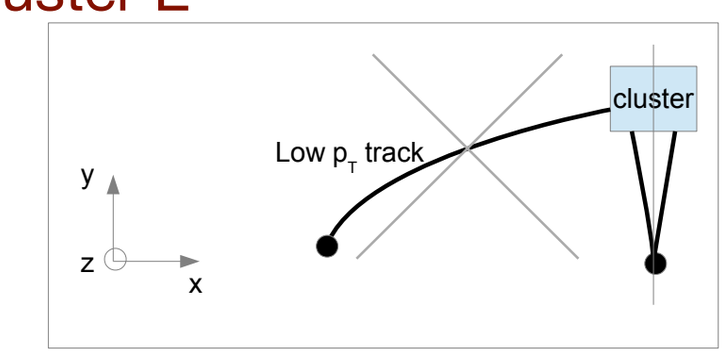
https://twiki.cern.ch/twiki/bin/view/AtlasProtected/EGammaSoftware

eγ reconstruction work-flow I

1. Select EM clusters and tracks
   - Cluster reconstructed with sliding window algorithm
     - Et > 2.5 GeV in EM calorimeter on towers equivalent to 3x5 cells in 2nd sampling
   - Track particles that loosely match EM cluster refitted with GSF algorithm
     - TRT-only tracks (used only for photons) simply copied



2. Match clusters with track particles and conversion vertices
   - Electrons: charged extrapolation using track-Pt or cluster E
   - Photons: straight-line extrapolation



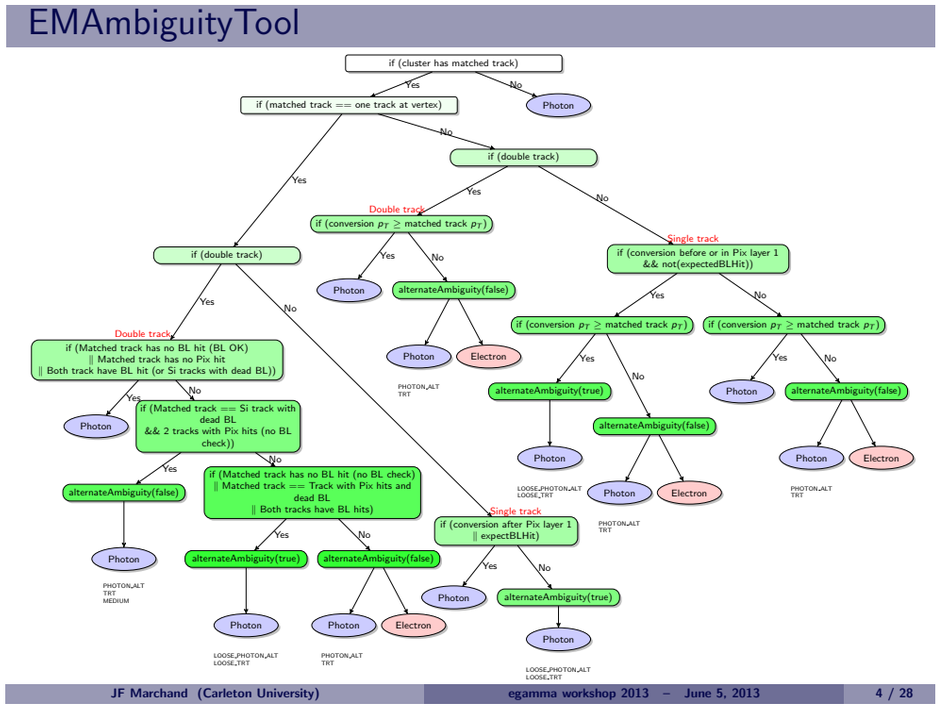
In Release 21 a new clustering algorithm has been developed!

https://twiki.cern.ch/twiki/bin/view/AtlasProtected/EGammaSoftware

eγ reconstruction work-flow II

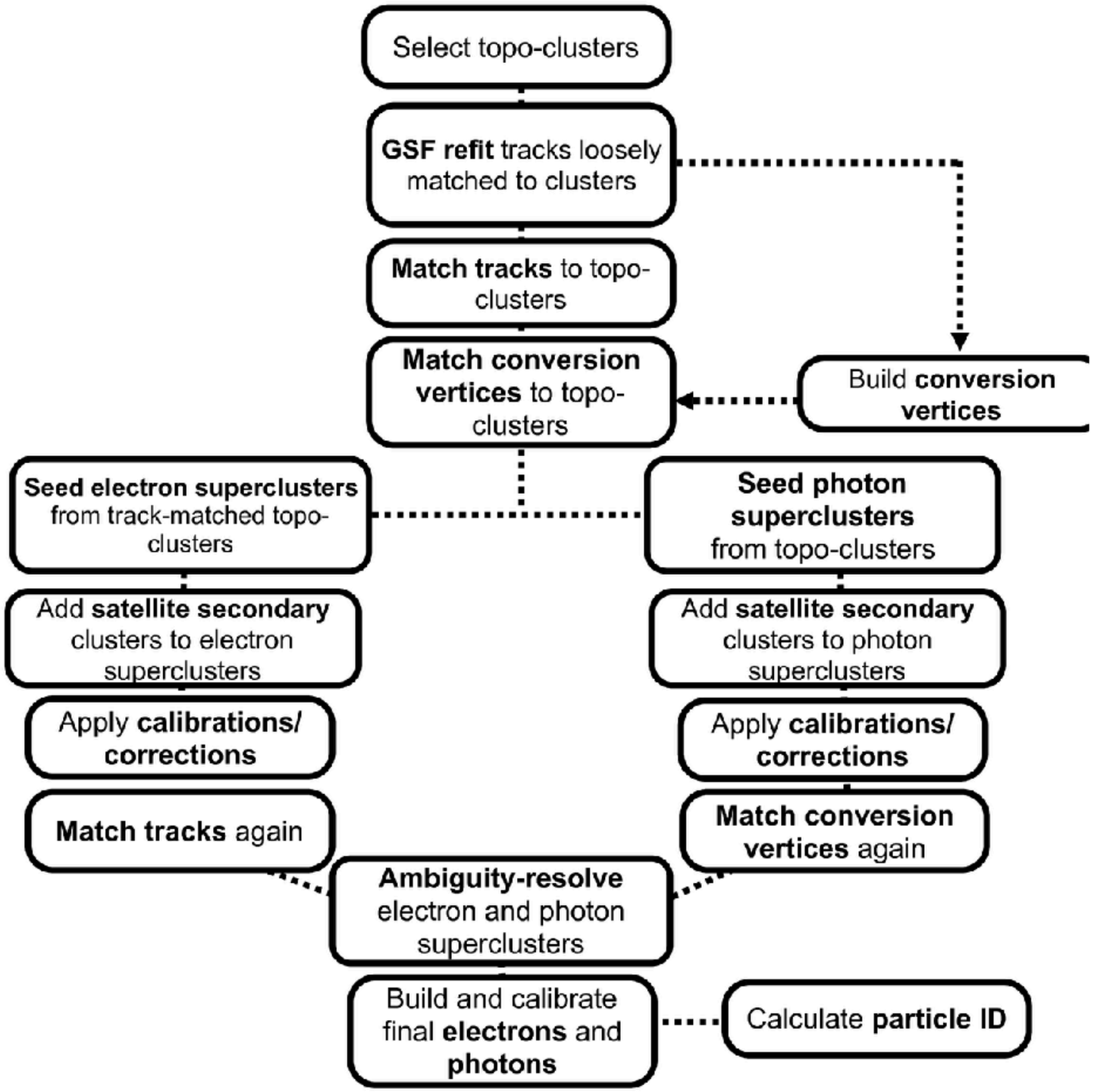
3. Ambiguity resolution (electron and/or photon?)

- Same cluster can give rise to both e/γ objects
  - Electron only: no vertex matched, track with E/p < 10 and pT > 2 GeV, b-layer hits or min. number of pixel hits and no Si+Si vertex
  - Photon only: no track with Si hits matched, Si+Si vertex and no track with min. number of pixel hits
  - Ambiguous (e + gamma): everything else (AuthorAmbiguous)
- Greatly simplified w.r.t. run-1



https://twiki.cern.ch/twiki/bin/view/AtlasProtected/EGammaSoftware

eγ reconstruction work-flow III

4. Create and decorate electrons and photons
   - Add photons built from topological EM clusters (mostly low-Et, AuthorCaloTopo35)
     - No overlaps with sliding window clusters
     - No dedicated conversion reconstruction
     - Clusters (3x5 cells in 2nd sampling) now stored in separate collection
   - Redefine cluster size (except topo-seeded): 3x7 (5x5) cells in 2nd sampling for barrel (endcap). Apply cluster corrections (position) and MVA calibration
     - Run-1: 3x5 for unconverted photons in the barrel
   - Energy/momentum: energy from cluster; η,ϕ from cluster or tracks (for electrons and conversions with 2 tracks with hits in Si detectors)
     - Run-1: track/vertex + cluster combination
   - Decorate with object quality, shower shapes, identification criteria

(new) eγ reconstruction work-flow I