



# Turnkey Software Stack Vision

Plans and Status

André Sailer

CERN-EP-LCD

Future Collider Software WS

June 12–13, 2019

Bologna, Italy



## Strategic R&D Programme on Technologies for Future Experiments

### Chapter 10.7 Turnkey Software Stacks for Future Experiments

- ... The goal of this project is the development of a single turnkey software stack that can be used for the detector studies of both FCC and CLIC communities. . .
- ... In a second phase, the resulting software stack should be prepared for reuse by new detector study groups. . .

# The Vision for the Turnkey Software Stack



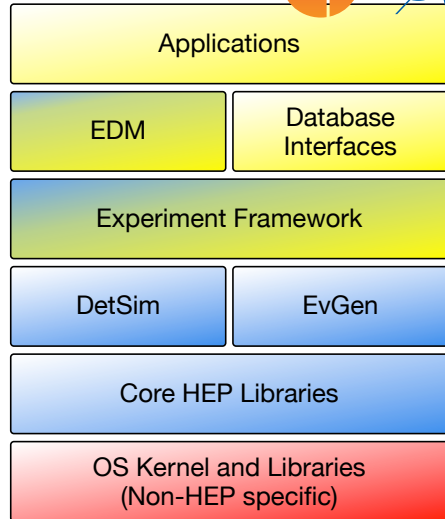
LCG Stack provides 100s of libraries, on top of which experiment frameworks are developed

## The turnkey stack

- connects and extends the individual packages
- enables a complete data processing framework

## and should be

- easy to use: for librarians, developers, users
- easy to set up
- easy to extend
- full of functionality





To enable interoperability and long term maintainability

- Build system

- ▶ Share build results, but also knowledge of how to build
- ▶ HSF Packaging working group studying various options

- Testing

- Licensing and copyright

- Documentation

- ▶ If it is not documented, it doesn't exist

Ensure as much uniformity as reasonable

E.g., through [https://github.com/HSF/tools/blob/master/create\\_project.py](https://github.com/HSF/tools/blob/master/create_project.py)



# Next Steps and Current Status

# Leverage Common Solutions



## Framework

- Gaudi: benefit from use and developments in ATLAS and LHCb

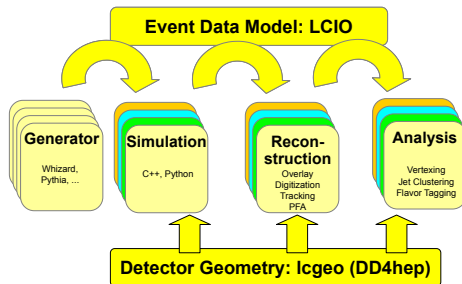
## Detector Geometry Description

- DD4hep: used by almost everybody

## Event Data Model

- Common **interface** allows for sharing of algorithm implementations
- Very successfully practised in iLCSoft for many years

Components need to offer agility and flexibility, especially for detector design studies



# Next Steps



## Make the reconstruction of CLIC (in iLCSoft), available in Gaudi for FCC(ee)

Adiabatic changes for CLIC reconstruction; re-factoring rather than re-writing

- Wrapper for Marlin processor to Gaudi algorithm
- One by one make them native Gaudi algorithms
- Adapt from iLCSoft::LCIO to podio::LCIO
- Integrate into the turnkey software stack

## Provide a way to build, deploy or use

- Build system
- CVMFS
- Containerization



Proof of concept for for Gaudi–Marlin Wrapper exists <https://github.com/andresailer/GMP>

- *Very* minimal changes needed in Marlin
- Make `marlin::Processor::setParameters` and `marlin::Processor::setName` public
- Rename `marlin::EventSelector` to avoid conflict with `gaudi::EventSelector`

## Build system

- FCC stack is being build with `spack`





## Benefits for CLIC and FCC

- Re-use and exchange reconstruction algorithms
- Share interfaces to common solutions, e.g., PandoraPFA, ACTS
- Based on multi-threading capable framework
- Use of a more wide-spread build system