dRICH Geometry and Optics in DD4hep

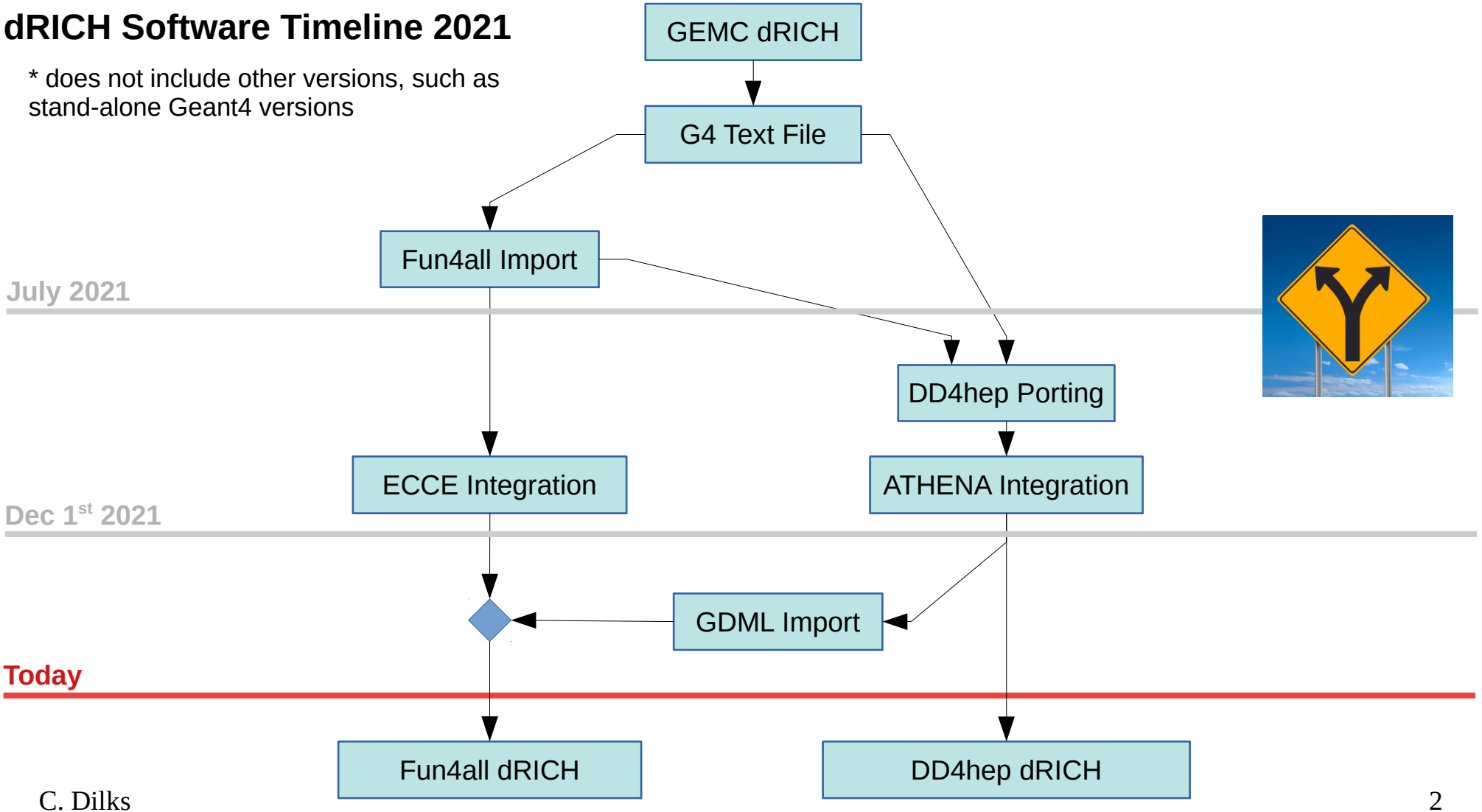
Christopher Dilks

dRICH Software Kickoff Meeting

4 May 2022

dRICH Software Timeline 2021

* does not include other versions, such as stand-alone Geant4 versions



Available dRICH Software

- Fun4all Standalone: https://github.com/cisbani/dRICh
  - Geometry G4 text file: https://github.com/cisbani/dRICh/tree/main/share/config
    - Updated ECCE Versions: https://github.com/ECCE-EIC/calibrations/tree/main/dRICH/mapping
  - Optics and Material Properties Generation: https://github.com/cisbani/dRICh/tree/main/share/source

- DD4hep (ATHENA): https://eicweb.phy.anl.gov/EIC/detectors/athena **(use branch 144-irt-geometry)**
  - Geometry:
    - Compact XML file (constants): https://eicweb.phy.anl.gov/EIC/detectors/athena/-/blob/144-irt-geometry/compact/drich.xml
    - Placement Algorithms: https://eicweb.phy.anl.gov/EIC/detectors/athena/-/blob/144-irt-geometry/src/DRICH_geo.cpp
  - Optical / Material Property Tables: https://eicweb.phy.anl.gov/EIC/detectors/athena/-/blob/144-irt-geometry/compact/optical_materials.xml
  - GDML Files available in CI artifacts, for example:
    - https://eicweb.phy.anl.gov/EIC/detectors/athena/-/jobs/638410/artifacts/file/geo/drich_only.gdml (link may expire)
  - IRT, documentation, and analysis: https://eicweb.phy.anl.gov/EIC/irt
  - Development scripts + more documentation: https://github.com/c-dilks/drich-dev

- GEMC: https://github.com/EIC-eRD11/dualRICH_inMEIC

- Send additional links / corrections
- Start our own wiki page under https://wiki.bnl.gov/eic-project-detector/index.php/CherenkovPID

DD4hep dRICH Fork Developments

- Integration with ATHENA (resizing, optical focusing, etc.)
- Generalize sensor tiling on a sphere
- Maximize azimuthal and polar angle acceptance
- Algorithm for focal region steering
- Algorithm for 2 mirrors per sector (and possibly more)
- Connection with IRT algorithm + numerous additional updates to geometry, materials, and optics

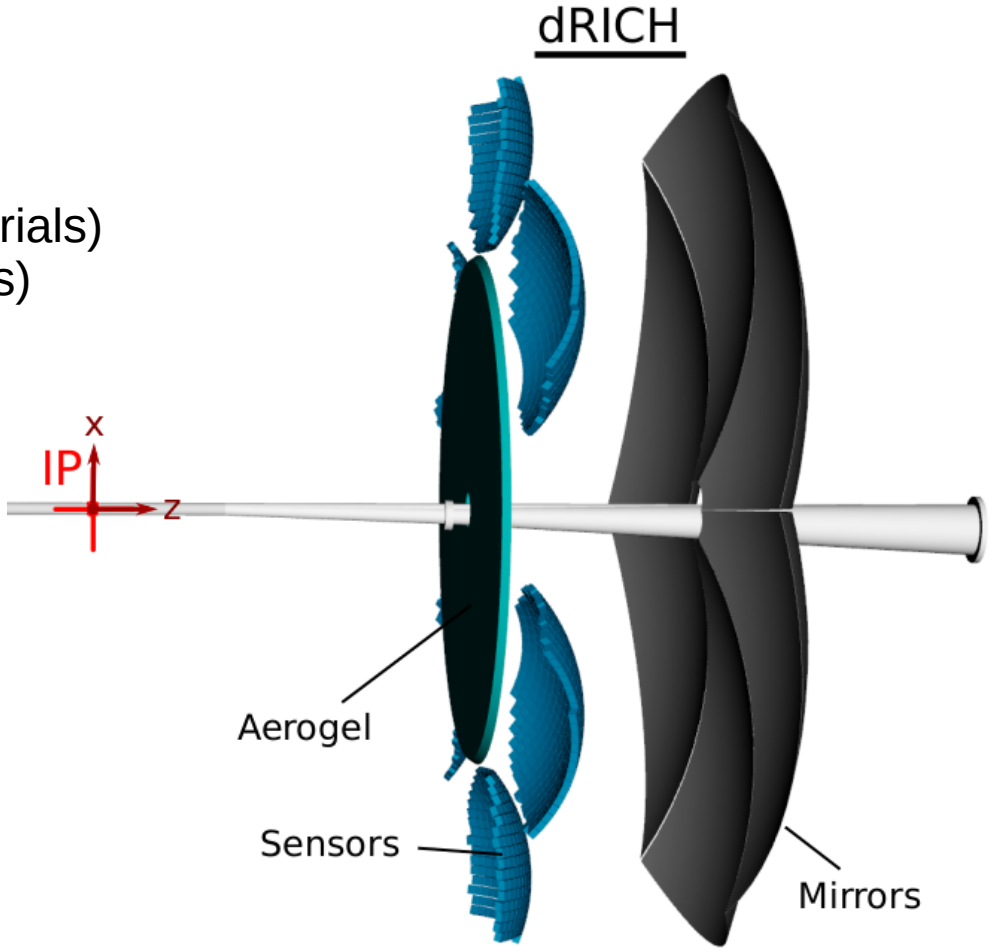
dRICH in DD4hep

Gitlab Server
https://eicweb.phy.anl.gov/EIC

Relevant repositories:
- detectors/athena (geometry, optics, materials)
- Project Juggler (reconstruction algorithms)
- IRT (indirect ray tracing)

3D Interactive View of dRICH (jsROOT)

https://eic.phy.anl.gov/geoviewer/index.htm?file=https://eicweb.phy.anl.gov/EIC/detectors/athena/-/jobs/artifacts/master/raw/geo/drich_only_geo.root?job=dump_geometry&item=default;1&opt=clipx;clipy;transp30;zoom75;ROTY290;ROTZ350;trz0;trr0;ctrl;all

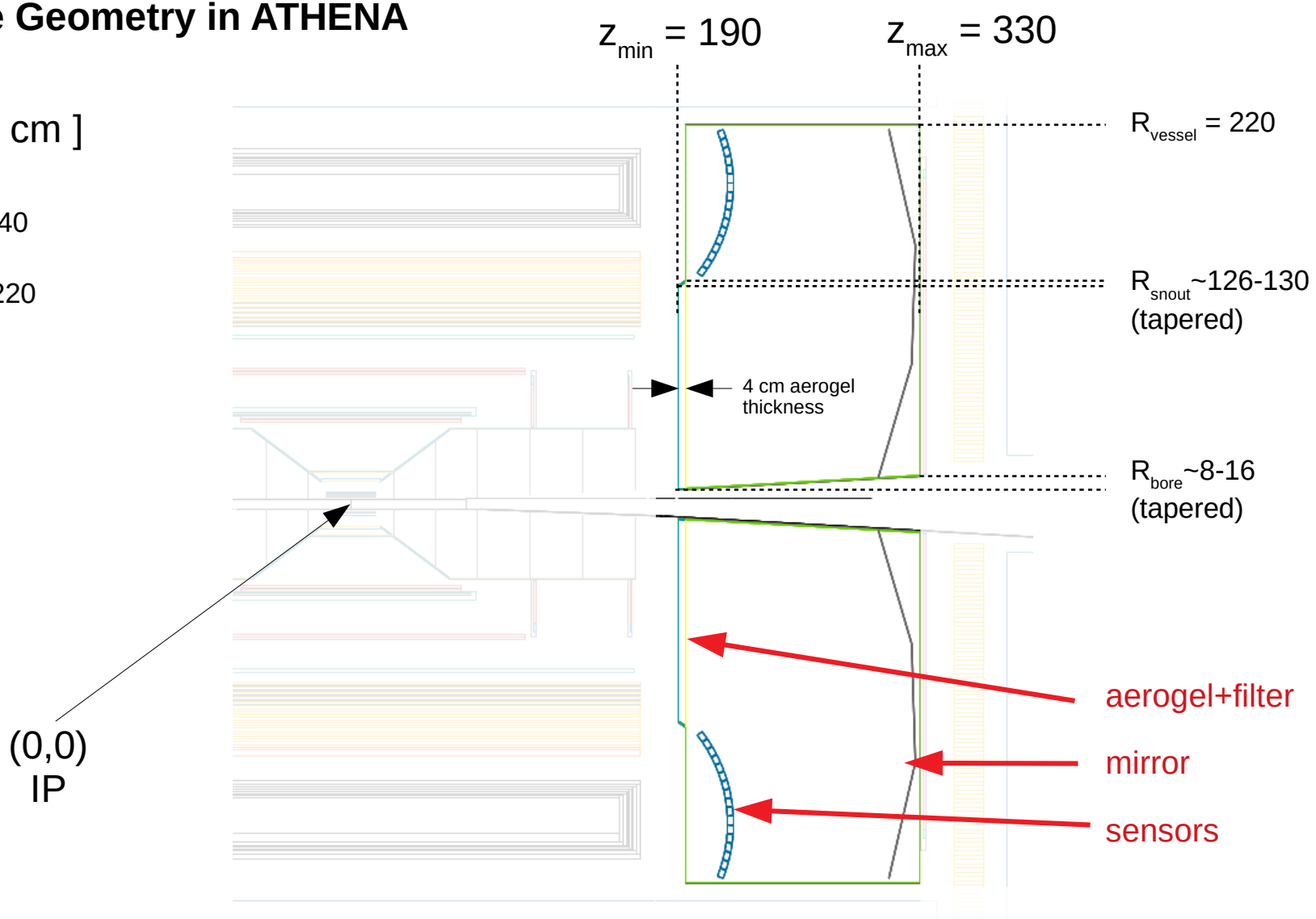


Envelope Geometry in ATHENA

[units = cm]

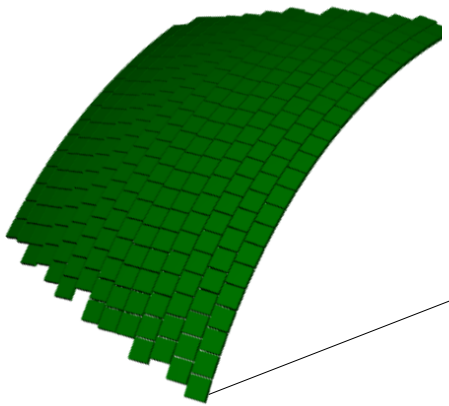
z_{length} = 140

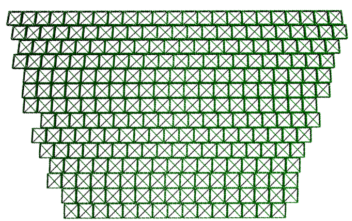
R_{vessel} = 220



Sensor Placement: Spherical Tiling

- Tile sensors like a disco ball, take a patch avoiding “seam” and “pole”
- Each sector has a sensor sphere, with 3 parameters: center (z_S,x_S) and radius (r_S)
- Sensor normal vectors are along sphere radii
- **Sphere may not be the optimal geometry for sensor placement, however**



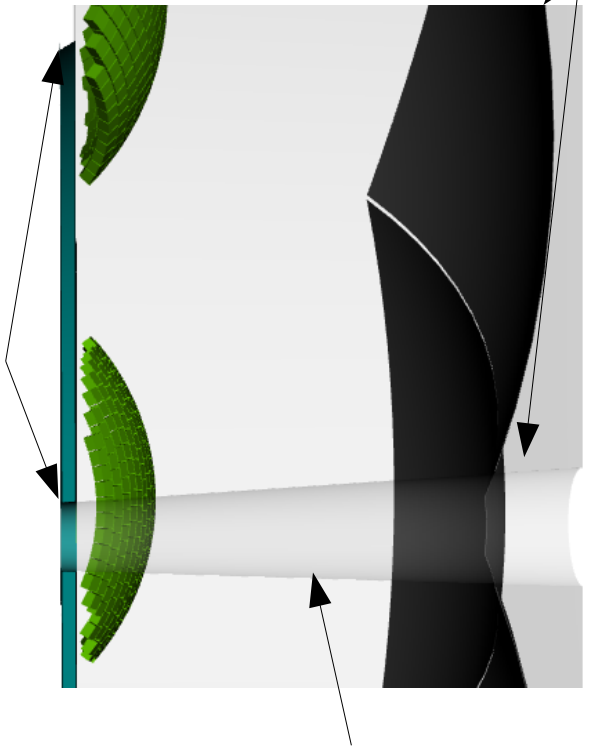


note: images for demonstration only, this is not the current number and size of sensors

Increase Polar Acceptance

mirror fills available polar space

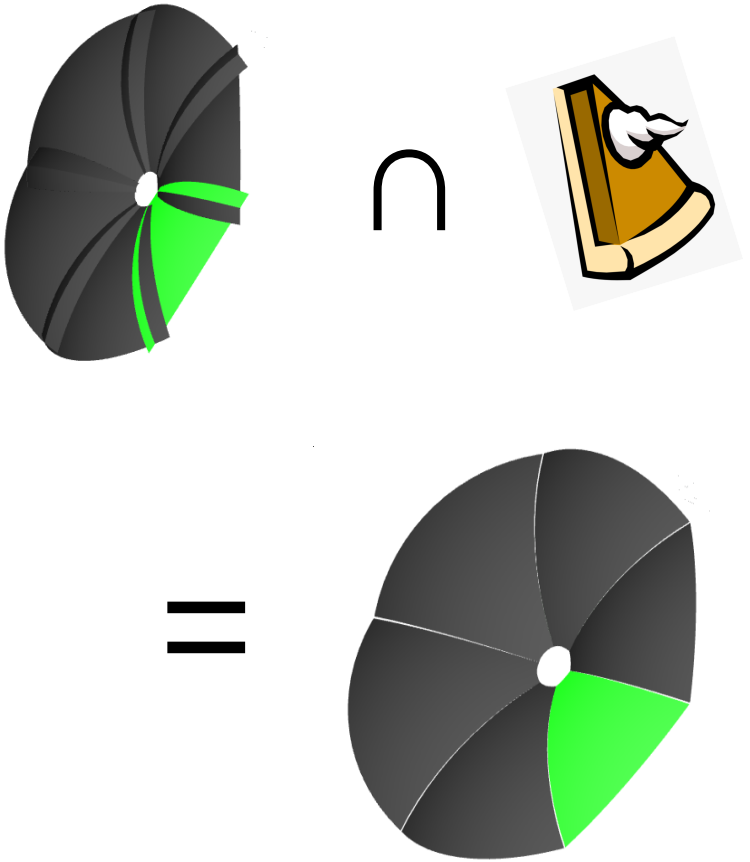
conical aerogel fills entire “snout” region; projective from IP



tapered bore

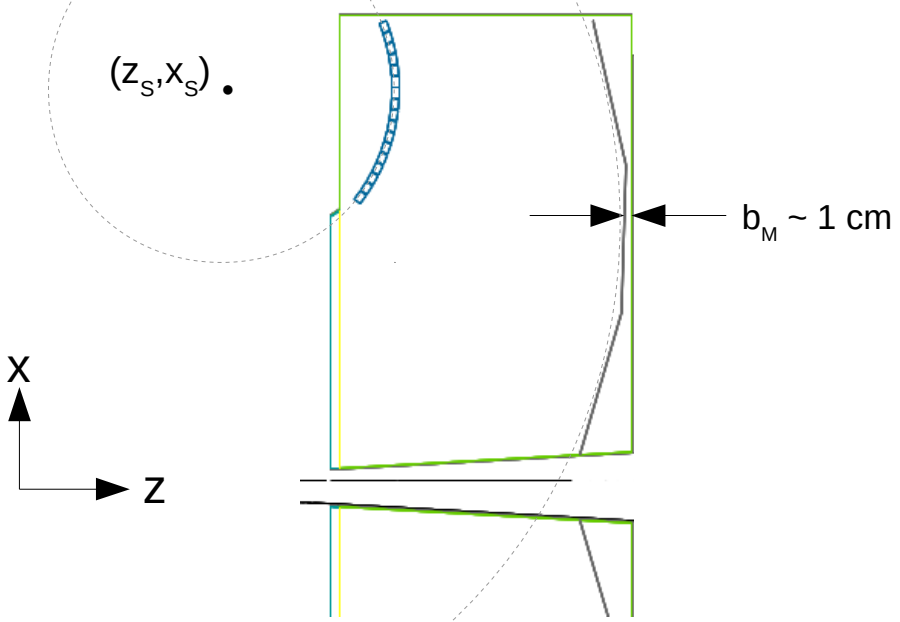
Increase Azimuthal Acceptance

Allow mirrors to have full azimuthal overlap, then take boolean intersection with a cylindrical sector



Spherical Mirror Focusing

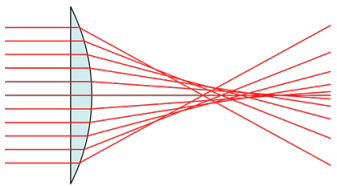
Point-to-Point focusing → Steer Parallel-to-point focusing



- Mirror sphere parameters:
  - Center (z_M,x_M)
  - Radius (r_M)
- Re-parameterized for focus tuning:
  - Back-plane: $b_M = z_{max} - z_M - r_M$
  - Focus tunes (z_F,x_F)

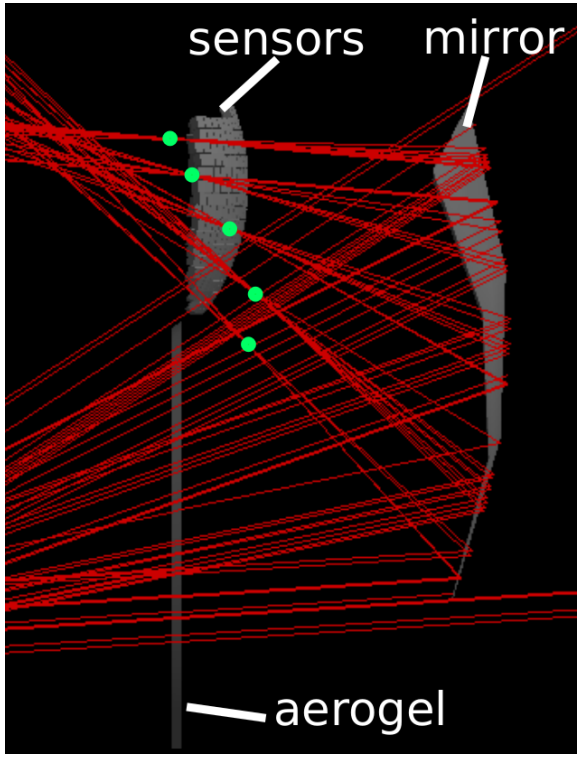
- Given the parameters $\{b_M,z_F,x_F\}$, along with the sensor sphere center $\{z_S,x_S\}$, the mirror sphere parameters $\{z_M,x_M,r_M\}$ are determined from the point-to-point focusing of the IP at the point (z_S+z_F, x_S+x_F)
- Spherical aberrations "blur" point-to-point focusing
- Focusing of Cherenkov rings requires **parallel-to-point focusing**
- Parallel-to-point focal region can be steered by changing $\{z_F,x_F\}$, as well as $\{z_S,x_S\}$

spherical aberrations from a lens

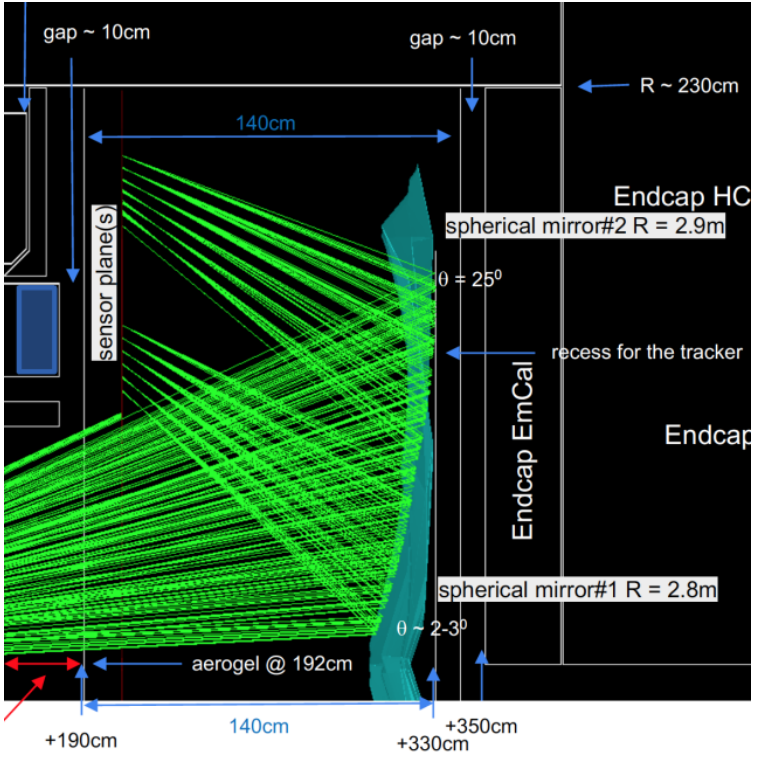


dRICH Dual Mirrors

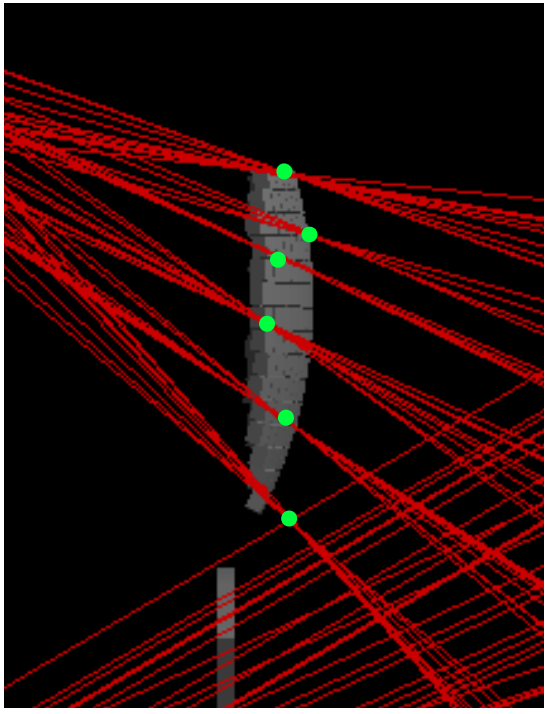
single-mirror config, 5 collimated photon beams; this was the configuration for the ATHENA proposal performance studies



Alexander's dual mirror configuration, in standalone Geant4 sandbox



current status of dual mirror configuration in DD4hep:



still plenty of room for improvement!!

Sensors tiled on a sphere may not be ideal...

dRICH in DD4hep: Software details

DD4hep Compact Files

- Compact files: generalized, *fully customizable*, XML geometry description with **numbers and parameterizations**
- **Algorithms** (e.g. tiles on a sphere) are in the C++ source code: “Detector Constructors”
- XML APIs →friendly to external configuration (optimization, machine learning)
  - change a number, no need to recompile
  - can use XML file to specify usage of a different C++ algorithm, again no need to recompile
  - XML file is limited only by imagination
- Fun4all version of dRICH uses a text file configuration file for geometry and placement
  - text file configuration must follow allowed syntax and specification

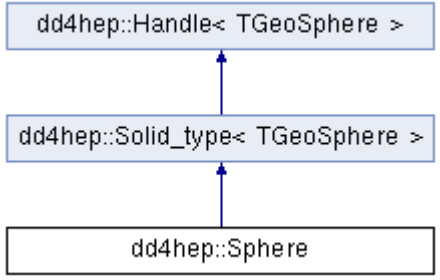
```
<radiator
  rmin="DRICH_rmin0 + DRICH_wall_thickness + 0.2*cm"
  rmax="DRICH_rmax0 - DRICH_wall_thickness - 0.2*cm"
  phiw="60*degree"
  frontplane="DRICH_window_thickness + 0.5*DRICH_aerogel_thickness"
  pitch="0*degree"
  >
  <aerogel
    material="Aerogel_DRICH"
    vis="DRICH_aerogel_vis"
    thickness="DRICH_aerogel_thickness"
    />
  <filter
    material="Acrylic_DRICH"
    vis="DRICH_filter_vis"
    thickness="0.3*mm"
    />
</radiator>
```

detectors/athena/compact/drich.xml

```
<mirror
  material="Acrylic_DRICH"
  surface="MirrorSurface_DRICH"
  vis="DRICH_mirror_vis"
  backplane="DRICH_window_thickness + 1.0*cm"
  rmin="DRICH_rmin1 + DRICH_wall_thickness - 1.0*cm"
  rmax="DRICH_rmax2 - DRICH_wall_thickness - 1.0*cm"
  phiw="59.5*degree"
  thickness="0.2*cm"
  focus_tune_x="30.0*cm"
  focus_tune_z="-40.0*cm"
  debug="DRICH_debug_mirror"
  />
```

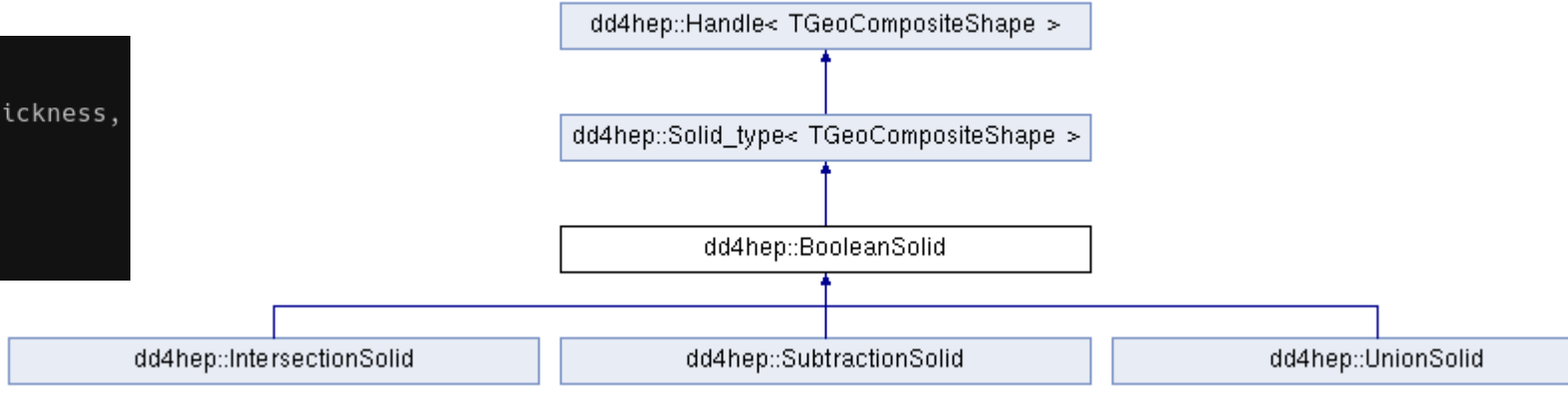
DD4hep Detector Constructors

- Compact file has numbers, C++ source file describes how to use them to build a dRICH (detectors/athena/src/DRICH_geo.cpp)
- DD4hep objects derive from TGeo objects; conversion to Geant4 objects happens internally
- C++ file also includes optics geometry details needed for the IRT algorithm



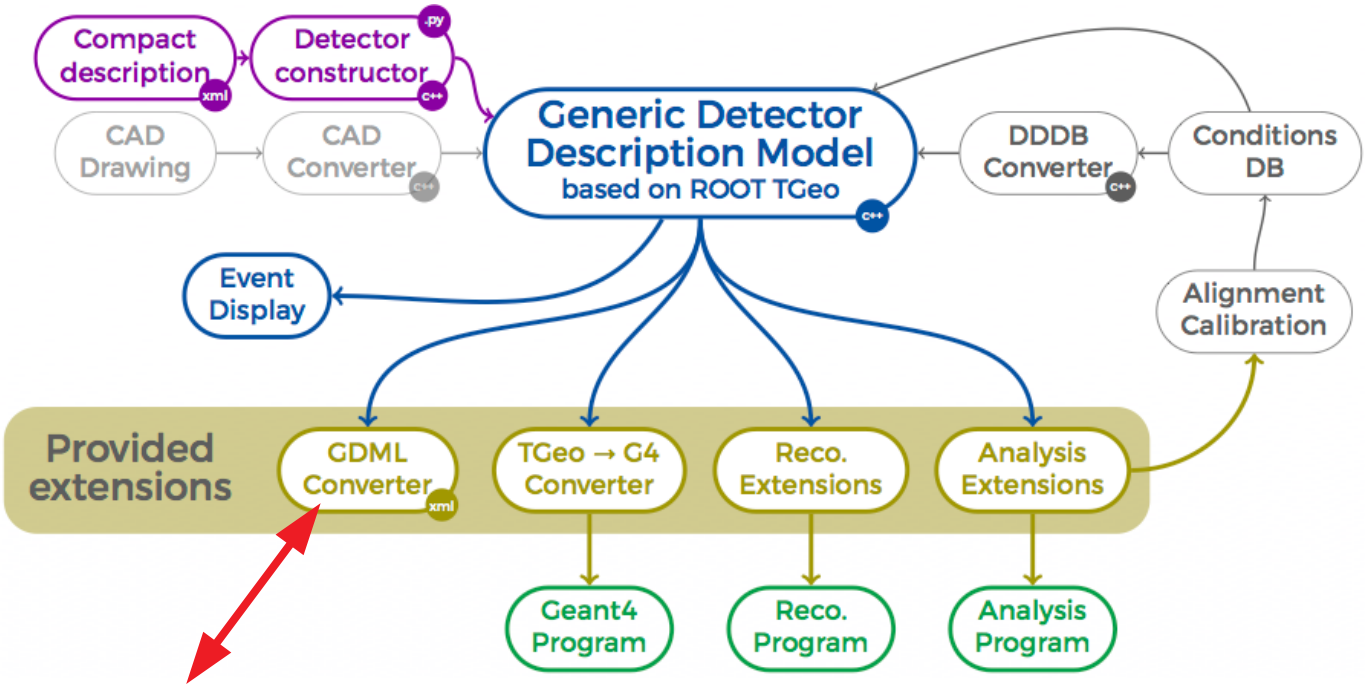

```
// derive spherical mirror parameters `(zM,xM,rM)`, for given image point
// coordinates `(zI,xI)` and `dO`, defined as the z-distance between the
// object and the mirror surface
// - all coordinates are specified w.r.t. the object point coordinates
// - this is point-to-point focusing, but it can be used to effectively steer
//   parallel-to-point focusing
double zM,xM,rM;
auto FocusMirror = [&zM,&xM,&rM](double zI, double xI, double dO) {
  zM = dO*zI / (2*dO-zI);
  xM = dO*xI / (2*dO-zI);
  rM = dO - zM;
};
```

```
Sphere mirrorSolid1(
    mirrorRadius,
    mirrorRadius + mirrorThickness,
    mirrorTheta1,
    mirrorTheta2,
    -40*degree,
    40*degree
    );
```



DD4hep Modularity

https://dd4hep.web.cern.ch/dd4hep/page/about/



Fun4all GDML Import

- Using GDML as a 'bridge' between DD4hep and Fun4all is possible
- This would suffice as a **temporary** 'framework independent' approach to maintain productivity while we await decisions for how the full simulation framework will proceed

C. Dilks

Preparing for the Future

Current dRICH Implementations
- DD4hep
- Fun4all, via text file
- Fun4all, via GDML import from DD4hep
- Standalone Geant4 (sandbox)

```
if future == dd4hep:

     port f4a developments to dd4hep   # originally ported Jul 2021, `diff` text config files?
     optimize()



else if future == fun4all:

     if fun4all_gdml_import_works_well AND dd4hep_is_preferred:
          make standalone dd4hep dRICH   # or integrate with other dd4hep detectors, if any
     else:
          port dd4hep developments to f4a   # time + patience

     optimize()

else:

     adapt()
```

N.B. ideally we want to be running and tuning **`optimize()`** before **`future`** is known
→ short term plans – today's open discussion