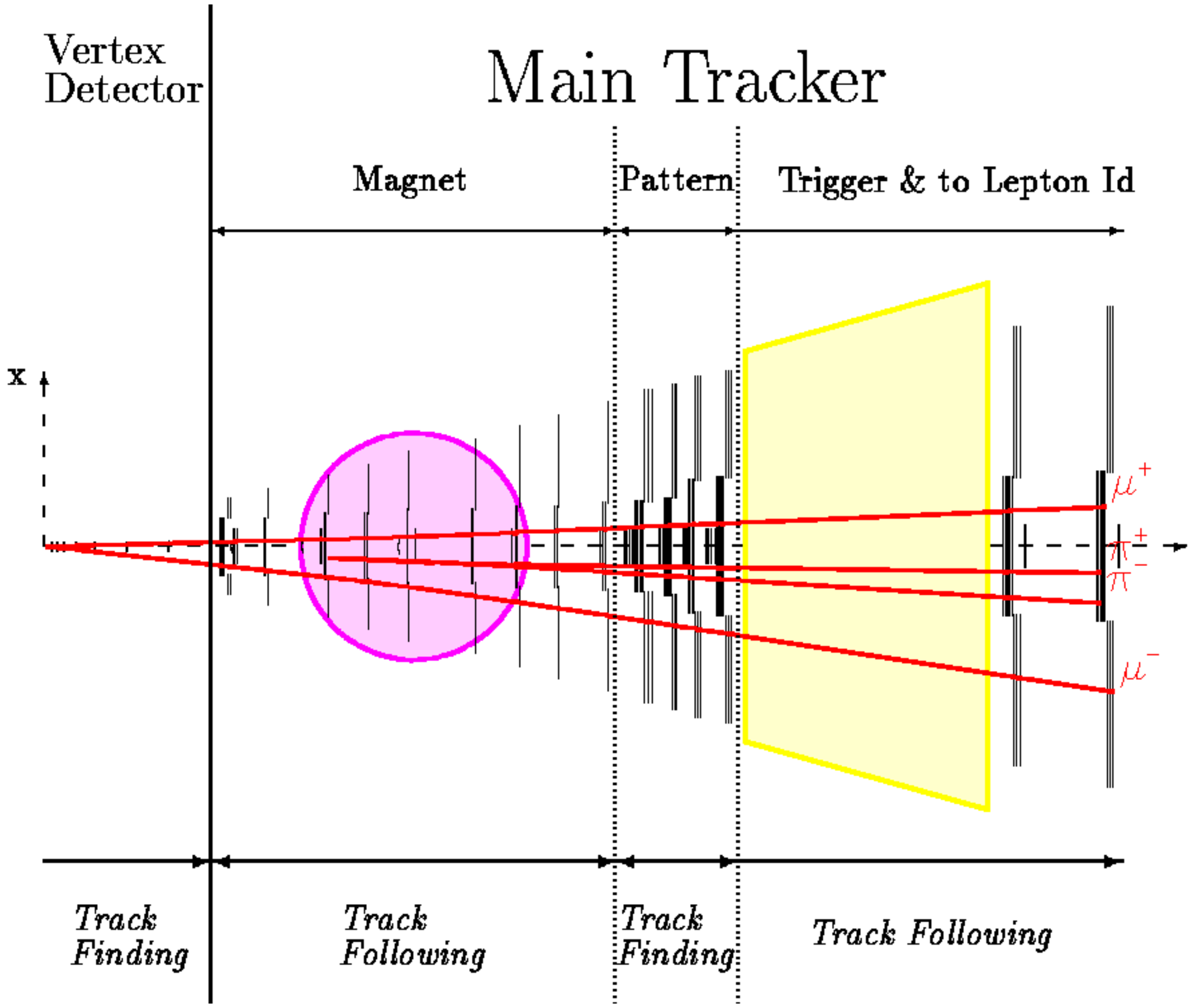
Status of Main Tracker Reconstruction

Rainer Mankel

Humboldt University Berlin

for the HERA-B collaboration

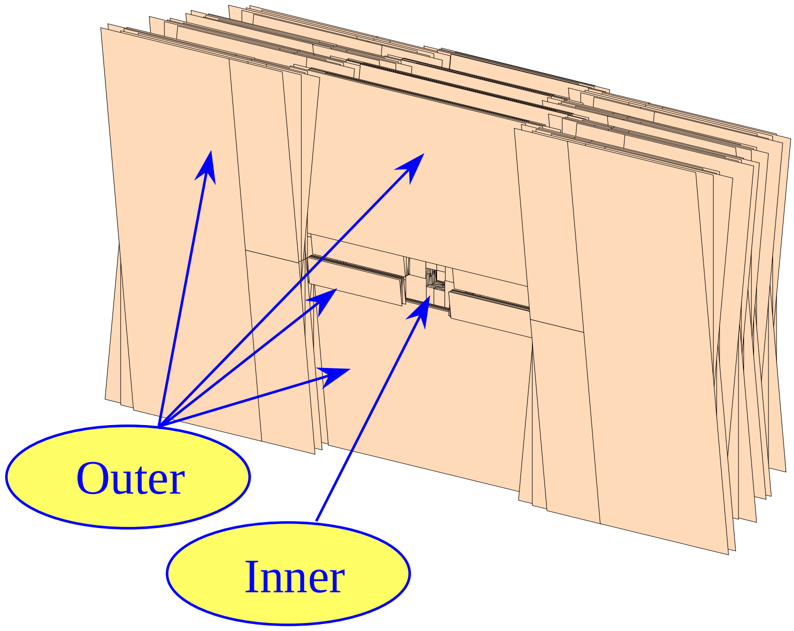
Track Reconstruction Concept

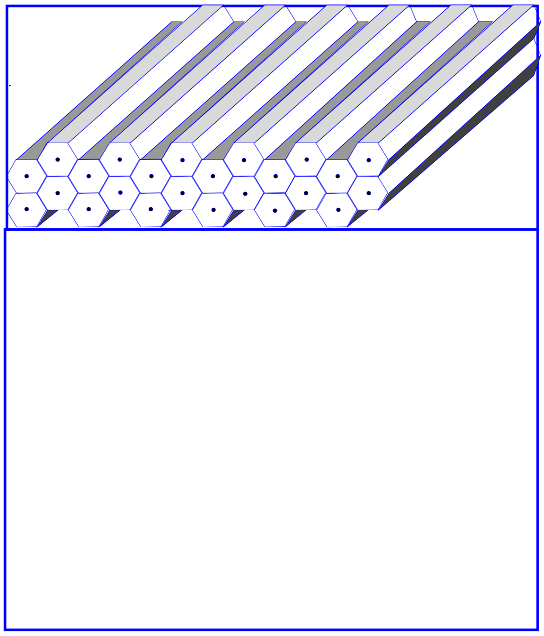


Main Tracker Part:

- Pattern Tracker track finding
- Magnet Tracker propagation
- Trigger Chamber Propagation

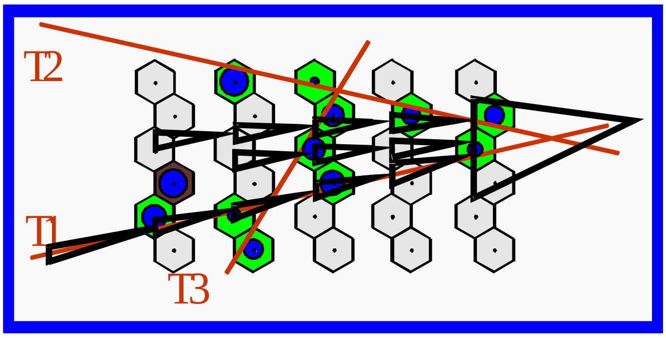
Pattern Tracker

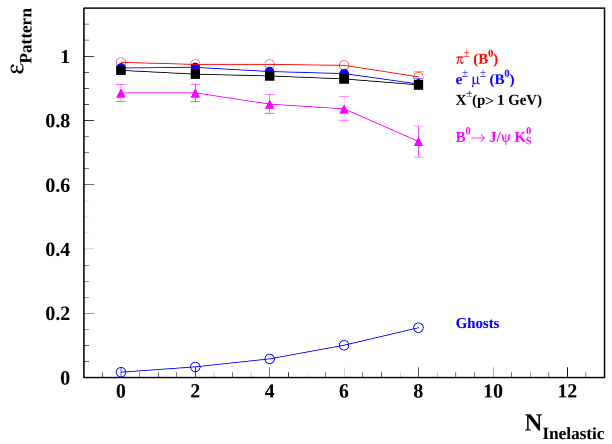




- efficiency was verified in ARTE-01-08
  - slight bug in ARTE-01-08-r1 was corrected
- mean efficiency for tracks from the golden B decay (5 superimposed interactions) **> 96%**
- fraction of "ghost" tracks **< 8%**

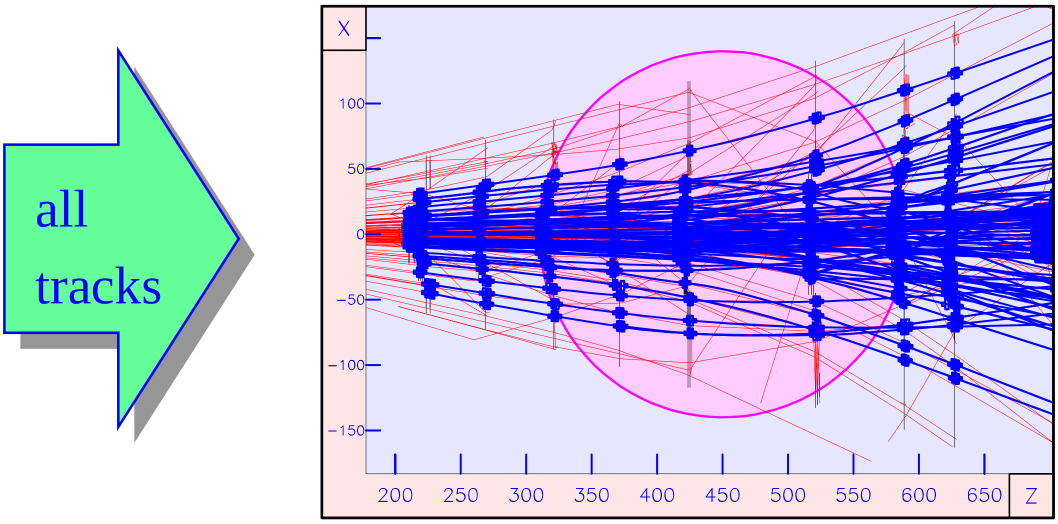
Concurrent Track Evolution

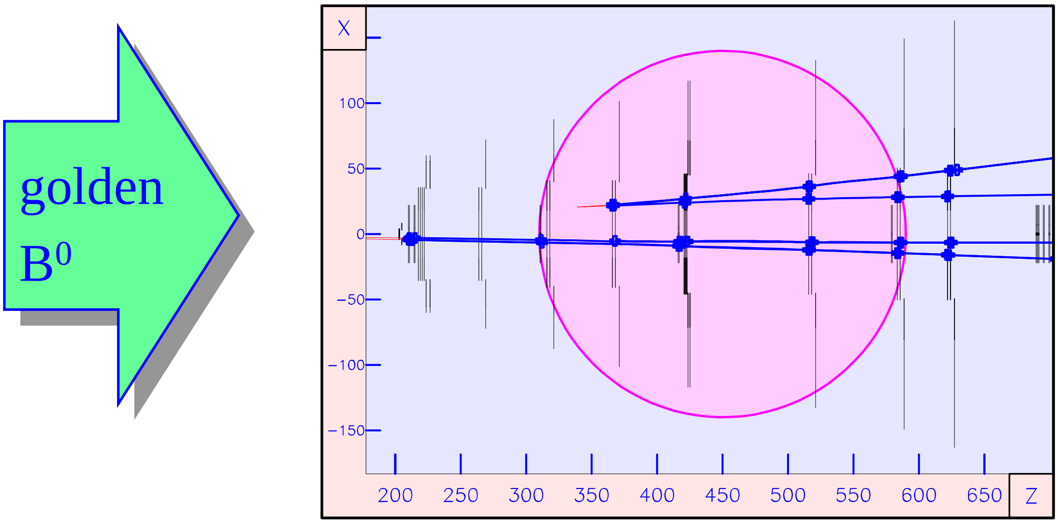




Track Propagation in the Magnet

- Extend concurrent track evolution to 3D in inhomogeneous magnetic field
- Optimized fast transport





➔ propagation efficiency 97–99% for tracks from the golden B decay, ghost rate 2%

➔ total main tracker pattern recognition efficiency for golden B decay ~83%

Magnet Propagation (cont'd)

- Geometrical acceptance (pattern+magnet tracker)

Particle	Geometrical Acceptance
$\mu^{\pm}_{J/\psi}$	$(80.2 \pm 1.3)\%$
$e^{\pm}_{J/\psi}$	$(66.6 \pm 1.9)\%$
$\pi^{\pm}_{K^0_S}$	$(66.8 \pm 1.1)\%$
$J/\psi \rightarrow \mu^+\mu^-$	$(63.4 \pm 1.6)\%$
$J/\psi \rightarrow e^+e^-$	$(45.4 \pm 2.9)\%$
$K^0_S \rightarrow \pi^+\pi^-$	$(54.8 \pm 1.6)\%$
$B^0 \rightarrow J/\psi K^0_S \rightarrow \mu^+\mu^-\pi^+\pi^-$	$(34.0 \pm 1.6)\%$
$B^0 \rightarrow J/\psi K^0_S \rightarrow e^+e^-\pi^+\pi^-$	$(28.6 \pm 2.6)\%$

- Efficiency vs kind of track

Particle	ϵ Pattern + Magnet	ϵ Magnet
$e^{\pm}_{J/\psi}$	$(96.8 \pm 0.9)\%$	$(98.7 \pm 1.1)\%$
$\mu^{\pm}_{J/\psi}$	$(97.3 \pm 0.4)\%$	$(99.6 \pm 0.6)\%$
$\pi^{\pm}_{K^0_S}$	$(93.3 \pm 0.7)\%$	$(97.4 \pm 0.9)\%$
$J/\psi \rightarrow e^+e^-$	$(93.5 \pm 2.1)\%$	$(97.3 \pm 2.6)\%$
$J/\psi \rightarrow \mu^+\mu^-$	$(94.5 \pm 0.9)\%$	$(99.3 \pm 1.3)\%$
$K^0_S \rightarrow \pi^+\pi^-$	$(88.0 \pm 1.5)\%$	$(94.8 \pm 1.9)\%$
$B^0 \rightarrow J/\psi K^0_S \rightarrow e^+e^-\pi^+\pi^-$	$(83.9 \pm 3.9)\%$	$(92.3 \pm 4.9)\%$
$B^0 \rightarrow J/\psi K^0_S \rightarrow \mu^+\mu^-\pi^+\pi^-$	$(83.0 \pm 2.1)\%$	$(93.6 \pm 2.9)\%$
$X^{\pm}$(p $>$ 1 GeV/c)	$(88.1 \pm 0.2)\%$	$(94.1 \pm 0.2)\%$
Ghosts	$(2.0 \pm 0.1)\%$	

- still some improvement wrt earlier preliminary results
- much faster than old algorithm ($\leq$ ranger 4.8)
- electron propagation works reasonably well even without a-priori identification

Total Efficiency?

- Rough (!) estimation of total efficiency
  - golden B^0 in muon channel:

ε_{total} = 34% · 83% · ε_{SVD} · ε_{match} · [ε_{TC}]

geom patt+mag

· 55 % / 63%

trigger geom(ll)

~ 25% · ε_{SVD} · ε_{match} · [ε_{TC}]

  - golden B^0 in electron channel:

ε_{total} = 28% · 83% · ε_{SVD} · ε_{match} · [ε_{TC}]

geom patt+mag

· 35 % / 45%

trigger geom(ll)

~ 18% · ε_{SVD} · ε_{match} · [ε_{TC}]

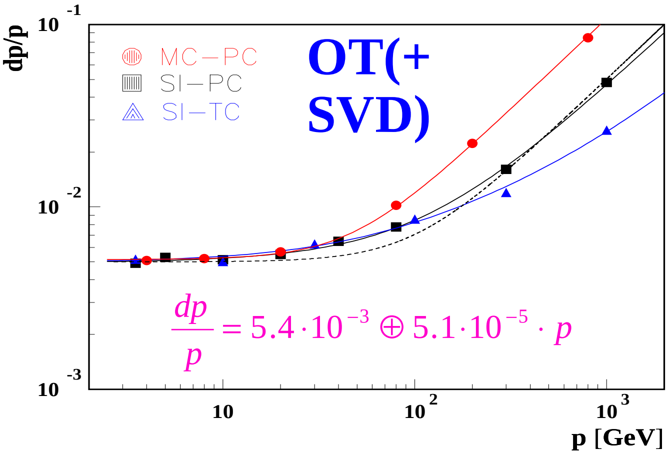
- Important to know the missing factors...

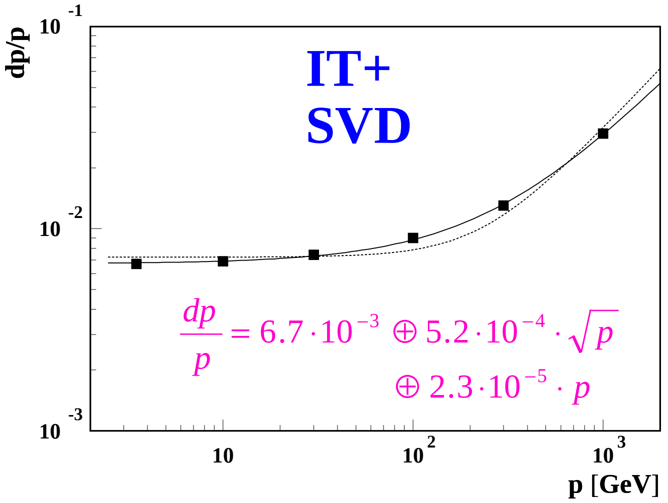
Track Fit & Momentum Resolution

- Kalman filter based track fit
- corrections for multiple scattering, energy loss

➔ magnet & pattern tracker dominate resolution up to p=30 GeV

➔ above 30 GeV, coordinate resolution of vertex detector improves momentum resolution





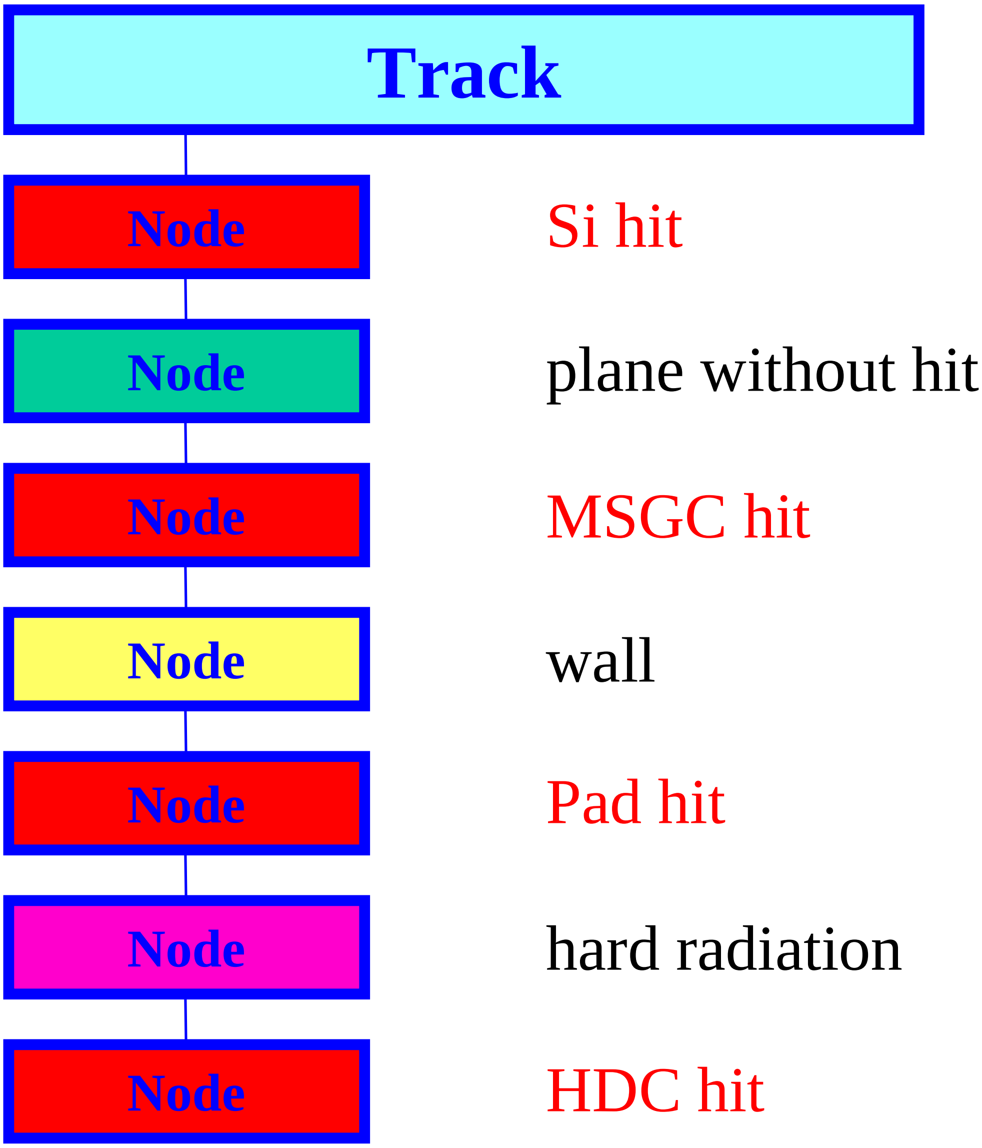
➔ Mean relative momentum resolution for muons from golden B decay (main tracker only):

method	δp/p
pattern recognition	$8.7\cdot10^{-3}$
“ “ + refit	$8.1\cdot10^{-3}$
ideal (using MC info)	$7.1\cdot10^{-3}$

Main Features of the ranger fit

- goal: best possible parameter estimate
- entirely written in C++
- various interfaces to Arte tables
- iterative Kalman filter & smoother
- in-situ insertion of “dead material“ (incl box, epi, trd1/2, tube, tubs, cone shapes)
- full covariance matrix multiple scattering correction for thin & thick scatterers
- stochastic correction for radiative energy loss (for electrons)
- correction for ionization energy loss
- fit modes with/without momentum determination
- fitting with interaction time shift
- various types of measurement: strip, wirechamber, pad, ECAL cluster
- basic classes designed to be reusable for pattern recognition

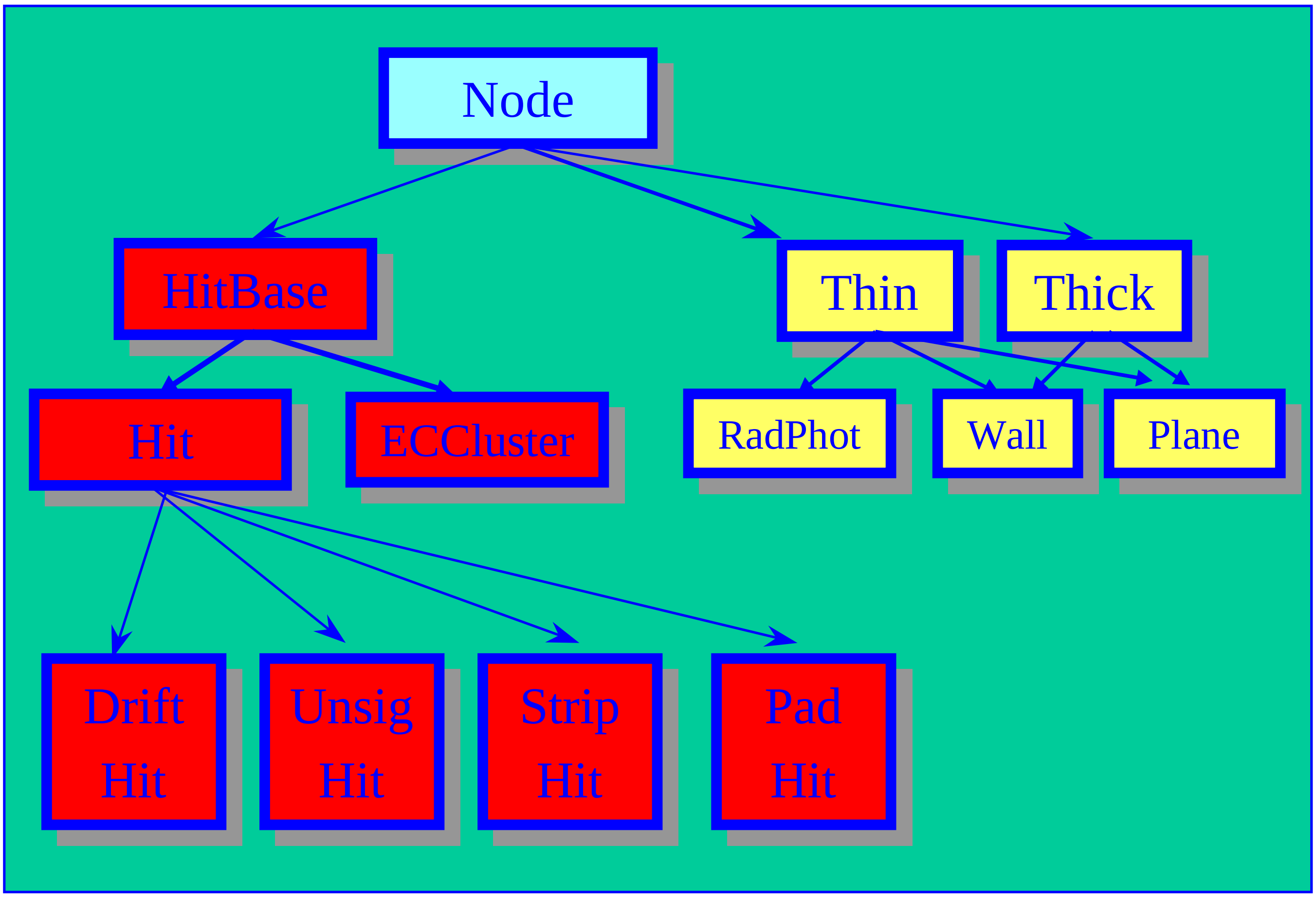
<u>Base object of fit: *rangerTrack*</u>



- track consists of "nodes"
- filter formalism in each node depends on <u>type</u> of node

Node Classes (revisited)

Inheritance scheme



- Added features:
  - pad/pixel hits for HiPT chambers
  - ECAL clusters (RCCL) with measured position and momentum
  - photon radiation (as process noise)
  - generation of additional points (RPNT) for RICH, TRD and muon system (in case of RTRA)

“Plane Vanilla”Application

- invoke rangerFit from C++

```
int rangerFit(ArtePointer<RSEG> rseg,
          rangerParticleType pType= MUON);
int rangerFit(ArtePointer<RTRA> rtra,
          rangerParticleType pType= MUON);
int rangerFit(vector<ArtePointer<RSEG> > rsegVec,
          rangerParticleType pType= MUON);
```

- invoke rangerFit from Fortran

```
integer function rgfit(int rseg,int elec_corr)


integer function rgfitsv
               (int* segVec,   // integer array
                int  nSeg,     // length of array
                int  pType)    // particle type
```

➔ any interface missing?

Advanced Application

- create a rangerTrack

(a) from RSEG:

```
rangerTrack(ArtePointer<RSEG> rseg, rangerFitMode fm,
         rangerParticleType pType= MUON,
         rangerRMap* rmap=0);
```

(b) from several RSEGs

```
  rangerTrack(const vector<ArtePointer<RSEG> >& rsegVec,
         rangerFitMode fm,
         rangerParticleType pType= MUON,
         vector<rangerRMap>* rmapVec=0);
```

(c) from RSEGs + RCCL

```
  rangerTrack(const vector<ArtePointer<RSEG> >& rsegVec,
         ArtePointer<RCCL> rccl,int charge,
          rangerFitMode fm,
         rangerParticleType pType= MUON,
         vector<rangerRMap>* rmapVec=0);
```

(d) from RTRA

```
  rangerTrack(ArtePointer<RTRA> rtra, rangerFitMode fm,
         rangerParticleType pType= MUON,
         rangerRMap* rmap=0);
```

Advanced Application (cont'd)

- fit rangerTrack

```
(a) all in one
int fit(const int firstPass=0,
        const int thirdPass=1,
        const int insertMaterial=1,
        const int updateInternal=0); // (re)fit
                                         track

(b) stepwise
...
```

- save rangerTrack

```
void storeback(rangerRMap* rmap);  // store track
   back using rmap info
void storeback(vector<rangerRMap>* rmapVec);  //
   same for multiple rmaps
ArtePointer<RTRA> storeRtra() const;  // store
   into new RTRA entry
ArtePointer<RSEG> storeRseg() const;  // store
   into new RSEG entry
```

Advanced Application (cont'd)

- simplest example:

```
rangerTrack t(rseg);
t.fit();
cout << "chi2=" << t.totalChiSq() << endl;
t.storeRtra();
```

➔ RPNTs are automatically generated/reused

for "Fast Reconstruction"

```
int  rgideal_(const float& pMin, // minimal
                      // momentum to be considered
   const int& hitMin,  // minimal number of hits
        const float& zMin, // lower z bound
        const float& zMax, // upper z bound
        const int& elecCorr // electron
                      // bremsstrahlung corr flag
        );
```

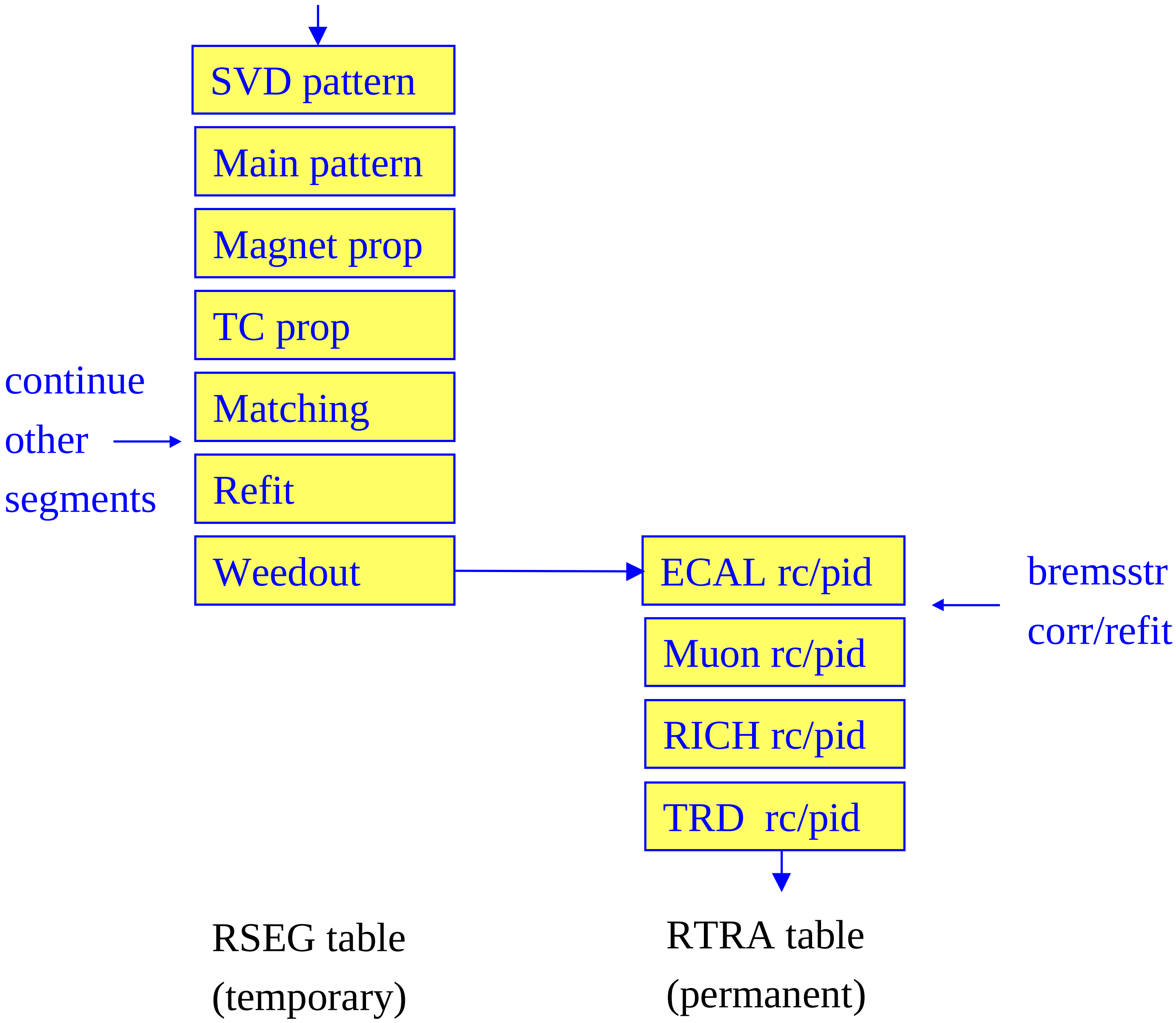
Release Schedule

- ranger 4.9 with new magnet propagation and new fit package
  - ready for release (to which ARTE version?)
  - tested with ARTE-01-08 (hydra+Linux)
- Caution: FIT flags (RTRA+RSEG) and CMP flag (RSEG) have been modified (adapted to standards)
- Old entries (`tfideal, tfrfseg`) will become obsolete

  ➔ use `rgideal, rgfit,` … instead

Implications for Full Reconstruction

- magnet tracking fast enough for “baseline approach“ (pattern → magnet →TC →matching)
- no iteration, no refit needed prior to matching
- separation tracking ⇔ particle id possible:



- is this viable?

Remarks on ARTE Tables

- Operations which may lead to rejection of tracks should be restricted to RSEG table
- need possibility to mark "rejected" tracks in the RSEG table
  - set rseg->fit to negative value?
    - not sensible if rseg->fit=0 indicates "good track"
  - rseg->cmp: most significant bit is still unused
    - mark rejection via

      rseg->cmp = rseg->cmp | Rsegc::bitsign ?
- Particle id: new likelihoods in RTRA table
  - 5 for RICH (lre,…lrp)
  - 2 for TRD (lte, lth)
  - 3 for ECAL (lee, lemi, leh)
  - 2 for muon (lmmu, lmh)

  plus flags for applicability ⇒ should be filled and used in the analysis software

Bits for rseg->cmp

```
#define _RSEG_cmp_hh_


struct Rsegc
{
  enum Cmp {
    vxd     =        0x1, /* Vertex Detector */
    patt    =        0x2, /* Pattern Tracker*/
    magt    =        0x4, /* Magnet Tracker*/
    muon    =        0x8, /* Muon Detector*/
    ecal    =       0x10, /* Ecal*/
    tar     =       0x20, /* Target Constraint*/

    bit6    =       0x40, /* not assigned yet =first bit is bit0,*/
    bit7    =       0x80, /* not assigned yet =first bit is bit0,*/
    bit8    =      0x100, /* not assigned yet =first bit is bit0,*/
    bit9    =      0x200, /* not assigned yet =first bit is bit0,*/
    bit10   =      0x400, /* not assigned yet =first bit is bit0,*/
    bit11   =      0x800, /* not assigned yet =first bit is bit0,*/
    bit12   =     0x1000, /* not assigned yet =first bit is bit0,*/
    bit13   =     0x2000, /* not assigned yet =first bit is bit0,*/
    bit14   =     0x4000, /* not assigned yet =first bit is bit0,*/
    bit15   =     0x8000, /* not assigned yet =first bit is bit0,*/
    bit16   =    0x10000, /* not assigned yet =first bit is bit0,*/
    bit17   =    0x20000, /* not assigned yet =first bit is bit0,*/
    bit18   =    0x40000, /* not assigned yet =first bit is bit0,*/
    bit19   =    0x80000, /* not assigned yet =first bit is bit0,*/
    bit20   =   0x100000, /* not assigned yet =first bit is bit0,*/
    bit21   =   0x200000, /* not assigned yet =first bit is bit0,*/
    bit22   =   0x400000, /* not assigned yet =first bit is bit0,*/
    bit23   =   0x800000, /* not assigned yet =first bit is bit0,*/
    bit24   =  0x1000000, /* not assigned yet =first bit is bit0,*/

    trd     =  0x2000000, /* forward TRD*/
    rich    =  0x4000000, /* Cherenkov Counter*/
    ptc     =  0x8000000, /* PTC Chambers =pi-pi trigger,*/
    msg     = 0x10000000, /* microstrip chambers*/
    wch     = 0x20000000, /* wire chambers*/
    sil     = 0x40000000, /* silicon detector*/
    bitsign = 0x80000000  /* Do not use sign bit to avoid
complications*/
  };
};
```

Summary

- completion of main tracker reconstruction well in progress
- magnet propagation note (Hera-B 98-154) available on the web
- next step: trigger chamber propagation (see talk by A. Spiridonov)
- still missing:
  - new *navigator* (needed for modular geometry)
  - propagation tools for leftover segments (eg. SVD)
- should converge further on full reconstruction scheme