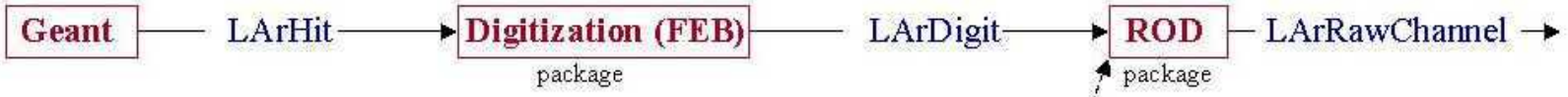
Moving towards an OO Data Format

- Atlas data will be stored in an OO format
- Simple overview of data flow
- Testbeam data will be a good opportunity to gain experience with OO data
- hec_adc - laid out in an OO philosophy "road map" is intended to be a guide for C++ code development
- First step 'EPIO converter' by Naoko Kanaya (next talk) already accomplished

Data Flow

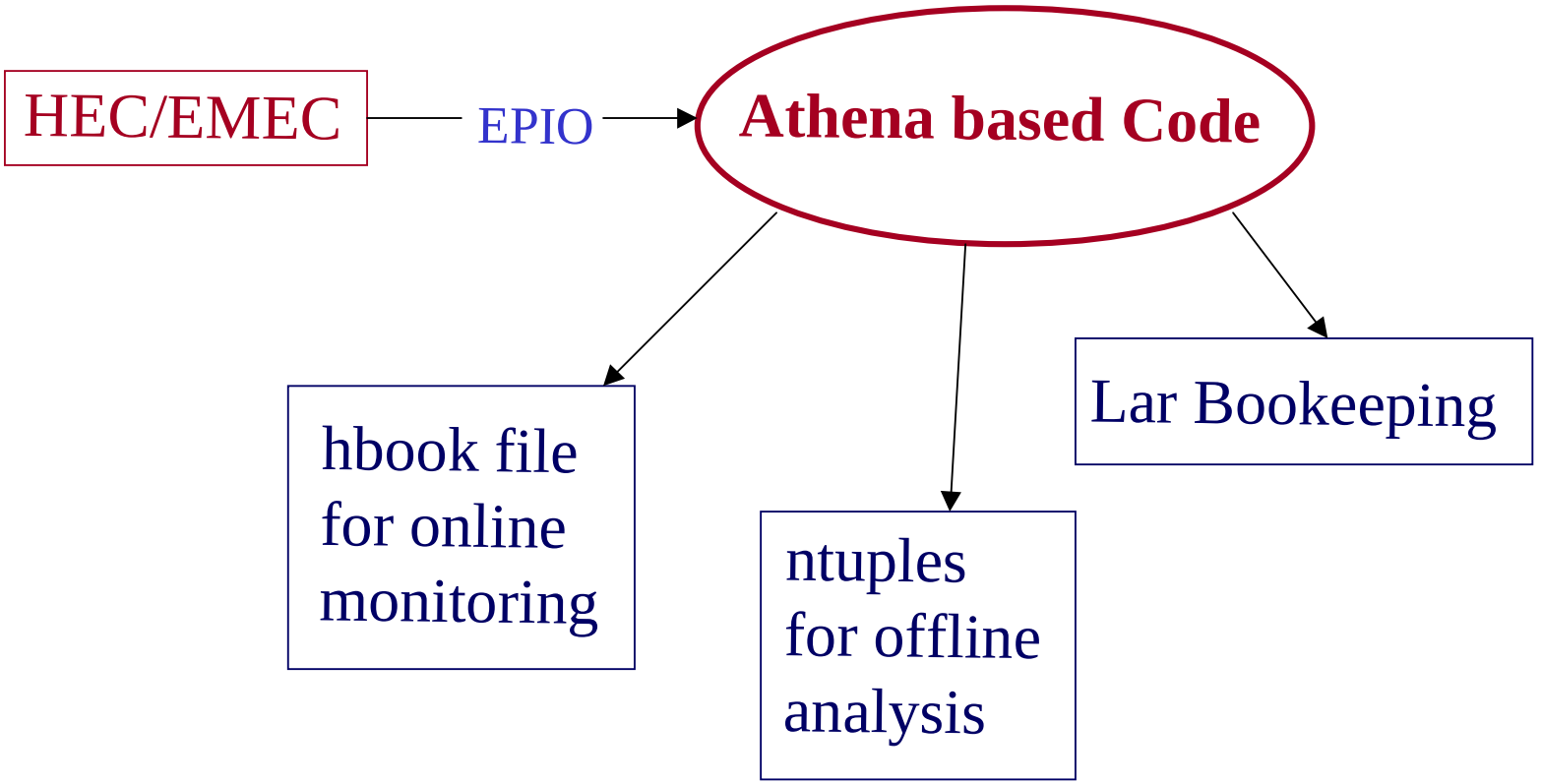
Atlas simulation:



Testbeam (aim for next year):

Atlas data:

Goal for OO Testbeam package:



Milestones:

1. EPIO converter ✓
2. Hec_adc ntuple (Dec.?)

Click here ---> []
to start hec_adc road map

Some useful hacd map overview figures!
hec_adc top level ps file
hec_online tree ps file
hec_online_pass1 ps file
packages 1st page , 2nd page , 3rd page , 4th page , 5th page

hec_adc.f [#]

^^

is the overall steering routine. It calls:

hec_ini.f

hec_offline.f or: hec_online.f

hec_ter.f

hec_offline.f [#]

^^

(is called by: **hec_adc.f**)

Includes the following commons:

hec_par.inc
hec_datacard.inc

This is the steering routine for running things offline.
The following routines are called:

hec_offline_ini.f

This calls routines which read the run header, slow control info and
set the tdc wrap-around constants.

hec_packages

This calls a number of subroutines that deal among others with:
pedestals, pulseheights, correlations, beam chamber reconstruction, calibration

user_ini

This does nothing unless the user changes it

hec_pass1 hec_pass2 hec_pass3

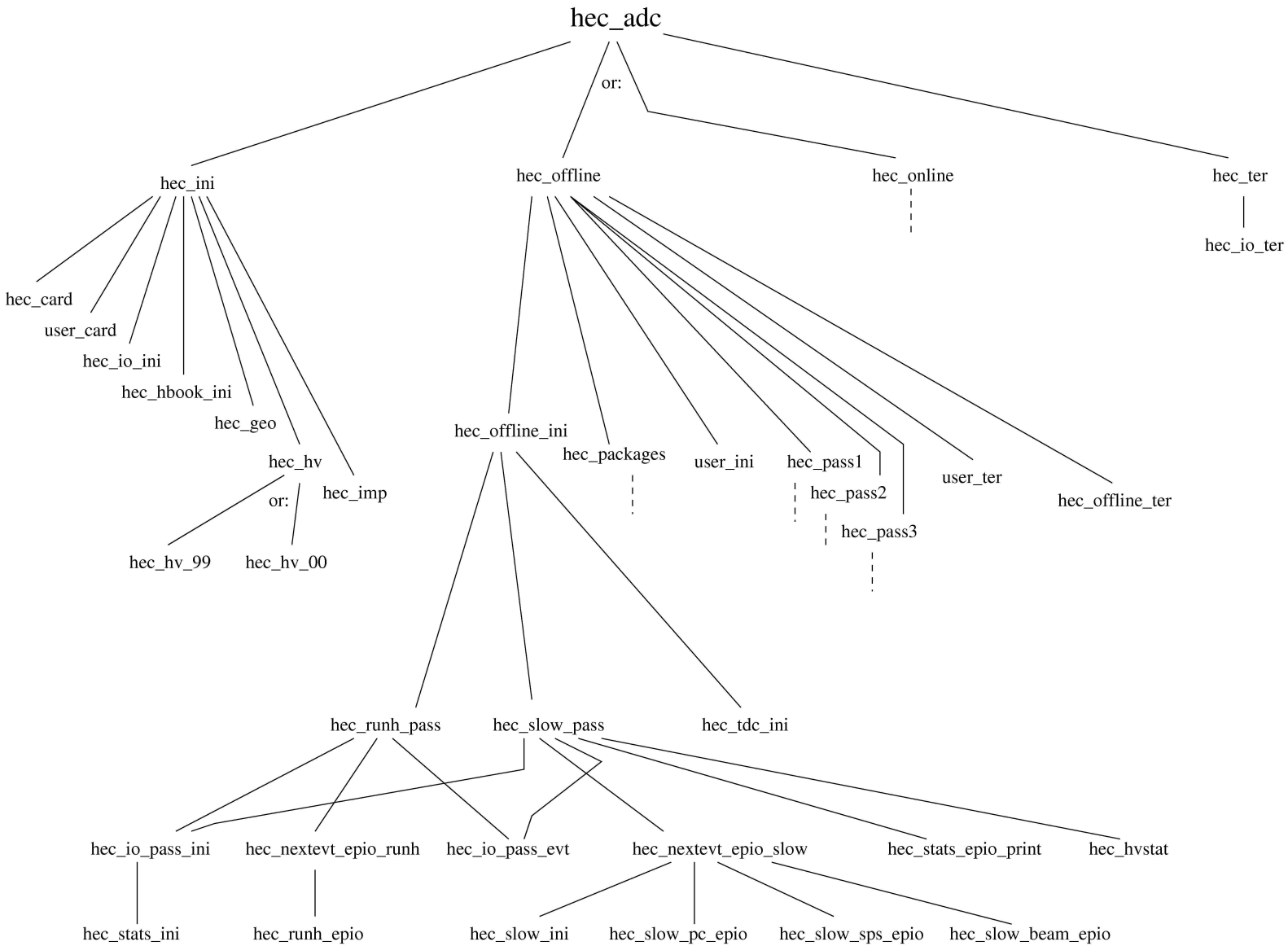
These routines do seperate passes over the data.
The user can include his or her stuff into the user routines here.

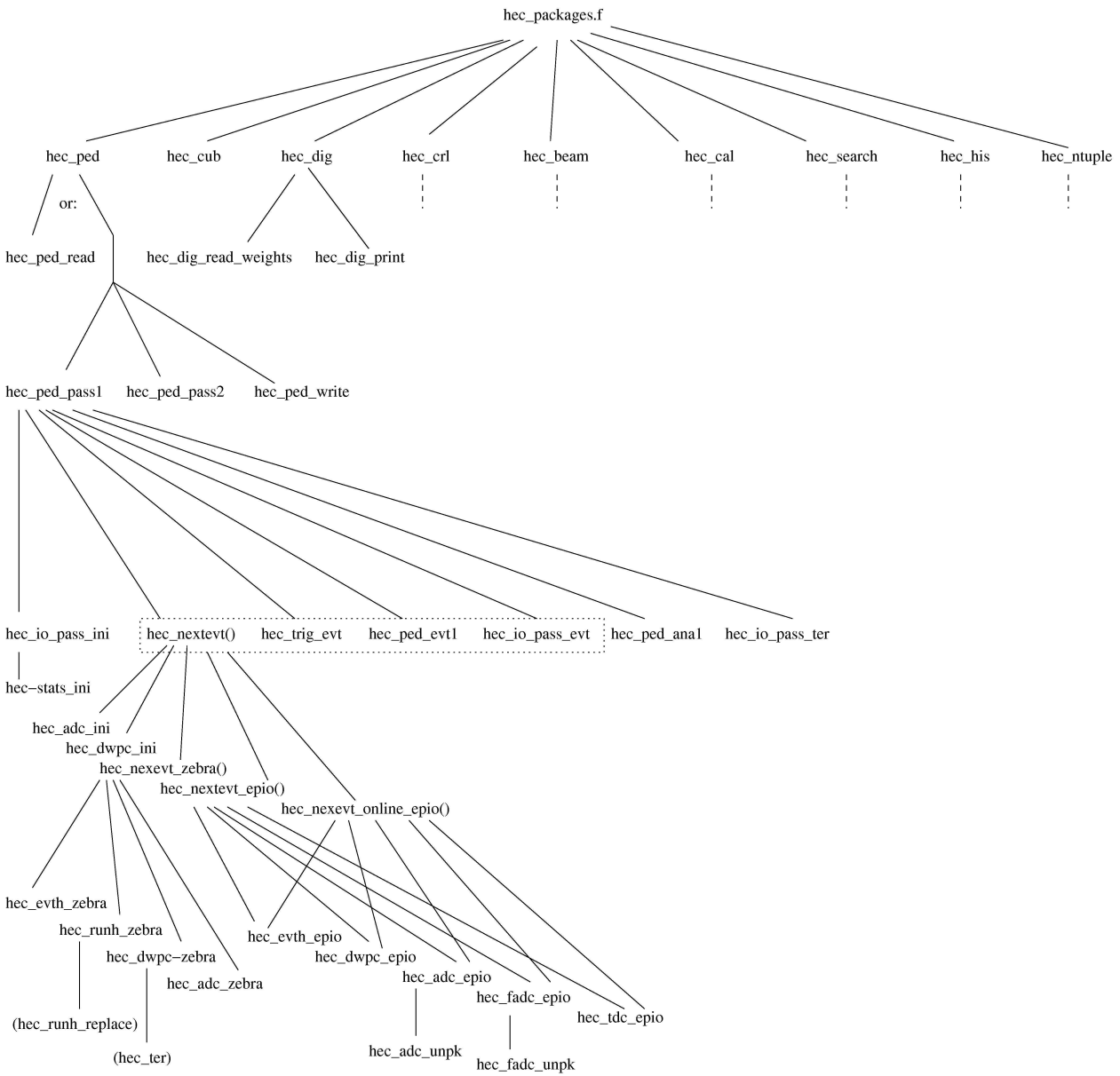
user_ter

This does nothing unless the user changes it.

hec_offline_ter

```
      subroutine hec_offline
c
c offline mode of hec_adc
c
      implicit none
c
c user datacard values common
c
      include'hec_par.inc'      !shared parameters
      include'hec_datacard.inc' !datacard values
c
c offline initialization
c
      call hec_offline_ini
c
c call system packages main routine
c
      call hec_packages
c
c user global initialization
c
      call user_ini
c
c pass 1 over data set for user
c
      if (n_pass .ge. 1) call hec_pass1
c
c pass 2 over data set for user
c
      if (n_pass .ge. 2) call hec_pass2
c
c pass 3 over data set for user
c
      if (n_pass .ge. 3) call hec_pass3
c
c user termination
c
      call user_ter
c
c offline termination
c
      call hec_offline_ter
c
      end
```


hec_adc
or:
hec_ini
hec_offline
hec_online
hec_ter
hec_io_ter
hec_card
user_card
hec_io_ini
hec_hbook_ini
hec_geo
hec_hv
hec_imp
or:
hec_hv_99
hec_hv_00
hec_offline_ini
hec_packages
user_ini
hec_pass1
hec_pass2
hec_pass3
user_ter
hec_offline_ter
hec_runh_pass
hec_slow_pass
hec_tdc_ini
hec_io_pass_ini
hec_nextevt_epio_runh
hec_io_pass_evt
hec_nextevt_epio_slow
hec_stats_epio_print
hec_hvstat
hec_stats_ini
hec_runh_epio
hec_slow_ini
hec_slow_pc_epio
hec_slow_sps_epio
hec_slow_beam_epio


hec_packages.f
hec_ped
hec_cub
hec_dig
hec_crl
hec_beam
hec_cal
hec_search
hec_his
hec_ntuple
or:
hec_ped_read
hec_dig_read_weights
hec_dig_print
hec_ped_pass1
hec_ped_pass2
hec_ped_write
hec_io_pass_ini
hec_nextevt()
hec_trig_evt
hec_ped_evt1
hec_io_pass_evt
hec_ped_ana1
hec_io_pass_ter
hec−stats_ini
hec_adc_ini
hec_dwpc_ini
hec_nexevt_zebra()
hec_nextevt_epio()
hec_nexevt_online_epio()
hec_evth_zebra
hec_runh_zebra
hec_dwpc−zebra
hec_adc_zebra
hec_evth_epio
hec_dwpc_epio
hec_adc_epio
hec_fadc_epio
hec_tdc_epio
(hec_runh_replace)
(hec_ter)
hec_adc_unpk
hec_fadc_unpk

Summary

- hec_adc - “road map” has been created to be a guide for C++ code development

- Next talk by Naoko Kanaya: EPIO converter