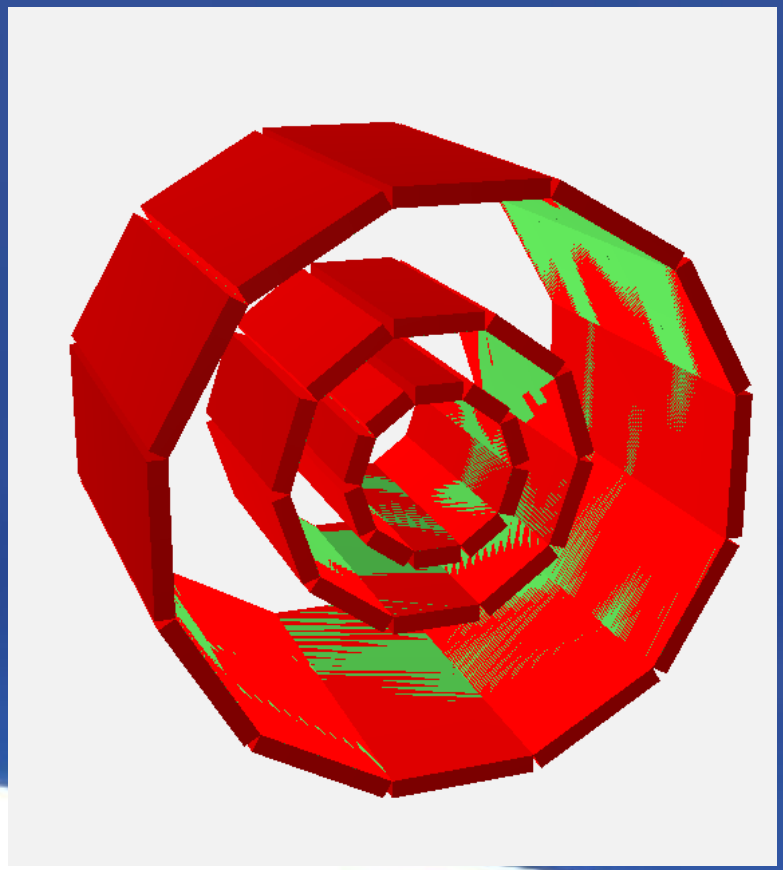
# TGeo geometry for ITS Upgrade

Mario Sitta
Univ. Piemonte Orientale
INFN GC Alessandria

# Summary - The Goal

A flexible geometrical description of the upgraded ITS using TGeo, to exploit the AliRoot framework for reconstruction

- Need “modules”



# Summary - Current Status

- Framework to create geometry with TGeo **DONE**
- Integration in AliRoot **DONE**
- Particle propagation **DONE**
- Hit generation **DONE**
- Segmentation **ONGOING**
- Charge sharing **TO BE DONE**
- Digitization **TO BE DONE**

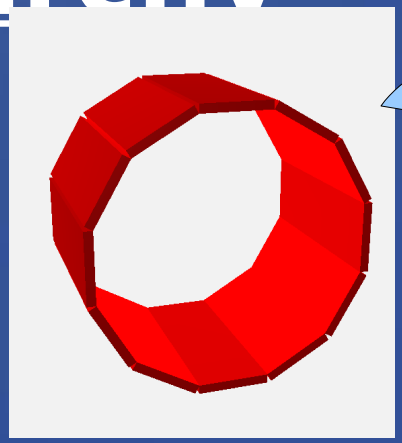
**Ready to be committed**

# Summary - Geometry With TGeo

A possible tool

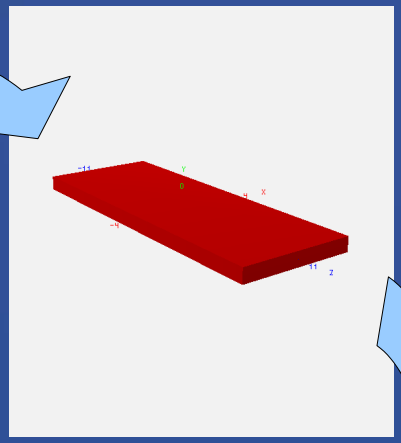
- layers – ladders – modules
  - ITSup contains *k* layers
  - each layer has *m* “ladders” (i.e. linear structures)
  - each ladder has *n* “modules” (basic building block)
  - a module has both sensitive and passive elements
- set parameters with methods (no need to recompile)
- can be used with Root alone (no need for AliRoot)

# Summary - The Implemented Hierarchy



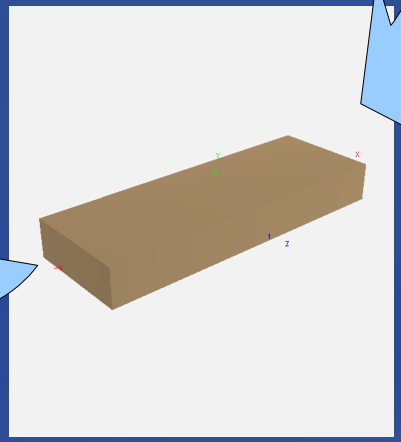
Layer

Air - Insensitive



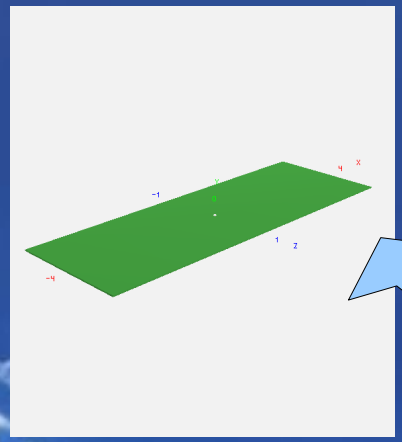
Ladder

Air - Insensitive



Module

Air - Insensitive



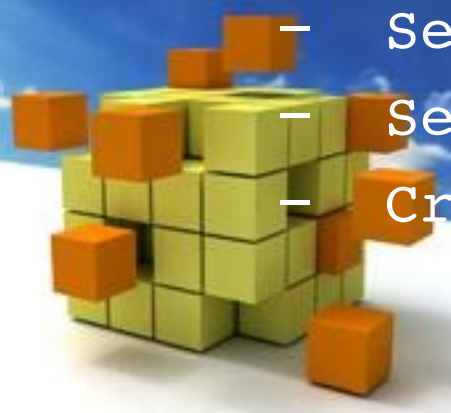
Sensor

Silicon - Sensitive

# Summary - The Basic Geometry Class

Creates a single layer

- Class name: `AliITSv11GeometryUpgrade`
- Currently implemented methods:
  - Constructor sets layer number
  - `SetRadius(Double_t)/GetRadius()`
  - `SetZLength(Double_t)/GetZLength()`
  - `SetNLadders(Int_t)/GetNLadders()`
  - `SetNModules(Int_t)/GetNModules()`
  - `SetLadderThick(Double_t)/GetLadderThick()`
  - `SetSensorThick(Double_t)/GetSensorThick()`
  - `CreateLadder(TGeoVolume*)`



# Summary - Integration Into AliRoot

- A new class: `AliITSvUpgrade` (on the model of `AliITSv11`)
  - can be used inside `Config.C`
  - allows users to fully configure the detector
- Modified classes: `AliITSInitGeometry` (`.h` and `.cxx`) and (marginally) `AliITSgeom` (`.h`)
  - duplicate the methods for geometry initialization and handling

# Summary - Example Of Use

In a standard `Config.C`

```
if (iITS)
  {
    //=================== ITS parameters ============================
    Bool_t isUpgrade = kTRUE;
//      AliITS *ITS = 0x0;
    if(isUpgrade) {
      AliITSvUpgrade *ITS  = new AliITSvUpgrade("ITS","ITS Upgrade",4);
      ITS->DefineLayer(0, 4.0, 14.1, 10, 6);
      ITS->DefineLayer(1, 7.6, 14.1, 10, 6);
      ITS->DefineLayer(2,14.9, 22.2, 12, 8);
      ITS->DefineLayer(3,23.8, 29.7, 16, 8);
    }
    else Ali     *ITS = new AliITSv11Hybrid("ITS","ITS v11Hybrid");

  }
```

Number of layers

Layer parameters

```
ITS->DefineLayer(0, 4.0, 14.1, 10, 6);
```

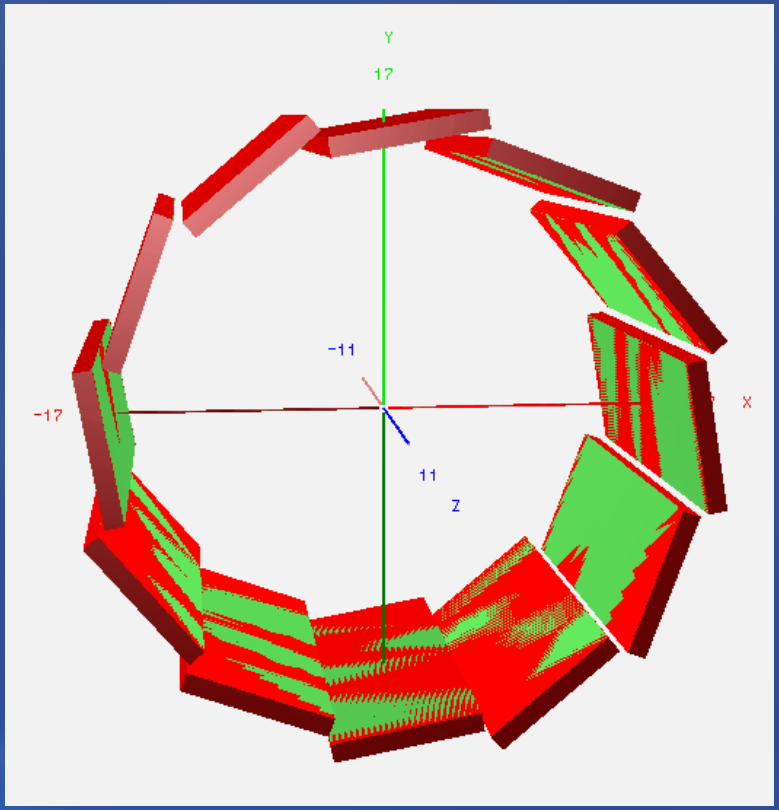
Inner radius

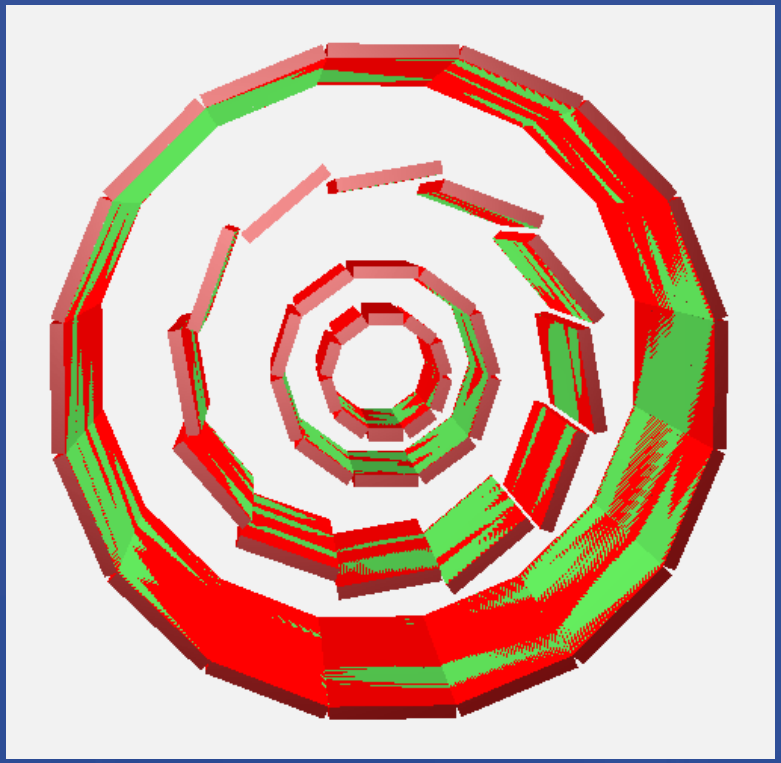
Z length

Number of ladders

Modules per ladder

# New: “Turbo” Layers



# Example Of Use

In a standard `Config.C`

```
if (iITS)
  {
    //=================== ITS parameters ============================
    Bool_t isUpgrade = kTRUE;
//      AliITS *ITS = 0x0;
    if(isUpgrade) {
      AliITSvUpgrade *ITS  = new AliITSvUpgrade("ITS","ITS Upgrade",4);
      ITS->DefineLayer(0, 4.0, 14.1, 10, 6);
      ITS->DefineLayer(1, 7.6, 14.1, 10, 6);
      ITS->DefineLayerTurbo(2,14.9, 22.2, 12, 8, 4.8, 10);
      ITS->DefineLayer(3,23.8, 29.7, 16, 8);
    }
    else Ali   *ITS = new AliITSv11Hybrid("ITS","ITS v11Hybrid");

  }
```

Number of layers

Layer parameters

Width

Tilt angle

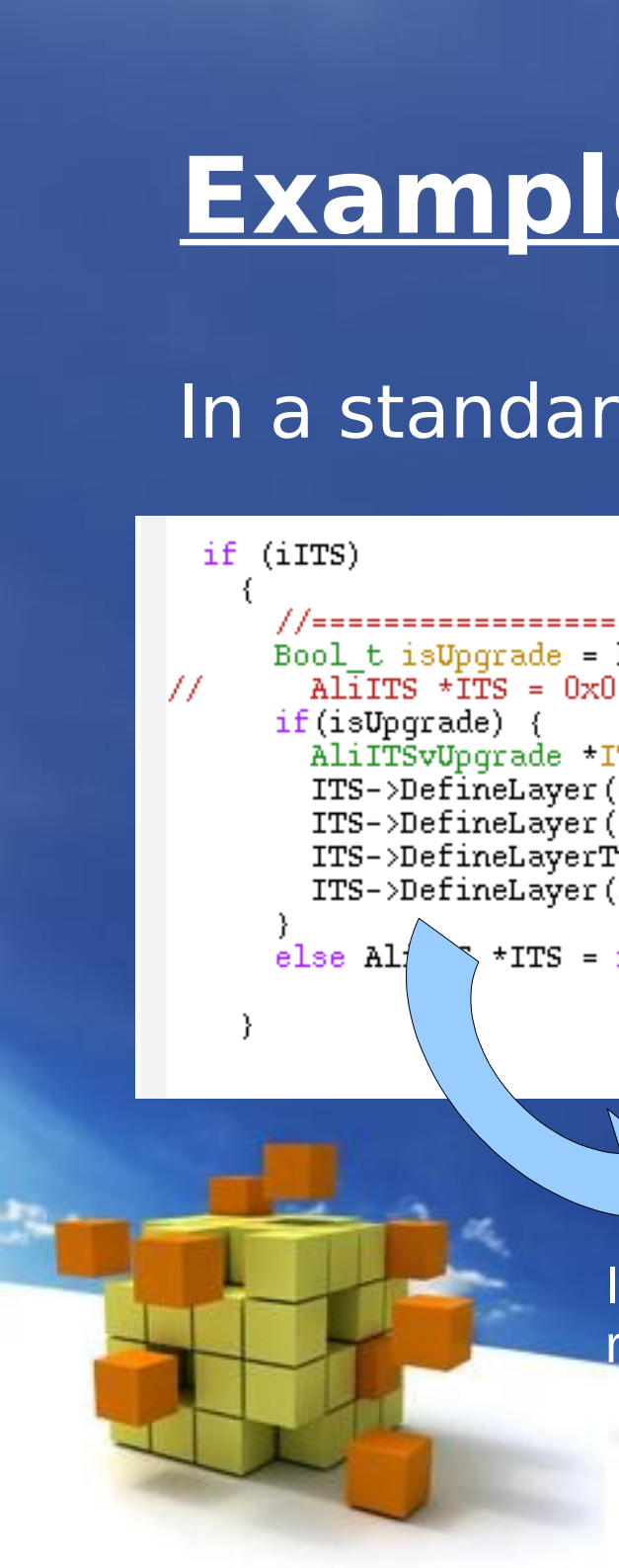
```
ITS->DefineLayerTurbo(2,14.9, 22.2, 12, 8, 4.8, 10);
```

Inner radius

Z length

Number of ladders

Modules per ladder



# Cloned Classes

- Till hits generation: small mods in current `AliITSInitGeometry`:
  - Pros: no need to modify other classes
  - Cons: pollute standard classes
- With digitization: too many classes need to be modified! So better to have “new” (well, cloned) classes
  - Pros: can keep current/upgrade well separated
  - Cons: many duplications needed (starting from “master” class)

# New/Cloned Classes

Till now

- `AliITSUpg` (from `AliITS`)
  - Calls `AliITSLoader`, `AliITSDetTypeSym`, `AliITSInitGeometry`
- `AliITSInitGeometryUpg` (from `AliITSInitGeometry`)
  - Calls to new geometry (no hardcode)
- `AliITSLoaderUpg` (from `AliITSLoader`)
  - Must use new `AliITSInitGeometryUpg` and new digits
- `AliITSDetTypeSymUpg` (from `AliITSDetTypeSim`)
  - Must not use SPD/SDD/SSD, instead new detector type, avoid hardcode
- `AliITSsegmentationUpgDet` , `AliITSsimulationUpgDet`
  - Define characteristics of the new detector

# To Be Done

- Geometry
  - Add passive layers
- Digitization
  - PreDigits, SDigits **ALMOST DONE**
  - Charge sharing
  - Digits
- General code
  - Try to simplify class calls
- Independent tests are more than welcome!

# Backups