



# ACORDE OFFLINE STATUS

April 7<sup>th</sup> 2008  
Offline Week

Presented by:

Mario Rodríguez Cahuantzi (FCFM-BUAP)



# Outline

## Status of:

✓ Digitalization (August 2007)

✓ Geometry

✗ Reconstruction

✓ Raw Data Reading

✗ ConvertDigits

✗ FillESD

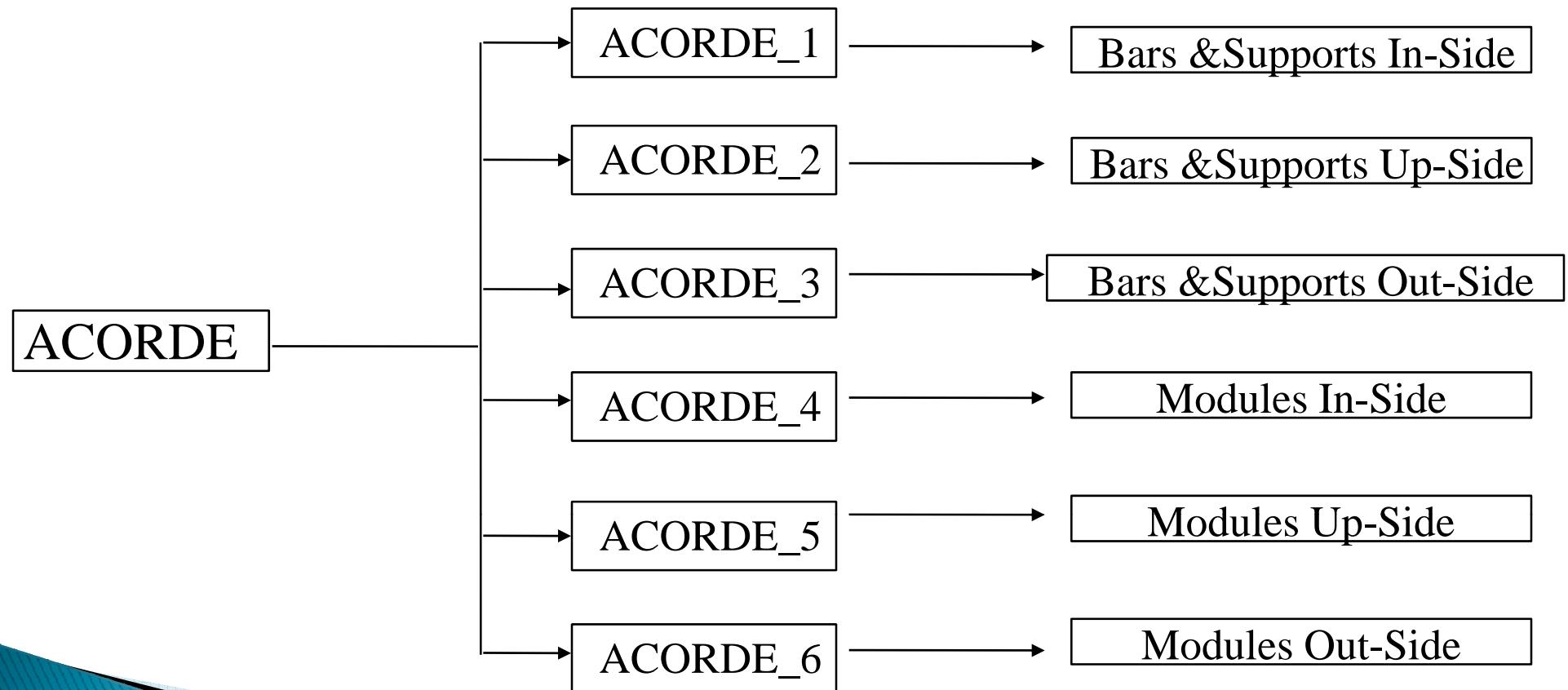
✗ Calibration

✗ Alignment



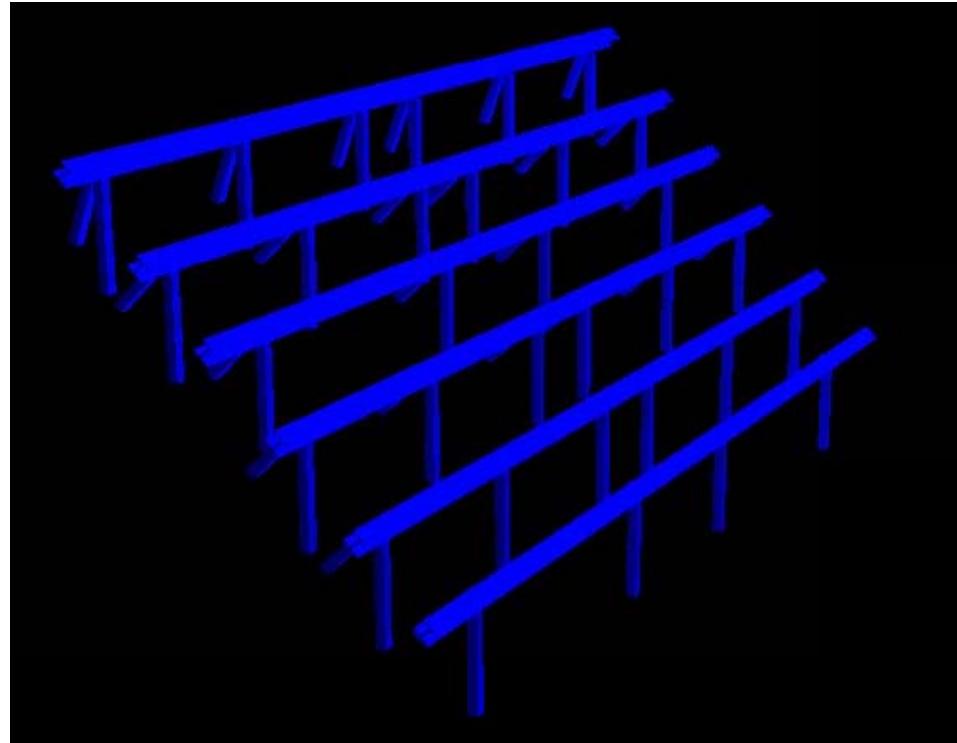
# Geometry of Acorde (AliACORDEv1)

Gometry described by **TGeo Objects**. It is implemented on six parts:





# ACORDE bars and supports (In-Side)



In-Side

ACORDE\_1:

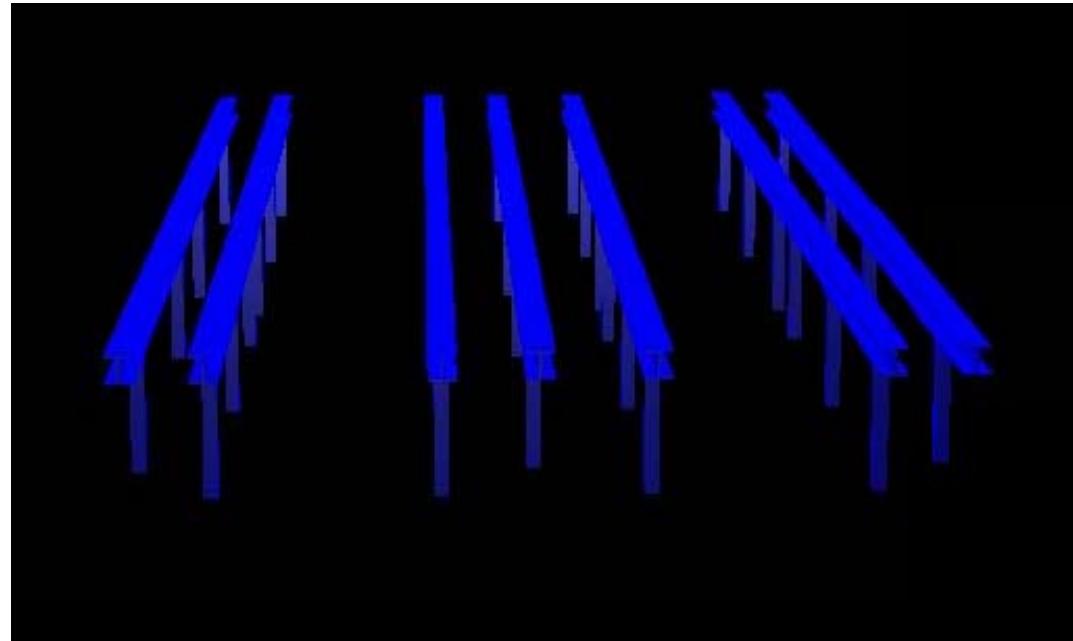
ACORDE7

ACORDE7\_1

ACORDE\_SUPPORT



# ACORDE bars and supports (Up-side)



Up-Side

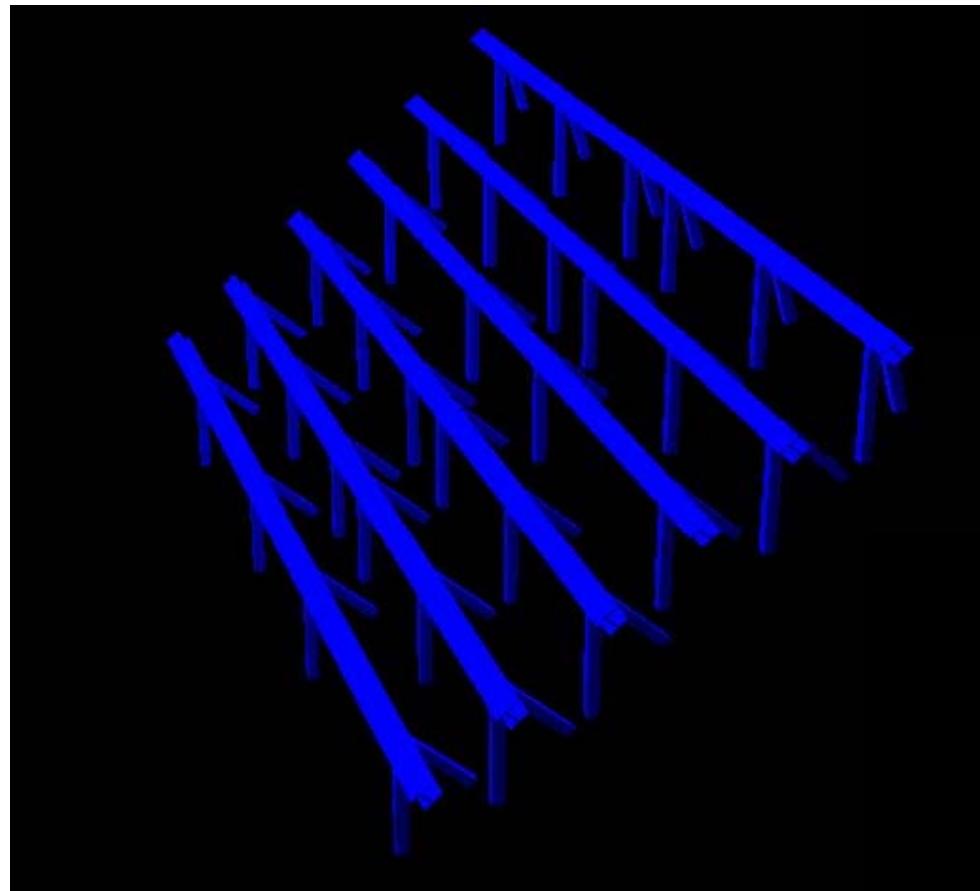
ACORDE\_2:

ACORDE00

ACORDE\_SUPPORT



# ACORDE bars and supports (Out-side)



Out-Side

ACORDE\_3:

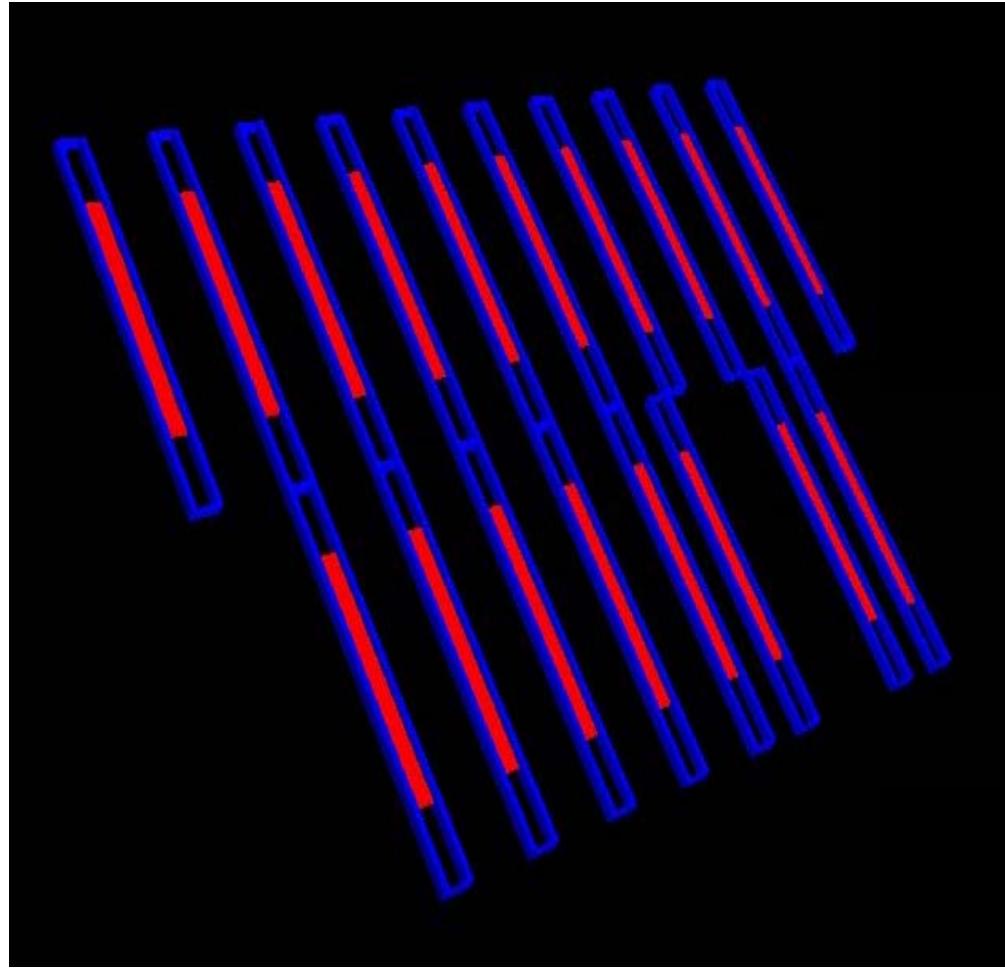
ACORDE7

ACORDE7\_1

ACORDE\_SUPPORT



# ACORDE Modules (In-Side)



In-Side

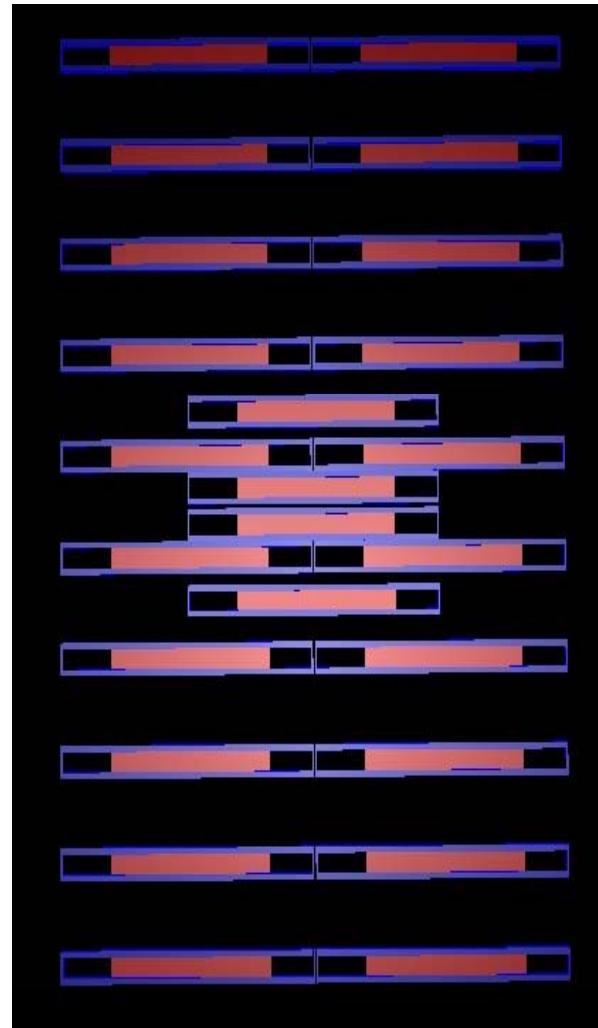
ACORDE\_4:

ACORDE\_MODULE

**18 modules**



# ACORDE Modules (Up-Side)



Up-Side

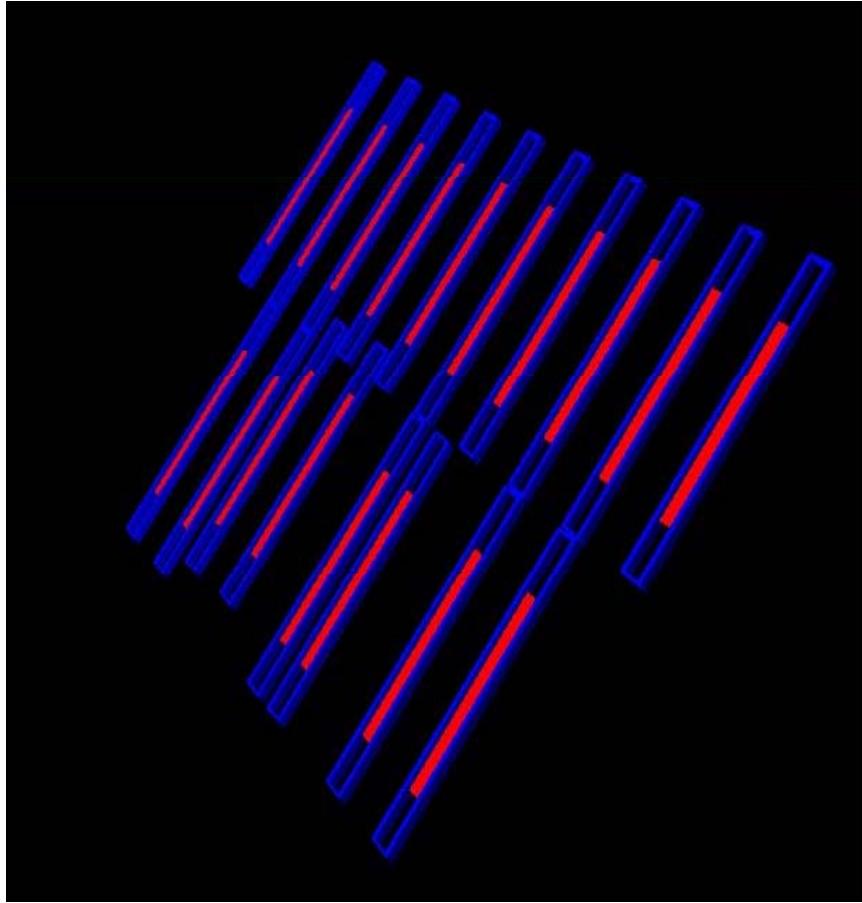
ACORDE\_5:

ACORDE\_MODULE

**24 modules**



# ACORDE Modules (Out-Side)



Out-Side

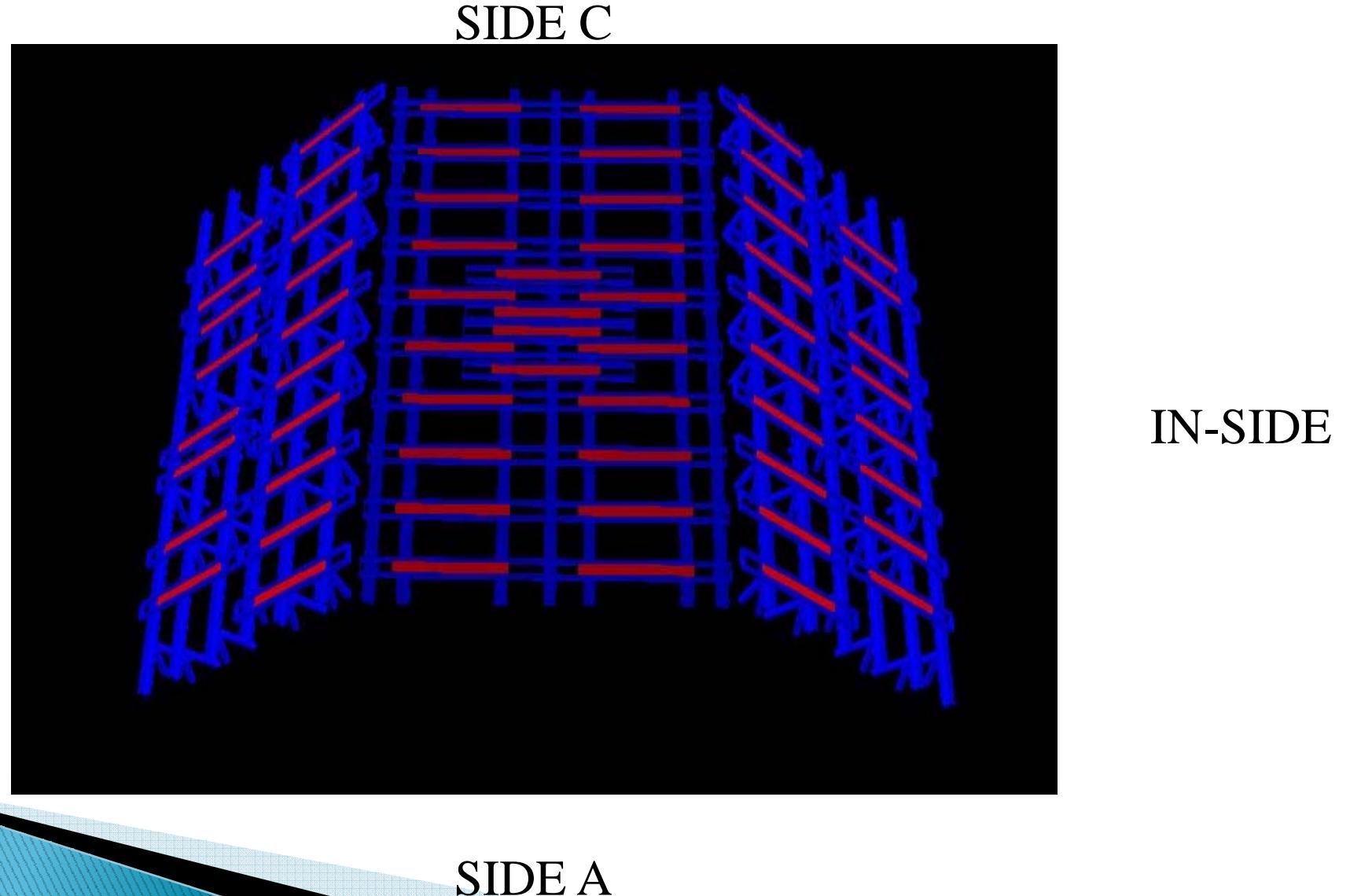
ACORDE\_6:

ACORDE\_MODULE

**18 modules**



# Full ACORDE geometry: Supports, bars and modules (**60 modules**)





# Reconstruction

ACORDE works as a trigger detector, consequently the reconstruction means to read the single muons and multicoincidence signal distributions.

The macro AcoReco.C was created to fill Histograms with the Acorde's BitPattern information.

AcoReco.C implements the needed Reconstruction for ACORDE.

Used CDB for parameters:

Procedure:

“decoding data” of AcoReco → ConvertDigits

store ESDs → ConvertDigits



# Acorde Raw Data (AliACORDERawStream)

## FOUR WORDS OF 32 BITS CONSIDERED

word[0] --> Modules 1 to 30 --> bits 31,32='00' --> **0000.....0**

word[1] --> Modules 31 to 60 --> bits 31,32='01' --> **0100.....0**

word[2] --> Modules 1 to 30 --> bits 31,32='10' --> **1000.....0**

word[3] --> Modules 31 to 60 --> bits 31,32='11' --> **1100.....0**



# Reconstruction: Procedure

AcoReco ----> based on AliACORDERawStream

P.E: **00000000000010000000100000000001-->dyWord[0]**

AliACORDERawStream\* rawStream = new AliACORDERawStream(rawReader) .

.....

dyWord[0]=rawStream->GetWord(0); --> tmpDy

dyWord[1]=rawStream->GetWord(1); --> tmpDy

dyWord[2]=rawStream->GetWord(2); --> tmpDy

dyWord[3]=rawStream->GetWord(3); --> tmpDy

.....

.....

loop over the first 30 bits of each word

{

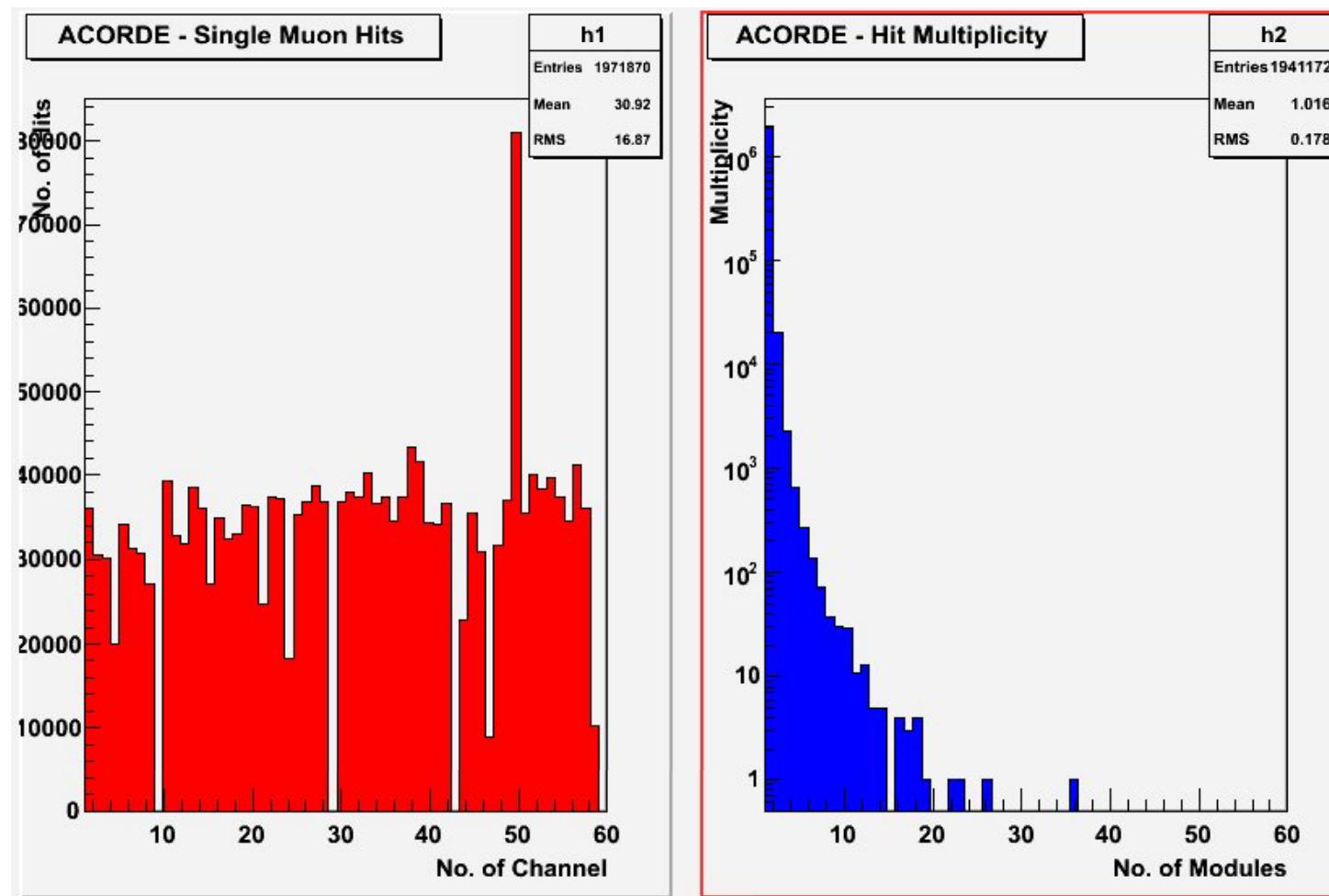
kroSingle[r] = tmpDy & 1; (or kroMulti[r] = ...)

tmpDy>>=1;

}



# Real data: single muon hits and multiplicity





# Calibration

The calibration for ACORDE is the monitoring of the rate per single module. We will not expect a big change in all of them.

Details should be presented on Friday session .



# Alignment

To take into account possible variation of the ideal geometry (during simulation & reconstruction) and then correctly associate signals to position in space (for several Alice-subdetectors).

## **MakeACORDEFullMisAlignment.C :**

- **Load Acorde's geometry.**
- **Produce a transformation accordingly to the user.**
- **Produce new geometry .**
- **Once we have created a root file with misalignment objects, we need to simulate and verify the changed geometry.**
- **Effect on measurements from the reconstruction.**



# Summary

- The geometry is finished: **committed to svn**.
- Reconstruction: We decode the data (bit pattern) and store the information in a standar output format (root files): **to implement AliESDACORDE class**.
- Raw data Reading: Finished. **committed to svn**.
- Calibration: done and **not committed to svn**