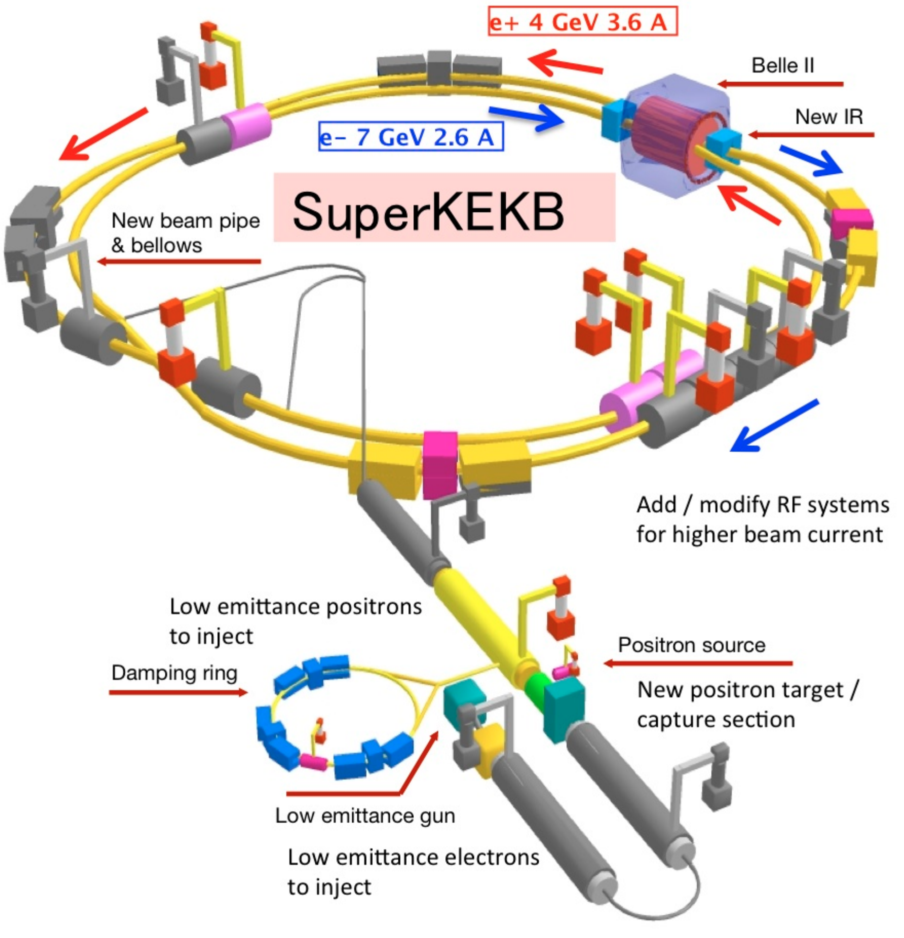
# The Belle II Software
## From Detector Signals to Physics Results

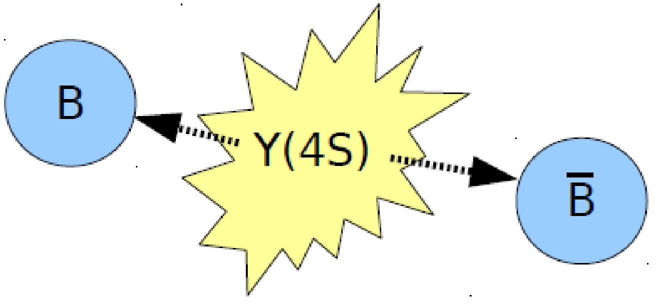
Thomas Kuhr
LMU Munich

Bundesministerium
für Bildung
und Forschung

INSTR17
2017-02-28

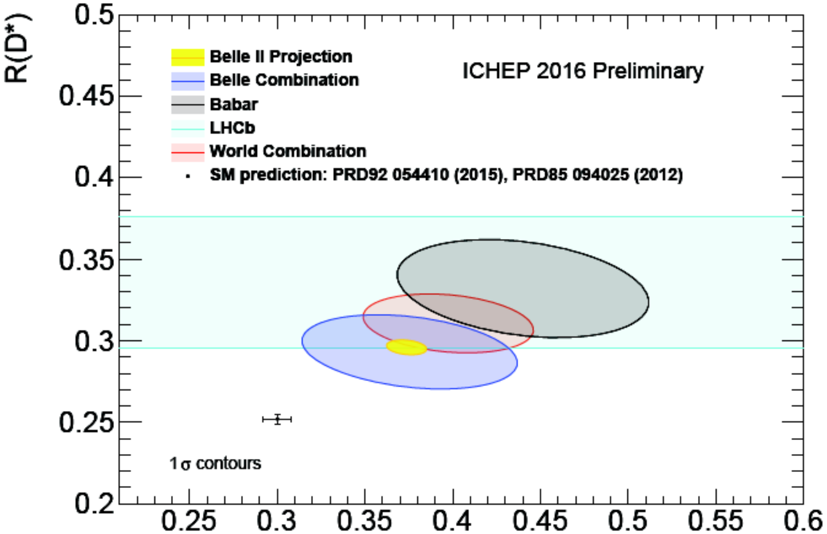
# Belle II @ SuperKEKB



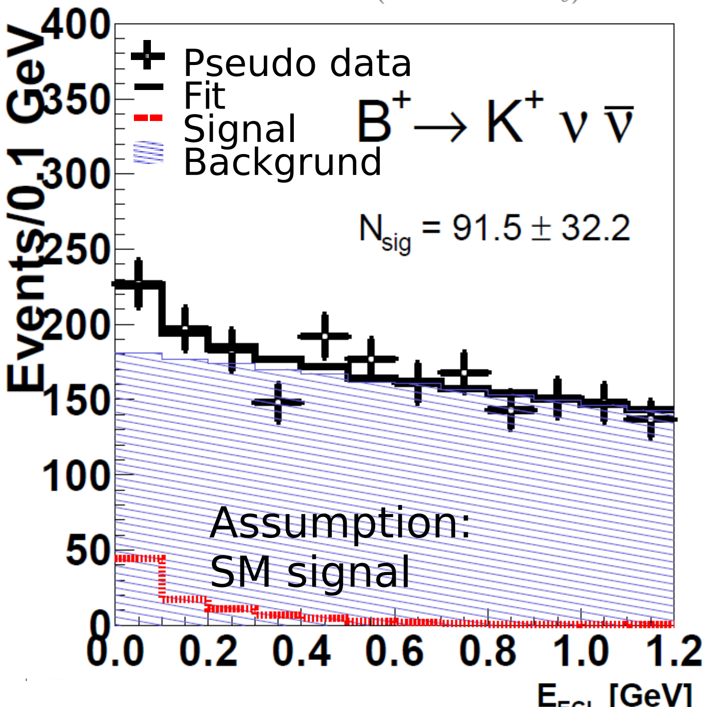




- B, charm, τ physics
  - 40 higher luminosity than KEKB
  - Aim: 50 times more data than Belle
  - ➔ Significantly increased sensitivity to new physics

# Physics @ Belle II



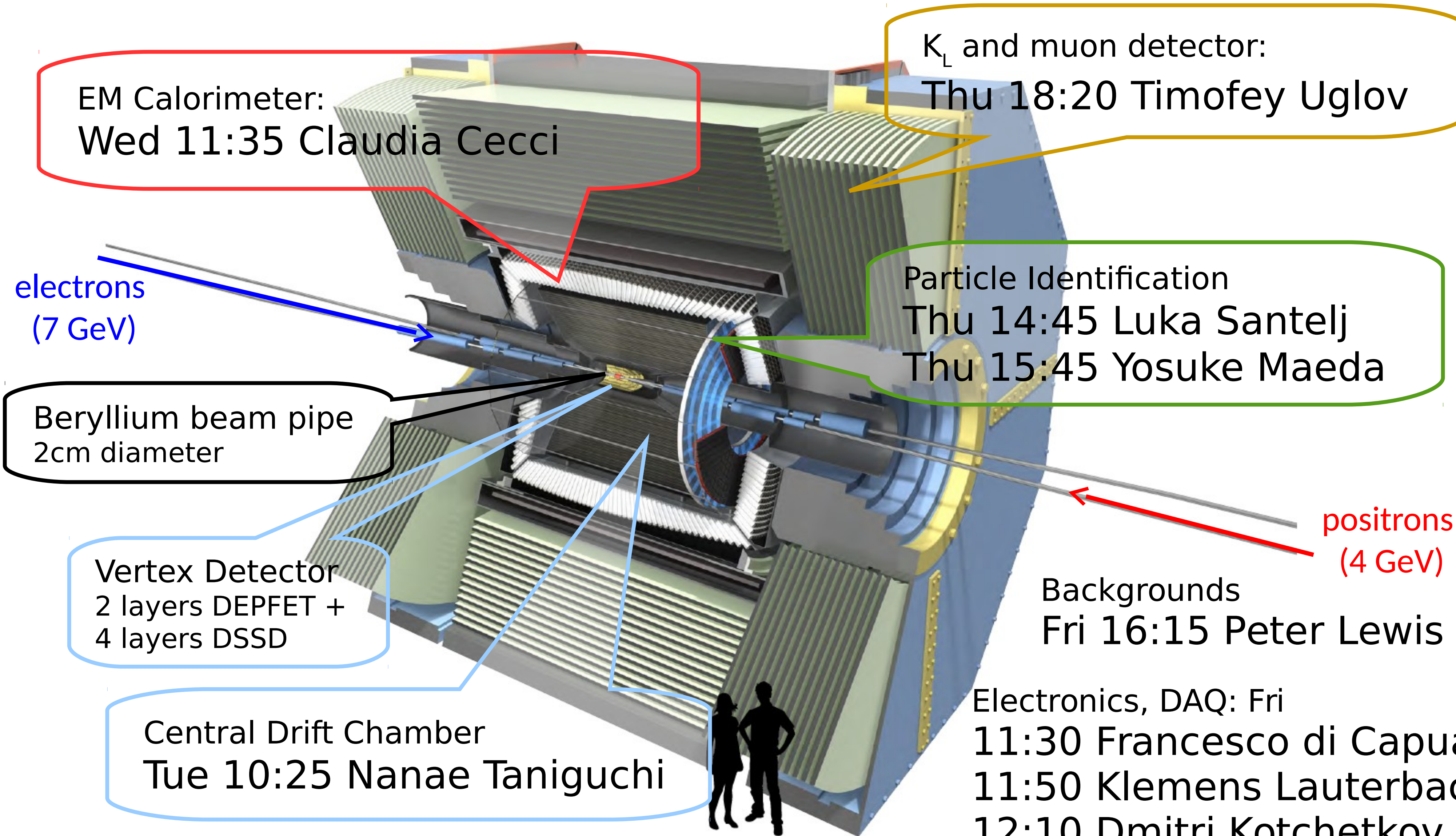

$$R_{D^{(*)}} = \frac{\mathrm{Br}(B \to D^{(*)}\tau\nu_\tau)}{\mathrm{Br}(B \to D^{(*)}\ell\nu_\ell)}$$




| | Observables | Belle (2014) | Belle II 5 ab$^{-1}$ | Belle II 50 ab$^{-1}$ |
|---|---|---|---|---|
| WG1 | $\|V_{cb}\|$ incl. | $41.6\cdot10^{-3}(1\pm1.8\%)$ [1] | 1.2% | |
| | $\|V_{cb}\|$ excl. | $37.5\cdot10^{-3}(1\pm3.0\%_{\rm ex.}\pm2.7\%_{\rm th.})$ [2] | 1.8% | 1.4% |
| | $\|V_{ub}\|$ incl. | $4.47\cdot10^{-3}(1\pm6.0\%_{\rm ex.}\pm2.5\%_{\rm th.})$ [3] | 3.4% | 3.0% |
| | $\|V_{ub}\|$ excl. (had. tag.) | $3.52\cdot10^{-3}(1\pm9.5\%)$ [4] | 4.4% | 2.3% |
| | $\mathcal{B}(B\to\tau\nu)$ $[10^{-6}]$ | $96(1\pm27\%)$ [5] | 10% | 5% |
| | $\mathcal{B}(B\to\mu\nu)$ $[10^{-6}]$ | $<1.7$ [6] | 20% | 7% |
| | $R(B\to D\tau\nu)$ | $0.440(1\pm16.5\%)$ [7]† | 5.2% | 3.4% |
| | $R(B\to D^*\tau\nu)$† | $0.332(1\pm9.0\%)$ [7]† | 2.9% | 2.1% |
| WG2 | $\mathcal{B}(B\to K^{*+}\nu\overline{\nu})$ $[10^{-6}]$ | $<40$ [8] | $<15$ | 20% |
| | $\mathcal{B}(B\to K^{+}\nu\overline{\nu})$ $[10^{-6}]$ | $<55$ [8] | $<21$ | 30% |
| | $A_{CP}(B\to X_{s,d}\gamma)$ $[10^{-2}]$ | $2.2\pm4.0\pm0.8$ [?] | 1 | 0.5 |
| | $S(B\to K_S^0\pi^0\gamma)$ | $-0.10\pm0.31\pm0.07$[9] | 0.11 | 0.035 |
| | $S(B\to\rho\gamma)$ | $-0.83\pm0.65\pm0.18$ [10] | 0.23 | 0.07 |
| | $C_7/C_9$ $(B\to X_s\ell\ell)$ | ~20% [11] | 10% | 5% |
| | $\mathcal{B}(B_s\to\gamma\gamma)$ $[10^{-6}]$ | $<8.7$ [12] | 0.3 | – |
| | $\mathcal{B}(B_s\to\tau\tau)$ $[10^{-3}]$ | – | $<2$ [13]‡ | – |
| WG3 | $\sin 2\beta$ | $0.667\pm0.023\pm0.012$ [14] | 0.012 | 0.008 |
| | $\alpha$ [°] | $85\pm4$ (Belle+BaBar) [?] | 2 | 1 |
| | $S(B\to\phi K^0)$ | $0.90^{+0.09}_{-0.19}$ [15] | 0.053 | 0.018 |
| | $S(B\to\eta' K^0)$ | $0.68\pm0.07\pm0.03$ [?] | 0.028 | 0.011 |
| | $S(B\to K_S^0K_S^0K_S^0)$ | $0.30\pm0.32\pm0.08$ [16] | 0.100 | 0.033 |
| WG4 | $\gamma$ [°] | $68\pm14$ [17] | 6 | 1.5 |
| WG5 | $\mathcal{A}(B\to K^0\pi^0)$ | $-0.05\pm0.14\pm0.05$ [18] | 0.07 | 0.04 |
| WG6 | $\mathcal{B}(D_s\to\mu\nu)$ | $5.31\cdot10^{-3}(1\pm5.3\%\pm3.8\%)$ [19] | 2.9% | 0.9% |
| | $\mathcal{B}(D_s\to\tau\nu)$ | $5.70\cdot10^{-3}(1\pm3.7\%\pm5.4\%)$ [19] | 3.5% | 3.6% |
| | $\mathcal{B}(D^0\to\gamma\gamma)$ $[10^{-6}]$ | $<1.5$ [20] | 30% | 25% |
| | $A_{CP}(D^0\to\pi^0\pi^0)$ $[10^{-2}]$ | $-0.03\pm0.64\pm0.10$ [21] | 0.29 | 0.09 |
| | $A_{CP}(D^0\to K_S^0\pi^0)$ $[10^{-2}]$ | $-0.21\pm0.16\pm0.09$ [21] | 0.08 | 0.03 |
| | $x(D^0\to K_S^0\pi^+\pi^-)$ $[10^{-2}]$ | $0.56\pm0.19\pm{}^{0.07}_{0.13}$ [22] | 0.14 | 0.11 |
| | $y(D^0\to K_S^0\pi^+\pi^-)$ $[10^{-2}]$ | $0.30\pm0.15\pm{}^{0.05}_{0.08}$ [22] | 0.08 | 0.05 |
| | $\|q/p\|(D^0\to K_S^0\pi^+\pi^-)$ | $0.90\pm{}^{0.16}_{0.15}\pm{}^{0.08}_{0.06}$ [22] | 0.10 | 0.07 |
| WG7 | $\tau\to\mu\gamma$ $[10^{-9}]$ | $<45$ [23] | $<14.7$ | $<4.7$ |
| | $\tau\to e\gamma$ $[10^{-9}]$ | $<120$ [23] | $<39$ | $<12$ |
| | $\tau\to\mu\mu\mu$ $[10^{-9}]$ | $<21.0$ [24] | $<3.0$ | $<0.3$ |

# Belle II Detector

Fri 15:45 Peter Krizan

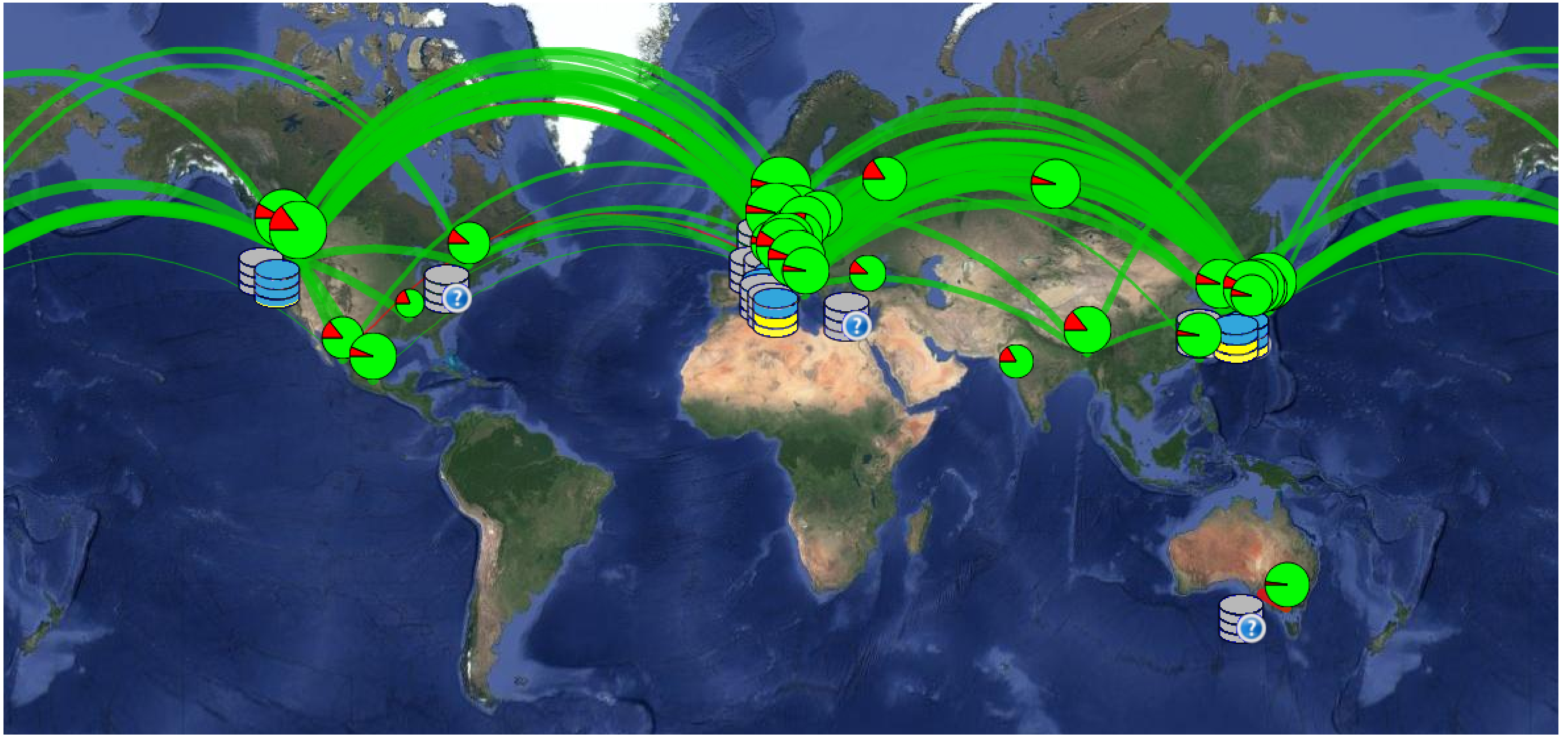



Backgrounds
Fri 16:15 Peter Lewis

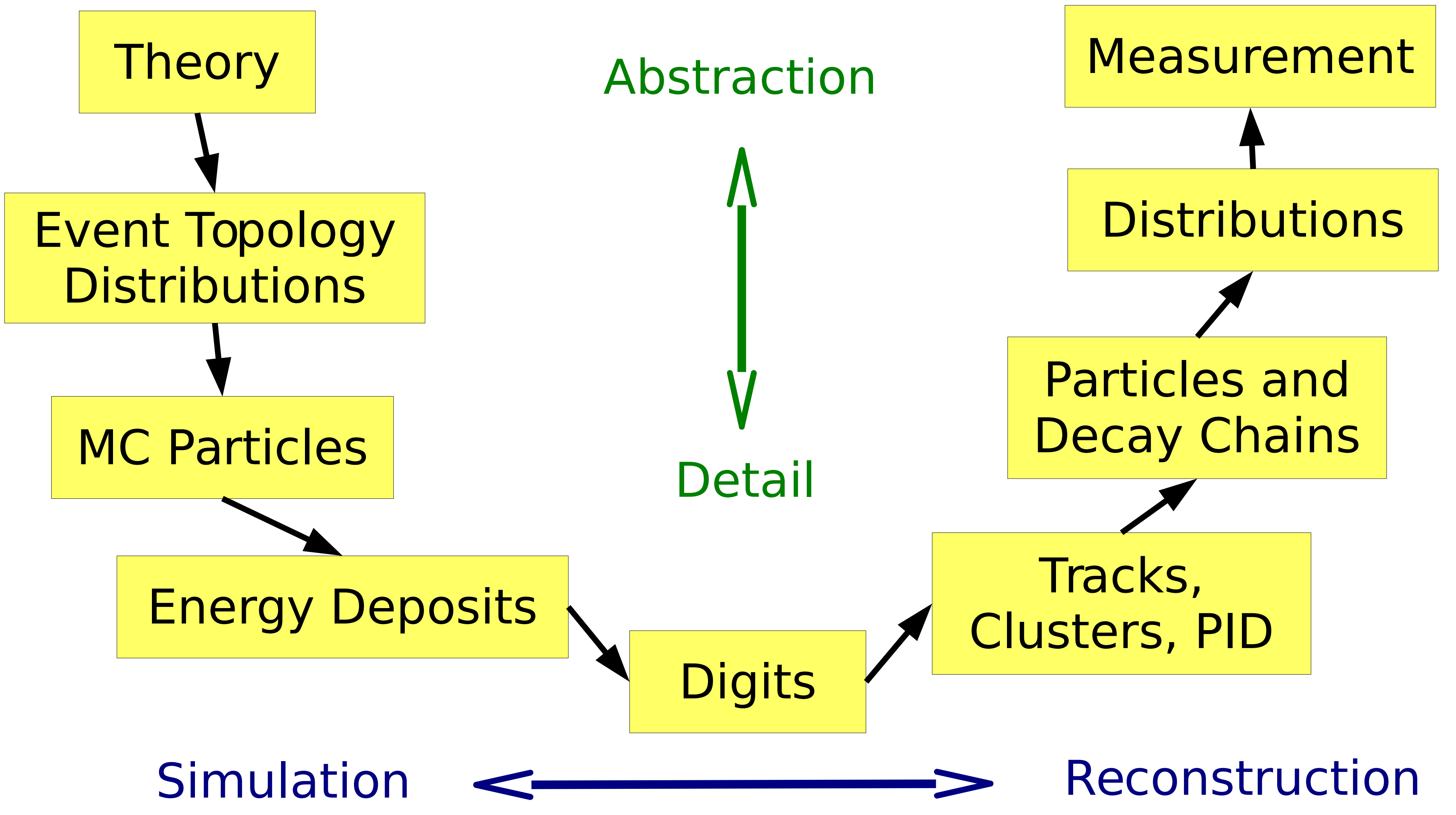
Electronics, DAQ: Fri
11:30 Francesco di Capua
11:50 Klemens Lauterbach
12:10 Dmitri Kotchetkov

# Belle II Data

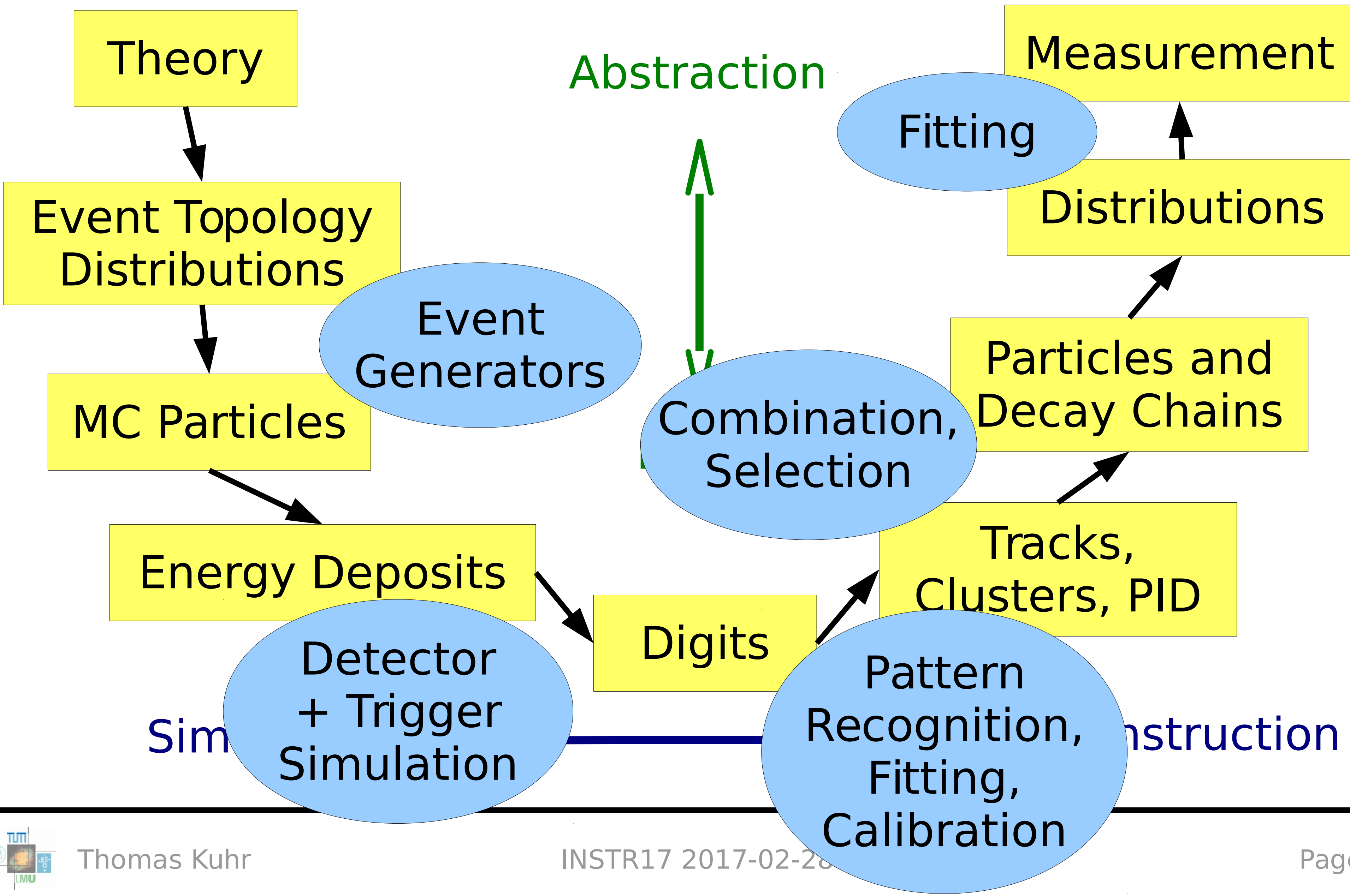
- O(50) larger data volume than Belle
  - ➢ Storage and CPU requirements similar to LHC experiments
  - ➔ Distributed computing model



# Information Flow

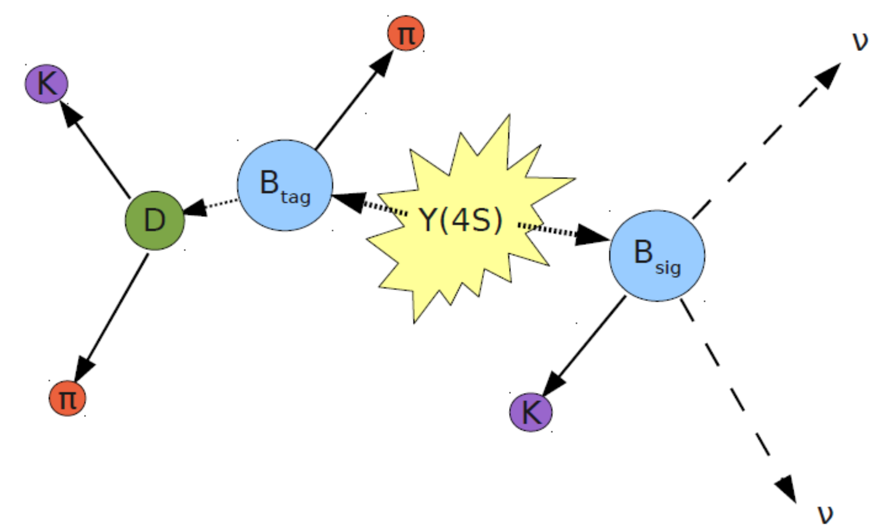


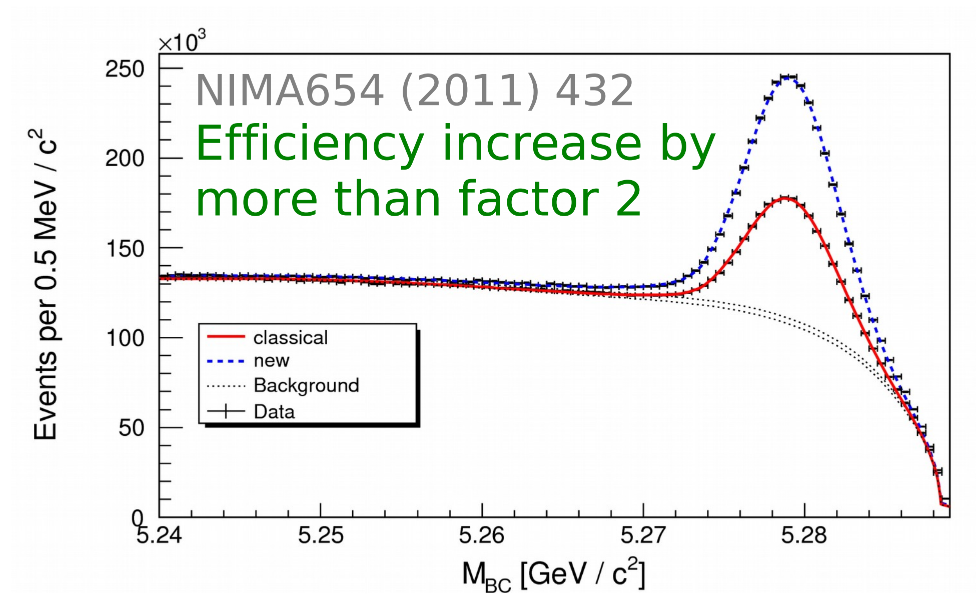
# Information Flow



# Importance of Software

- ✔ Essential for obtaining physics results from detected signals
- ✔ Important factor for computing resource demands
- ➔ Full potential of complex detectors can only be exploited with sophisticated software
- ➢ Example: Full reconstruction of B mesons at Belle



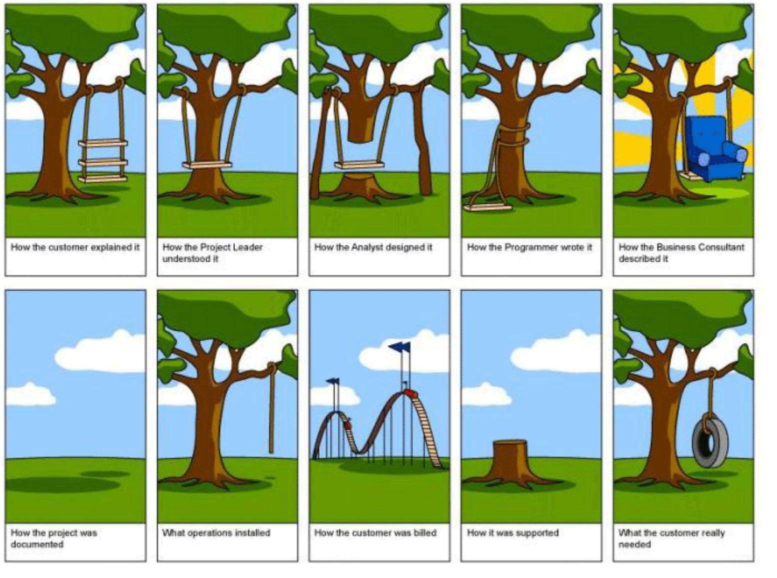
# Software Development at Belle II

Aim:

- Reliable, sophisticated, and easy-to-use software for acquisition, simulation, reconstruction, and analysis of Belle II data

Challenge:

- Regional distribution, different (cultural) backgrounds and skills of developers




- ✔ State-of-the-art tools
- ✔ Commonly accepted rules and guidelines
- ✔ Well defined procedures
- ✔ Efficient communication channels

# Code Structure

- Tools: scripts for installation and environment setup
- Externals: software from others that we use

```
COMMON_PACKAGES:=pkg-config-wrapper gcc binutils zlib bzip2 curl git libxml2 libxslt \
    cmake freetype sqlite python gdb boost gtest eigen astyle scons madgraph
COMMON_OPTIONAL:=clang cppcheck valgrind
PACKAGES:=clhep geant4 postgresql neurobayes xrootd root nbplugin fastbdt vgm rave \
    MillepedeII hepmc pythia photos tauola evtgen phokhara cry exrootanalysis \
    flc nsm2 belle_legacy fann
export PYTHON_PACKAGES:=autopep8==1.2.1a0 CherryPy==4.0.0 decorator==4.0.4 ipython==4.0.0 \
    ipython-genutils==0.1.0 lxml==3.4.4 numpy==1.9.2 path.py==8.1.1 pep8==1.6.3a0 \
    pexpect==3.3 pickleshare==0.5 requests==2.7.0 simplegeneric==0.8.1 setuptools-scm==1.7.0  \
    traitlets==4.0.0 cycler==0.10.0 matplotlib==1.5.1 pyparsing==2.1.0 python-dateutil==2.5.0 \
    pytz==2015.7 six==1.10.0
```

- Belle II software basf2: our code
  - C++11, python
  - SCons build system

```
Import('env')

env['LIBS'] = [
    'framework',
    'analysis_dataobjects',
    'mdst_dataobjects',
    '$ROOT_LIBS',
    'TMVA',
    'EG',
    'RooFit',
    'RooFitCore',
    'RooStats',
    'skim_L1Emulator_dataobjects'
]
Return('env')
```

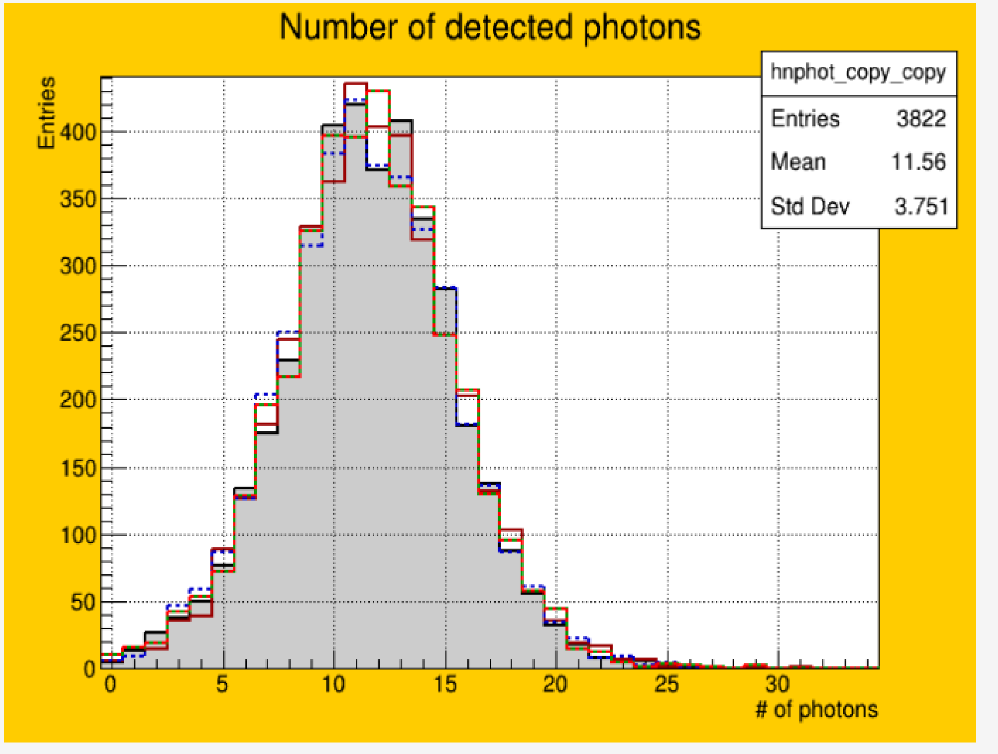
https://bitbucket.org/scons/scons/wiki/SconsVsOtherBuildTools: To sum up, my very subjective opinion is that scons is a better idea, but CMake has a stronger implementation

# Software Quality Control

## Automated checks:

- code style
- gcc/clang/icc
- cppcheck,
  clang static analyzer
- unit/execution tests
- Doxygen
- geometry overlaps
- valgrind memcheck
- execution time and
  output size monitoring
- high level validation plots
  using simulated samples

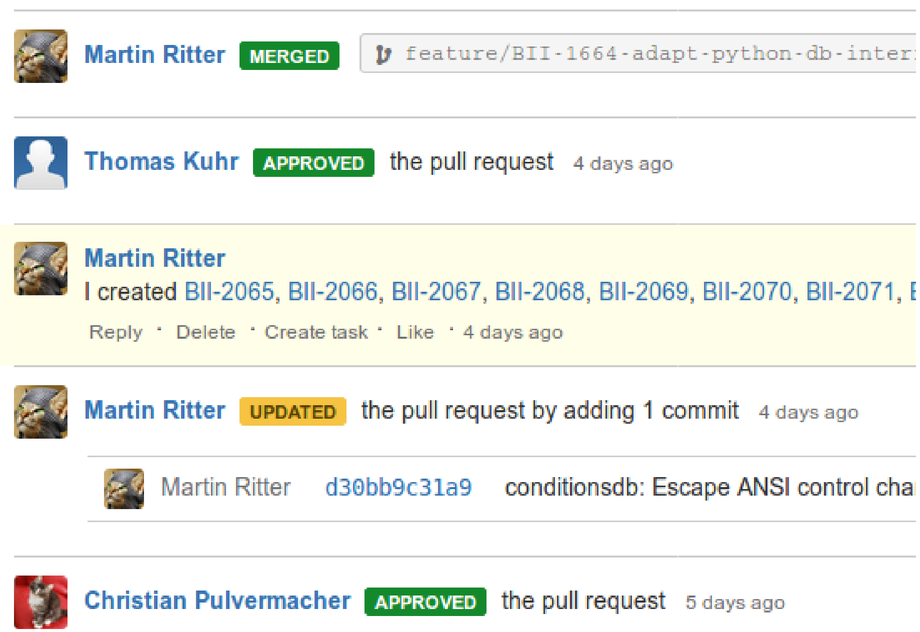
**Plot: ARICHValidate_hnphot**

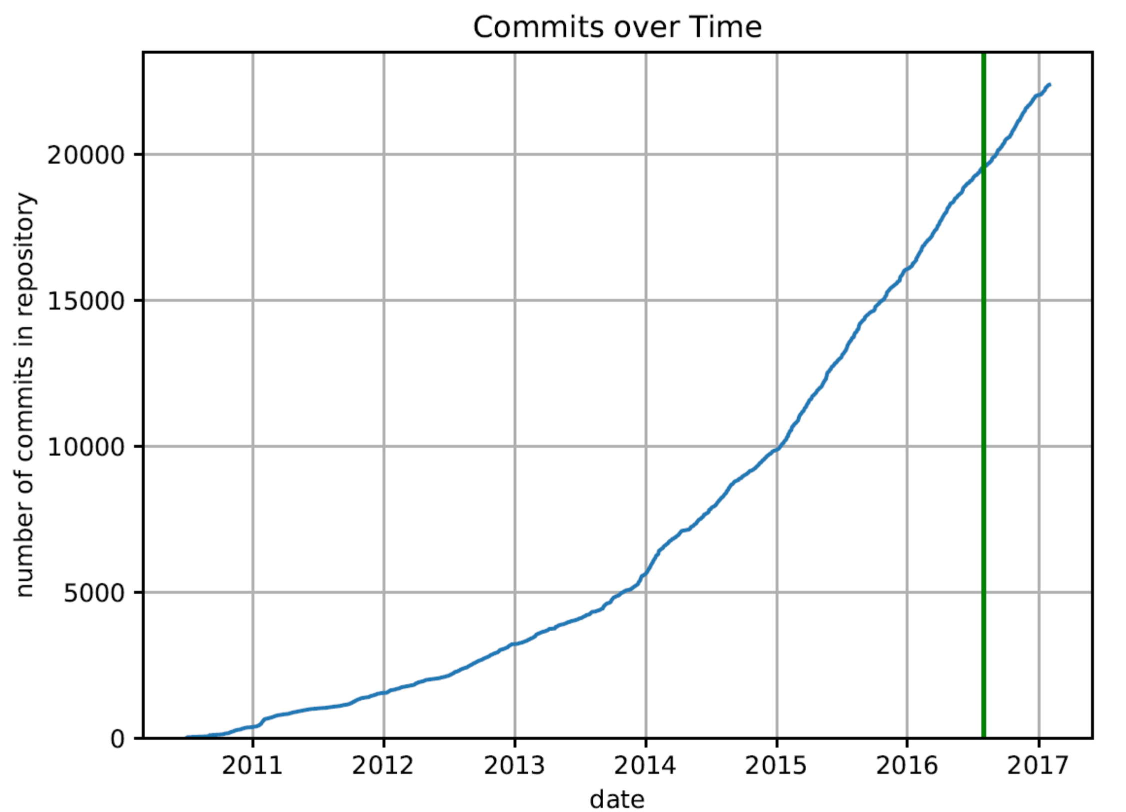



**Package:** arich
**Description:** Number of detected photons in a 3#sigma band around the expected Cherenkov angle, for #pi with momenta 3.0-3.5 GeV.
**ROOT file:** ARICHValidate.root
**Chi^2 Test:** Performed Chi^2-Test between reference and 29456 (Chi^2 = 32.79074530336273 NDF = 28 Chi^2/NDF = 1.1710980465486691)
**Comparison p-value:** 0.2435353507685487 (Warning below 1 | Error below 0.01)
**Contact:** luka.santelj@kek.jp
**Check for:** Poissonian peak with average at ~11.5.

# Migration svn → git

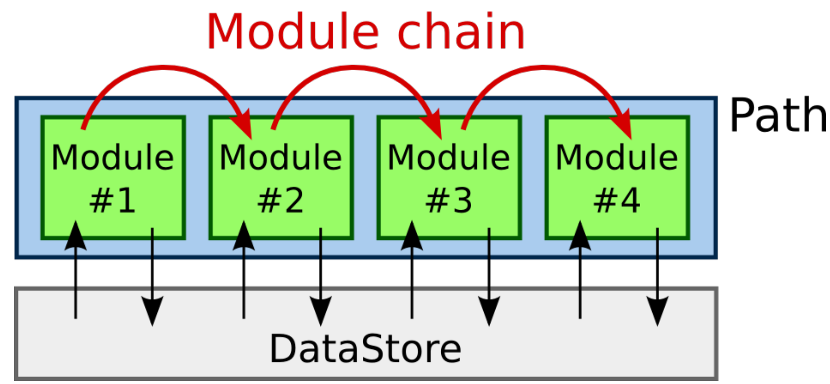
- Belle II decided last year to migrate collaborative services from KEK to DESY
  - ➢ We used that opportunity to switch from svn to git
    - ➔ Adjustment of procedures and tools required



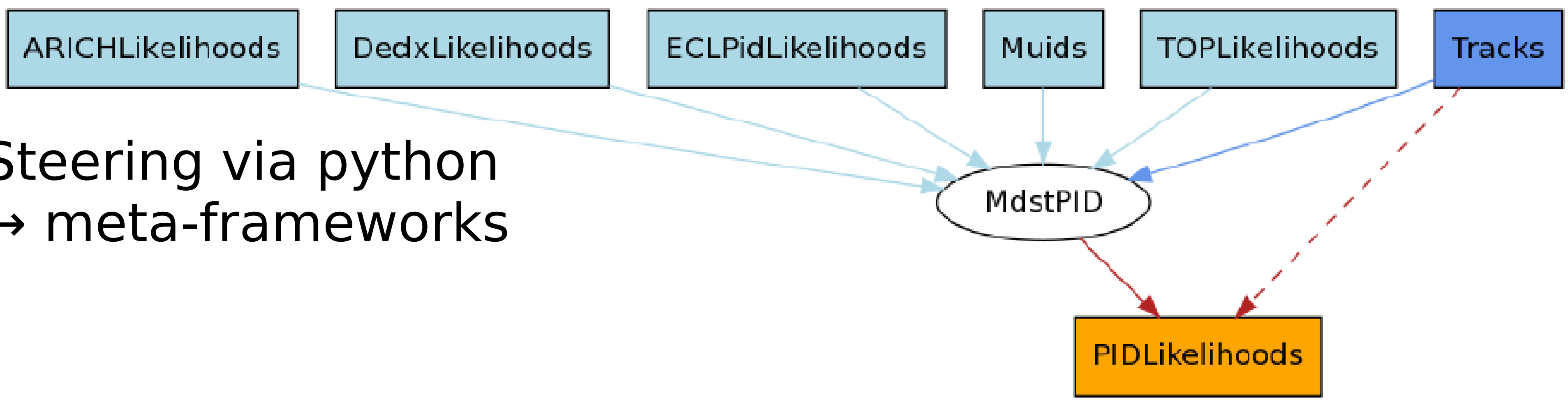



# Framework

- Dynamic loading of modules
- Data exchange via DataStore
- Relations
- Conditions data interface
- Root I/O
- Parallel processing



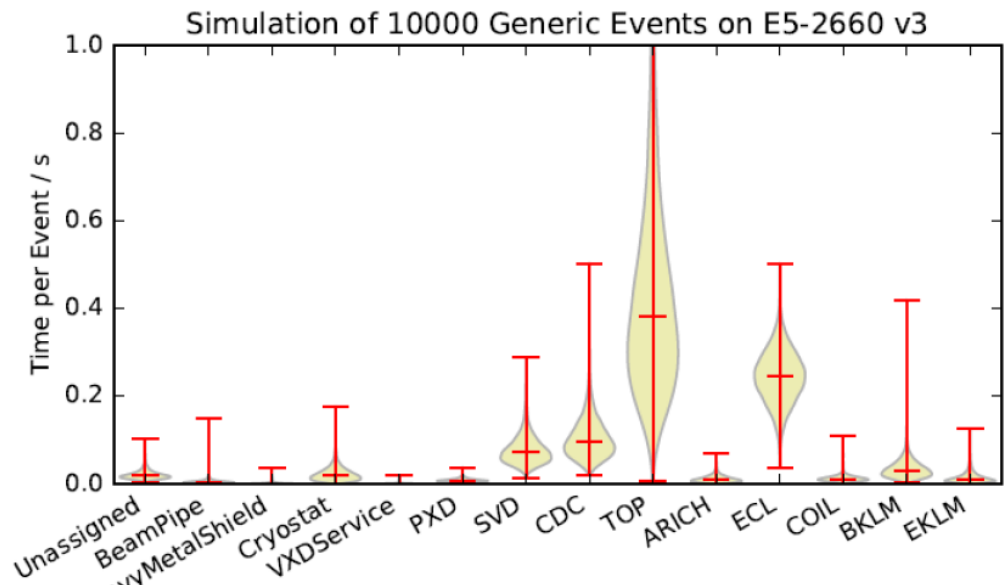

```
StoreArray<Track> tracks;
for (const Track& track: tracks) {
  const PIDLikelihood* pid =
    track->getRelated<PIDLikelihood>();
}
```

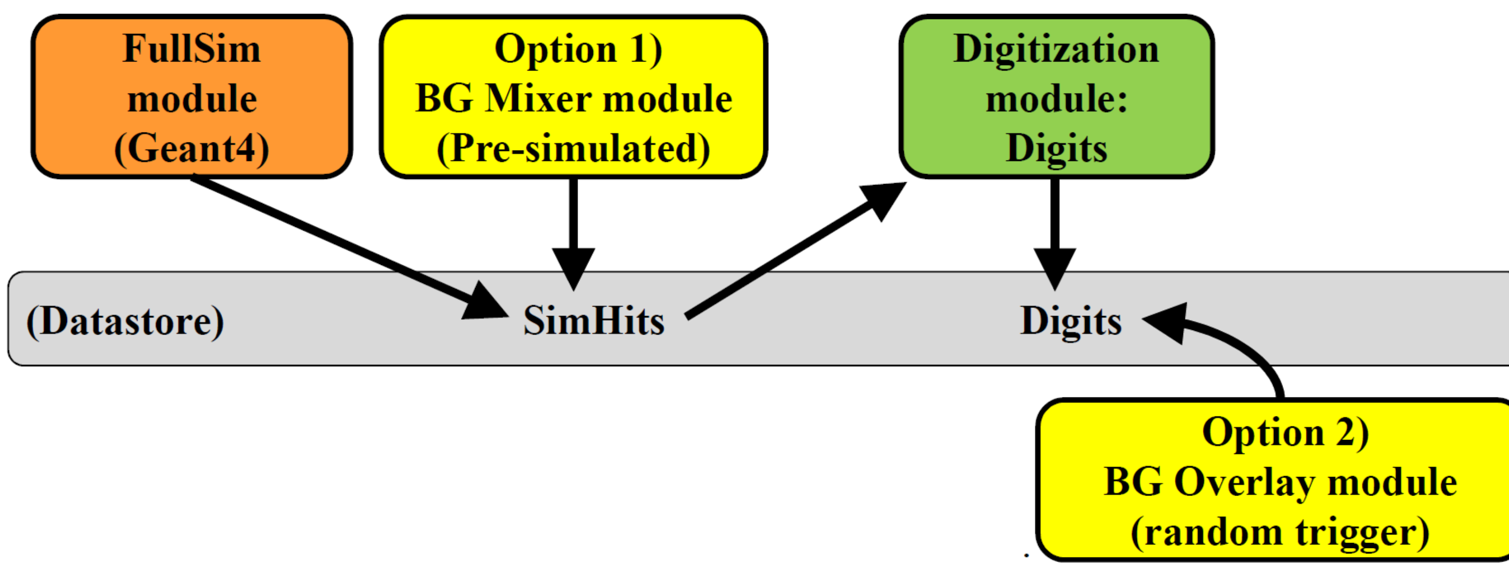



- Steering via python
  → meta-frameworks

# Simulation

- Detector geometry implemented in Geant4
- Parameters obtained from xml file/database
- Energy deposits stored as SimHits
- Digitization in modules
- Background mixing
- Back-ground overlay

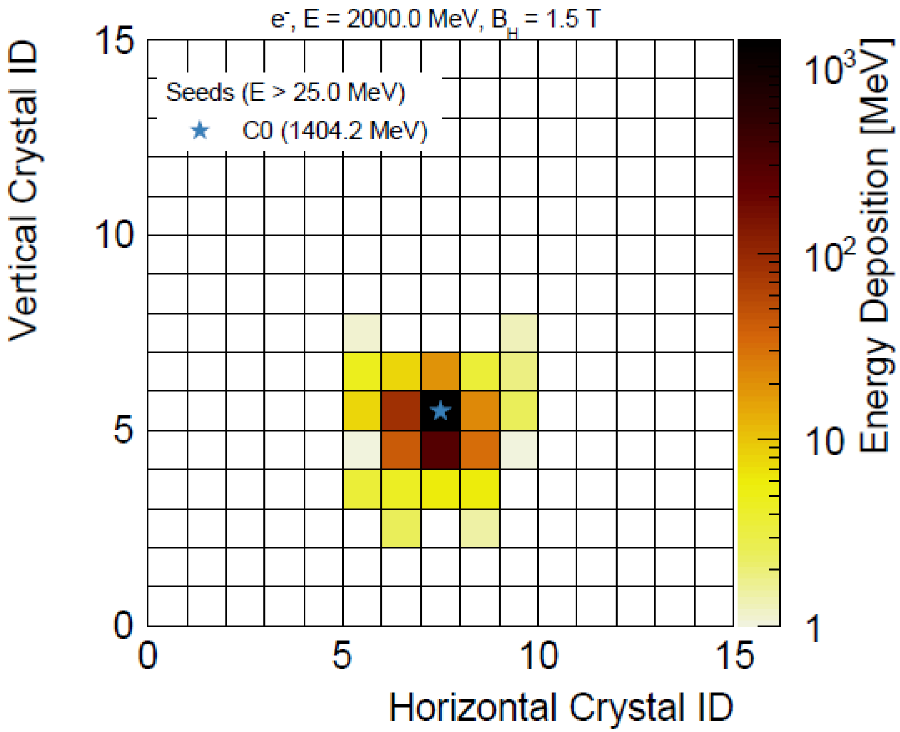


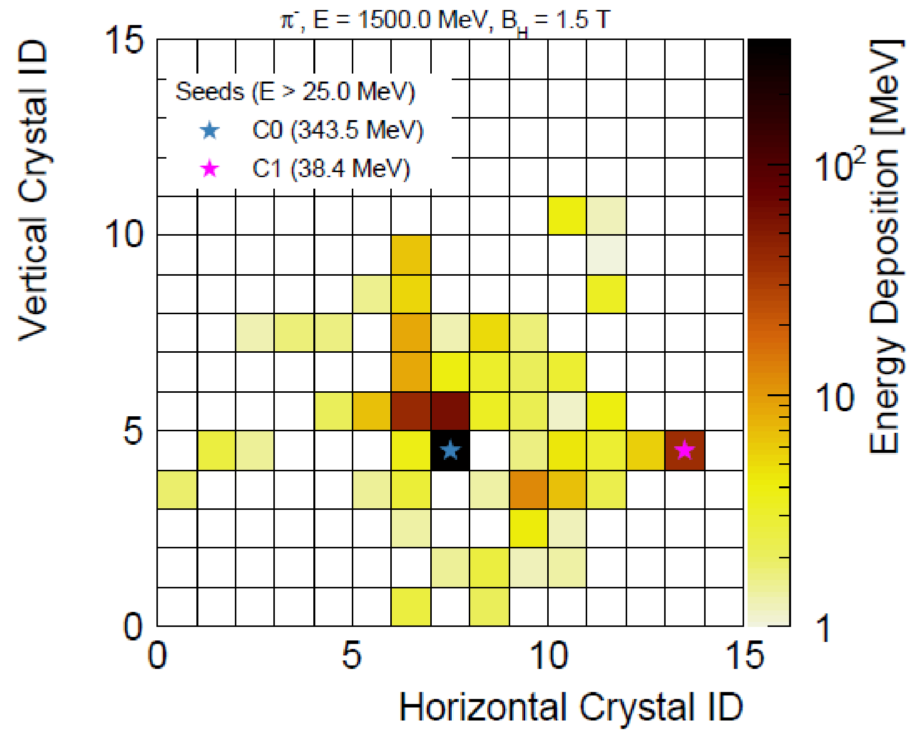



# ECL Reconstruction

- Higher background level than at Belle/BaBar requires development of new clustering algorithm

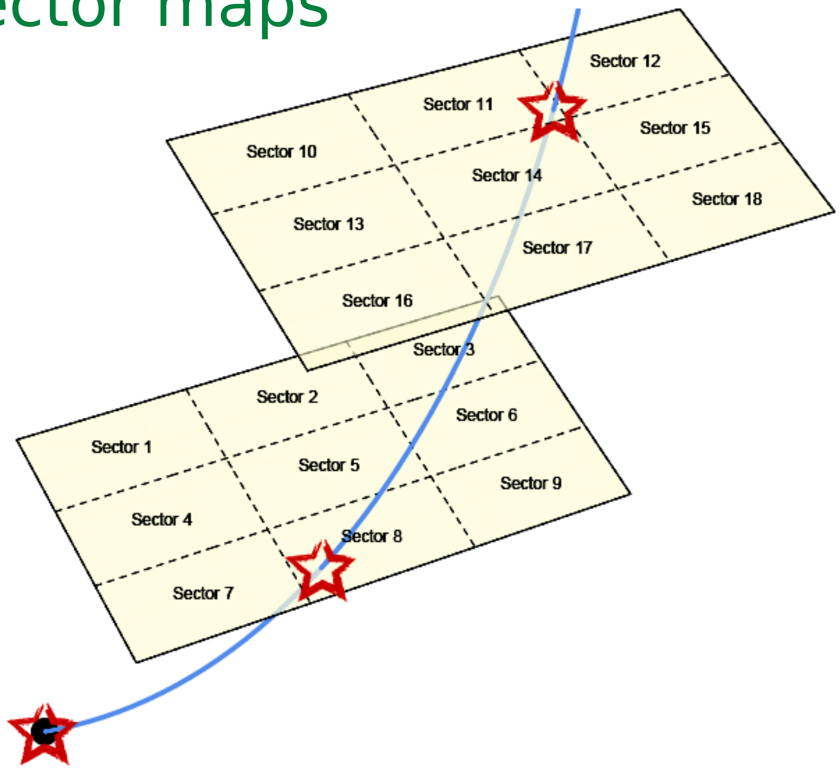
→ Hypothesis dependent reconstruction

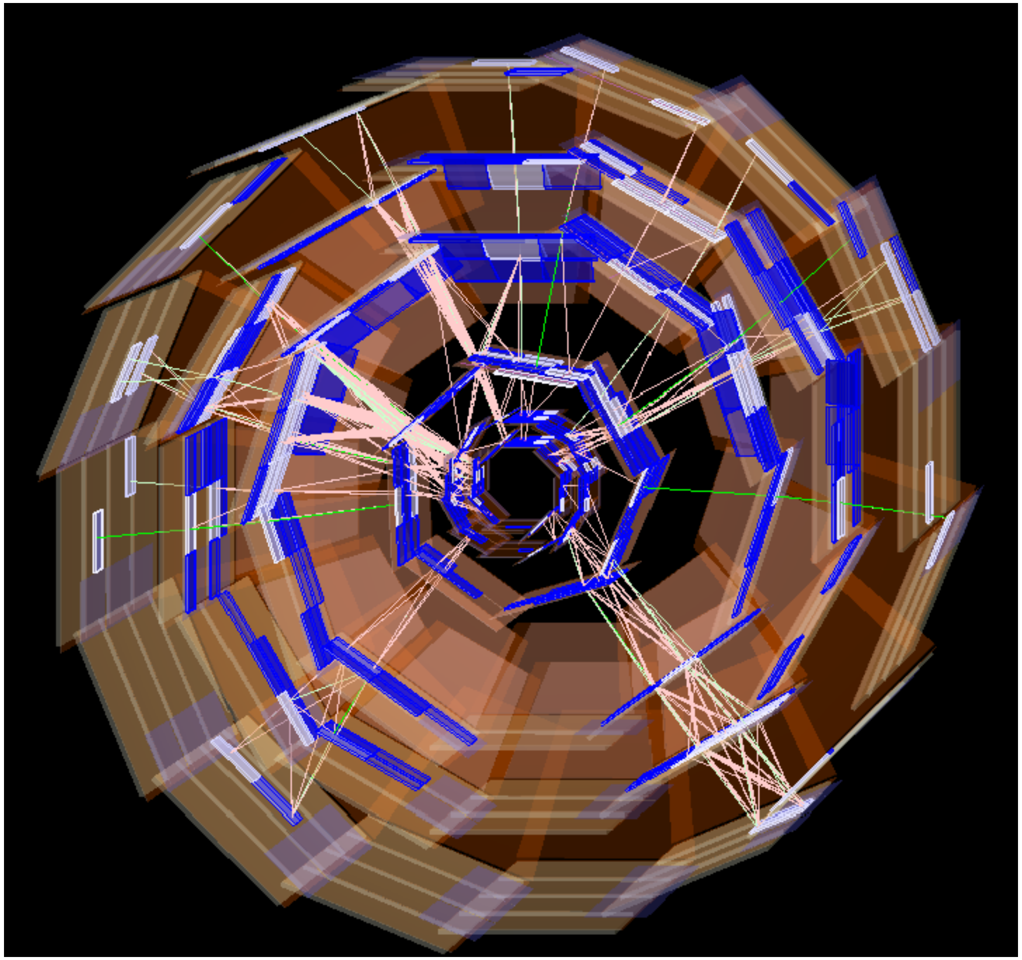


# Tracking

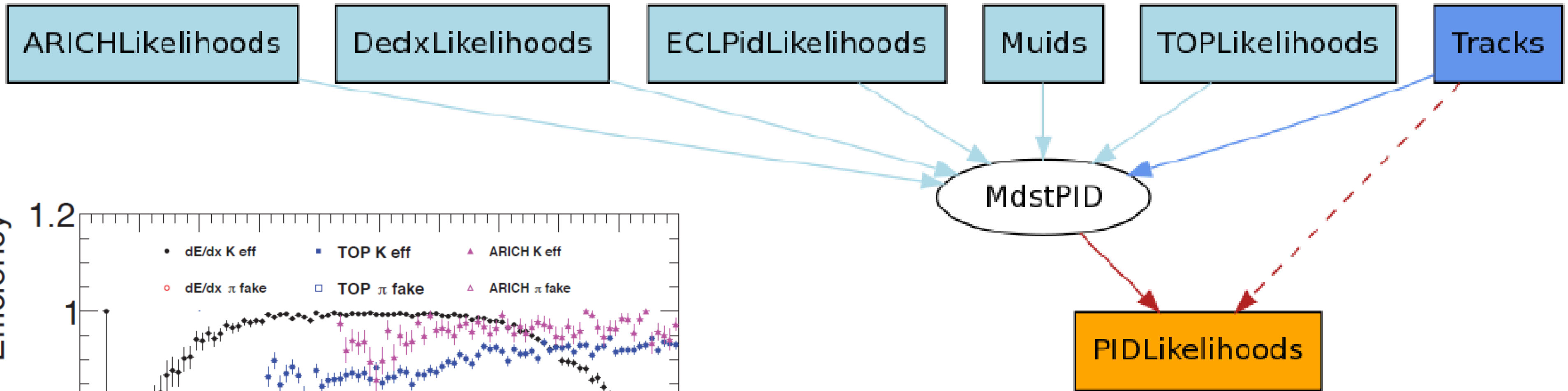
- Combinatorial problem of track finding in the vertex detector
- → Sector maps

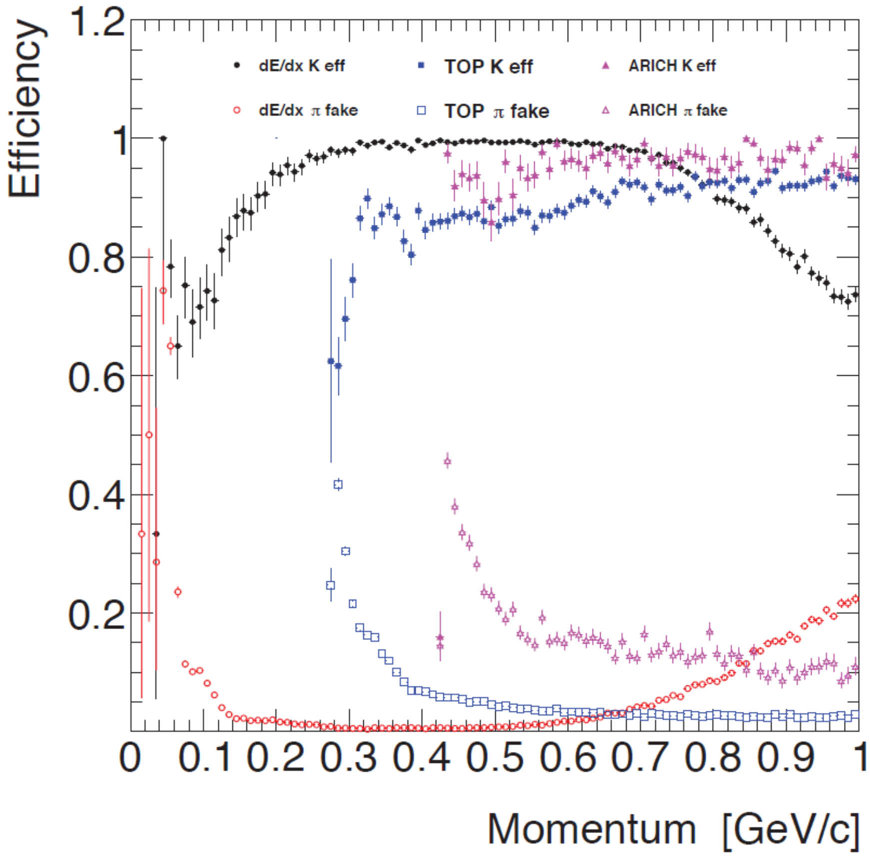


- ✗ No symmetries to be exploited

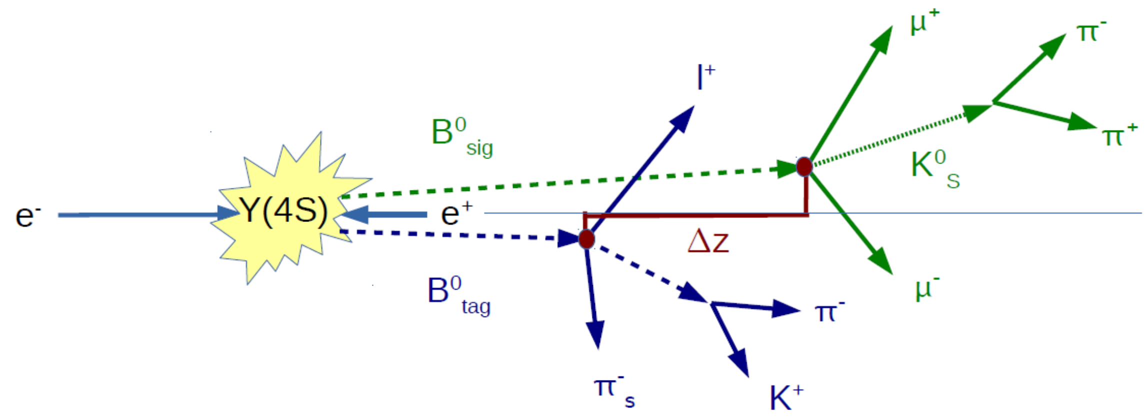
# Charged Particle Identification




- Neyman Pearson lemma
- Likelihood for each detector: L(detector response|part. type)
- Combination: product of likelihoods
- Probability can be calculated with analysis dependent priors

# Modular Analysis



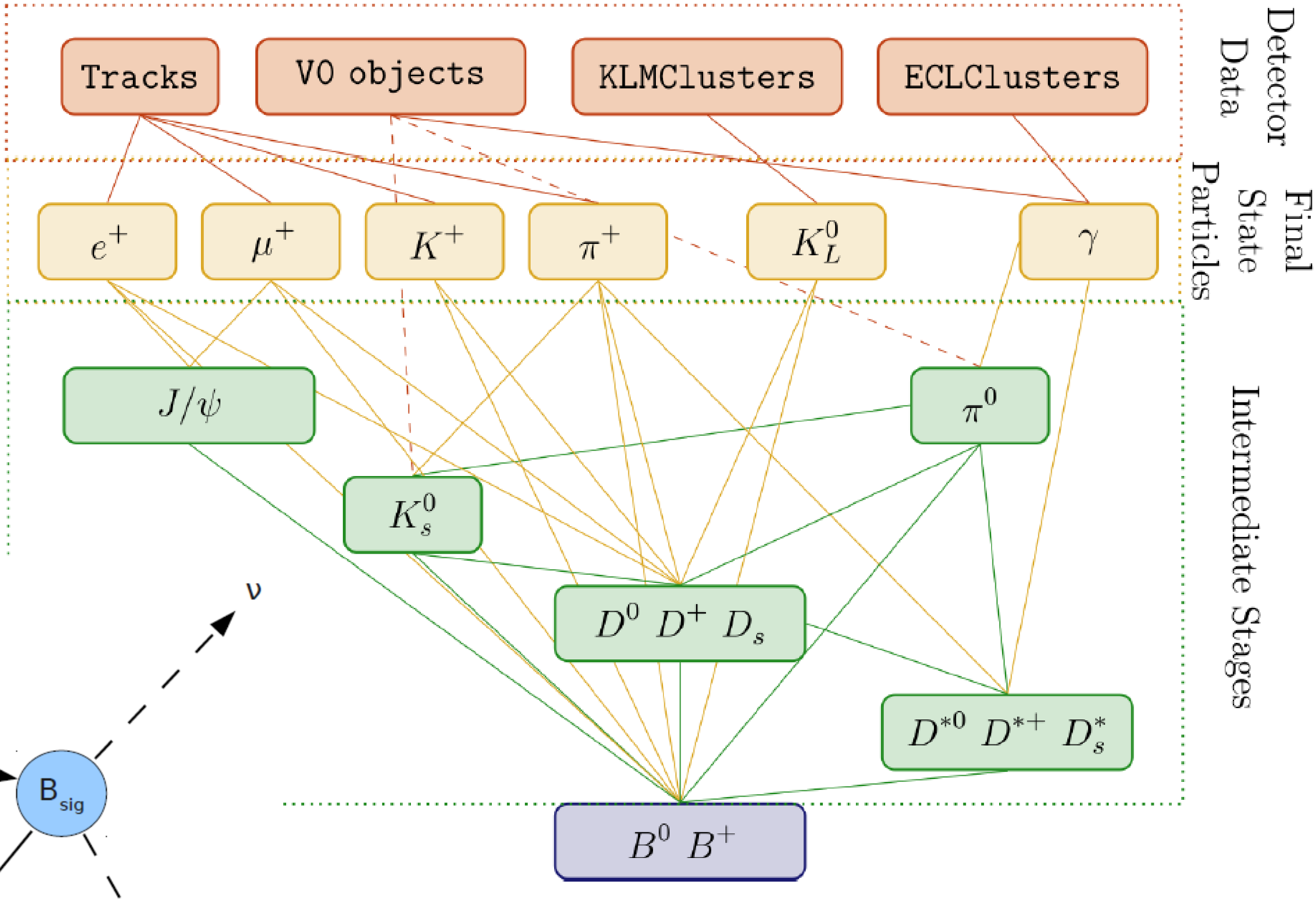

```
inputMdst(...)

# create "mu+:loose" ParticleList (and c.c.)
stdLooseMu()

# create Ks -> pi+ pi- list from V0
# keep only candidates with 0.4 < M(pipi) < 0.6 GeV
fillParticleList('K_S0:pipi', '0.4 < M < 0.6')

# reconstruct J/psi -> mu+ mu- decay
# keep only candidates with 3.0 < M(mumu) < 3.2 GeV
reconstructDecay('J/psi:mumu -> mu+:loose mu-:loose', '3.0 < M < 3.2')

# reconstruct B0 -> J/psi Ks decay
# keep only candidates with 5.2 < M(J/PsiKs) < 5.4 GeV
reconstructDecay('B0:jspiks -> J/psi:mumu K_S0:pipi', '5.2 < M < 5.4')

# perform B0 kinematic vertex fit using only the mu+ mu-
# keep candidates only passing C.L. value of the fit > 0.0 (no cut)
vertexRave('B0:jspiks', 0.0, 'B0 -> [J/psi -> ^mu+ ^mu-] K_S0')

# build the rest of the event associated to the B0
buildRestOfEvent('B0:jspiks')

# perform MC matching (MC truth asociation)
matchMCTruth('B0:jspiks')

# calculate the Tag Vertex and Delta t (in ps)
# breco: type of MC association.
TagV('B0:jspiks', 'breco')

# create and fill flat Ntuple with MCTruth, kinematic information and D0 FlightInfo
toolsDST = ['EventMetaData', '^B0']
toolsDST += ['MCTruth', '^B0 -> [^J/psi -> ^mu+ ^mu-] [^K_S0 -> ^pi+ ^pi-]']
toolsDST += ['Vertex', '^B0 -> [^J/psi -> mu+ mu-] [^K_S0 -> pi+ pi-]']
toolsDST += ['DeltaT', '^B0']
toolsDST += ['MCDeltaT', '^B0']

# write out the flat ntuples
ntupleFile('B2A410-TagVertex.root')
ntupleTree('B0tree', 'B0:jspiks', toolsDST)
```

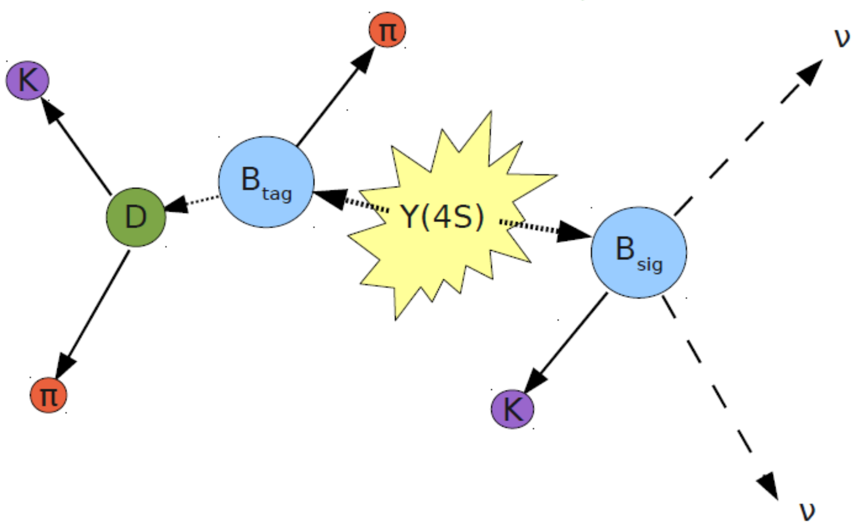
- Analysis on steering file level using decay strings

- ✔ Particle reconstruction and selection
- ✔ MC matching
- ✔ Vertex fits
- ✔ Flavor tagging
- ✔ Continuum suppression

# Full Event Interpretation

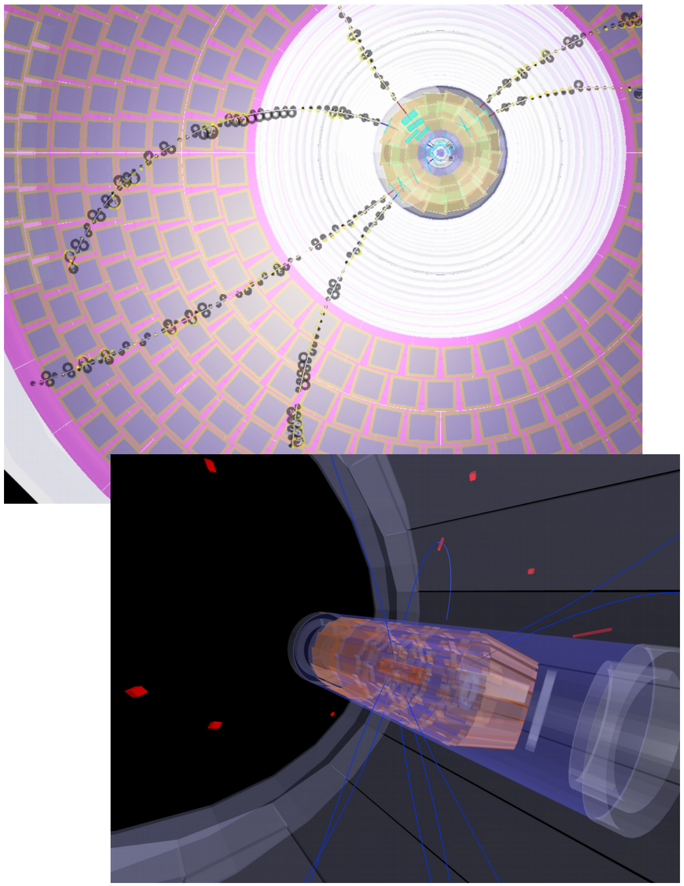
- Huge number of B meson decay modes
  - → Hierarchical reconstruction
  - → Multivariate classifiers



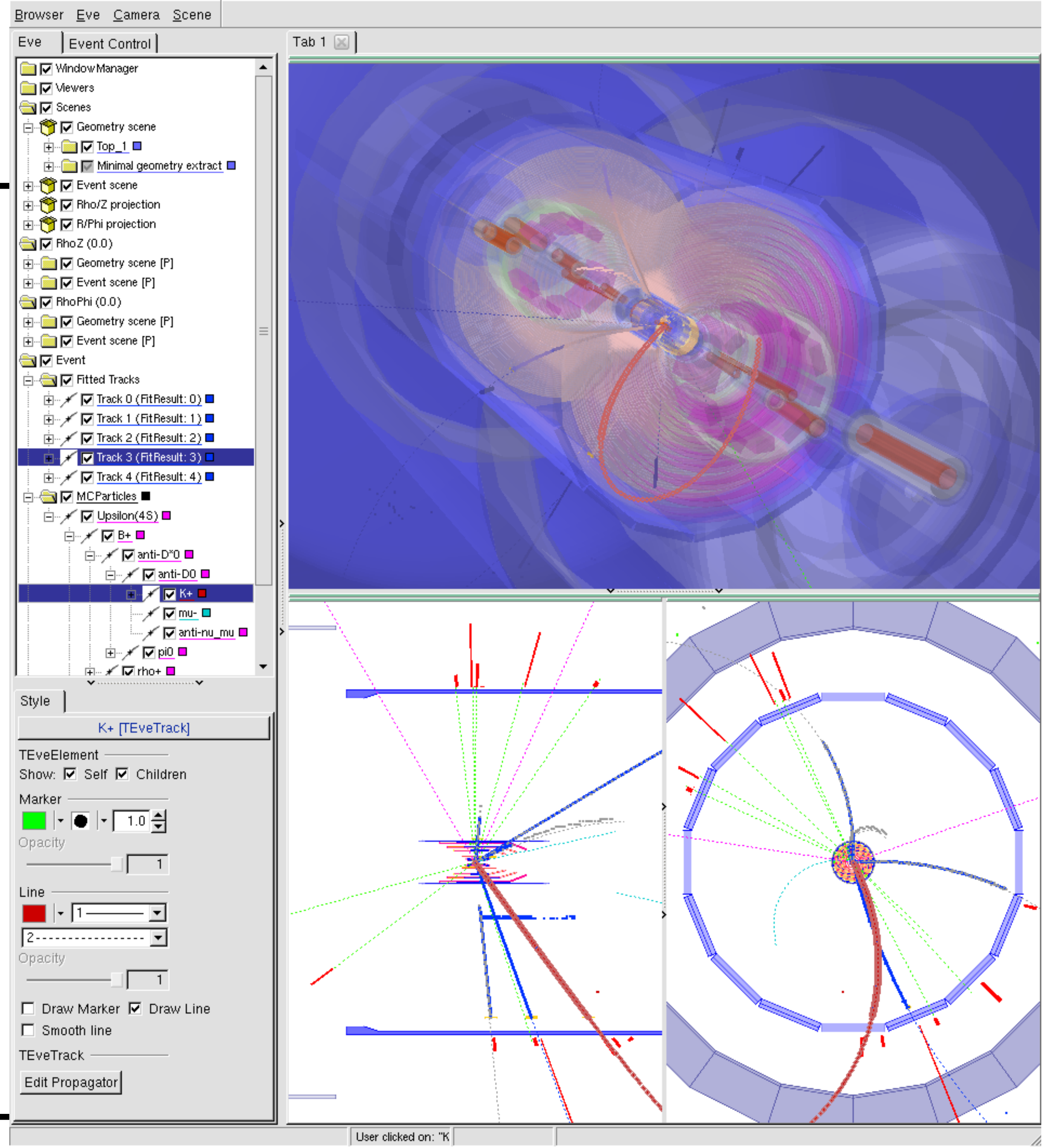




→ Tools for analysis specific training of classifiers

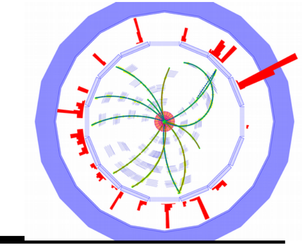
# Event Display



- Virtual reality: https://vimeo.com/185549878



# Summary



- Full potential of Belle II detector components can only be exploited if complemented by corresponding simulation and reconstruction algorithms
- Large data volume requires huge computing resources
- ➔ Challenge: algorithms with high physics performance and low computing resource demand
- ✔ State of the art development tools and various software quality monitoring measures used at Belle II
- ✔ Significant improvements compared to Belle achieved
- ✔ On track for delivering software for first physics data

➔ Take home message: Consider implications on software and computing resources already at detector design stage