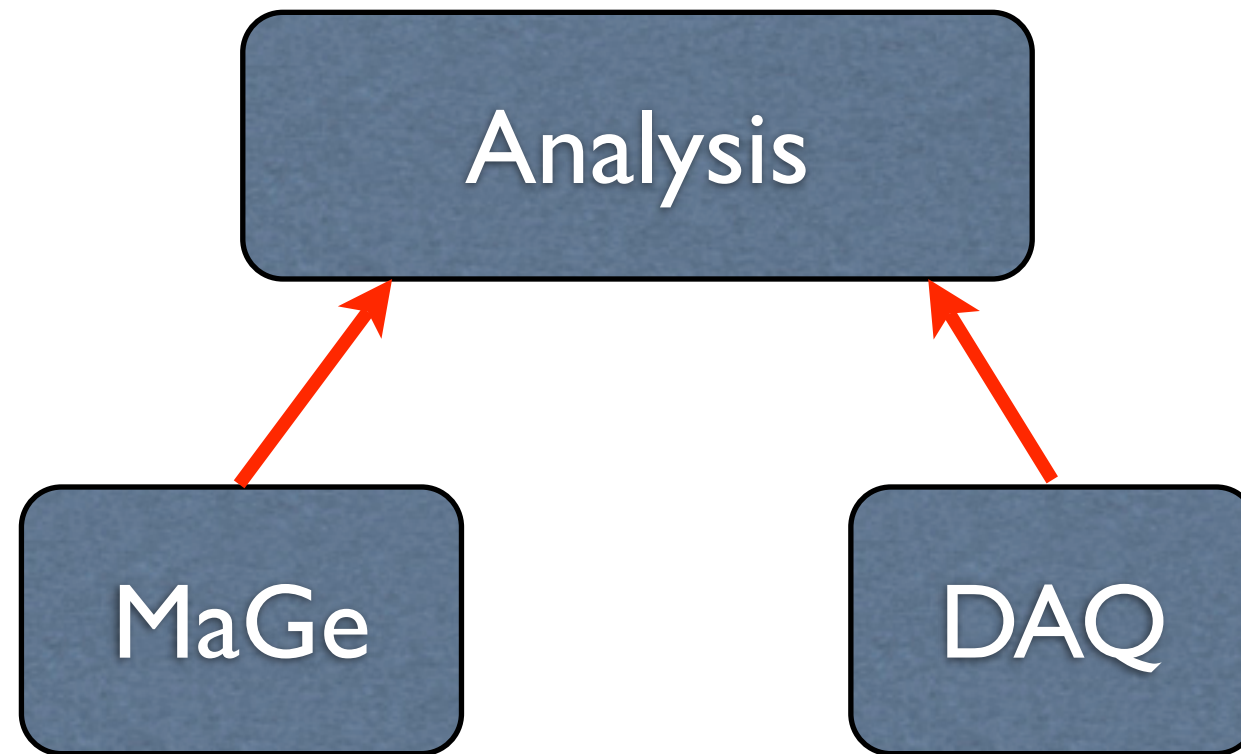


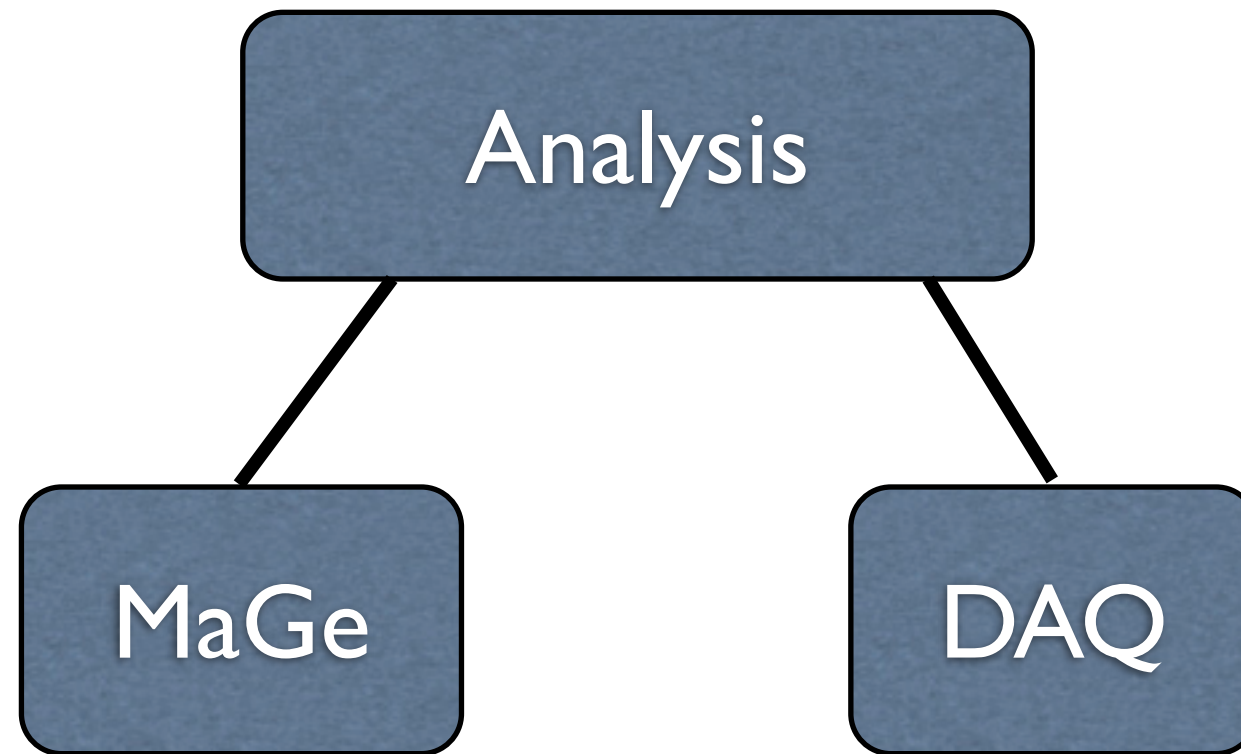
# MGDO: MAJORANA-GERDA Data Objects

Jason Detwiler  
MaGe Workshop 2010  
18-20 January, Munich

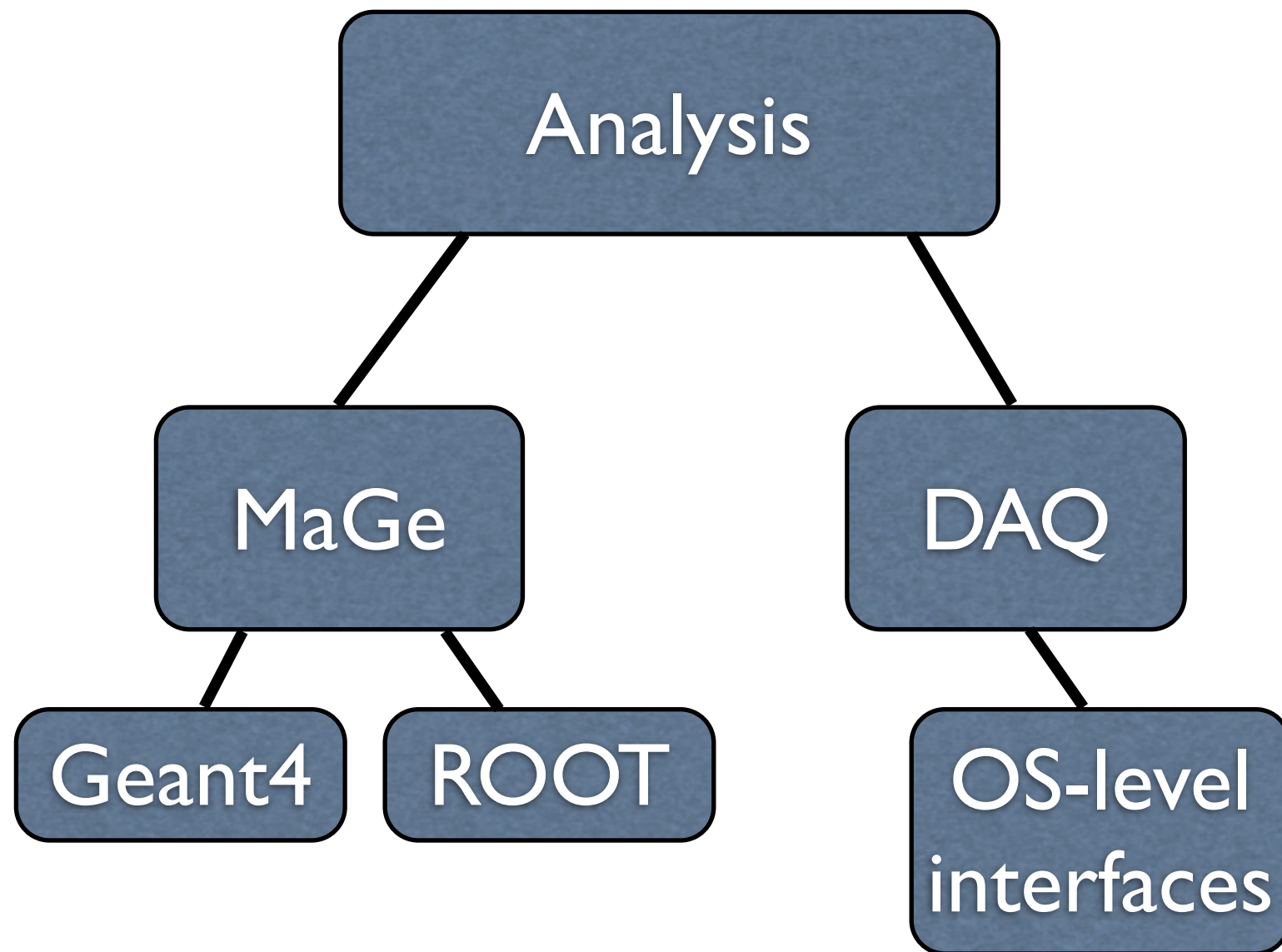
# Data Flow



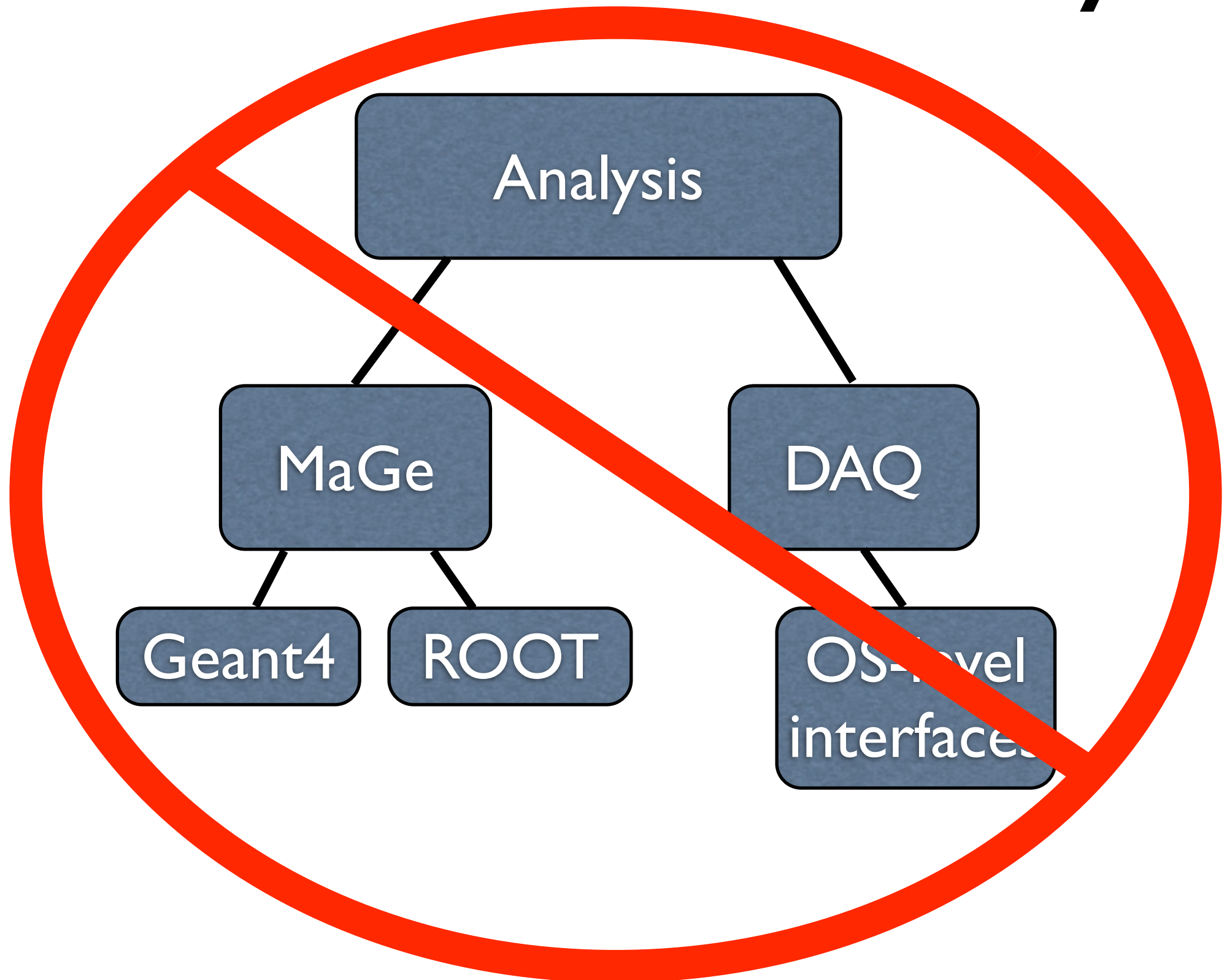
# Software Hierarchy



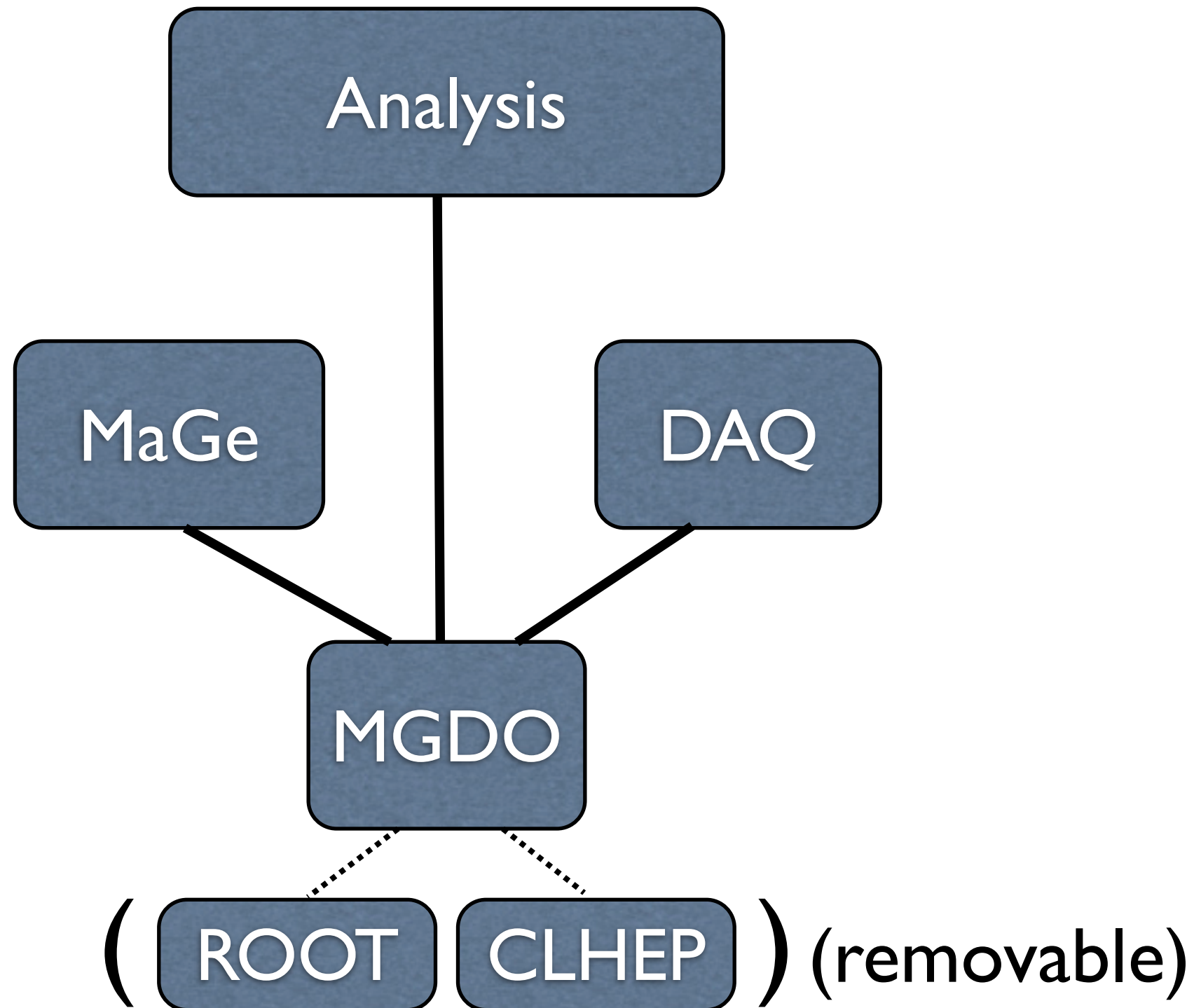
# Software Hierarchy



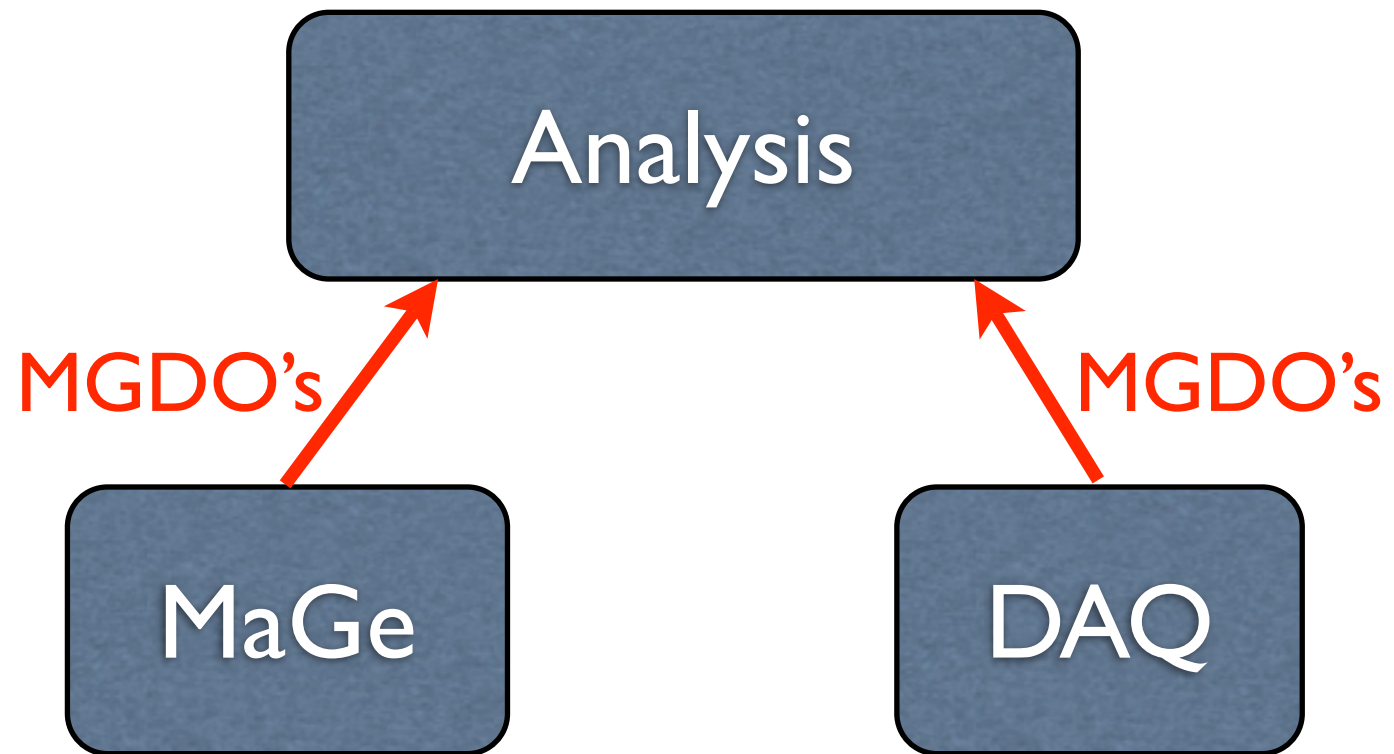
# Software Hierarchy



# Software Hierarchy



# Data Flow



# MGDO Packages

- **Base**: the basic data objects (waveforms, DFTs, digitizer data, crystal information, ...)
- **Root**: ROOT wrappers for DO's in Base
- **Gerda**: GERDA-specific data objects
- **Majorana**: MAJORANA-specific data objects
- **Transforms**: operations on waveforms. Put them here so that MaGe and the analysis code can share the same DSPs



# Conventions

- Use CLHEP system of units and physical constants (plus a few more units, e.g. GHz)
- Standard error / warning reporting:  
MGDOerr << “error here” << endl;  
Output: “Error at file.cc:28 - error here”  
MGDOwarn << “warning!!!” << endl;
- ROOT/Taligent coding conventions
- autoconf + standardized make procedure  
(via files in MGDO/buildTools directory)

# MGWaveform

- Data Members

- `vector<double> fData`: the waveform
- `double fSampFreq`: sampling frequency in CLHEP units
- `double fTOffset`: (global) time of sample 0
- `EWFFType fWFFType`: kCharge, kCurrent, kADC, or kNoType
- `int fID`: ID of crystal / contact that generated the waveform

- Important concepts

- `Similarity`: same `fSampFreq`, `fTOffset`, `fWFFType`
- `MGWaveformRegion`: for limiting operations to a region of a waveform
- `Interpolation`: endpoints can be flat, zero, or interpolated

# MGWaveform

- MGTWaveform

- **THID\* GimmeHist()**: return a THID\* with bin centers set to waveform samples. The THID is named according to the waveform ID. Allocates a new THID if necessary.
- **THID\* GimmeUniqueHist()**: same as GimmeHist(), but always returns a (new) unique histogram
- **TFI\* GetFunction()**: return a TFI\* representing the waveform

- Waveform Transforms

- **void Transform(&wfIn, (&wfOut))**: apply a transformation to the waveform, optionally storing the transformation in wfOut (leaving wfIn untouched)
- See the next talk by Marino

# MGWaveformFT

- Discrete Fourier Transform of a MGWaveform
- FFT performed by  
MGVWFFastFourierTransform  
(MGWFFastFourierTransformDefault or  
MGWFFastFourierTransformFFTW)
- MGWaveformFT's have their own
  - Transforms
  - ROOT wrapper (MGTWaveformFTs)

# MGVDigitizerData

- Data Members

- **double fEnergy**: the digitizer's estimate of the event energy
- **double fTimeStamp**: the timestamp of the event (clock tick, trigger time, etc)
- **int fID**: ID of the crate / card / channel of the digitizer (note: distinct from the crystal ID)

- Important concepts

- This class is pure virtual; derived classes must be defined specific to particular piece of hardware. Example:  
MGDO/Majorana/MJGretina4DigitizerData.hh
- A default packaging of crate / card / channel into fID is provided in the base class but may be overloaded
- fTimeStamp may correspond to the time of sample 0 (MGWaveform::fTOffset) or may be some other trigger-related time

# MGCrystalData

- Shape Information (inner / outer radius, height, well depth)
- Dead layer info (geometry/depth)
- Segmentation info (number / type heights of segments)
- Also: passivation depth, impurity gradient, crystal axes, voltage, IDs, diode type, global position
- Possibly should be in a database eventually?
- MGCrystalDataContainer: a vector of MGCrystalData's

# Other Data Objects

- MGElectricField
- MGTEvent: see later talk (by me)
- Possible future objects:
  - Hit pattern object (detector level, segment level)
  - Single / multi-site tag
  - PSA object(s) (parameter list?)
  - Veto information
  - MC information (see A. Schubert's talk)

# Using MGDO

- ROOT interactive session:

```
.x $MGDODIR/Root/LoadMGDOClasses.C
MGTWaveform w(100)
for(int i=0; i<100; i++) w.SetValue(i, gRandom->Gaus())
TH1D* h = w.GimmeUniqueHist()
h->Draw("L")
MGWFSmoothing smoother(10)
MGTWaveform wSmooth
smoother.Transform(&w, &wSmooth)
TH1D* hSmooth = wSmooth.GimmeUniqueHist()
hSmooth->SetLineColor(kRed)
hSmooth->Draw("L SAME")
```



# Using MGDO

- **ROOT macros: add**

```
gROOT->ProcessLine( ".x $MGDODIR/Root/LoadMGDOClasses.C" );
```

**or**

```
gROOT->ProcessLine( ".x $MGDODIR/Majorana/LoadMGDOMJClasses.C" );
```

**to the top of your macro file**

- **Compiled code: see**

[http://mjwiki.npl.washington.edu/bin/view/MaGe/MGDO#Example\\_Makefile\\_and\\_Application](http://mjwiki.npl.washington.edu/bin/view/MaGe/MGDO#Example_Makefile_and_Application)