LHCb Detector Description and Alignment

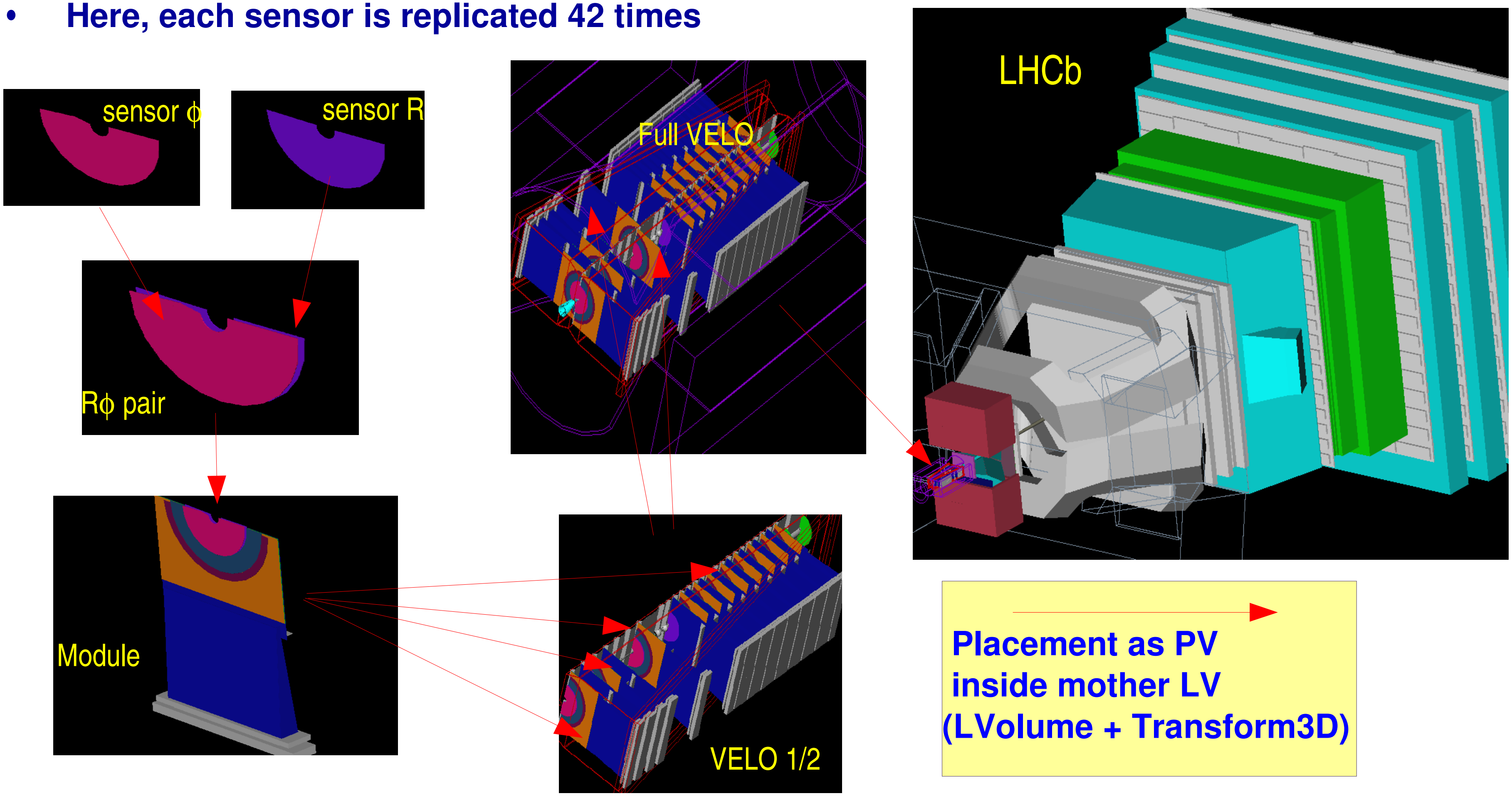
- Overview of LHCb detector description
- Use and handling of misalignment information
- Conclusions

LHCb Detector Description overview

- **Two hierarchical structures**
  - Geometrical tree
    - Re-usable blocks of geometry description: ***Logical Volumes*** or LVs
      - Volumes with shape, material, surface properties...
      - Hierarchy from positioning of LVs within LVs (***Placed Volumes*** or PVs)
  - Logical tree (AKA Detector structure)
    - Coupled to physical structure of LHCb
    - Hierarchy of "interesting" ***detector elements***
    - One-to-one correspondence to "interesting" components of real detector
    - Handle very sophisticated information

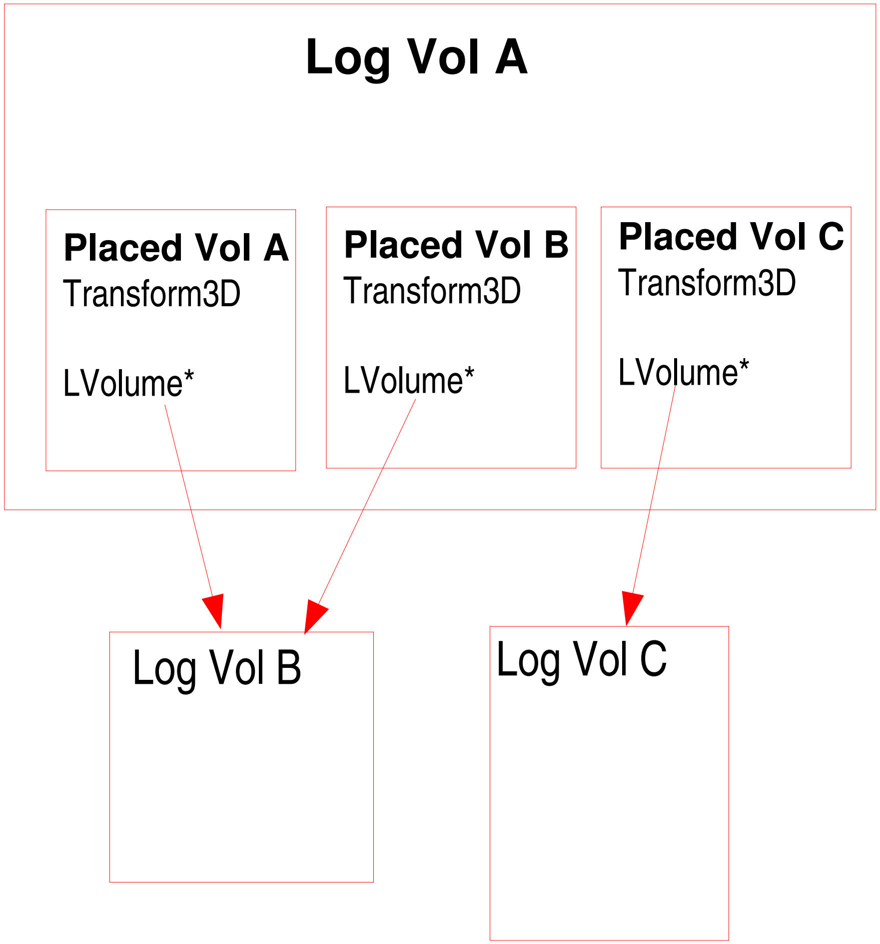
Geometrical hierarchy illustration

- Geometrical hierarchy elements (LVs) can be seen as replicable generic components that can be used many times
- Here, each sensor is replicated 42 times



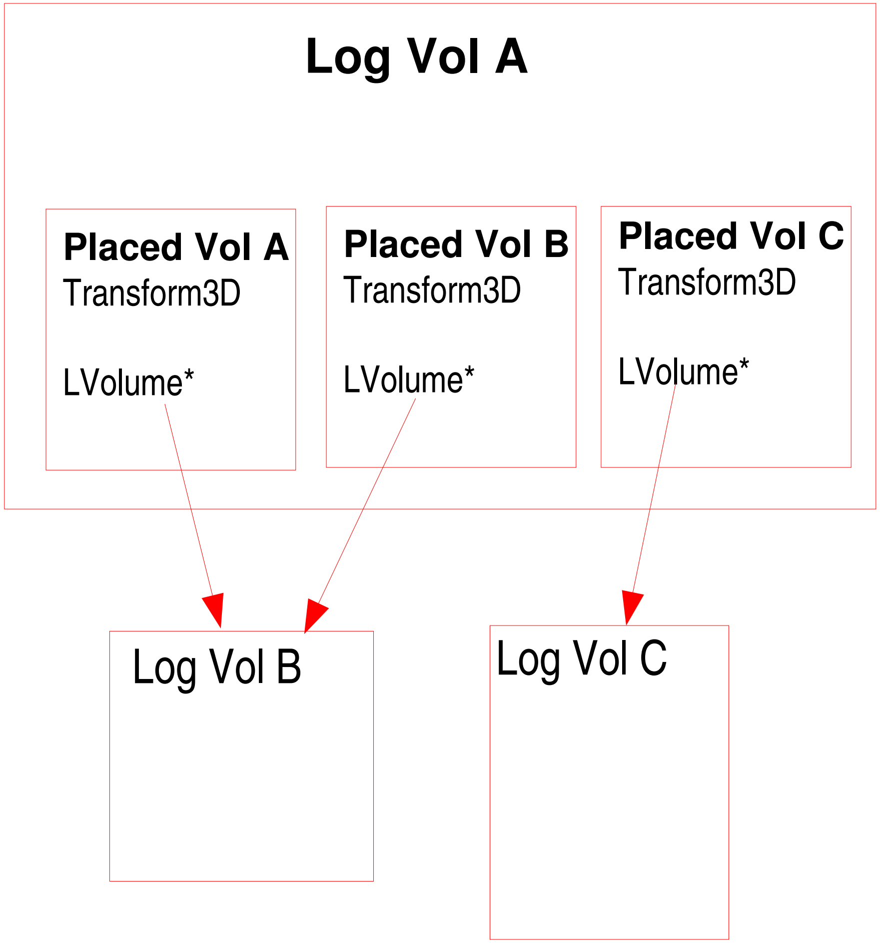
Geometrical hierarchy: Logical Volumes

- **Defines geometrical shape, material, optical properties of surfaces**
- **Hierarchy from placement inside other LVs:**
  - Logical volume + placement via 3D transformation
  - One logical volume can be placed an arbitrary number of times, within different Lvs
  - Many placements coupling same logical volume to parent(s) via transformation matrices
  - Logical volume only occurs once in memory



Geometrical hierarchy: Logical Volumes

- **Knows of LVs (daughters) placed within it**
- **Only link in hierarchy is from parents to children**
  - Parent fixes reference frame of children
  - For children, parents don't exist!
  - This is essential for replicability
- **All computations performed in the frame of reference of the parent**
  - For placed volumes, there is only one frame

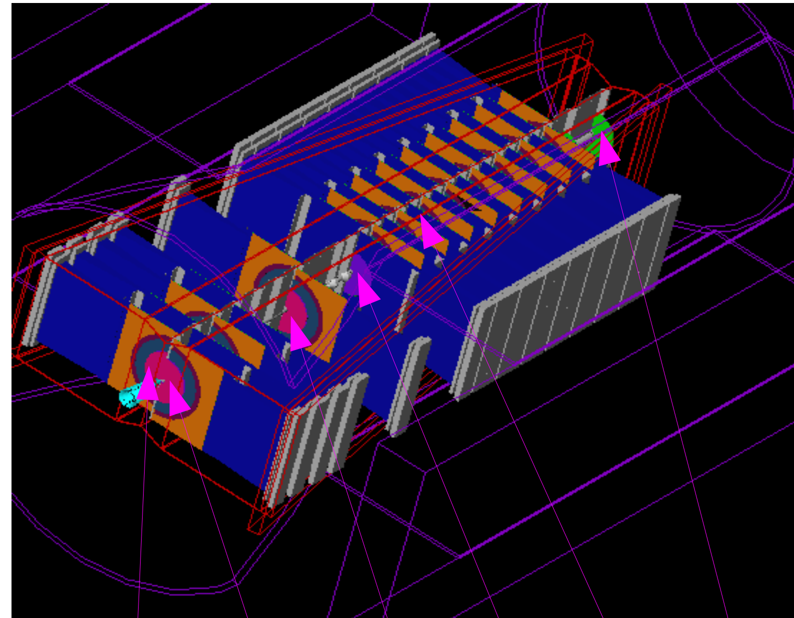


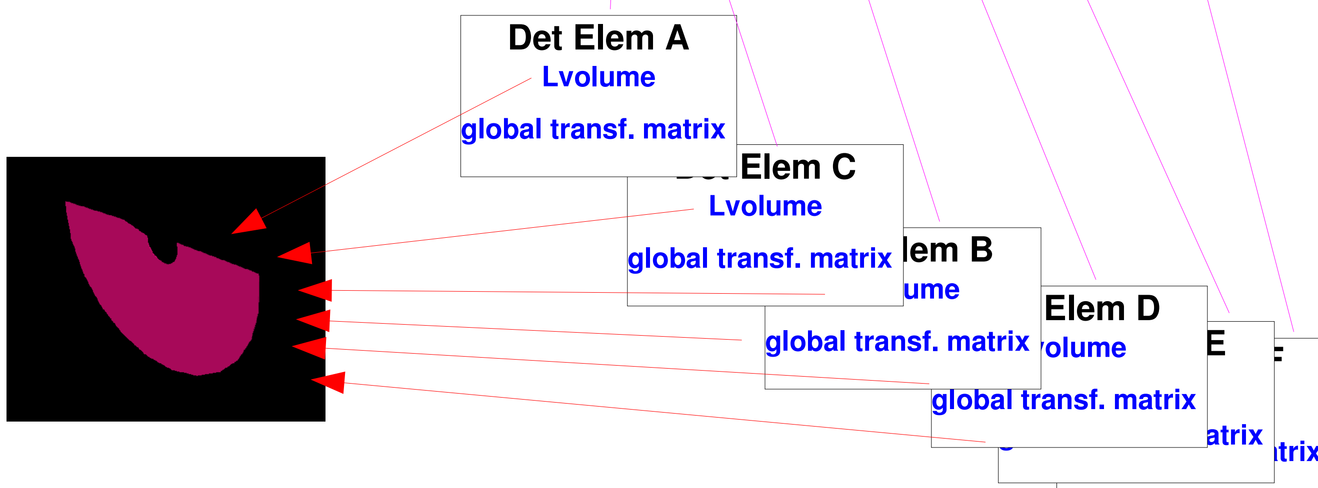
Logical hierarchy: Detector Elements

- **DetectorElement class:**
  - handles complex information to do with geometry, readout, calibrations, alignment, detector response...
- **Detector structure represented as hierarchy of DEs**
  - Sub-detectors provide specialisations with arbitrary level of complexity and granularity
  - Coupled but not directly mapped to geometrical hierarchy
  - Allow navigation within hierarchy
- **Users of LHCb detector description interact only with detector element hierarchy**

Logical hierarchy: Detector Elements

- ***One-to-one mapping*** **to interesting components of the real detector**
- **Contain *unique, identifiable* attributes**
- **The granularity and hierarchy wrt. geometrical description is arbitrary**
  - Can decide what are interesting layers
    - sub-detectors, readout regions, Si, sensors, straw chamber planks...

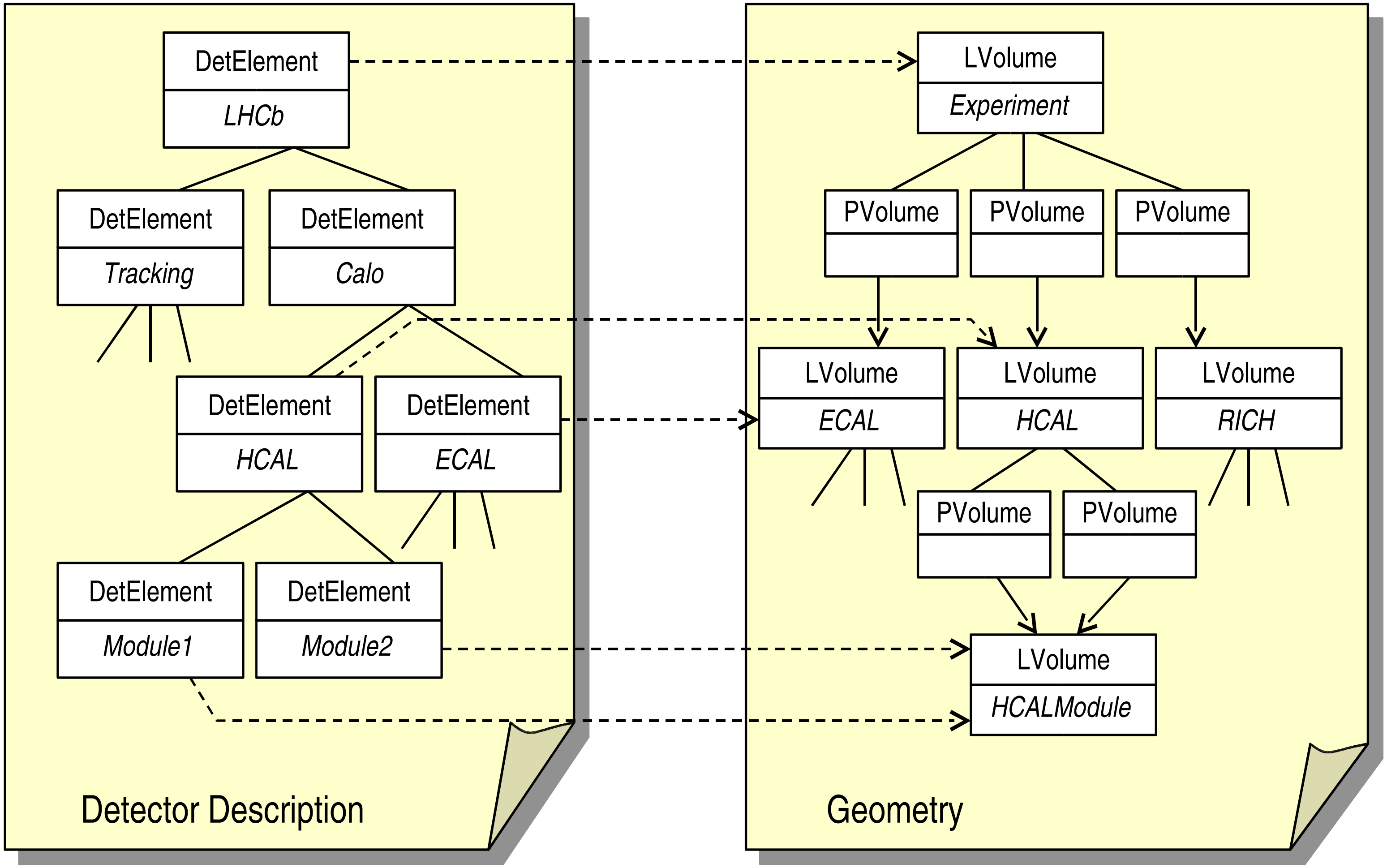




Logical and geometry hierarchies

Logical hierarchy (detector elements)

Geometry hierarchy (replicable, nestable elements)



Detector structure

- **DEs combine knowledge of definition and placements of logical volumes, plus UP and DOWN links, to obtain**
  - One-to-one mapping with LHCb components
  - Knowledge of global position in space
    - position in global reference frame (transf. matrix)
    - position in parent volume (link to placement transformation)
  - Knowledge of place in hierarchy
  - Knowledge of daughter placed volumes (not necessarily detector elements)
  - Many other useful things
    - Gain calibrations, strip capacitances, S/N, etc.

Misalignments in the detector structure

- **Misalignments are handled in the detector element**
  - Combine ***local misalignment*** with ***local transf. matrix*** to obtain ***new local position***
    - Local wrt ***parent detector element***
  - Use links to parents to establish *global* position after *local* misalignment
  - Use links to daughters to propagate misalignments to daughters' global position matrices
- **Misalignments “exported” to client code behind the scenes**

Misalignments in the detector structure

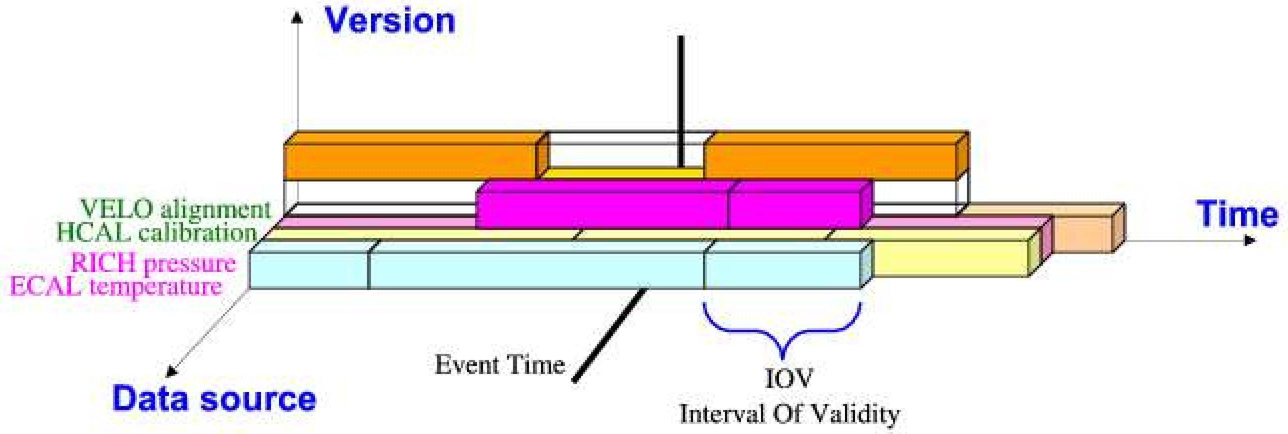
- **Represents change from nominal alignment in the reference frame of the detector element**
  - ie relative to its ***parent*** detector element
- **Handled as a *condition***
- **Stored in conditions database (CondDB)**
  - Nine parameters: X,Y,Z of displacement, and pivot point, α, β, γ of rotation about cartesian axes.
- **Stored as an XML string in the DB.**
- **Convertors instantiate Alignment objects from XML**
  - XML string + convertor gives great flexibility: the concept of ***Alignment Condition*** can easily be extended
    - Of course client code needs to use the information!

Misalignments and Conditions database

- **“Misalignment stored as *condition*...”**
- **But what is a “Condition”?**
  - Non-event, time-varying data needed for ***reconstruction and/or analysis***
    - Single-version (temperature, atmospheric pressure, ...)
    - Multiple-version (Alignment!)
- **Two main challenges for condition handling:**
  - Handling of ***Interval of Validity*** (IOV) + ***versions***
  - Updates and synchronisation in event loop
- **If this is handled correctly then misalignments are too!**

Conditions DB: IOVs and Versions

- **Handled through the API provided by the LCG common project COOL**
  - See http://lcgapp.cern.ch/project/CondDB
- **This is really all I need to know!**
  - The LHCb detector description is a client of this and is unconcerned by the details!



Conditions DB: Updates and synchronisation

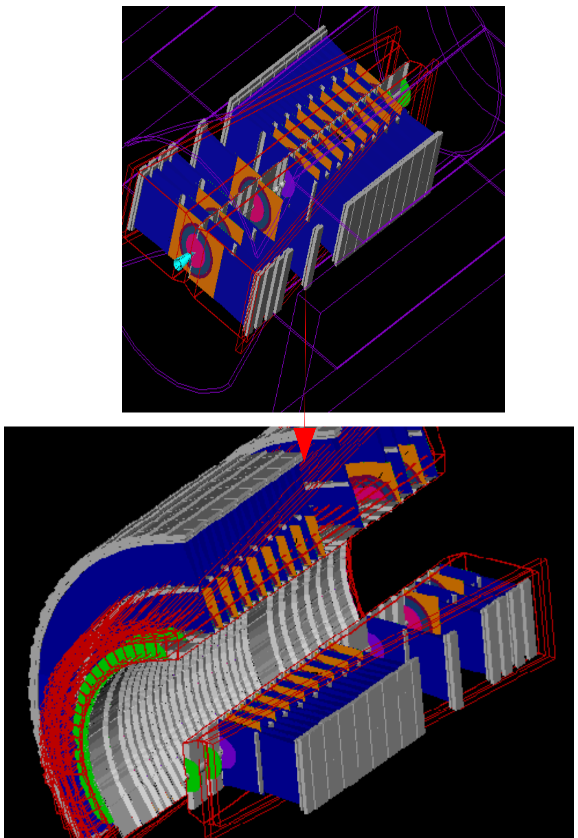
- **During the event loop all conditions and dependent information must be valid**
  - All quantities calculated from conditions must also be updated when conditions are
- **Handled via the UpdateManagerSvc**
  - LHCb-specific solution integrated in the Detector Description framework (all of Gaudi actually)
  - Takes care of
    - Loading new conditions when needed
    - Call user-defined callback functions depending on the changes
      - Used for updating condition-dependent information
    - Handling dependencies between objects
- **Ultimate responsibility lies on users to correctly register condition dependent caching functions**
  - This is actually very easy

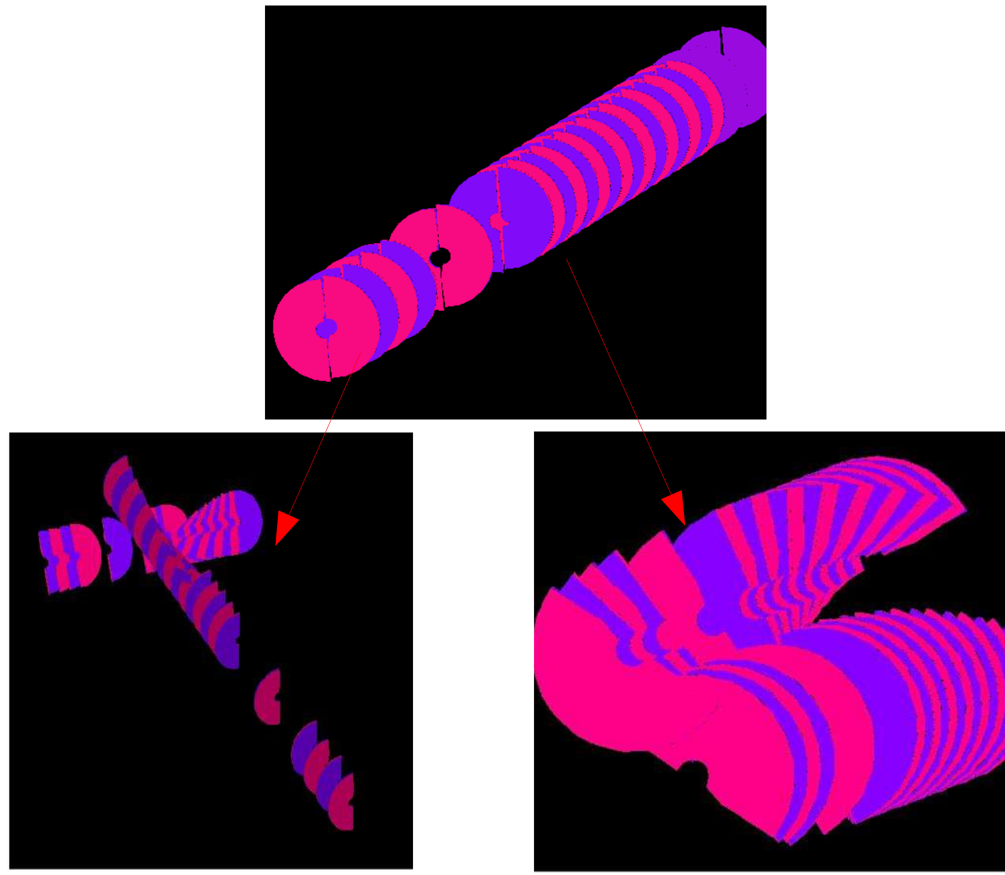
Access and use of misalignments

- **Detector description users see the off-nominal alignment automatically**
  - Local -> global and global -> local transformations now contain misalignments
  - Users must take care of binding all caching of geometrical information to validity of dependent parameters
    - In practice most of this already done in “core” components
  - Misalignments dependent on parent's misalignments
    - potentially many matrix calculations after each change
- **Users can modify misalignments at run-time**
  - changes are propagated and should be seen by all other users
  - This is a transient change in memory, not to be confused with updating database
- **“Users” include simulation, digitization, reconstruction, physics analysis, event display...**

Some (extreme) misalignment illustrations

- **Misalignment allows for rotation about arbitrary pivot point + translation**
- **Arbitrary misalignments applied at different levels of hierarchy during course of SW application**





Use of misalignment information in LHCb

Nominal + off-nominal alignment information accessible and usable everywhere

- **Tests of misalignment systematics**
  - Simulate misaligned detector
  - Digitise hits using correct “misalignments”
  - Reconstruct tracks, vertices, etc. with different alignment constants
  - Gauge the effect on physics performance
- **Alignment algorithms and strategy (see following LHCb talks)**
  - Potential running scenario for detector alignment
    - Apply alignment procedure
    - New alignment constants to lightweight DB slice
    - Validation process
    - Tagged copy to master DB at CERN
    - Replicated in tier 1 centres

Conclusions

- The LHCb detector description framework models and handles run-time misalignments to detector components
- Misalignments are tied in to the Conditions Database framework to allow both automatic run-time updating and propagation of changes, plus versioning and time dependence of alignment parameters
- The functionality has been tested within the LHCb reconstruction chain
- The concept of alignment condition is flexible and can be extended if so required
- LHCb sub-detectors are using it to investigate detector alignment procedures and strategies, systematic effects, etc.

References

- **Gaudi user guide chapter 8**
  - http://lhcb-comp.web.cern.ch/lhcb-comp/Frameworks/Gaudi/Gaudi_v9/GUG/Output/GUG_DetDescription.html
- **LHCb misalignments in detector description Wiki page**
  - https://twiki.cern.ch/twiki/bin/view/LHCb/GeometryFramework
- **LCG conditions database page**
  - http://lcgapp.cern.ch/project/CondDB
- **LHCb conditions database Wiki page**
  - https://twiki.cern.ch/twiki/bin/view/LHCb/CondDBGettingStarted
- **LHCb detector conditions**
  - http://lhcb-comp.web.cern.ch/lhcb-comp/Frameworks/DetCond/