Introduction to B2BII

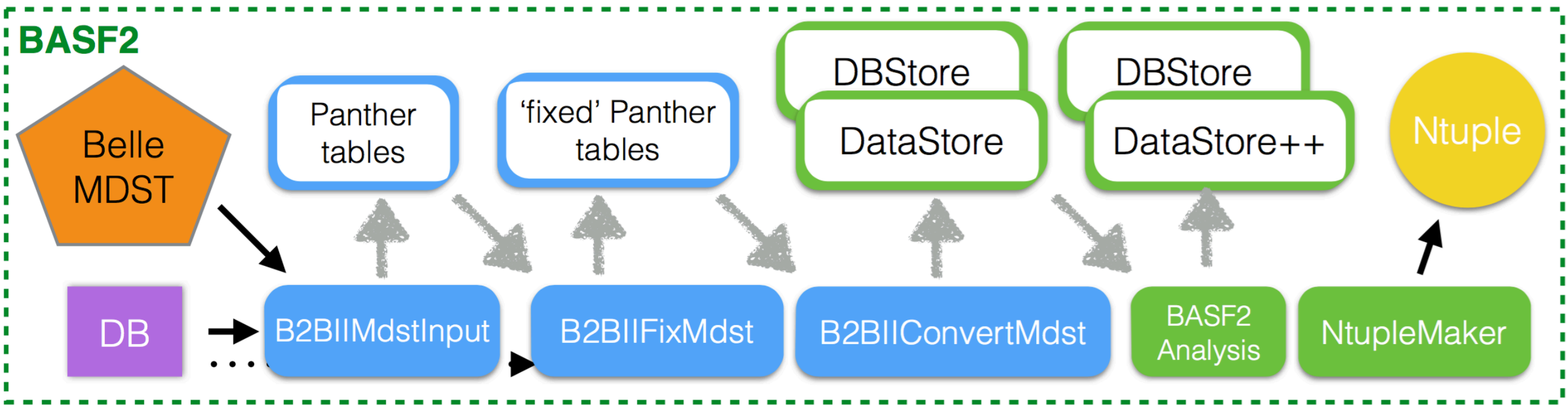
Chia-Ling Hsu
University of Sydney
Belle II Starter Kit Workshop
30th Jan ~ 1st Feb 2020

Outline

- ✦ What's B2BII
- ✦ Introduction to Belle experiment
  - ✦ Belle Data Stream
    - ✦ MDST structure
    - ✦ Physics skims
    - ✦ How to find Generic MC and Data?
- ✦ MDST conversion in B2BII
  - ✦ Particle identification at Belle
  - ✦ Belle-specific variables
- ✦ Conversion Monitoring

What's B2BII

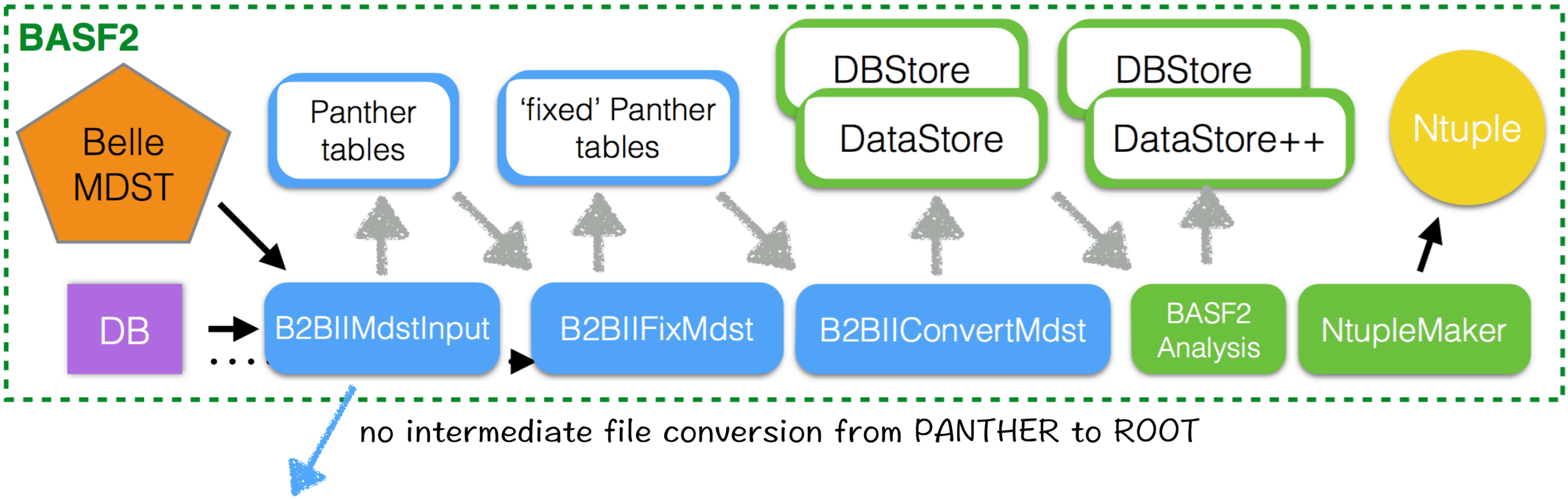
✦ Read and analyze Belle MDST format in basf2.



no intermediate file conversion from PANTHER to ROOT

What's B2BII

✦ Read and analyze Belle MDST format in basf2.

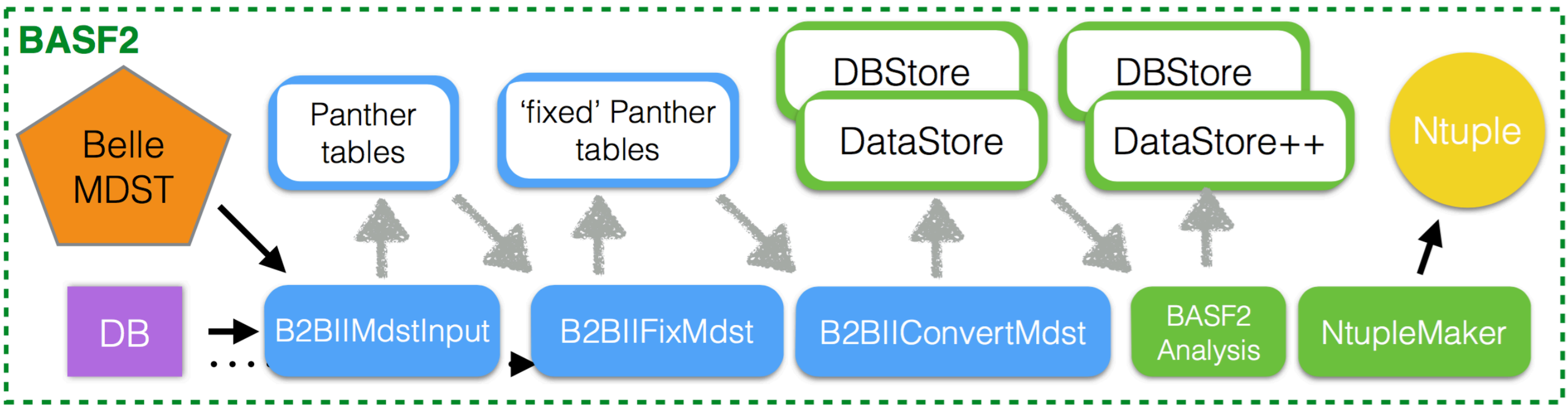


✓ **Reads Belle mdst file**

✓ **All Belle mdst data objects accessible within BASF2**

What's B2BII

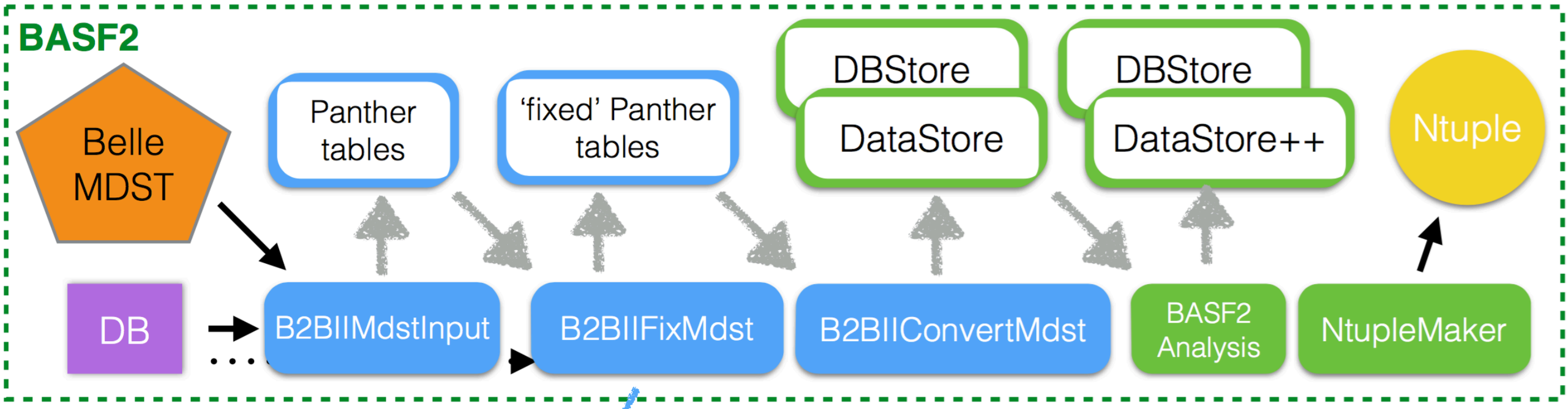
✦ Read and analyze Belle MDST format in basf2.



no intermediate file conversion from PANTHER to ROOT

What's B2BII

✦ Read and analyze Belle MDST format in basf2.

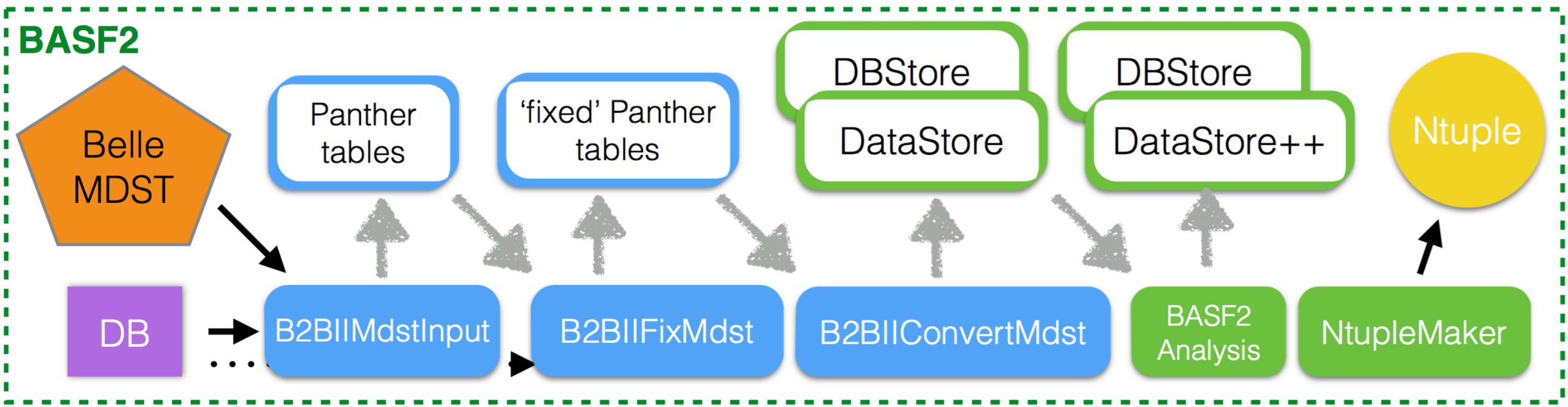


✓ **Performs corrections to the Belle mdst data objects ("fix_mdst")**

✓ **Applies HadronB(J) skim by default**

What's B2BII

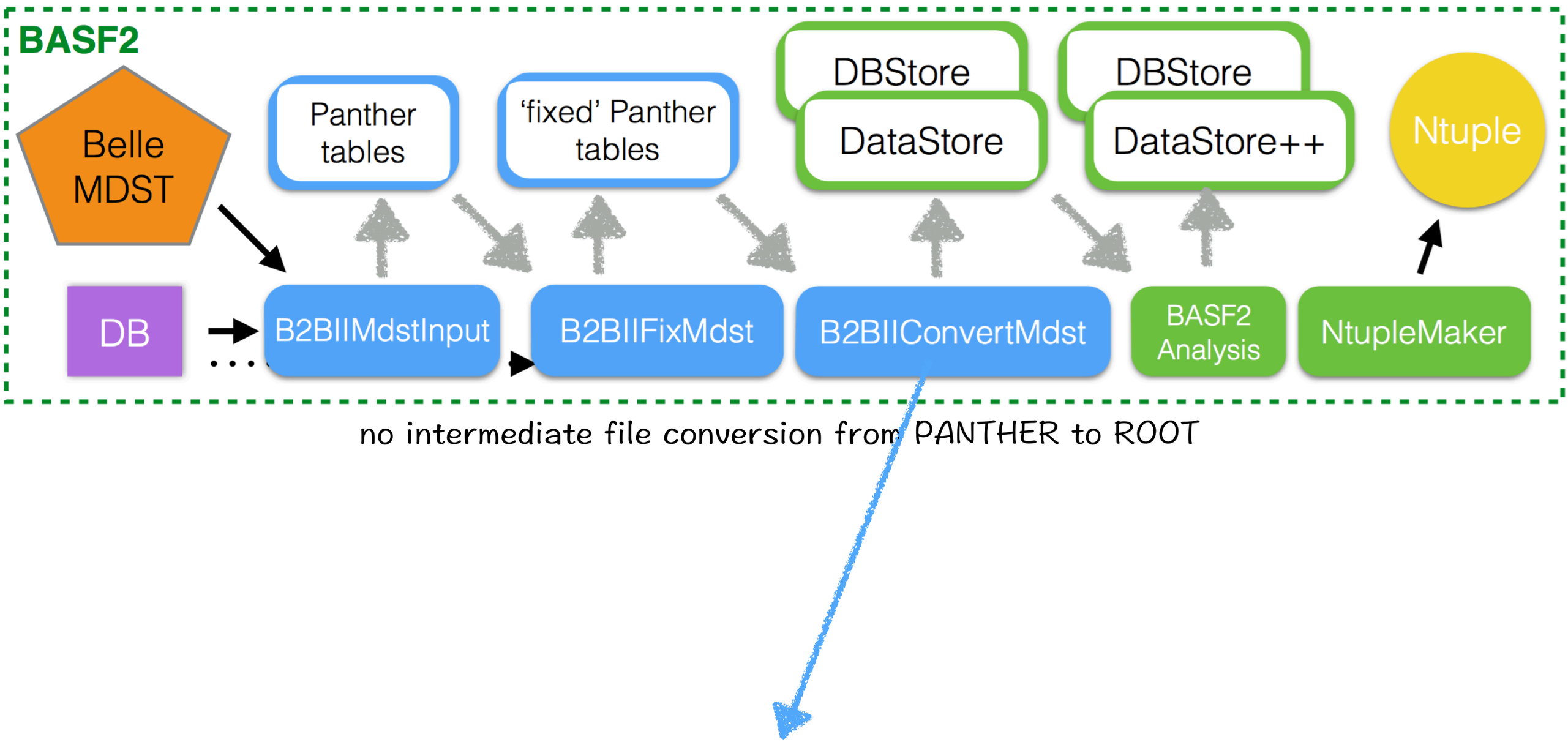
✦ Read and analyze Belle MDST format in basf2.



no intermediate file conversion from PANTHER to ROOT

What's B2BII

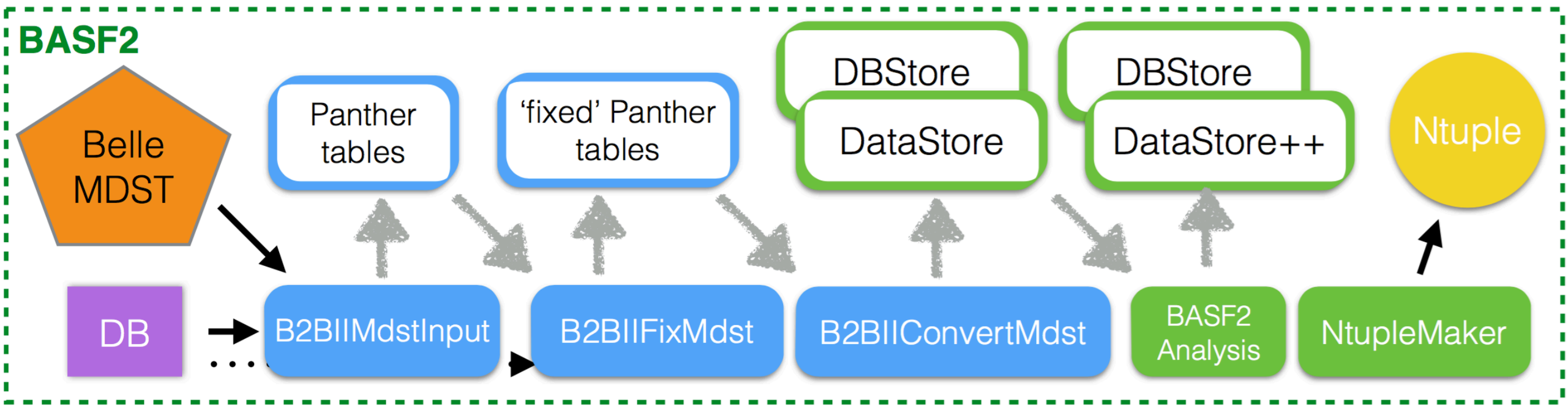
✦ Read and analyze Belle MDST format in basf2.



✓ **Performs Belle mdst to Belle II mdst conversion**

What's B2BII

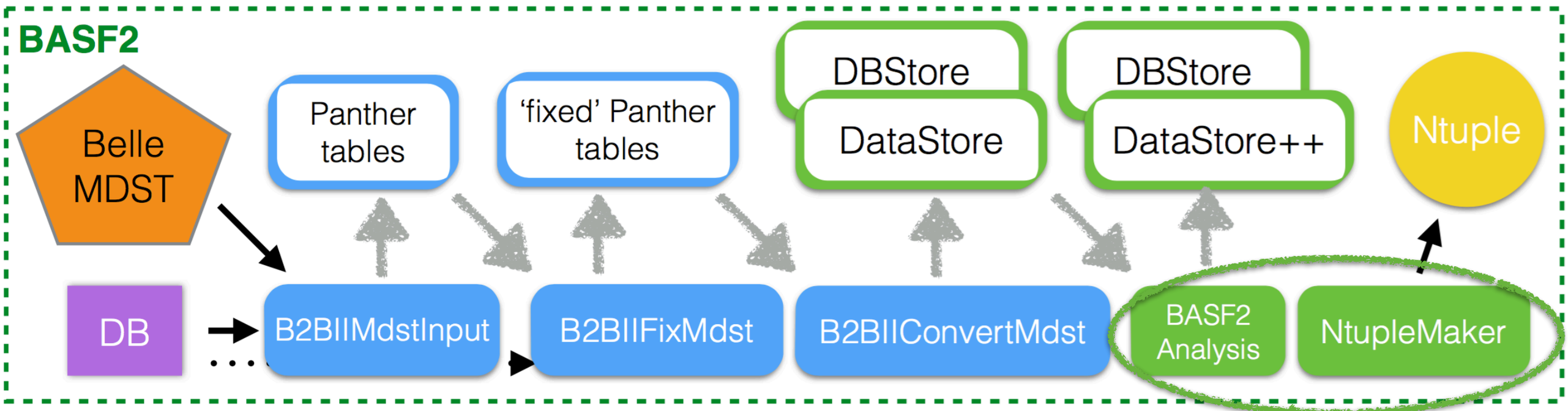
✦ Read and analyze Belle MDST format in basf2.



no intermediate file conversion from PANTHER to ROOT

What's B2BII

✦ Read and analyze Belle MDST format in basf2.

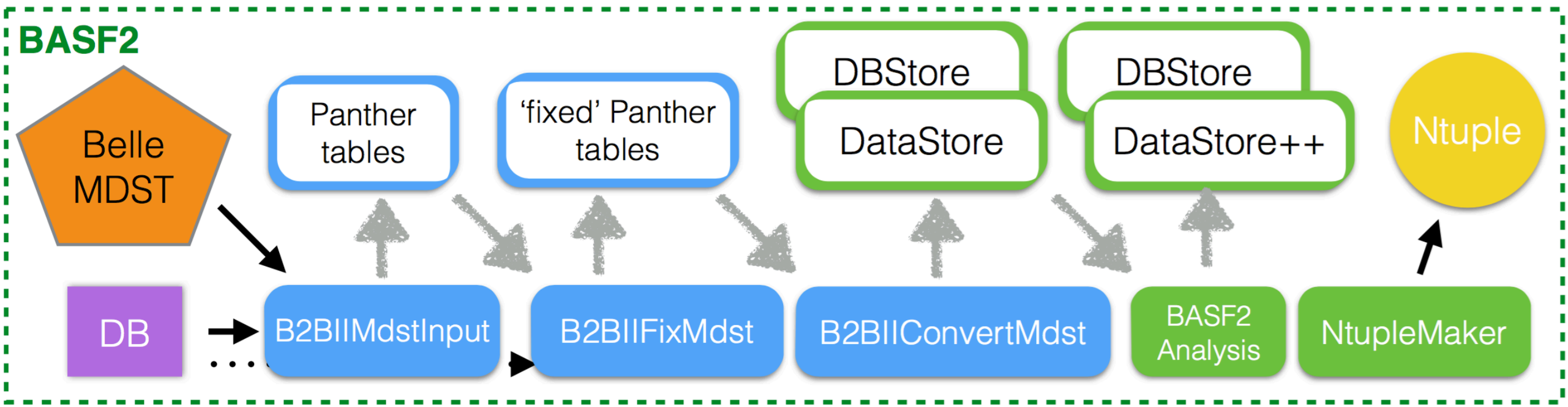


no intermediate file conversion from PANTHER to ROOT

✓ **Any basf2 steering script will work**

What's B2BII

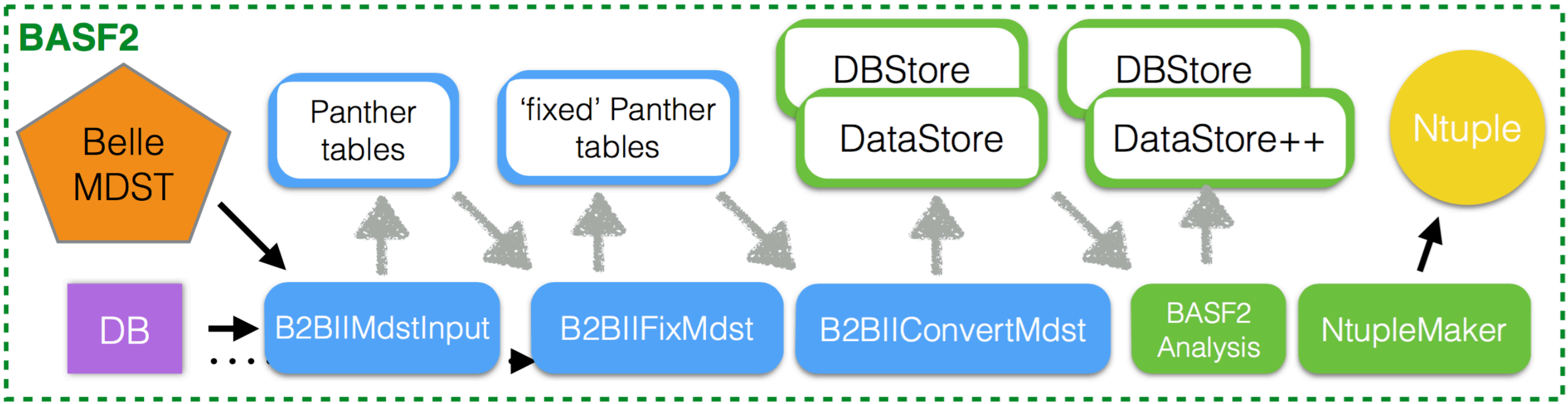
✦ Read and analyze Belle MDST format in basf2.



no intermediate file conversion from PANTHER to ROOT

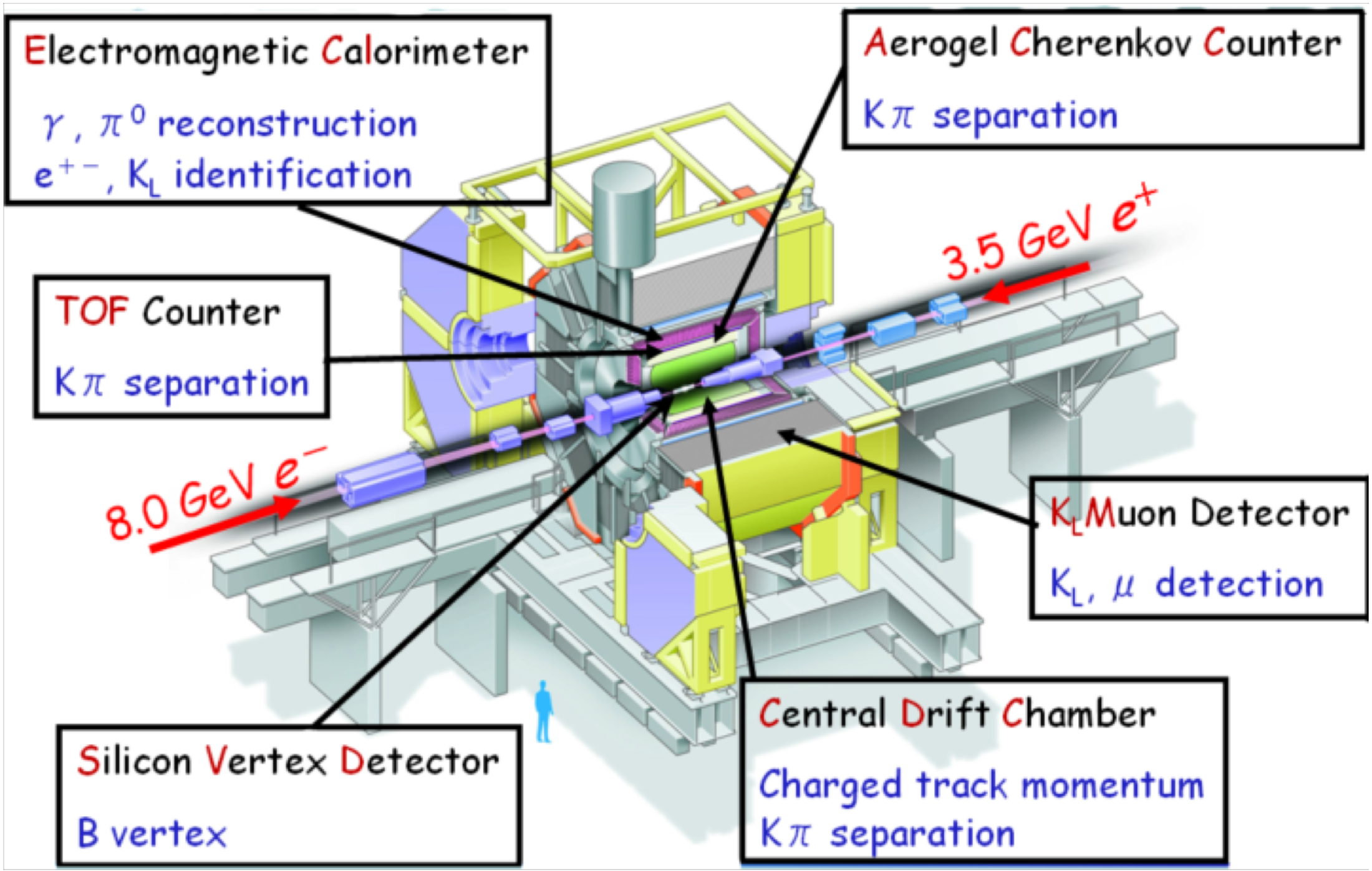
What's B2BII

✦ Read and analyze Belle MDST format in basf2.

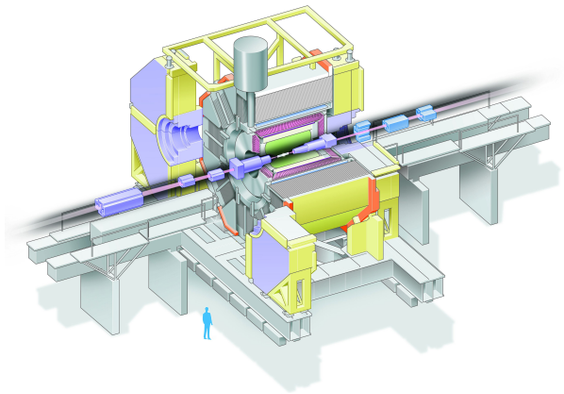


Before going to the details, let's learn what's the difference between Belle and Belle 2

Overview of Belle Experiment



Belle Data Stream



online reconstruction

raw data +
“DST” data

production

(Generic)
Monte Carlo

Evtgen
(pgen file)

GSIM

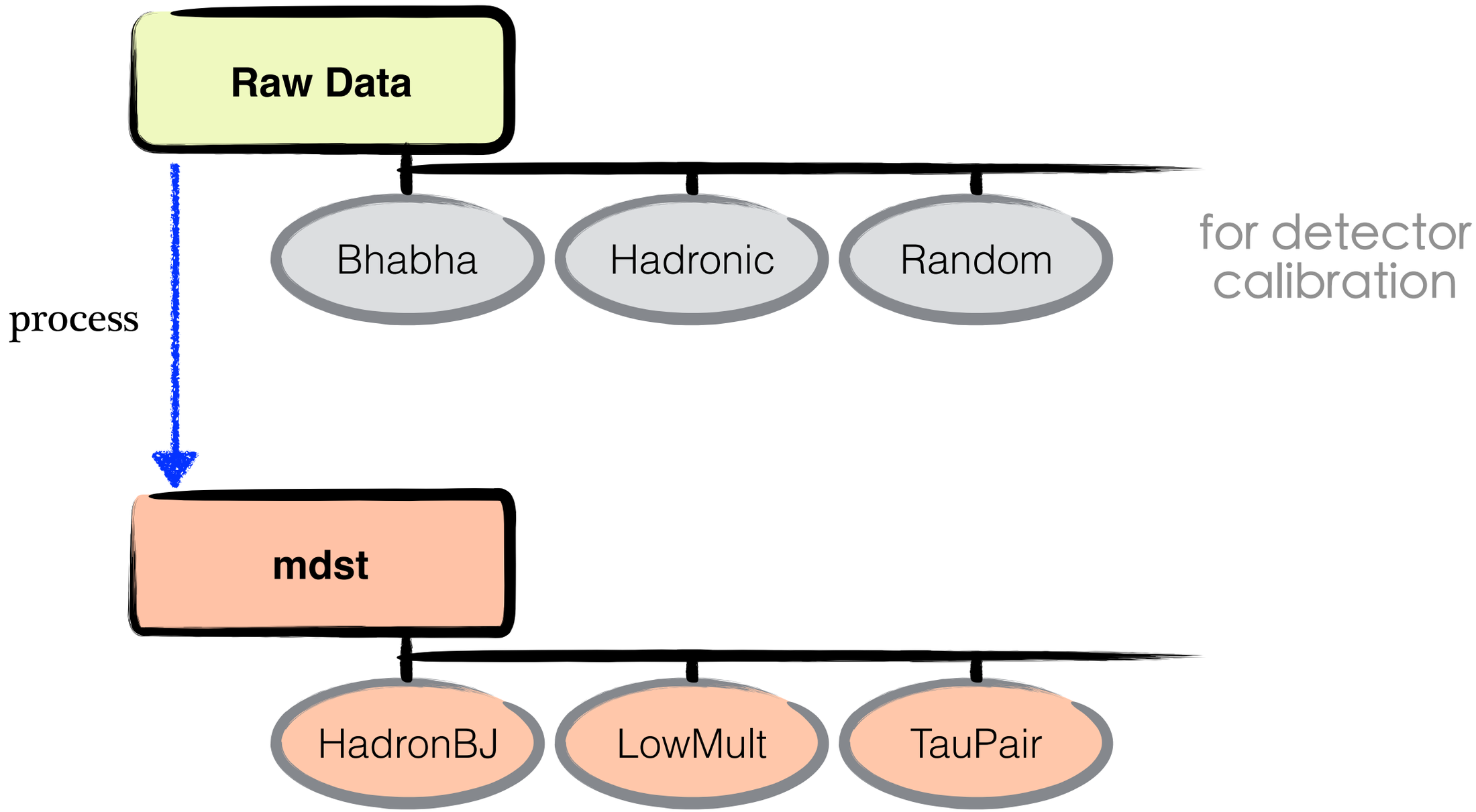
MC production

“MDST” data
(4-vector, Particle ID … etc.)

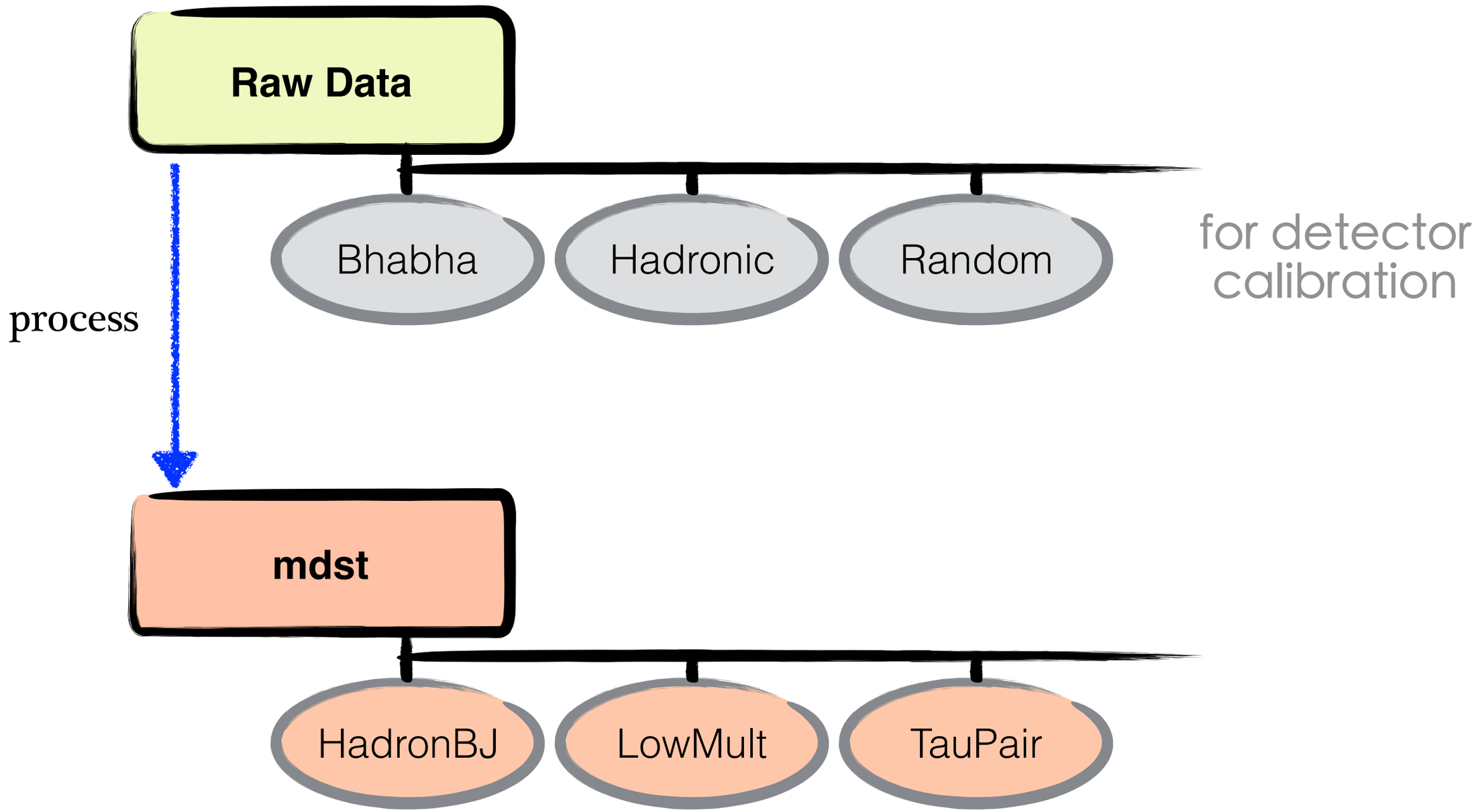
Users’ analyses

Ntuple…

Belle Data Stream



Belle Data Stream



✓ **These subsets of data are called "Skim".**

Skims in Belle

- ✦ Most B analyses use “**HadronB(J)**” samples.
  - ✓ HadronB (Standard hadronic selection) + psiskim (#J/psi >0)
  - ✓ fix_mdst applies HadronB(J) selection by default.
- ✦ Additional samples for low-multiplicity events “**LowMult**” and “**tau_skimA(B)**”.
  - ✓ Necessary for tau, two-photon group or ISR analysis.
  - ✓ Some of them can be picked up by evtcls flags.

    https://belle.kek.jp/secured/wiki/doku.php?id=software:event_classification

- ✦ non-4S data use “**ypipi**” skim or “**all-mdst**” (unskimmed).

MDST structure

✦ In a typical official mdst file, it contains the following tables:

```
BELLE_EVENT  (   1) BGTBL_INFO   (   1) EVTCLS_FLAG  (   1) EVTCLS_FLAG2 (   1) EVTCLS_HADRON_CHARGED          (   5)
EVTCLS_HADRON_INFO               (   1) EVTCLS_HADRON_NEUTRAL             (   4) EVTCLS_HADRONIC_FLAG           (   1)
EVTVTX_PRIMARY_VERTEX            (   1) EVTVTX_TRK    (   5) GEN_HEPEVT   (  90) GEN_PAKEVT    (  28) GEN_PAKVTX    (   6)
GSIM_RAND    (   1) L4_SUMMARY   (   1) MCTYPE        (   1) MDST_ACC     (  10) MDST_CDC_FIT  (  25) MDST_CHARGED  (   9)
MDST_ECL     (  28) MDST_ECL_AUX (  28) MDST_ECL_CR   (  25) MDST_ECL_TRK (   7) MDST_EVENT_ADD                  (   1)
MDST_GAMMA   (   6) MDST_KLM_CLUSTER                  (   7) MDST_KLM_CLUSTER_HIT          ( 176) MDST_KLM_MU   (   9)
MDST_KLM_MU_EX                   (   9) MDST_KLONG    (   3) MDST_MUID    (   9) MDST_PI0      (   4) MDST_SIM_ECL  (  28)
MDST_SIM_TRK (   9) MDST_SIM_TRK_EXTRA                 (   1) MDST_SVD_HIT (  46) MDST_SVD_HIT_EXTRA            (   8)
MDST_TOF     (   2) MDST_TRG     (   1) MDST_TRK      (   9) MDST_TRK_ADD (   9) MDST_TRK_EXTRA                (   1)
MDST_TRK_FIT (  26) MDST_VEE2    (   4) MDST_VEE2_EXTRA                          (   1) MDST_VEE_DAUGHTERS         (   2)
MDST_VEE_DAUGHTERS_ADD           (   2) RECCDC_TIMING(   6)
----- Crate=   0, Event No.=       1 -----
```

GSIM* and GEN_* only exist in MC.

✦ Information of charged and neutral particles are stored in the following tables:

`MDST_Charged`: charged tracks reconstructed at CDC+SVD

`MDST_Vee2`: "V"-particles reconstructed from 2 charged tracks.

(i.e. K_S, Λ, converted γ)

`MDST_Gamma`: photons reconstructed at ECL

`MDST_pi0`: π^0 reconstructed from 2 photons. Mass constraint fit is applied.

`MDST_Klong`: K_L etc reconstructed at KLM

Official Belle MC

- ✦ Generic MC:
  - ✓ **charged, mixed**, **uds, charm**, **bsbs, nonbsbs**
  - ✓ run-dependent beam energy
  - ✓ 1 stream MC ~ Belle data
- ✦ Special MC:
  - ✓ mixedrare, chargedrare, jpsi, psi2s, ulnu…
  - ✓ run-independent beam energy
  - ✓ 20 ~ 100 times of Y(4S) data
- ✦ Tau MC:
  - ✓ run-dependent beam energy
  - ✓ Available at Nagoya server, historical reason?
  - ✓ Copying to KEKCC now

https://belle.kek.jp/secured/wiki/doku.php?id=software:data_search

Official Belle MC

- ✦ Generic MC:
  - ✓ charged, mixed, uds, charm, bsbs, nonbsbs
    only b→c transition
  - ✓ run-dependent beam energy
  - ✓ 1 stream MC ~ Belle data
- ✦ Special MC:
  - ✓ mixedrare, chargedrare, jpsi, psi2s, ulnu…
  - ✓ run-independent beam energy
  - ✓ 20 ~ 100 times of Y(4S) data
- ✦ Tau MC:
  - ✓ run-dependent beam energy
  - ✓ Available at Nagoya server, historical reason?
  - ✓ Copying to KEKCC now

https://belle.kek.jp/secured/wiki/doku.php?id=software:data_search

Official Belle MC

- ✦ Generic MC:
  - ✓ **charged, mixed**, **uds, charm**, **bsbs, nonbsbs**
  - ✓ run-dependent beam energy
  - ✓ 1 stream MC ~ Belle data
- ✦ Special MC:
  - ✓ mixedrare, chargedrare, jpsi, psi2s, ulnu…
  - ✓ run-independent beam energy
  - ✓ 20 ~ 100 times of Y(4S) data
- ✦ Tau MC:
  - ✓ run-dependent beam energy
  - ✓ Available at Nagoya server, historical reason?
  - ✓ Copying to KEKCC now

https://belle.kek.jp/secured/wiki/doku.php?id=software:data_search

Official Belle MC

- ✦ Generic MC:
  - ✓ **charged, mixed**, **uds, charm**, **bsbs, nonbsbs**
  - ✓ run-dependent beam energy
  - ✓ 1 stream MC ~ Belle data
- ✦ Special MC:
  - ✓ mixedrare, chargedrare, jpsi, psi2s, ulnu…
  - **Decay tables are out-of-dated**
  - ✓ run-independent beam energy
  - ✓ 20 ~ 100 times of Y(4S) data
- ✦ Tau MC:
  - ✓ run-dependent beam energy
  - ✓ Available at Nagoya server, historical reason?
  - ✓ Copying to KEKCC now

https://belle.kek.jp/secured/wiki/doku.php?id=software:data_search

How to find data

✦ Belle File Search Engine: http://bweb3.cc.kek.jp/

Skim Type:
HadronB(J)
LowMult etc.

Data Type:
on-resonance,
continuum,
5S_scan etc

***Only accessible from within KEK or via VPN**

Conversion in B2BII

Belle

Belle II

MC Information

- Gen_hepevt
- MCParticle

Tracking output

- Mdst_charged
- Mdst_trk
- Mdst_trk_fit
- Track
- TrackFitResult

BeamParameters

- BeamEnergy
- IPProfile
- Collision*
- BeamSpot
- BeamParameters

PID information

- atc_pid/eID/muID
- PIDLikelihood

ECL output

- Mdst_ecl
- Mdst_ecl_aux
- Mdst_gamma
- Mdst_pi0
- ECLCluster

V0

- Mdst_vee2
- Mdst_vee2_daughter
- Mdst_vee2_extra
- V0
- TrackFitResult

Klong

- Mdst_klong
- KLMCluster

Conversion in B2BII

Belle

Belle II

MC Information

- Gen_hepevt
- MCParticle

Tracking output

- Mdst_charged
- Mdst_trk
- Mdst_trk_fit
- Track
- TrackFitResult

BeamParameters

- BeamEnergy
- IPProfile
- Collision*
- BeamSpot
- BeamParameters

PID information

- atc_pid/eID/muID
- PIDLikelihood

ECL output

- Mdst_ecl
- Mdst_ecl_aux
- Mdst_gamma
- Mdst_pi0
- ECLCluster

V0

- Mdst_vee2
- Mdst_vee2_daughter
- Mdst_vee2_extra
- V0
- TrackFitResult

Klong

- Mdst_klong
- KLMCluster

Evtcls

- Evtcls_flags
- ???

Conversion in B2BII

Belle

Belle II

MC Information

- Gen_hepevt
- MCParticle

Tracking output

- Mdst_charged
- Mdst_trk
- Mdst_trk_fit
- Track
- TrackFitResult

BeamParameters

- BeamEnergy
- IPProfile
- Collision*
- BeamSpot
- BeamParameters

PID information

- atc_pid/eID/muID
- PIDLikelihood

ECL output

- Mdst_ecl
- Mdst_ecl_aux
- Mdst_gamma
- Mdst_pi0
- ECLCluster

V0

- Mdst_vee2
- Mdst_vee2_daughter
- Mdst_vee2_extra
- V0
- TrackFitResult

Klong

- Mdst_klong
- KLMCluster

Evtcls

Save as ExtraInfo

- Evtcls_flags
- ???

Particle List for B2BII ana

basf	basf2	In basf2 analysis script
Mdst_charged	Track	*Create ParticleLists (w/ or w/o cuts) of Final State Particles: fillParticleList('K+:mine','atcPIDBelle(3,2)>0.6 and abs(d0)<2 and abs(z0)<4', path=mypath)*
Mdst_gamma	gamma:mdst ECLCluster	*Use already created 'gamma:mdst' ParticleList: cutAndCopyList('gamma:mine,'gamma:mdst','E > 0.1', path=mypath)*
Mdst_pi0	pi0:mdst (daughters link to gamma:mdst)	*Use already created 'pi0:mdst' ParticleList: reconstructDecay('rho+:myRho -> pi+:all pi0:mdst', '0.6 < M < 1.0', path=mypath)*
Mdst_vee2	K_S0:mdst Lambda0:mdst gamma:v0mdst	*Use already created 'K_S0:mdst', 'Lambda0:mdst', and 'gamma:v0mdst' ParticleLists: cutAndCopyList('K_S0:good', 'K_S0:mdst', cut='extraInfo(goodKs)==1', path=mypath)*
Mdst_klong	K_L0:mdst	*Use already created 'K_L0:mdst' ParticleList: reconstructDecay('B0:jpsikl-> J/psi:my K_L0:mdst', 'Mbc>5.0', path=mypath)*

Tracking in Belle

Particle Identification:

Aerogel Cerenkov Counter: Silica aerogel

Time of Flight: Plastic scintillator

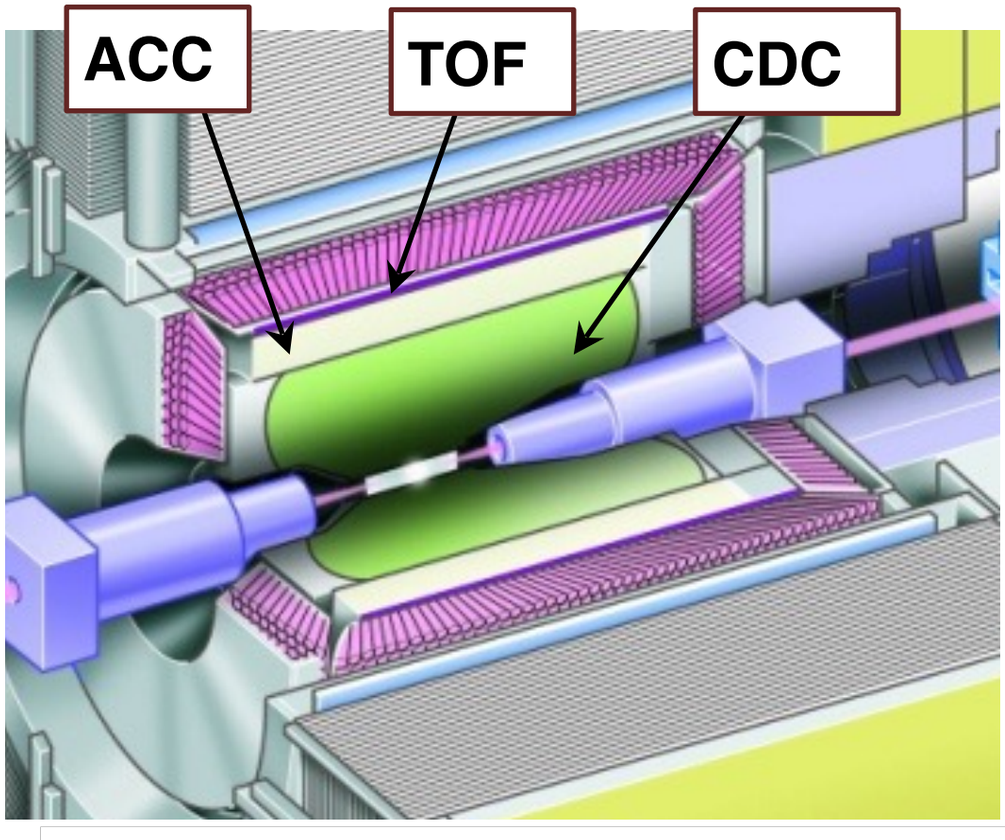
Central Drift Chamber (dE/dx): Small cell + He/C_2H_6

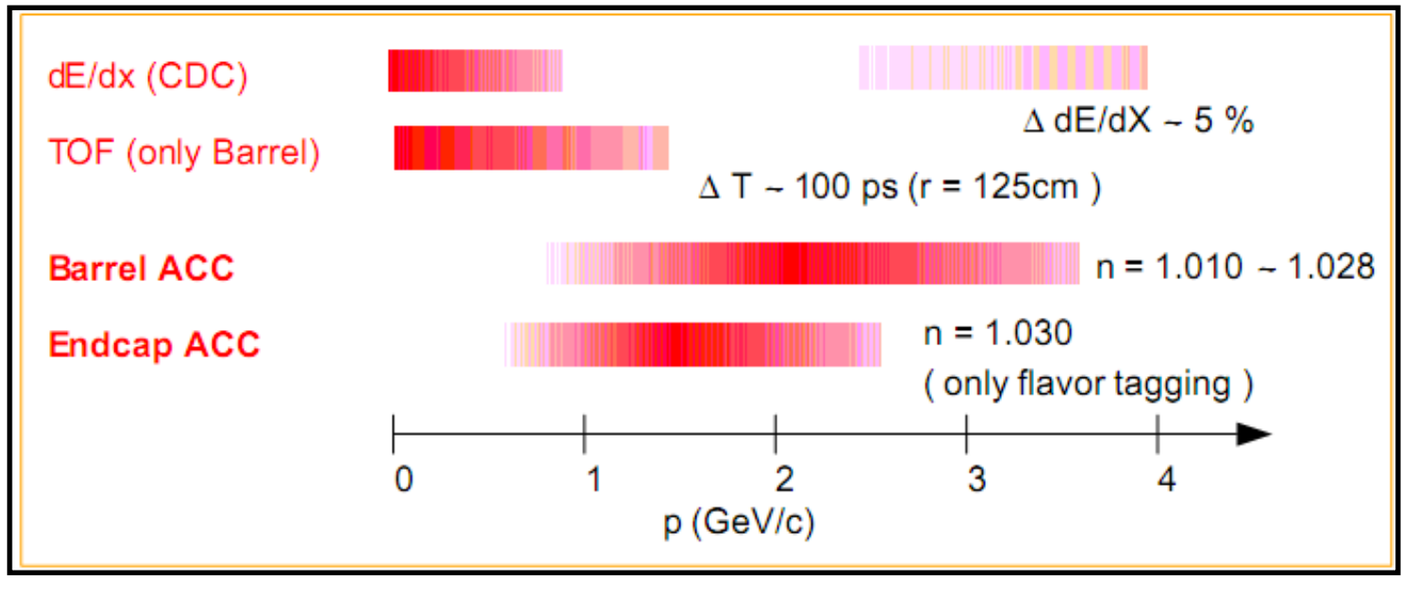
e, μ, π, K, p can be reconstructed

Cherenkov light detected by FM-PMT

Timing resolution~100ps

Momentum Measurement (CDC)
1. P_{X-Y}: Curvatures in the transverse plane
2. P_Z : Helical track information
dE/dx Information





$$\text{Likelihood: } \mathcal{L}_i = \mathcal{L}_i^{CDC} \times \mathcal{L}_i^{TOF} \times \mathcal{L}_i^{ACC}$$

Charged particle identification

- Charged final particles detected by Belle detector:
  e, μ, π, K, p (id = 0, 1, 2, 3, 4)
  - All of them are stored in mdst_charged.
  - Three different packages for the particle identification:
    - atc_pid - mostly for K/π separation, also used for proton id)
      - Combines information from **A**CC, **T**OF, **C**DC.
      - Returns a likelihood ratio that compares two hypotheses (binary PID).
    - eid - identify **electron** from hadrons
    - Muid_mdst - identify **muon** from hadrons

Charged particle identification

✦ Use predefined Belle-like PID Variables:

basf	basf2
atc_pid(3,1,5,3,2).prob(…)	atcPIDBelle(3,2)
eid.prob(3,-1,5)	eIDBelle
Muid_mdst.Muon_likelihood()	muIDBelle
Muid_mdst.Prerejection()	muIDBelleQuality

Charged particle identification

✦ Use predefined Belle-like PID Variables:

basf	basf2
atc_pid(3,1,5,3,2).prob(…)	atcPIDBelle(3,2)
eid.prob(3,-1,5)	eIDBelle
Muid_mdst.Muon_likelihood()	muIDBelle
Muid_mdst.Prerejection()	muIDBelleQuality

K v.s. pi

Charged particle identification

✦ Use predefined Belle-like PID Variables:

basf	basf2
atc_pid(3,1,5,3,2).prob(…)	atcPIDBelle(3,2)
eid.prob(3,-1,5)	eIDBelle
Muid_mdst.Muon_likelihood()	muIDBelle
Muid_mdst.Prerejection()	muIDBelleQuality

K v.s. pi

✦ A reasonable selection for PID (in B to hhh decays):
atcPIDBelle(3,2)>0.6 for K, <0.4 for π
eIDBelle>0.95 for electron veto
muIDBelle>0.95 and muIDBelleQuality=1 for muon veto

Vee2 particle variables for B2BII ana

- ✦ The quality indicators for Kshort and Lambda as estimated by the findKs, nisKsFinder, and findLambda are attached as extraInfo to converted particles:

basf	basf2
findKs.goodKs()	extraInfo(goodKs)
findLambda.goodLambda()	extraInfo(goodLambda)
nisKsFinder.nb_vlike()	extraInfo(ksnbVLike)
nisKsFinder.nb_nolam()	extraInfo(ksnbNoLam)
nisKsFinder.standard()	extraInfo(ksnbStandard)

- ✦ findKs and findLambda use cut-base optimisation, and nisKsFinder uses NeuroBayes.
- ✦ goodBelleKshort and goodBelleLambda now return **extraInfo(goodKs)** and **extraInfo(goodLambda)**.

Event variables for B2BII ana

✦ Evtcls flags stored in evtcls table in basf.

Important event classes are ;

evtcls_flag[?]	evtcls	evtcls_flag[?]	evtcls
0	hadronic	5	2 photon
1	bhabha	6	cosmic
2	γ pair	7	radiative bhabha
3	μ pair	11	radiative μ pair
4	τ pair	18	beam bg

basf	basf2
Evtcls_flag*	eventExtraInfo(evtcls_flag)
Evtcls_flag2†	eventExtraInfo(evtcls_hadronic_flag)

✦ Details can be found **here**.

*https://belle.kek.jp/~ichiro/evtcls/archives/panther_flag.text

†https://belle.kek.jp/~ichiro/evtcls/archives/hadronic_flag.text

Still some info not converted…

There is still some information not yet converted that is important in specific analyses.

- CDC hit pattern
- Trigger menu

If you need this info for your analysis, you are welcome to discuss with us!

Prerequisite

- ✦ basf2 release version:
  release-04-01-01
  light-1912-icarus (recommended)

- ✦ Set up environment variable after setting up basf2:

```
setenv LD_LIBRARY_PATH /sw/belle/local/neurobayes-4.3.1/lib/:$LD_LIBRARY_PATH
or
export LD_LIBRARY_PATH=/sw/belle/local/neurobayes-4.3.1/lib/:$LD_LIBRARY_PATH
```

- ✦ To access and read the Belle DB:

```
setenv PGUSER g0db
or
export PGUSER=g0db
```

How to use b2bii?

How to use b2bii?

- ✦ It’s very simple, just add one line in your script…

How to use b2bii?

✦ It’s very simple, just add one line in your script…

```
convertBelleMdstToBelleIIMdst(inputMDSTFile, path=mypath)
```

How to use b2bii?

✦ It's very simple, just add one line in your script…

```
convertBelleMdstToBelleIIMdst(inputMDSTFile, path=mypath)
```

✦ Okay… maybe two…

```
from b2bii import convertBelleMdstToBelleIIMdst
convertBelleMdstToBelleIIMdst(inputMDSTFile, path=mypath)
```

How to use b2bii?

✦ It's very simple, just add one line in your script…

```
convertBelleMdstToBelleIIMdst(inputMDSTFile, path=mypath)
```

✦ Okay… maybe two…

```
from b2bii import convertBelleMdstToBelleIIMdst
convertBelleMdstToBelleIIMdst(inputMDSTFile, path=mypath)
```

✦ There are some examples in
~capid/public/B2SKW at KEKCC.
(Also available in the master branch)

Example I: Convert and Dump data

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import os
import basf2
from b2biiConversion import convertBelleMdstToBelleIIMdst

os.environ['PGUSER'] = 'g0db'

main = basf2.create_path()

# add all modules necessary to read and convert the mdst file
inputfile = basf2.find_file('b2bii_input_evtgen_exp_07_BptoD0pip-D0toKpipi0-0.mdst', 'examples', False)
convertBelleMdstToBelleIIMdst(inputfile, applyHadronBJSkim=True, path=main)

# Print out the contents of the DataStore
main.add_module('PrintCollections')

# progress
main.add_module('Progress')

basf2.process(main)

# Print call statistics
print(basf2.statistics)
```

B2BII Conversion

To run it

basf2 ~capid/public/B2SKW/DumpData.py

Example II: Convert and reconstruct

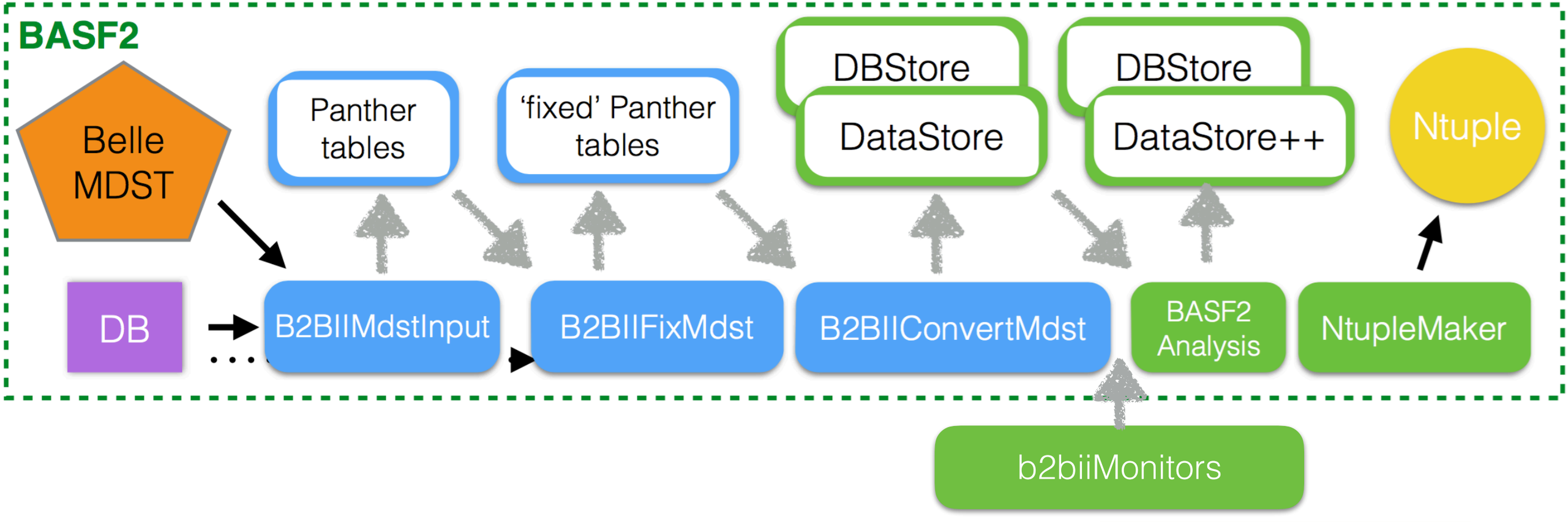
```python
# Only charged final state particles need to be loaded. The neutral particles
# gamma, pi0, K_S0, K_L0, and Lambda0 are already loaded to the 'gamma:mdst',
# 'pi0:mdst', 'K_S0:mdst', 'K_L0:mdst', and 'Lambda0:mdst' particle lists,
# respectively.
ma.fillParticleList('pi+:all', '', path=mypath)
ma.fillParticleList('K+:all', '', path=mypath)

# Let's have a look at the pi0 candidates in 'pi0:mdst' and print the values of some variables
# In order to access the MC information we need to run the MC matching first
ma.matchMCTruth('pi0:mdst', path=mypath)
ma.printVariableValues('pi0:mdst', ['mcPDG', 'p', 'M', 'InvM'], path=mypath)

# The Belle PID variables are: atcPIDBelle(sigHyp, bkgHyp), muIDBelle, and eIDBelle
va.variables.addAlias('Lkpi', 'atcPIDBelle(3,2)')

# Since we did not apply any PID requirement the 'pi+:all' particle list
# contains all type of charged final state particles. Have a look at the
# printOut and notice how the true identity correlates with the corresponding
# PID values.
ma.printVariableValues('pi+:all', ['mcPDG', 'p', 'Lkpi', 'Lppi', 'muIDBelle',
                                   'muIDBelleQuality', 'eIDBelle', 'nSVDHits'], path=mypath)

# Now, let's really reconstruct a B decay with an intermediate D meson:
ma.reconstructDecay('D0:Kpipi0 -> K-:all pi+:all pi0:mdst', '1.7 < M < 2.0', path=mypath)
ma.reconstructDecay('B+:D0pi -> anti-D0:Kpipi0 pi+:all', '4.8 < M < 5.5', path=mypath)

ma.matchMCTruth('B+:D0pi', path=mypath)

# create and fill flat Ntuple with MCTruth and kinematic information
kinematics_and_truth = vc.kinematics + vc.mc_truth
variables = vu.create_aliases_for_selected(kinematics_and_truth, '^B+ -> [ ^D0 -> ^K- ^pi+ ^pi0] ^pi+')

belle1pid = ['eIDBelle', 'muIDBelleQuality', 'muIDBelle', 'Lkpi', 'Lppi', 'Lpk']
variables += vu.create_aliases_for_selected(belle1pid, 'B+ -> [ D0 -> ^K- ^pi+ pi0] ^pi+')

ma.variablesToNtuple('B+:D0pi', variables, filename='B2BII_ConvertAndReconstruct_Example.root', path=mypath)
```

Attach this after b2bii conversion

~capid/public/B2SKW/ConvertAndReconstruct.py

Conversion Monitors

- ✦ To validate the conversion.
- ✦ Both basf and basf2 monitoring modules produce predefined histograms.
- ✦ Histograms of the same quantity filled within basf and basf2 have to match.



Conversion Monitoring Histograms

Belle MDST

basf

Monitoring basf module

- ✓ Tracks
- ✓ PID
- ✓ neutrals
- ✓ V0s
- ✓ MC Particles

basf2

b2biiMonitors (optional)

- ✓ Tracks
- ✓ PID
- ✓ neutrals
- ✓ V0s
- ✓ MC Particles

basf Monitoring Histograms

basf2 Monitoring Histograms

compare monitoring histograms obtained in basf and basf2

Conversion Monitoring Histograms

Belle MDST

basf

Monitoring basf module

- ✓ Tracks
- ✓ PID
- ✓ neutrals
- ✓ V0s
- ✓ MC Particles

Please see backup

basf2

b2biiMonitors (optional)

- ✓ Tracks
- ✓ PID
- ✓ neutrals
- ✓ V0s
- ✓ MC Particles

basf Monitoring Histograms

basf2 Monitoring Histograms

compare monitoring histograms obtained in basf and basf2

Example III: Monitoring

```
import os

import basf2
import ROOT
from ROOT import Belle2
from b2biiConversion import convertBelleMdstToBelleIIMdst
from b2biiMonitors import addBeamParamsConversionMonitors
from b2biiMonitors import addTrackConversionMonitors
from b2biiMonitors import addKshortConversionMonitors
from b2biiMonitors import addConvertedPhotonConversionMonitors
from b2biiMonitors import addLambda0ConversionMonitors
from b2biiMonitors import addNeutralsConversionMonitors
from b2biiMonitors import addMCParticlesConversionMonitors
from b2biiMonitors import addKlongConversionMonitors

os.environ['USE_GRAND_REPROCESS_DATA'] = '1'
os.environ['PGUSER'] = 'g0db'

# Convert
mypath = basf2.create_path()
inputfile = basf2.find_file('b2bii_input_evtgen_exp_07_BptoD0pip-D0toKpipi0-0.mdst', 'examples', False)
convertBelleMdstToBelleIIMdst(inputfile, path=mypath)

# Reconstruct
# Create monitoring histograms
addBeamParamsConversionMonitors(path=mypath)
addTrackConversionMonitors(path=mypath)
addKshortConversionMonitors(path=mypath)
addKlongConversionMonitors(path=mypath)
addLambda0ConversionMonitors(path=mypath)
addConvertedPhotonConversionMonitors(path=mypath)
addNeutralsConversionMonitors(path=mypath)
addMCParticlesConversionMonitors(path=mypath)
```

import b2biiMonitors

Create Conversion
Monitoring Histograms

~capid/public/B2SKW/Monitor.py

How to verify?

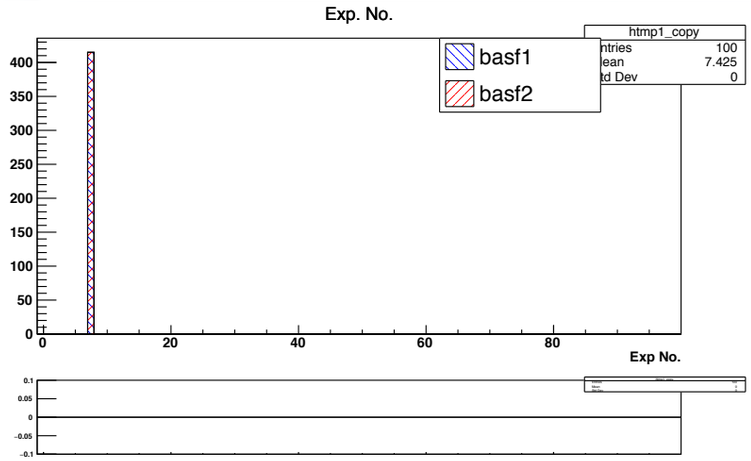
- ✦ Scripts in b2bii/tools
- ✦ Modify root macro `plotComparisonBASF1vs2.C` update the path of input root files.

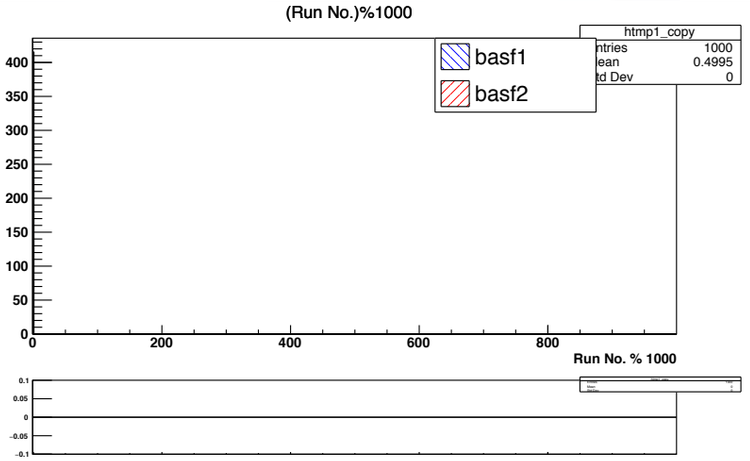
```
// ADD BASF1 FILE
TFile *gBASF = new TFile("BASFMonitors-testSample.root");

// ADD BASF2 FILES
TFile *gNeutralsGamma = new TFile("b2biiGammaConversionMonitors.root");
TFile *gNeutralsPi0 = new TFile("b2biiPi0ConversionMonitors.root");
TFile *gNeutralsMCPi0 = new TFile("b2biiMCPi0ConversionMonitors.root");

TFile *gBeamParams = new TFile("b2biiBeamParamsConversionMonitors.root");
TFile *gKSMonitors = new TFile("b2biiKshortConversionMonitors.root");
TFile *gKLMonitors = new TFile("b2biiKlongConversionMonitors.root");
```

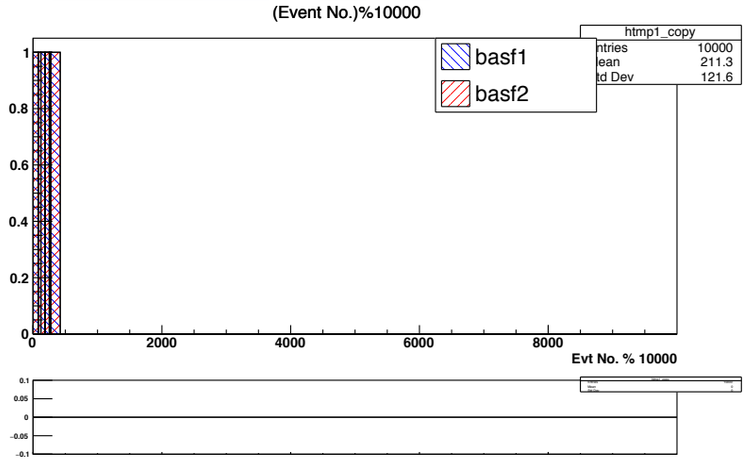
- ✦ Run the macro, you will have comparison plots in pdf format.

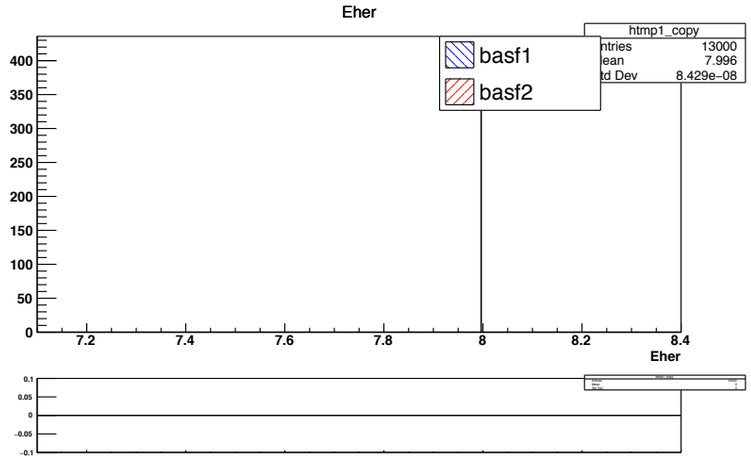
BeamParameters-MonitorPlots.pdf
KS-MonitorPlots.pdf
KS-MonitorPlots-AVF.pdf
KL-MonitorPlots.pdf
Gamma-V0-MonitorPlots.pdf
Gamma-V0-MonitorPlots-AVF.pdf
gammaMonitorPlots.pdf
pi0MonitorPlots.pdf
MCParticleMonitorPlots.pdf
TrackMonitorPlots.pdf

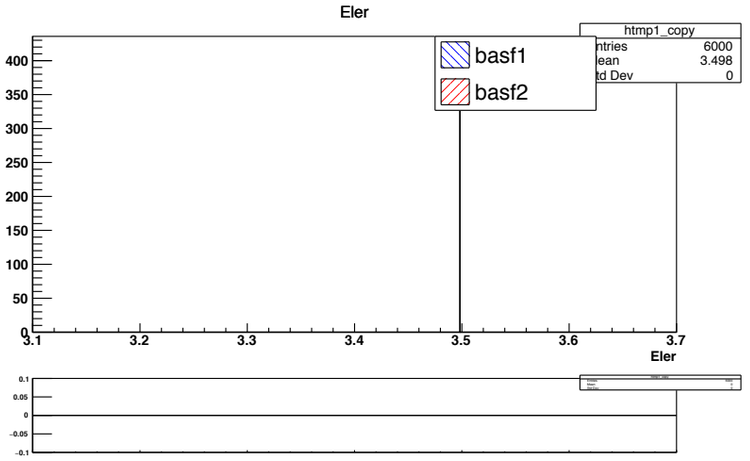
Examples of Monitoring Histograms

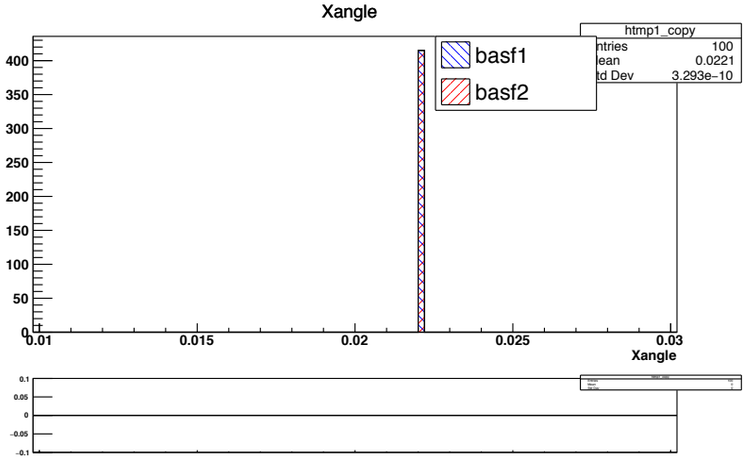


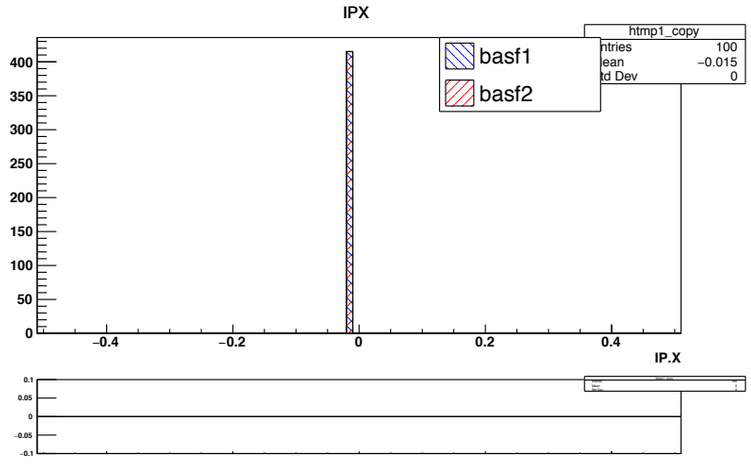


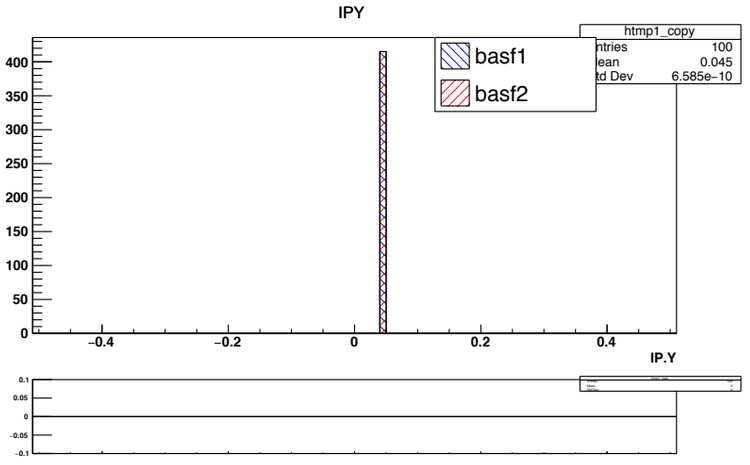


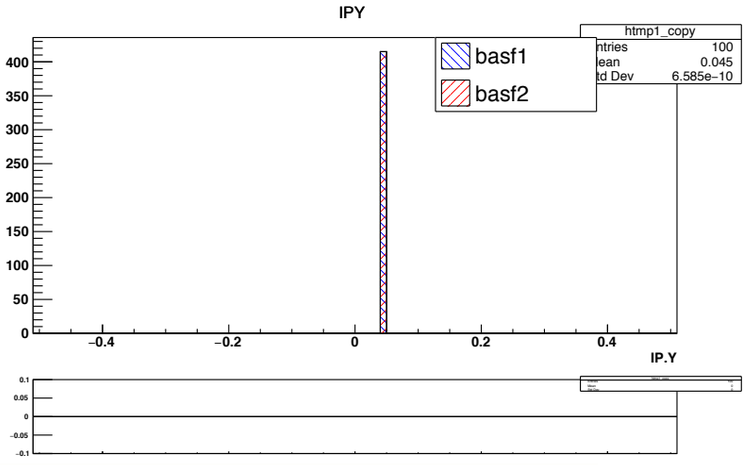












Something need to keep in mind…

✦ There are some functions that have belle specific option, please enable them in your b2bii analysis.

✓ `buildRestOfEvent('B+:sig', belle_sources=True)`

✓ `tagCurlTracks('pi+:all', belle=True)`
in `modularAnalysis.py`

✓ `flavorTagger(belleOrBelle2="Belle")`
in `flavorTagger.py`

✓ `get_default_channels(convertedFromBelle=True)`
`config.FeiConfiguration(prefix='FEIv4_2017_MCConverted_Track14_2',b2bii=True)`
in `fei` package

How to submit b2bii jobs to bqueues

How to submit b2bii jobs to bqueues

```
#bsub -q l “basf2 CovertAndReconstruct.py”
```

How to submit b2bii jobs to bqueues

```
#bsub -q l “basf2 CovertAndReconstruct.py”
```

Commands:

`bsub:` submit a job

`bjobs:` display information about jobs

`bkill:` kill a job

`bqueues:` show information about queues.

How to submit b2bii jobs to bqueues

```
#bsub -q l “basf2 CovertAndReconstruct.py”
```

Commands:

`bsub:` submit a job

`bjobs:` display information about jobs

`bkill:` kill a job

`bqueues:` show information about queues.

- ✦ Currently b2bii can only be used at KEKCC because the communication with Belle DB server is needed.
- ✦ Submitting large amount jobs might cause unforeseen segmentation faults, due to bandwidth limits of DB server.

Take home message

- ✦ B2BII is to analyze Belle MDST data in basf2 (with improved tools and algorithms, such as FEI, Flavour Tagger etc…).
  A few Belle analyses using B2BII already published.
- ✦ Beware of
  - ✦ **neutral particle** lists, only use the pre-made list in B2BII.
  - ✦ **Belle specific PID** should also be used in a B2BII analysis.
- ✦ What you can contribute?
  - ➡ Use it for your analysis, and give feedbacks!
  - ➡ Get involved in the development!
- ✦ More details can be found at

  https://confluence.desy.de/display/BI/Physics+B2BII

  Subscribe to software-b2bii@belle2.org to get latest news or to discuss in the community.

Acknowledgement

✦ Some slides have been stolen from

Anze Zupanc, Shohei Nishida, Hülya Atmacan

Thank you for your attention!

Backup

If you are using release-03…

- ✦ There is some information not available…
  - ☹️ goodBelleKshort
  - ☹️ buildRestOfEvent
  - 💀 goodLambda info
  - 💀 evtcls flags
- ✦ Alternatively, you can…
  - ✦ Call variable `extraInfo(goodKs)`
  - ✦ Use the `RestOfEventBuilder` directly
  - ✦ Modify b2bii module to include evtcls flags

But there is no way to get goodLambda info in release before release-04

How to generate signal MC

- Please refer to expert's talk:

  https://kds.kek.jp/indico/event/20711/contributions/54328/attachments/43769/51998/mcprod_belle_ss.pdf

How to create monitoring histograms

In basf

- Copy the directory in your working directory:
  `cp -r ~capid/public/B2SKW/b2biiBASFMonitors .`
- Go to **b2biiBASFMonitors/scripts/** directory
  `cd b2biiBASFMonitors/scripts/`
- Setup BASF environment
  `source setup_BASF.sh`
  or
  `source setup_BASF.csh`
- Compile the monitoring modules
  `make compile` Runs CompileAll.sh
- Run the monitors over the test sample
  `make run` Submits a job to run RunMonitorOnTestSample.csh
- Job will produce
  - BeamParamsKShotsLambdasConvertedPhotonsNeutralsTracksMonitors.hbook
  - MCParticlesMonitor.root
- Convert hbook file to root and merge two files.
  `make run`
  Finally you will get **BASFMonitors-testSample.root**

scripts folder contains:

- ✓ **CompileAll.sh**
- ✓ **RunMonitorOnTestSample.csh**
- ✓ **setup_BASF.csh**
- ✓ **setup_BASF.sh**
- ✓ **Makefile**

Data MDST file

- ✦ Data file contains less tables
  (no simulation info and evtgen info)

```
*** Record number = 1 ***
BELLE_EVENT  (   1) EVTCLS_FLAG  (   1) EVTCLS_FLAG2 (   1) EVTCLS_HADRON_CHARGED           (   3) EVTCLS_HADRON_INFO              (   1)
EVTCLS_HADRON_NEUTRAL           (   5) EVTCLS_HADRONIC_FLAG             (   1) EVTVTX_PRIMARY_VERTEX           (   1) EVTVTX_TRK   (   3)
L4_SUMMARY   (   1) MDST_ACC    (   6) MDST_CDC_FIT (  15) MDST_CHARGED (   6) MDST_CITIMING(   1) MDST_ECL    (  39) MDST_ECL_AUX (  39)
MDST_ECL_CR  (  38) MDST_ECL_TRK (   4) MDST_EVENT_ADD                  (   1) MDST_GAMMA   (  13) MDST_KLM_CLUSTER                (   2)
MDST_KLM_CLUSTER_HIT            (  17) MDST_KLM_MU  (   6) MDST_KLM_MU_EX                   (   6) MDST_MUID    (   6) MDST_PI0     (  24)
MDST_SVD_HIT (  28) MDST_TOF    (   2) MDST_TRG     (  15) MDST_TRK     (   6) MDST_TRK_ADD (   6) MDST_TRK_FIT (  15) RECCDC_TIMING(   6)
----- Crate=   0, Event No.=     73 -----
```