# Simulation of the LHeC detector

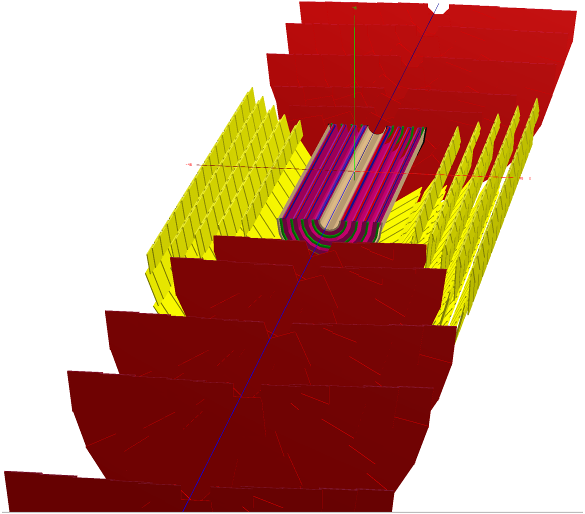
Christian Grefe, Peter Kostka

LHeC Workshop, Chavannes-de-Bogis

21. January 2014

# Outline

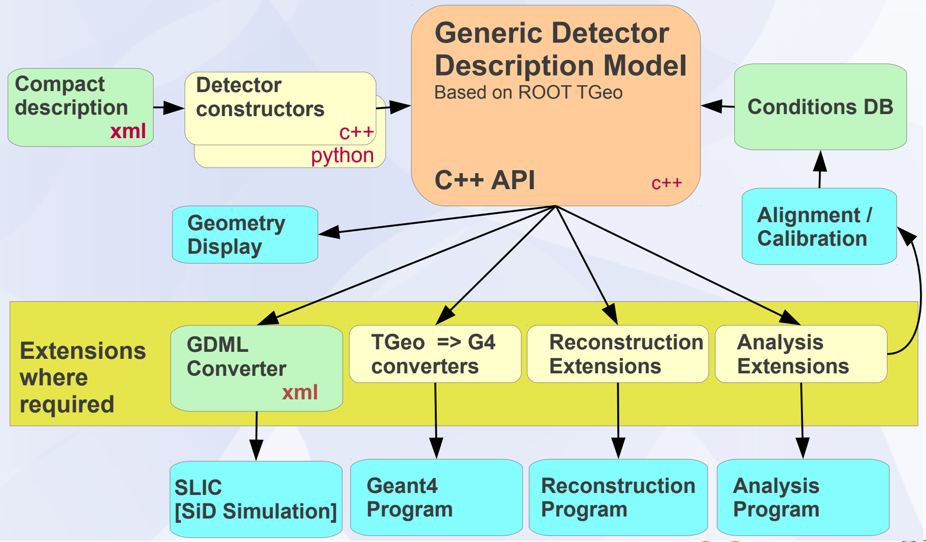
1. DD4hep
2. LHeC Detector Simulation



# DD4hep – Main Requirements

- 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

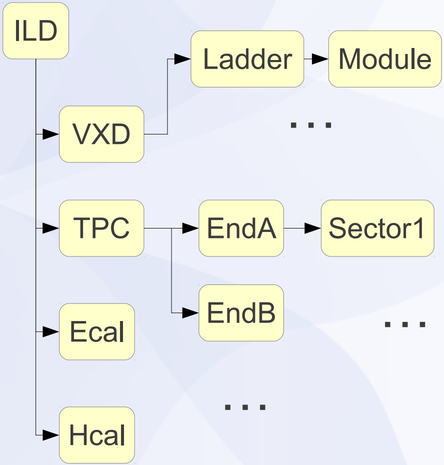
# DD4hep – The Big Picture




M. Frank

# Generic Detector Description Model

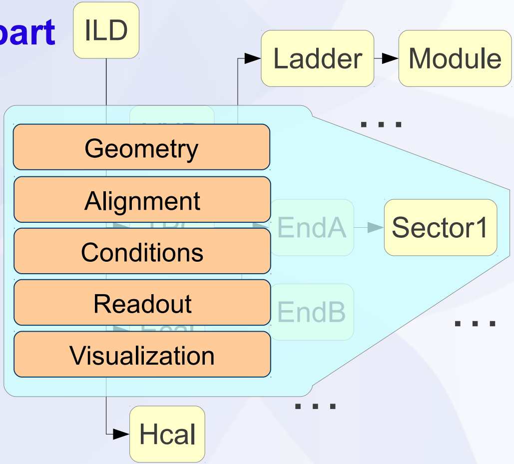
- **Description of a tree-like hierarchy of "detector elements"**
  - **Subdetectors or parts of subdetectors**
  - **Example:**
    **- Experiment**
    **- TPC**
    **- Endcap A/B**
    **- Sector**
    **...**




M. Frank

# Generic Detector Description Model

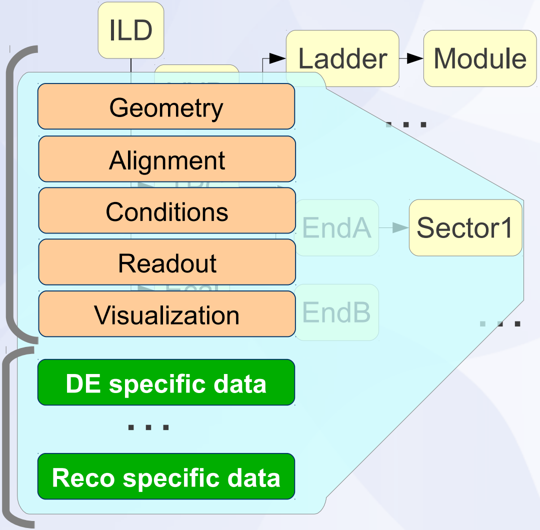
- **Subdetector or the part of a subdetector including the description of its state**
  - **Geometry**
  - **Environmental conditons**
  - **Properties required to process event data**




M. Frank

# Extending Detector Description: Detector Views

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




M. Frank

# What comes with DD4hep?

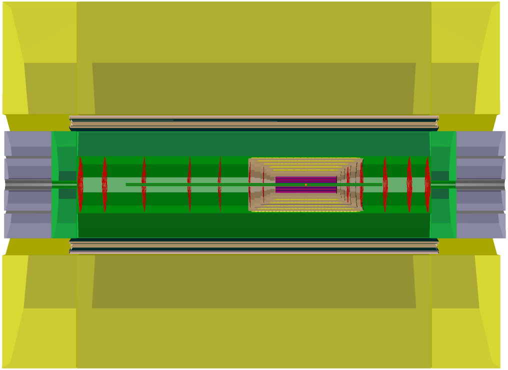
- Core package: detector description using DetElement and plug-ins for converting from compact XML and to GDML or LCDD
- DDSegmentation: provides virtual segmentation (position in volume to cell ID and inverse) with no dependencies; used by DD4hep and simulation and reconstruction tools (extendable via 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
- DDReconstruction: high level interface to geometry using views, extendable via plug-ins $\Rightarrow$ needs to match detector constructors

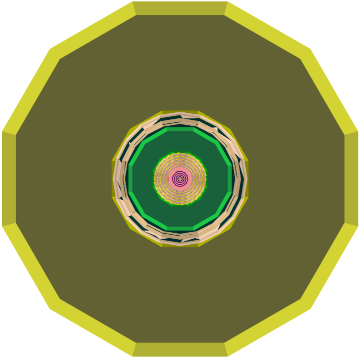
# Status and Plans for DD4hep

- DD4hep core package rather stable
- Documentation available (needs to improve)
- Growing number of (example) detector constructors
  $\Rightarrow$ Linear Collider detectors
- Early 2014: Finalize DDSegmentation and DDReconstruction packages
- Afterwards, review of design choices and beta release
- End 2014: full transition of ILD and CLIC detector concepts to DD4hep

- Website: `http://aidasoft.web.cern.ch/DD4hep`
- SVN repository: `https://svnsrv.desy.de/public/aidasoft/DD4hep`

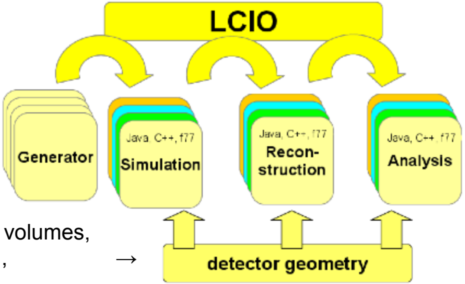
# LHeC Detector in DD4hep





# Detector design/sim/rec - AIDA / DD4Hep

## Common Software tools for Linear collider




DD4Hep: detector descr. - xml, volumes, alignment/calibration, DB cond., →

AIDA: Advanced Infrastructure for Detector development for future Accelerators

A common Event Data Model (LCIO) with persistency and a common detector geometry description are the requirements for the exchange and common development of software tools between detector concepts and working groups.
This can even work across languages (C++, Java, Fortran), provided EDM and Geometry provide interfaces for these languages.

# lhec-afs-project-space

## Installed on lxplus (SL6):

- /afs/cern.ch/project/lhec/scripts (setting the environment)
- /afs/cern.ch/project/lhec/bin
- /afs/cern.ch/project/lhec/software (many packages - generators, root, geantX, ...)
- /afs/cern.ch/project/lhec/software/aidasoft/DD4hep - simulation framework (pre-release still)
- /afs/cern.ch/project/lhec/software/aidasoft/DD4hep/DD4hep-t - installation dev. and tests
  - ReadMe.txt and ReadMe_lhec.txt
    DD4hep software repository on server at DESY:
    svn co https://svnsrv.desy.de/public/aidasoft/DD4hep/trunk target_directory (e.g. DD4hep)
  - Installation and running of lhec example(s) - (incl. cmake commands etc.)
- Project: http://aidasoft.web.cern.ch/DD4hep pre-release software (!)
- Installation up to date - svn revision 970.

## Generator input implemented in module LHeDSimul

- see S.Mandelli (http://indico.cern.ch/getFile.py/access contribId=15&sessionId=2&resId=3&materialId=slides&confId=281921)
  - formats: stdhep, hepevt, lhe (les houches), hepmc2, lcio
  - directly transformed into lcio → DD4hep/GEANT4
  - simulation result - output: lcio file

## git-svn repository/interfacing to be used for LHeC development
(recommended: SmartGitHg 5 - linux, windows, OSX)

# ep/eA Simulation in the LHeC Detector

## Demanding extensions needed

- besides ROOT and GEANT4 - FLUKA has to be incorporated into the DD4hep environment:
  - Generators -PYTHIA8, HERWIG, SHERPA- do not consider standardly ep and even less eA currently.
    FLUKA is handling nuclear evaporation/fragmentation
  - For eA we need a handle on radiative corrections, bigger than in ep.
  - see Néstor Armesto: eA at the LHeC: detector requirements and simulations: http://indico.cern.ch/getFile.py/access?contribId=8&sessionId=1&resId=0&materialId=slides&confId=281921
  - dedicated man power - interested group of Uladag University (Turkey)

# LHeC - Reconstruction Modules - Next

Based on DD4hep reconstruction interface (ready soon)

- Cell ID and detector ID encoding / decoding to allow look-up of closest detector element based on hit cell ID → find the sensitive layer to get the local coordinate system → reconstruction and keeping track of all properties
- Reconstruction and analysis coding for each subdetector (-system) based on DD4hep interfaces
- Experience and even code modules re-usable from other detectors (e.g. Pandora package for particle flow reconstruction)
- recruiting of man power urgently needed
  - apply for Horizon 2020: a topic for advanced community "Detector for future accelerators" (targeted to AIDA continuation)

# Join the Effort

Short list:

- EM calorimetry optimisation
  - granularity - installation requirements - modularity - not compromising the functionality!
  - very backward EM calorimetry, rates and geometry
- HAD calorimetry optimisation
  - longitudinal shower containment
  - effect of solenoid on hadronic scale
  - granularity needed
  - installation requirements - modularity - again: not compromising the functionality!
- Tracker - together with calorimetry: particle ID, energy flow ...
- Trigger
  - overall trigger geometry (trigger-less, round-robin L1, ...), how many levels?
  - do we need dedicated trigger detectors (up-stream background reduction)
  - heavy-flavour triggers, how, on which level?
  - trigger strategies and rates (inclusive electron+$E_{miss}$ on L1, rest on higher levels?)