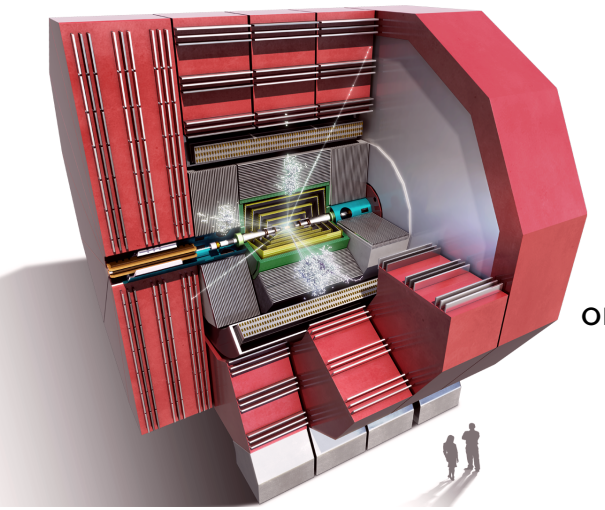
DD4hep and Status of CLIC Detector Implementation



Christian Grefe (CERN)

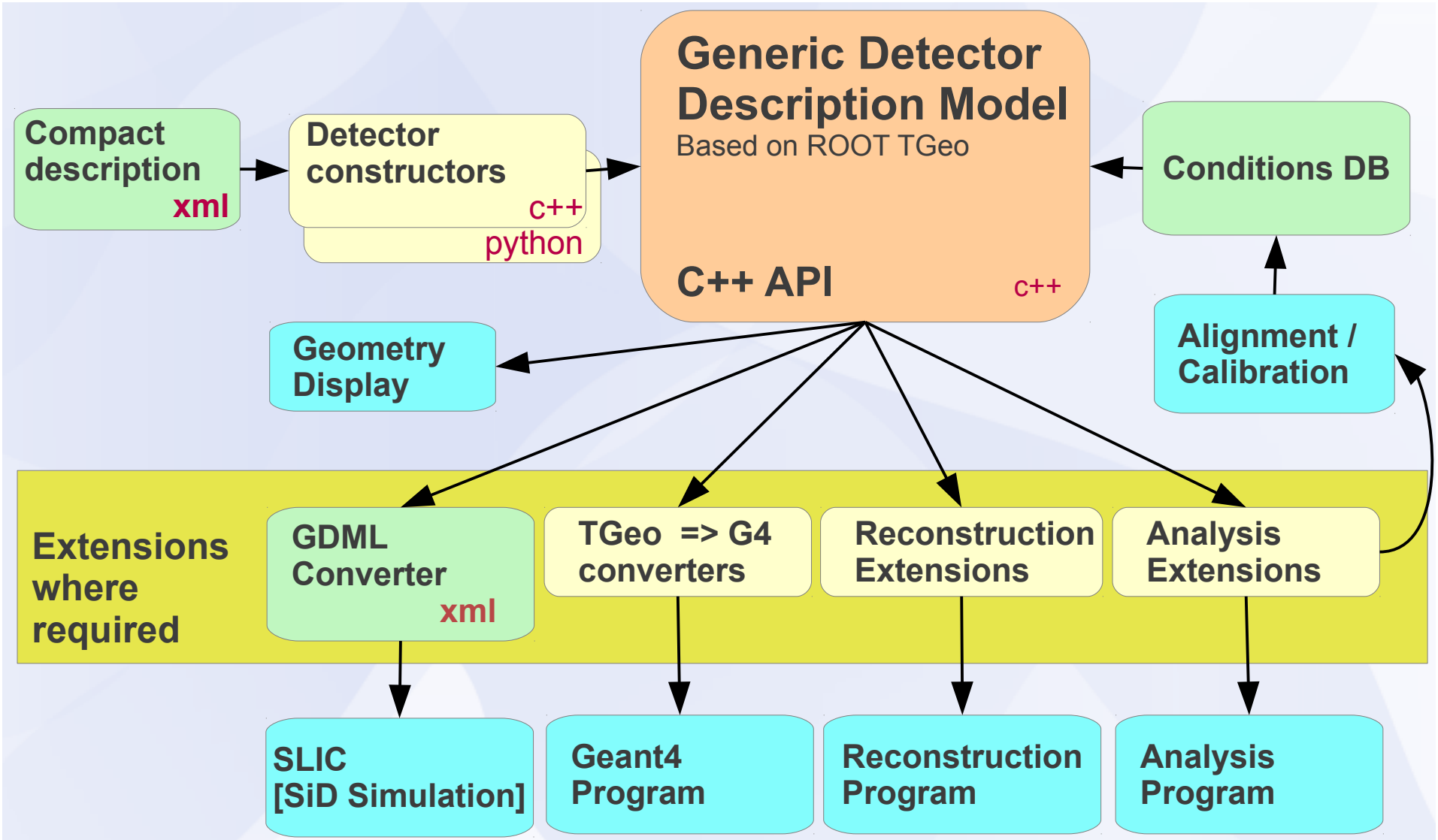
on behalf of the CLIC physics and detector study

10. June 2014

DD4hep – Overview

- Full detector description
  - Geometry, materials, visualization, readout, alignement, calibration, etc.
- Full experiment life cycle
  - Detector concept development, detector optimization, construction, operation
  - Easy transition from one phase to the next
- Consistent description
  - Single source of detector information for simulation, reconstruction, analysis
- Easy to use
  - Only few places to enter information
  - Minimal dependencies

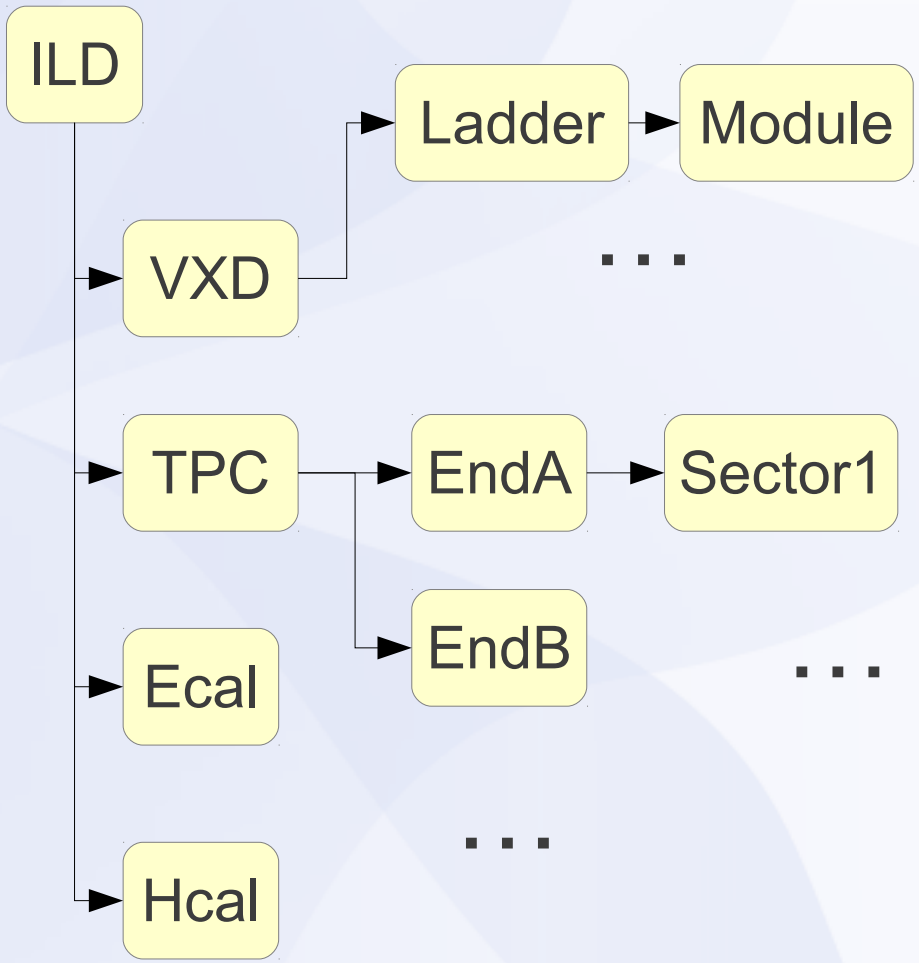
DD4hep – The Big Picture



M. Frank

Generic Detector Description Model

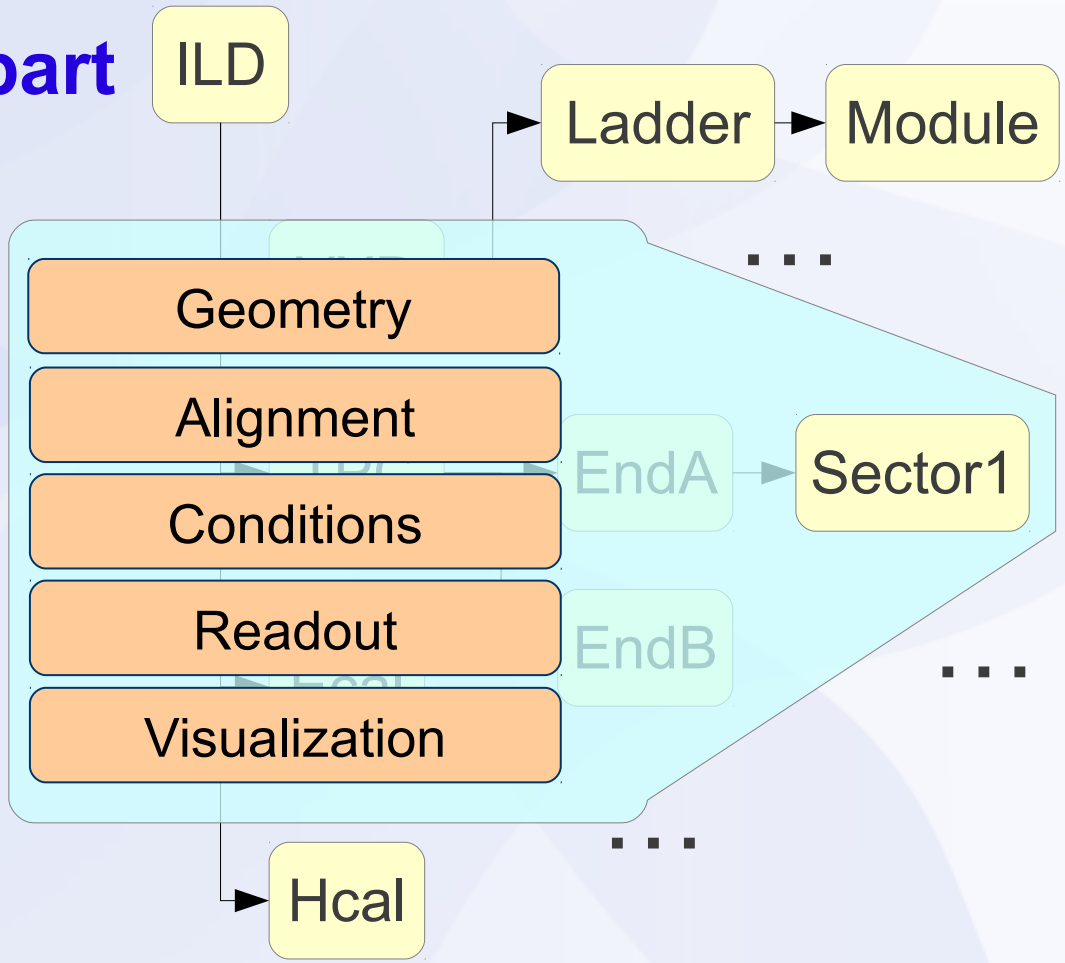
- **Description of a tree-like hierarchy of "detector elements"**
  - **Subdetectors or parts of subdetectors**
  - **Example:**
    - Experiment
    - TPC
    - Endcap A/B
    - Sector
    ...



M. Frank

Generic Detector Description Model

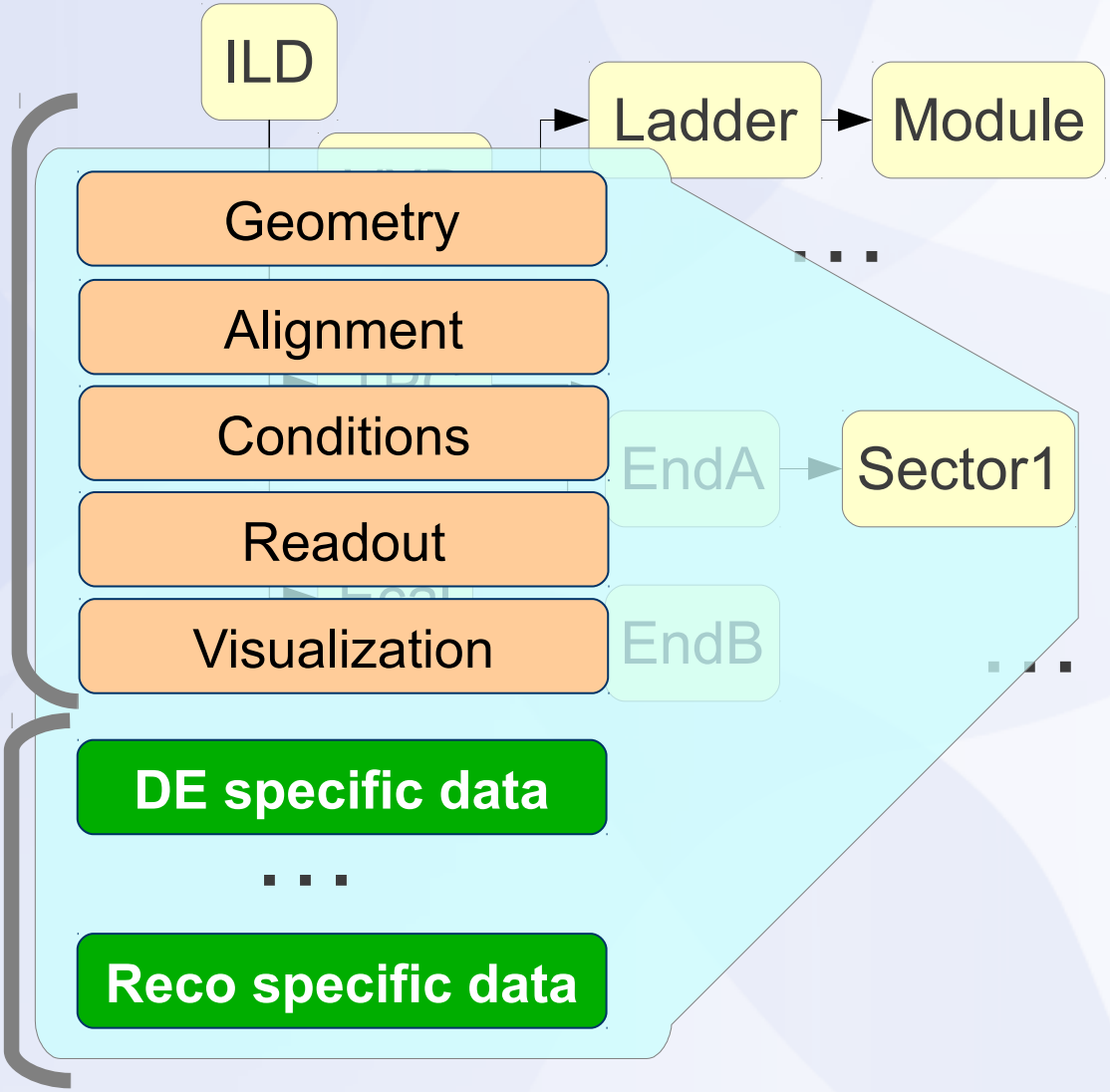
- **Subdetector or the part of a subdetector including the description of its state**
  - **Geometry**
  - **Environmental conditons**
  - **Properties required to process event data**



M. Frank

Extending Detector Description: Detector Views

- **Default DetElement data**
- **Added subdetector specific data**



M. Frank

What comes with DD4hep?

- Core package: detector description using DetElement and plug-ins for converting from compact XML and to GDML or LCDD
- DDSegmentation: provides virtual segmentation (position in volume to cell ID and inverse) with no dependencies; used by DD4hep and simulation and reconstruction tools (extendable via plug-ins)
- Detector constructors provided by user as plug-ins (many simple subdetectors and other examples available from Linear Collider studies)
- Geometry information in simulation: through linking, e.g. DD4G, or via export of geometry, e.g. SLIC via LCDD
- DDRec: high level interface to geometry using views, extendable via plug-ins
  $\Rightarrow$ needs to match detector constructors

Getting Started with DD4hep

- DD4hep project page:
  `http://aidasoft.web.cern.ch/DD4hep`
- Source code (SVN):
  `https://svnsrv.desy.de/public/aidasoft/DD4hep/trunk`
  - Contains user manual and examples
- Additional examples (implementation of ILD):
  `https://svnsrv.desy.de/public/ddsim/DDSim/trunk/`
- Bug tracker:
  `http://sft.its.cern.ch/jira/browse/DDFORHEP`
- Mailing list:
  `dd4hep-developers@cern.ch`
- Phone meetings:
  `http://indico.cern.ch/category/2742/`

Available Implementations and Plans

- Re-implementation of ILD (direct translation of Mokka implementations)
- Re-implementation of CLIC_SiD (direct translation from GeomConverter)
- Both versions lack validation and detail (especially the SiD version)
- What needs to be done:
  - Validate all existing implementations and verify that they can reproduce the previous detector geometries (given the same parameters)
  - Follow up with detector optimisation and engineering studies
  - Include enough flexibility for further optimisation studies
- Subdetector tasks:
  - Silicon tracker – N. Nikiforos (CERN)
  - ECAL – D. Protopopescu (Glasgow)
  - Forward calorimeters – A. Sailer (CERN)
  - Open issues: HCAL, solenoid, yoke & muon chambers
  - In case of specific implementations: LumiCal, vertex detector
  - Other material (usual simple shapes just defined in XML): beam pipe, supports, cables, MDI

Contributors are very welcome!