Possible alignment strategies

General requirements for alignment

- To have well-separated procedures but common format between survey-alignment and alignment with tracks.
- Survey-alignment should not only be the starting point for later alignment procedures but also should constrain them to its output ± its precision
- To have a uniform basic approach for all the detectors (e.g. format of geometry objects, how to access them), diverging only when it comes to detector specific issues (e.g. which parameters to consider alignable).
- To try to use only one format of geometry objects in all the alignment stages, in all simulation and reconstruction and when interfacing with the Conditions DataBase

Tasks for alignment (1)

Framework tasks:

- Allow storing and retrieving of the geometry
- Provide common tools (e.g. virtual methods to query the geometrical modeler or to store detector parameters, capability to align the geometry according to given alignment data)
- Allow consistent treatment of the geometry and of its updating (e.g.proper treatment of overlaps)
- Provide alignment procedures
  - Some will be detector specific
  - Others will be used by more detectors, possibly in a combined way

Tasks for alignment (2)

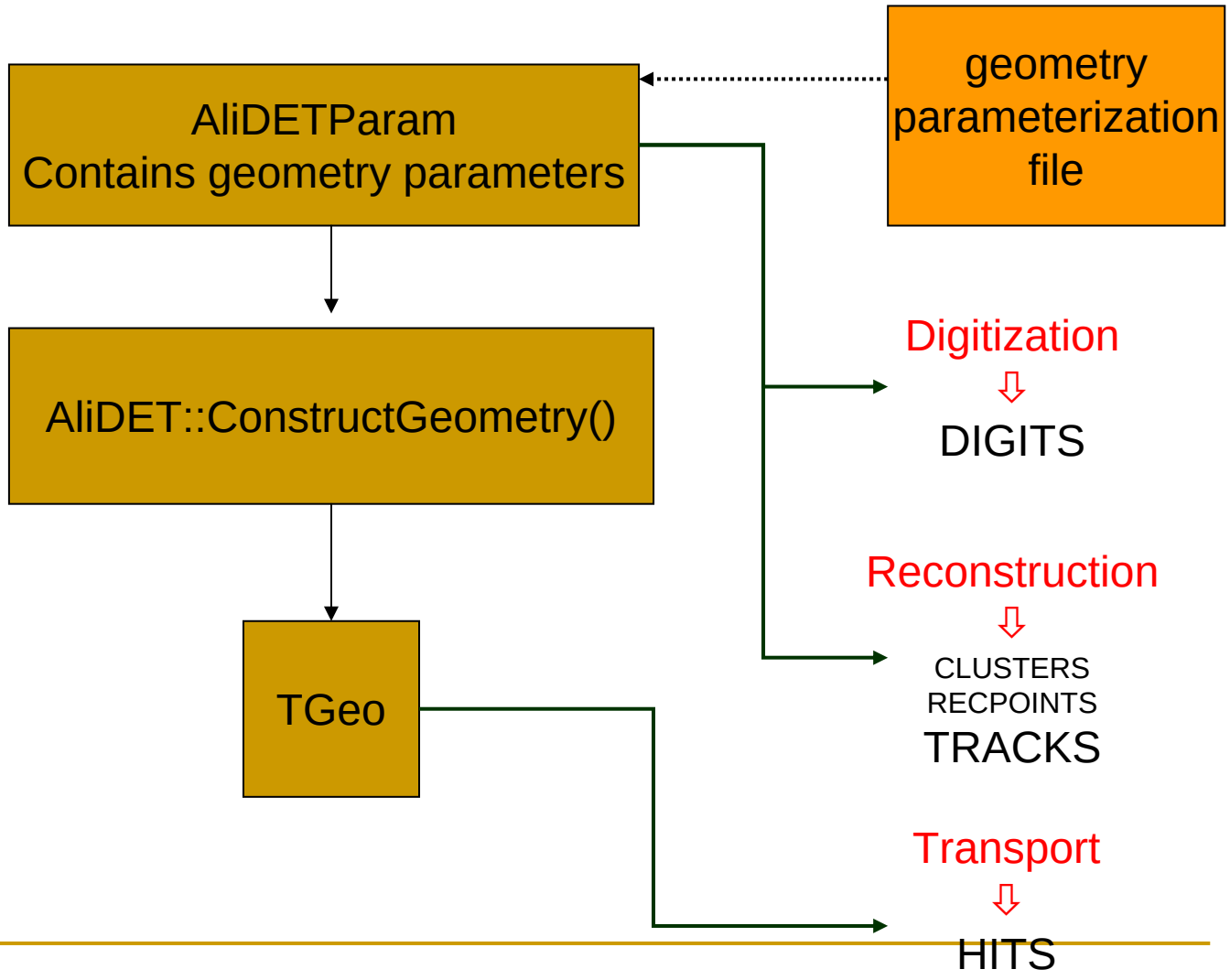
Detector's specific tasks:

- Which alignment information: convert detector expert's knowledge of alignable and assembly volumes into code ⇨ this is easier with TGeo
- How: clarify the type of alignable objects needed
- Substitute parametrized volume positioning with „volume-by-volume" positioning in the ConstructGeometry methods
- Enable digitization and reconstruction to use „volume-by-volume" information wherever needed ⇨ queries to TGeo

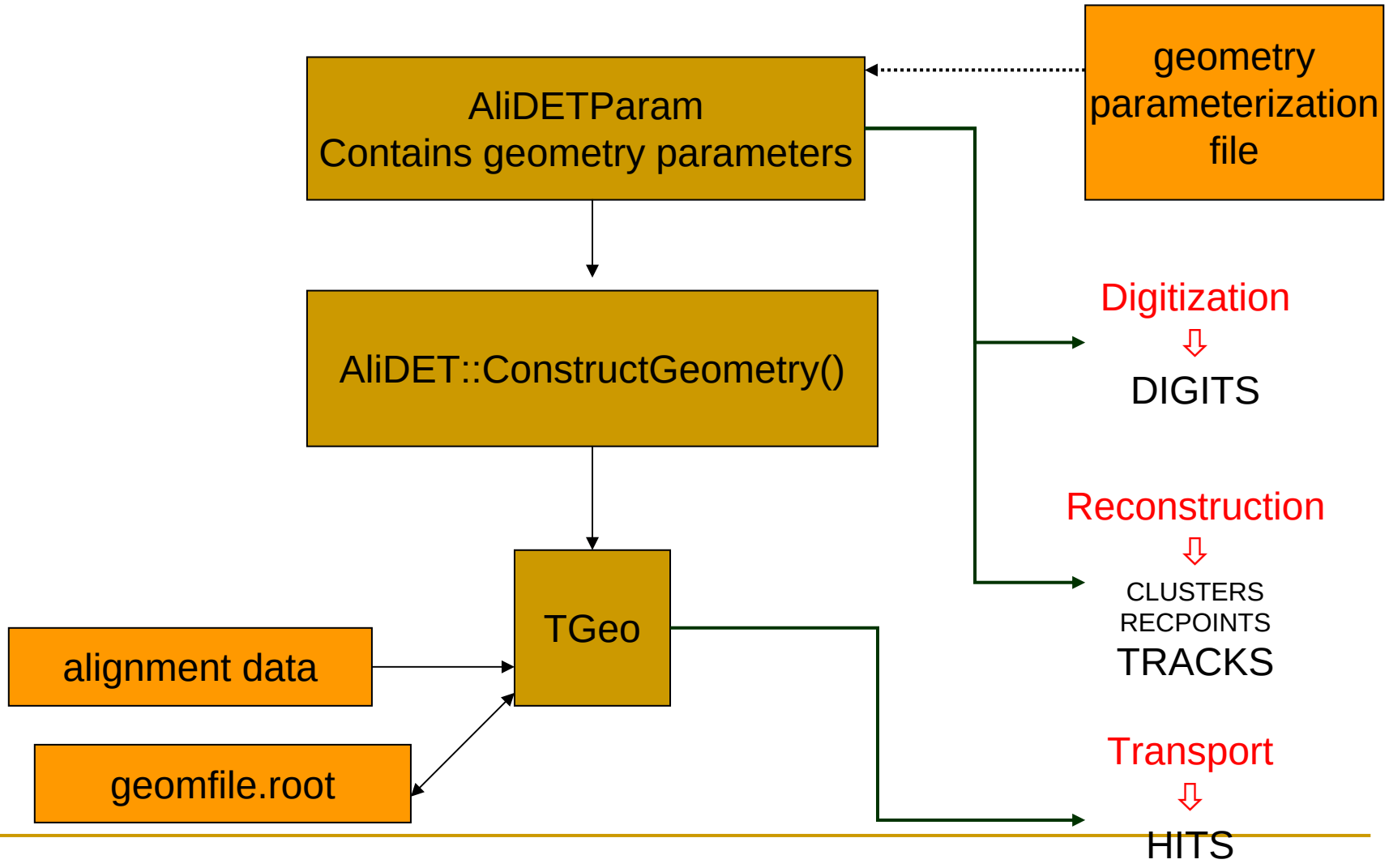
AliDETParam

- Build Ali(TPC, ITS, RICH,...)Param classes storing parameters where they are sharply devided between fixed and alignable ones. Alignable parameters have to be automatically updated by queries to TGeo if the geometry used is not the hard-coded one. These classes will derive from AliDETParam STEER-class.
- In my idea the virtual AliDETParam class should be almost empty, but provide a framework to separate fixed-alignable parameters and calibration-alignment parameters: suggestions and requirements for this class have to arrive as soon as possible to be effective.

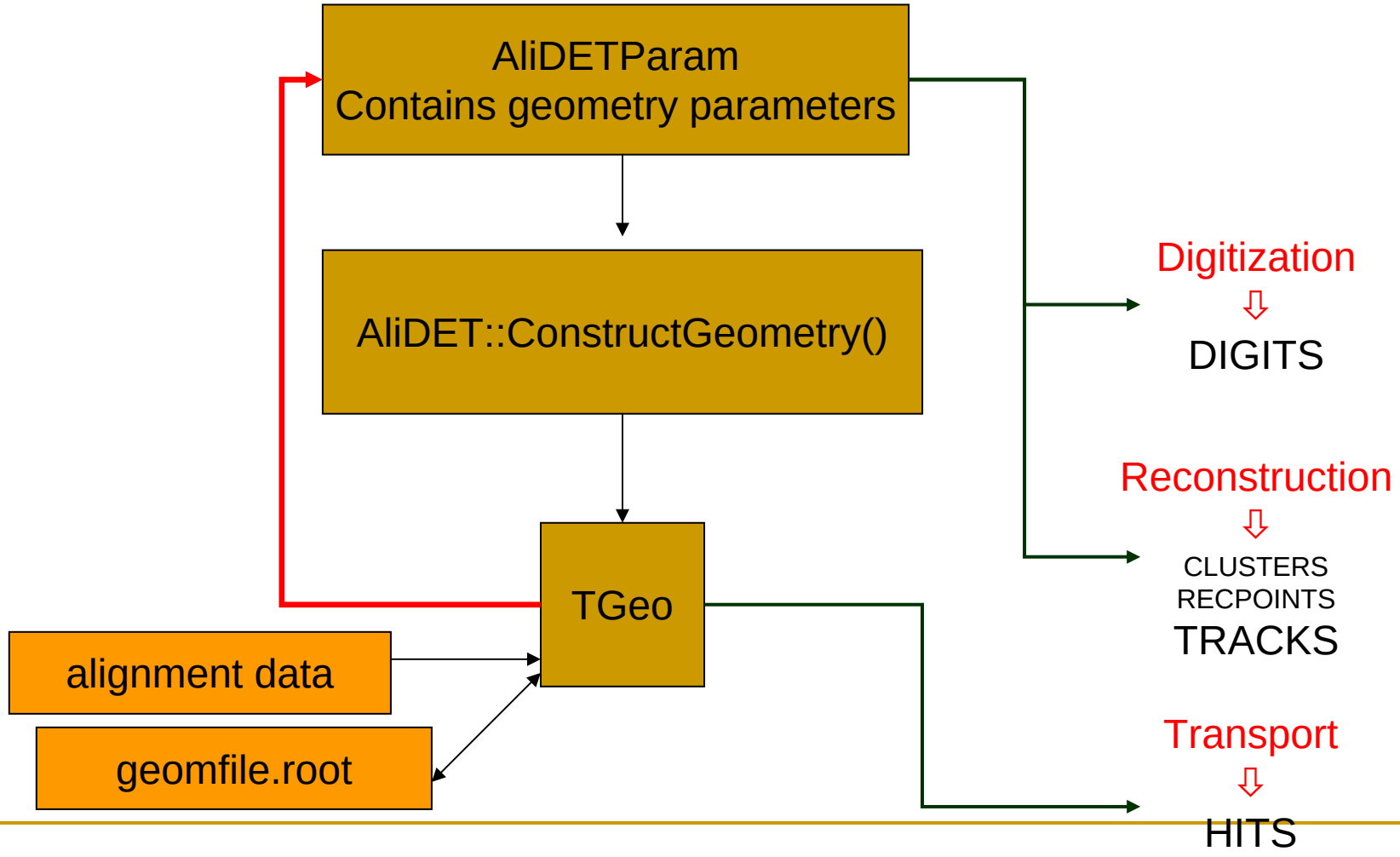
Geometry status: a few months ago



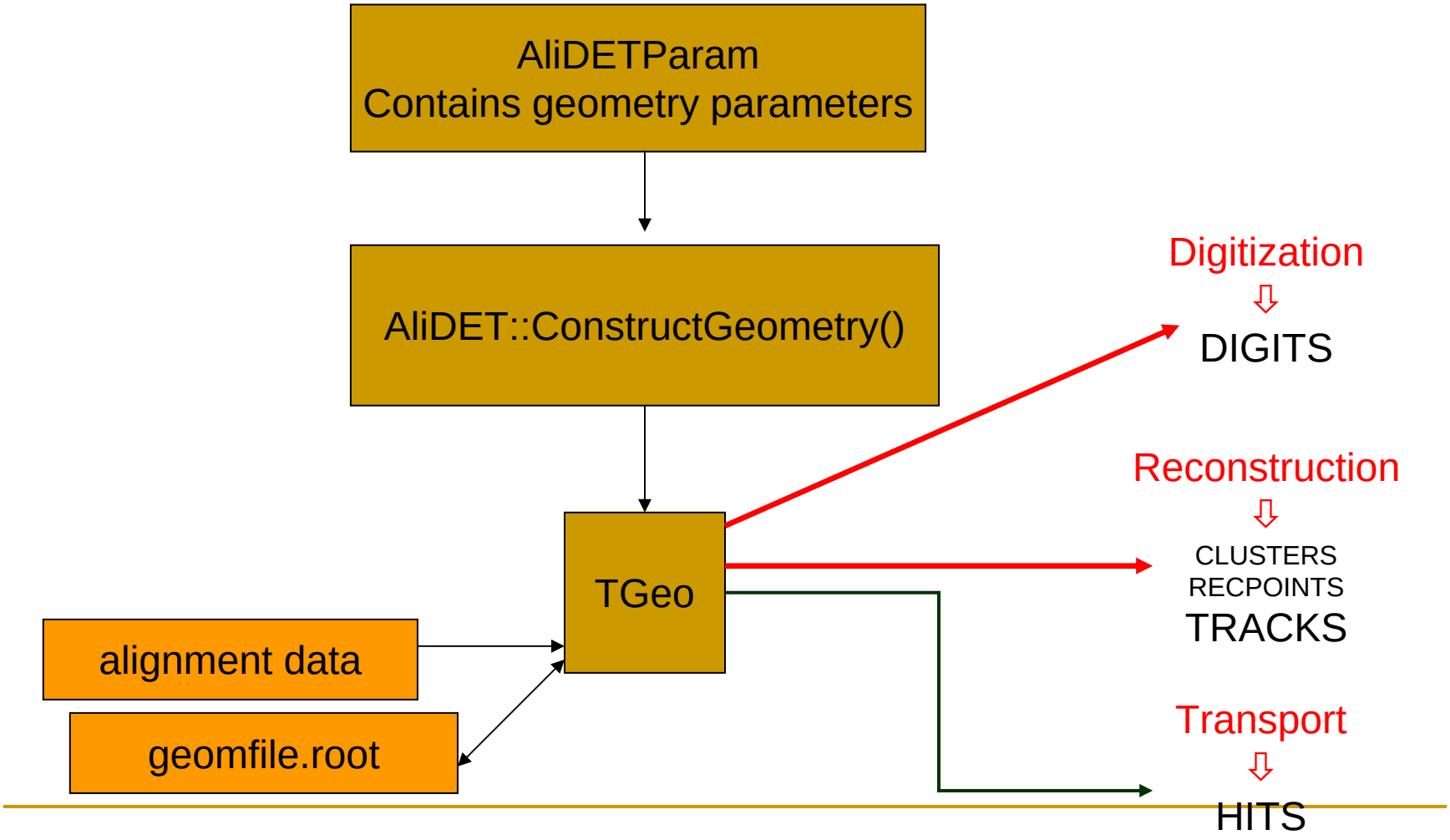
Geometry status: now



Alignment strategy: 1



Alignment strategy: 2



Short-term aim

- Simulation has to be aware of the geometry: ideal and aligned geometries have to result in different and consistent digits.
- Reconstruction has to be aware of the geometry:
  - reconstructing from same digits but different geometries has to result in different clusters and tracks
  - reconstructing with the 4 possible combinations of digits and geometries has to give better tracks in the cases:

    1) ideal digits + ideal geom

    2) aligned digits + aligned geom

Not-so-short-term aim

- Development of efficient alignment procedures

Examples up to now

- TPC is developing and testing alignment in digitization (Cvetan) and an alignment procedure minimizing residuals (Marian)
- This has allowed to point us to some possible problems arinsing in this step (see next presentation)
- Some other problems will be experienced in a detector-specific fashion, however those common problems should be solved and addressed in a similar manner

Open issue 1: which approach

- „Convolution“ or „addition“ between ideal geometry and geometry perturbation (cf. Previous Andrei‘s presentation): i.e. applying geometry patch to the geometry file or before creating it.
- Pros: we can hard-code alignability and alignment precision
- Contra:
  - heavier to develop
  - insecure reference to geometry because geometry is again hardcoded

Proposal of „theoric“ answer

- Approach 1 for application of dynamic perturbations (from tracking-based procedures)
- Approach 2 for application of static perturbations (from survey)

Open issue 2: Geometry objects

- Which geometry objects do we save; the Conditions DB stores objects deriving from TObject, e.g. storable objects are TClonesArrays of TGeoMatrixes
- Do we save „displacement“ objects (positions w.r.t. ideal position of the same object) or „position“ objects (positions w.r.t. container volume as in the ConstructGeometry methods)

Open issue 3: which reconstruction is sensitive to alignment

- After alignment can part of the reconstruction chain be skipped?
- In particular: is the reconstruction inside sensitive volumes (up to clusters/recpoints) indipendent of the position of the sensitive volume itself?
- For the detectors for which this approximation is good, moving recpoints instead of moving digits could imply an important (unavoidable) time saving (e.g. in TPC)

Conclusions

- Many practical things and even general strategy is open, but still there is much then detectors presently can (and should) do.
- This work will drive the alignment strategy