

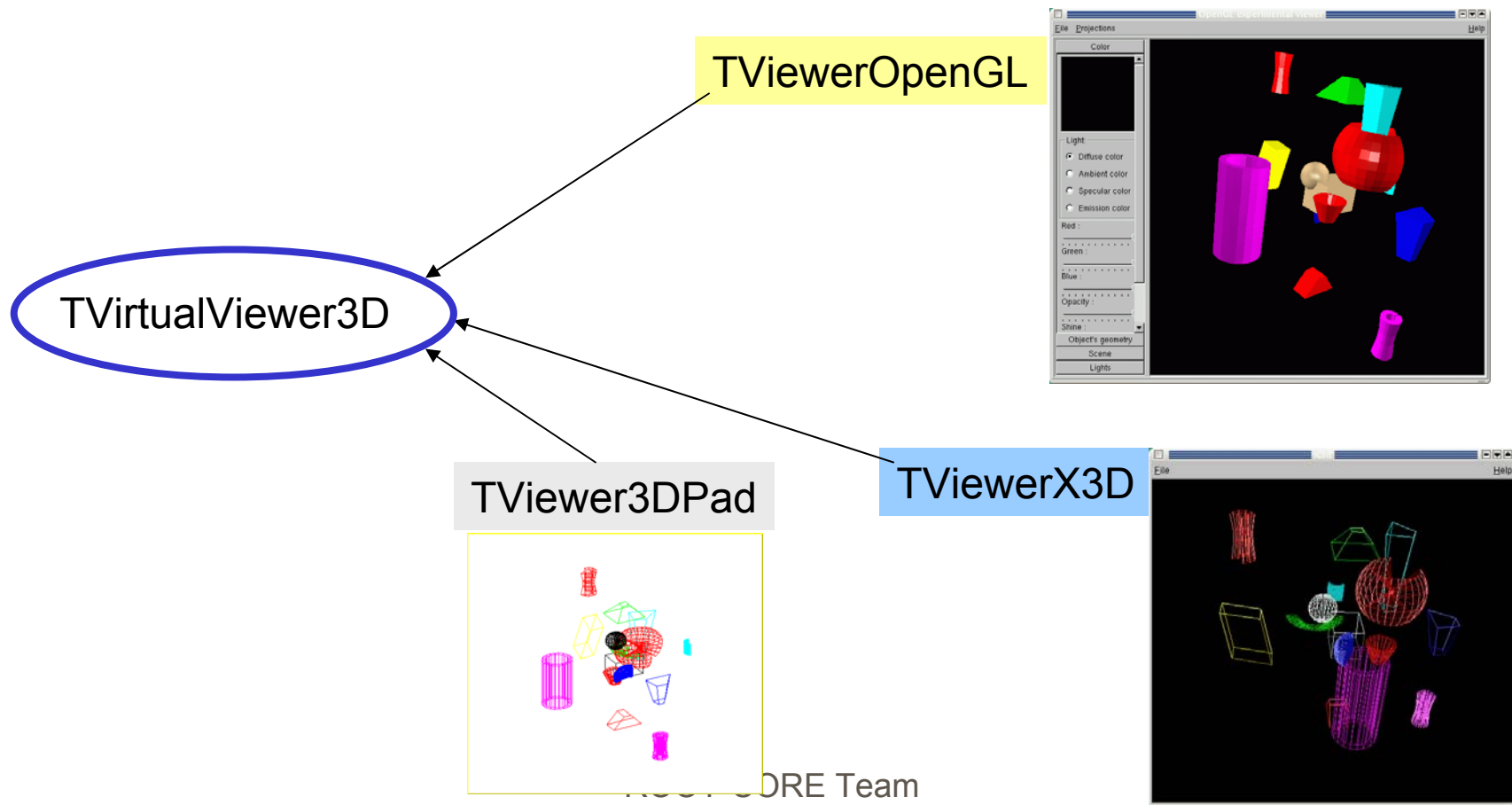
# ROOT Graphics

- Olivier Couet
- Valeri Fine
- Richard Maunder
- Valeriy Onuchin

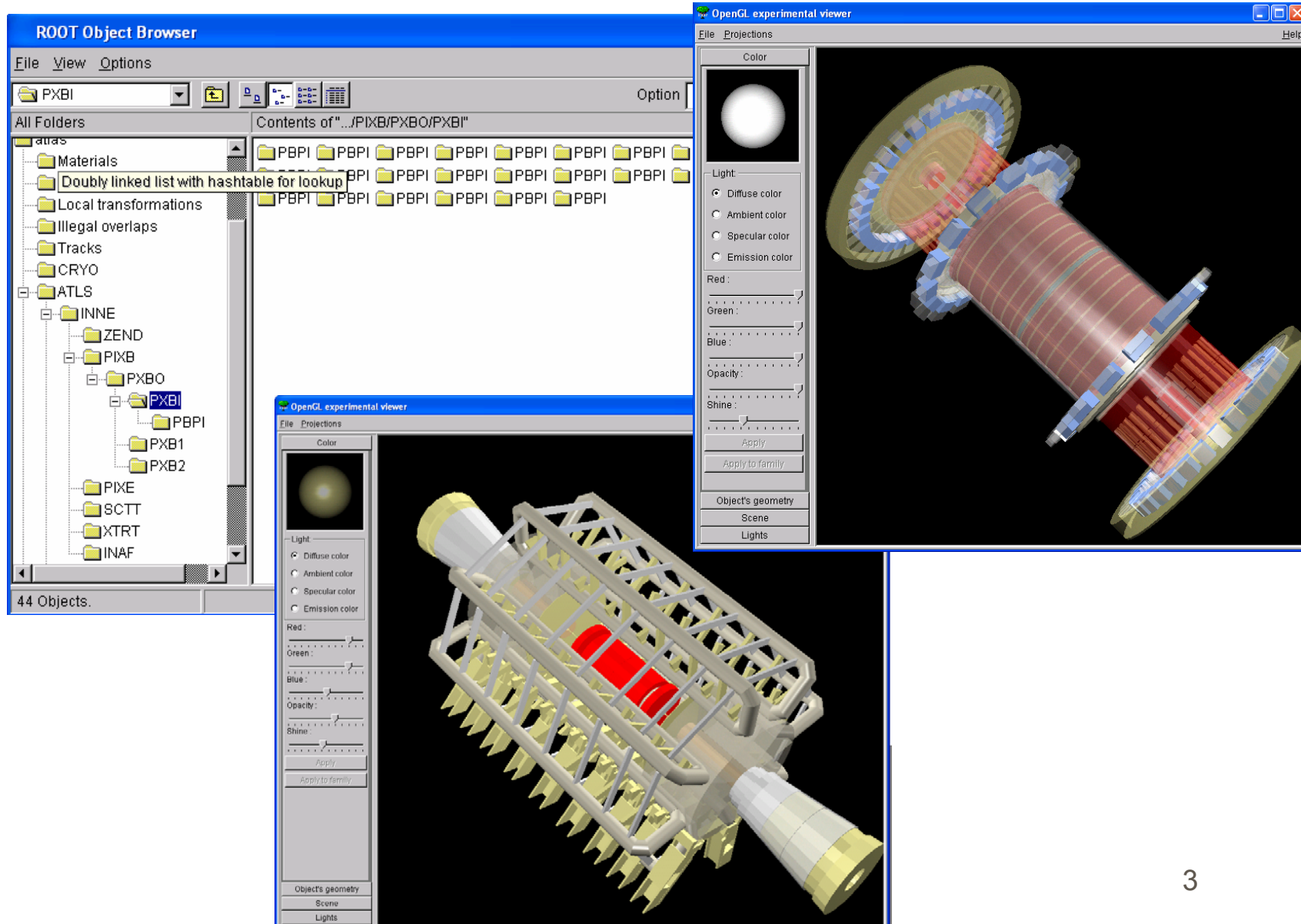
# 3D Viewers

---

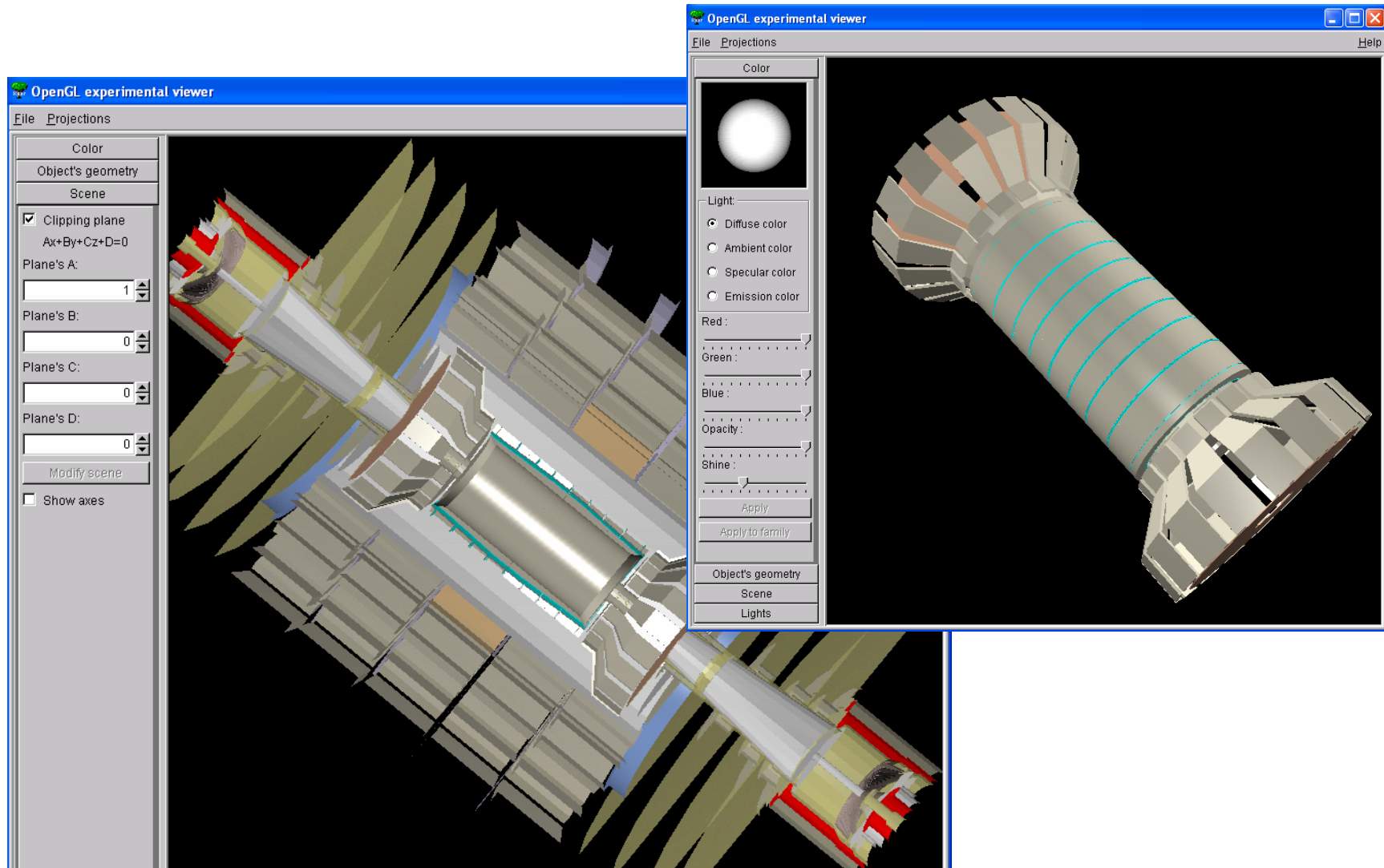
3 concrete viewers inherit from the virtual interface TVirtualViewer3D.



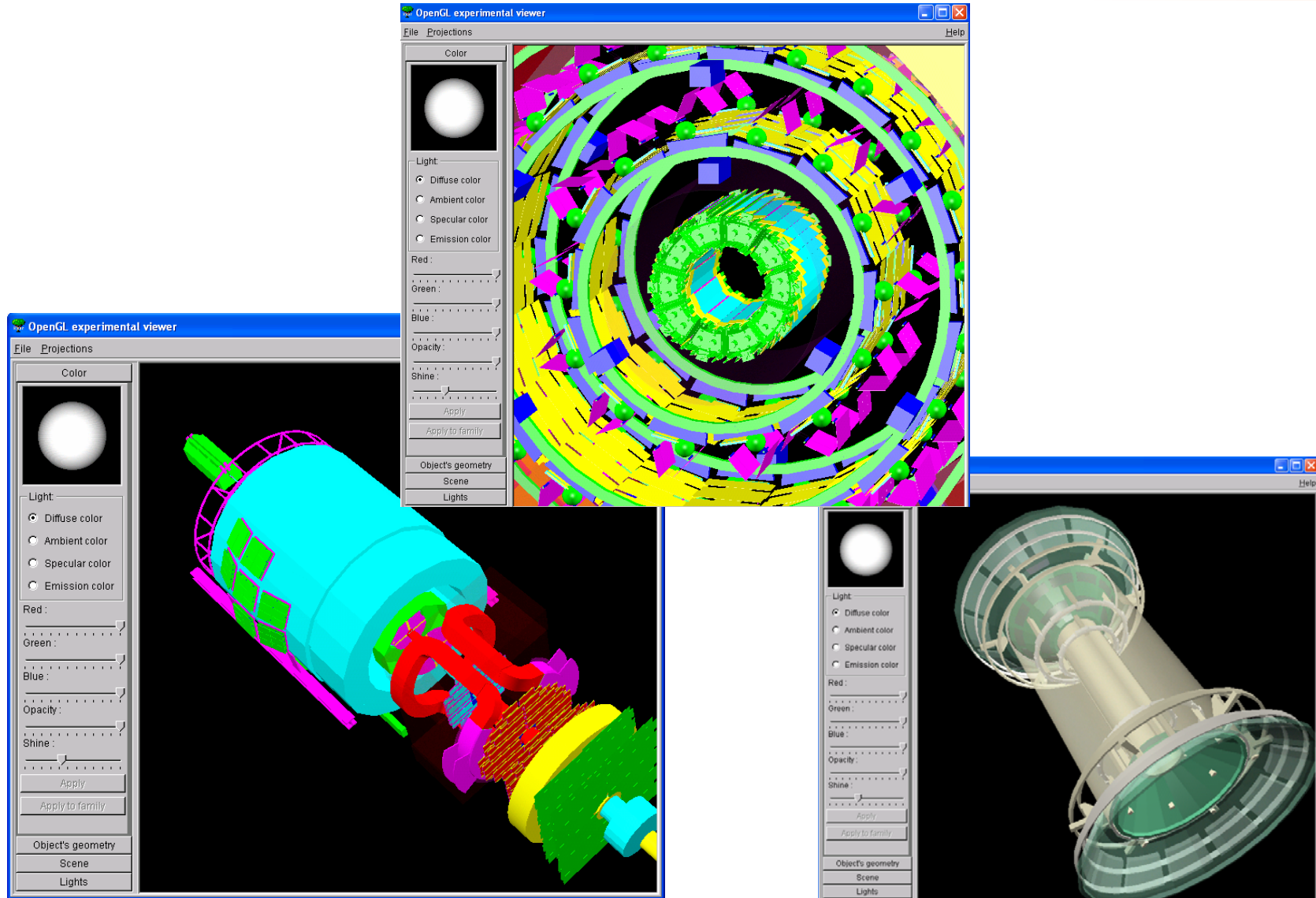
# ATLAS detector in TGeo classes



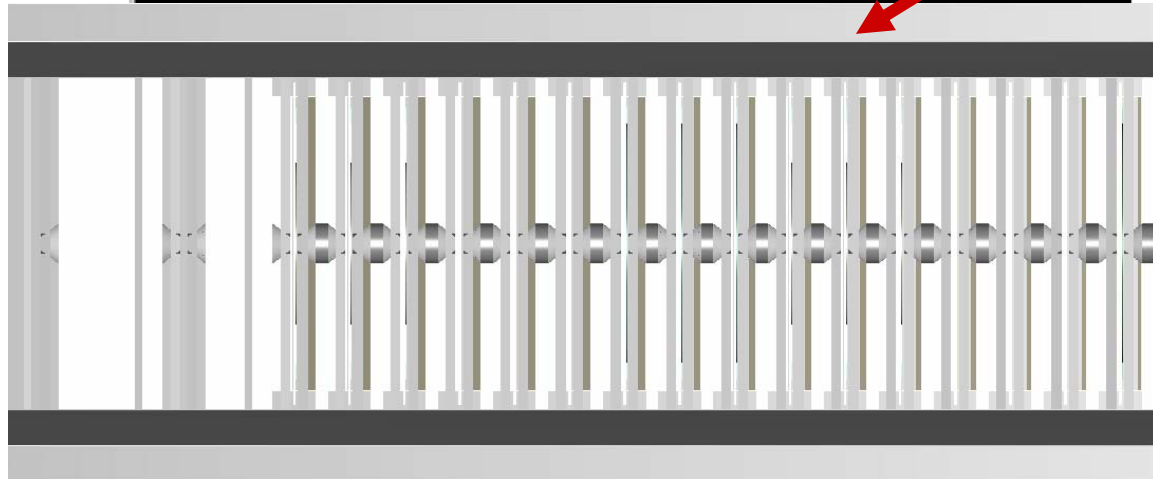
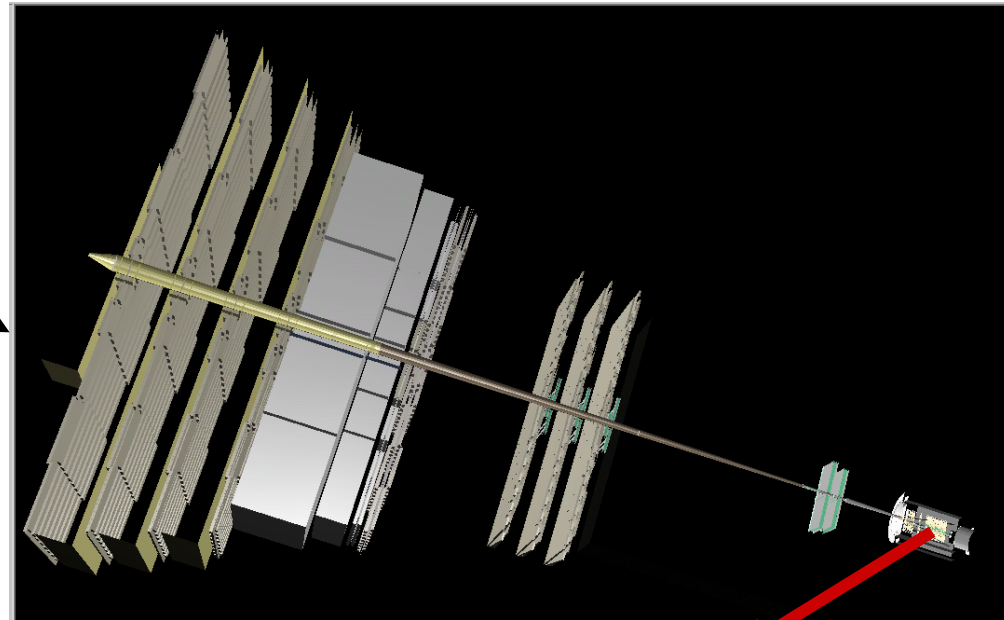
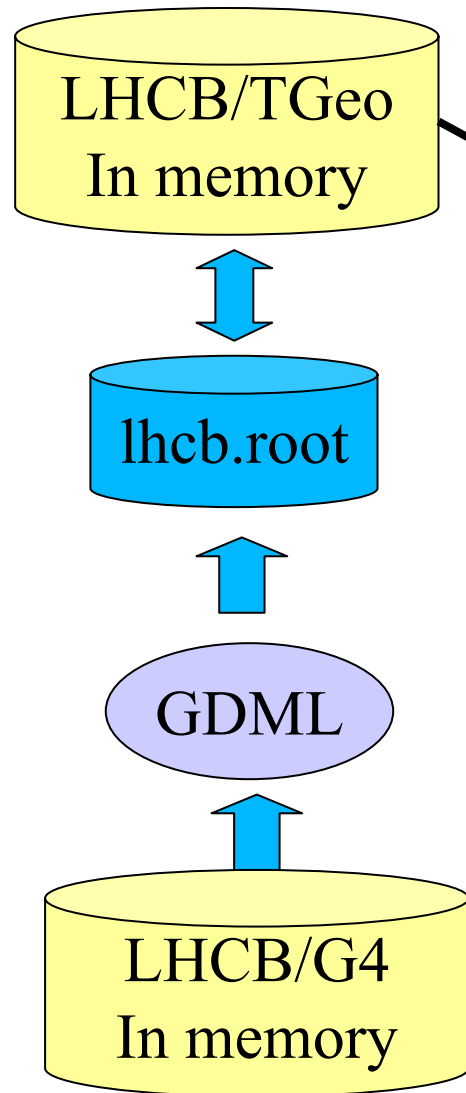
# CMS detector in TGeo classes



# Alice detector in TGeo classes



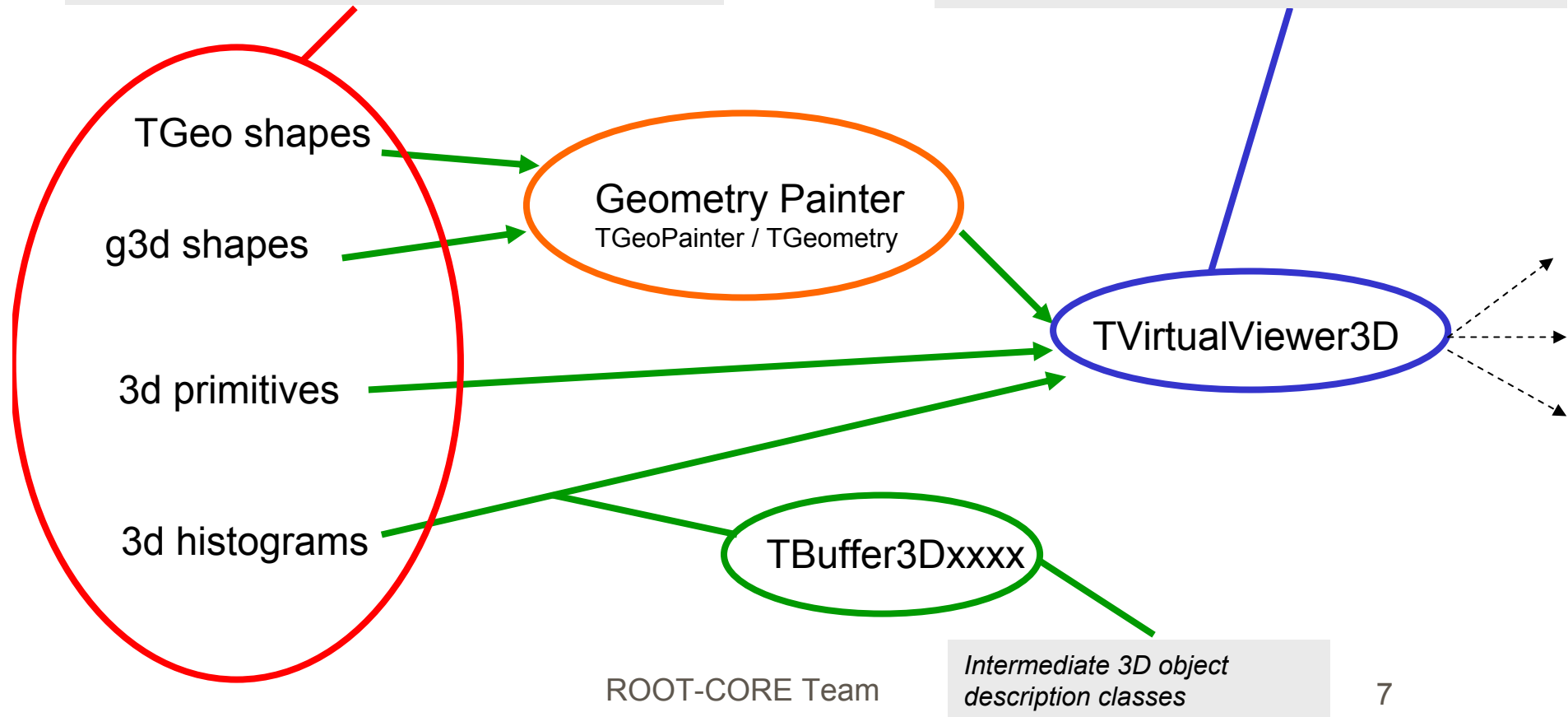
# LHCb detector in TGeo classes



# 3D scenes rendering

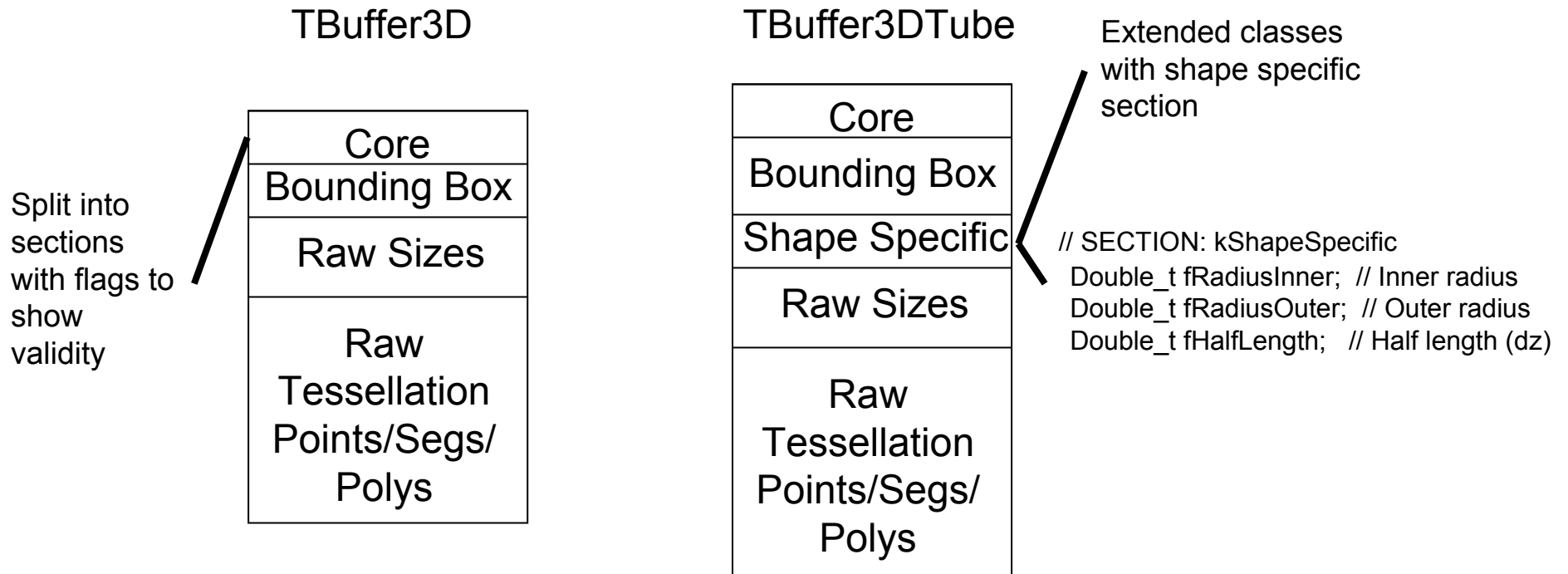
*Producers: 3D objects describe themselves in TBuffer3D (vertices, edges, polygons) and extended classes (TBuffer3DSphere, TBuffer3DTube etc).*

*Consumers: Extract shape description from TBuffer3D (see concrete implementations)*



# Viewer Infrastructure Changes

## 3D Buffer Classes



- Producers fill cheap sections automatically and expensive parts (tessellation) on demand of viewer.
- Addition of bounding box, local/master reference frame and translation matrix.



# Viewer Infrastructure Changes - cont

---

## TVirtualViewer3D

- Viewer preference for local frame positions – producer shapes not obliged to meet request.\*
- Viewer interest in child objects – should producer send?
- Simple objects: “3D primitive, at this 3D location”
- Placed & Template objects: “Placed copy (with unique ID) of this template 3D primitive” – enables viewer side caching of the unique shapes.

Equivalent to:

*TVV3D:*      *Template*      *Placed*

Geant4 :      Logical Volume      Physical Volume

TGeo:          Volumes          Nodes

\* All producing shapes must be able to generate buffer in master reference frame, and all consuming viewers be capable of displaying them.

# Viewer Infrastructure Changes - cont

---

Together these allow:

- Filling of only the sections a viewer needs for a specific shape.
- Rejection of objects off screen before tessellation.
- Efficient, high quality native viewer tessellation of supported shapes, with fall back of producer side tessellation for unsupported one.
- Repeated geometry expansions, with termination on viewer request.
- Various viewer side caching schemes – e.g. retain all large/costly shapes, extract finer details as current view requires and performance permits.
- Ensures code outside viewer is free of viewer specific dependencies/branches.
  
- == Higher quality, faster rendering and interaction in OpenGL and high performance viewers, + backward compatibility with pad and legacy x3d viewer.

# Viewer Infrastructure Changes

---

## TBuffer3D

- Split into sections – core, bounding box, shape specific, raw tessellation (points/segs/polys) with flags to show validity.

- Extended classes with abstract shape descriptions e.g. TBuffer3DTube:

```
// SECTION: kShapeSpecific  
Double_t fRadiusInner; // Inner radius  
Double_t fRadiusOuter; // Outer radius  
Double_t fHalfLength; // Half length (dz)
```

- Producers fill cheap parts automatically and expensive parts (tessellation) on demand of viewer.
- Addition of bounding box, local/master reference frame and translation matrix.

# OpenGL Viewer

---

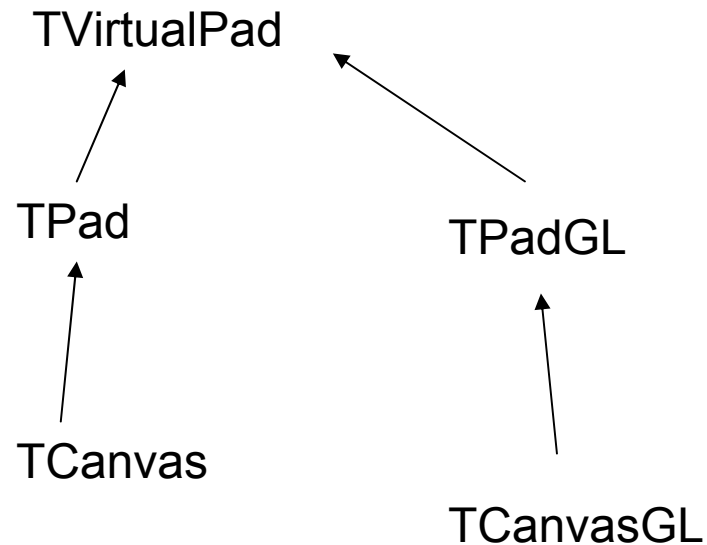
- In the future will be the main 3D viewer. Has to be updated to take advantage of the infrastructure changes.
- Use the full power of OpenGL (lights, transparency, interactivity, anti-aliasing, hardware acceleration ...)
- PS/PDF output using *gl2ps*
- Native rendering of uncut solid spheres and various tubes already in place.
- Soon composite shape support

## *In the future:*

- Convert to local frame, with template shape and OpenGL display list caching.
- Level of detail support – adjust tessellation for object size.
- Animation of objects.
- etc

# TPadGL / TCanvasGL

---



*Inheritance diagram*

Once TPad will be split, a new version Based on GL will be implemented. It will do:

- 2D graphics (*the equivalent of gVirtualX is needed for OpenGL*).
- 3D scene rendering (TGeo) (*see 3D viewers slides*)
- 3D representation (Lego etc ..)

## 2D graphics and others issues

# Graphics output

---

Graphics outputs can be generate using:

- PostScript: Stable. No major developments foreseen
- PDF: The future. Very likely will replace PS in the medium term.
- SVG: not complete yet. More and more requests.

*Vector*

- TAsimage package:
  - many pixel formats,
  - Works in batch mode,
  - Markers are missing,
  - Could be use to generate output for ray tracing.

*Pixel*

# Astronomers requests

---

- Reverse Y (and X) axis. This requires changes in many places.
- New projections: AITOFF, MERCATOR etc... It is available for some representations but still missing for COL plots for instance.



# TLatex

---

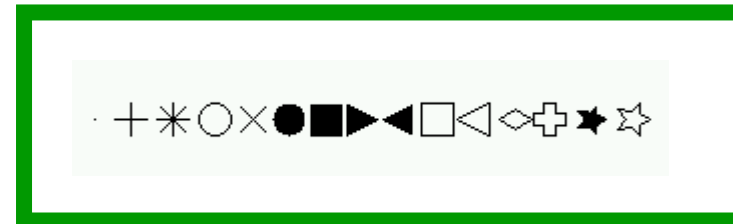
- `#ell` Calligraphic “l” : l
- `#matcal{}`: Calligraphic font
- `#v{}`
- `#perthousand{}` : ‰
- German umlaut

These symbols are not easy. Either the font is not (yet) in our TTF and PostScript drivers, or the character is available in one but not in the other.

# Markers

---

- We need more markers. The current list is not enough.
- User defined markers
- 3D markers



*Markers currently available*

# Graphics test suite

---

- Automatic check: generate PostScript and compare the number of lines with a reference. Not very precise.
- Visual check: More accurate but need more time and manpower.