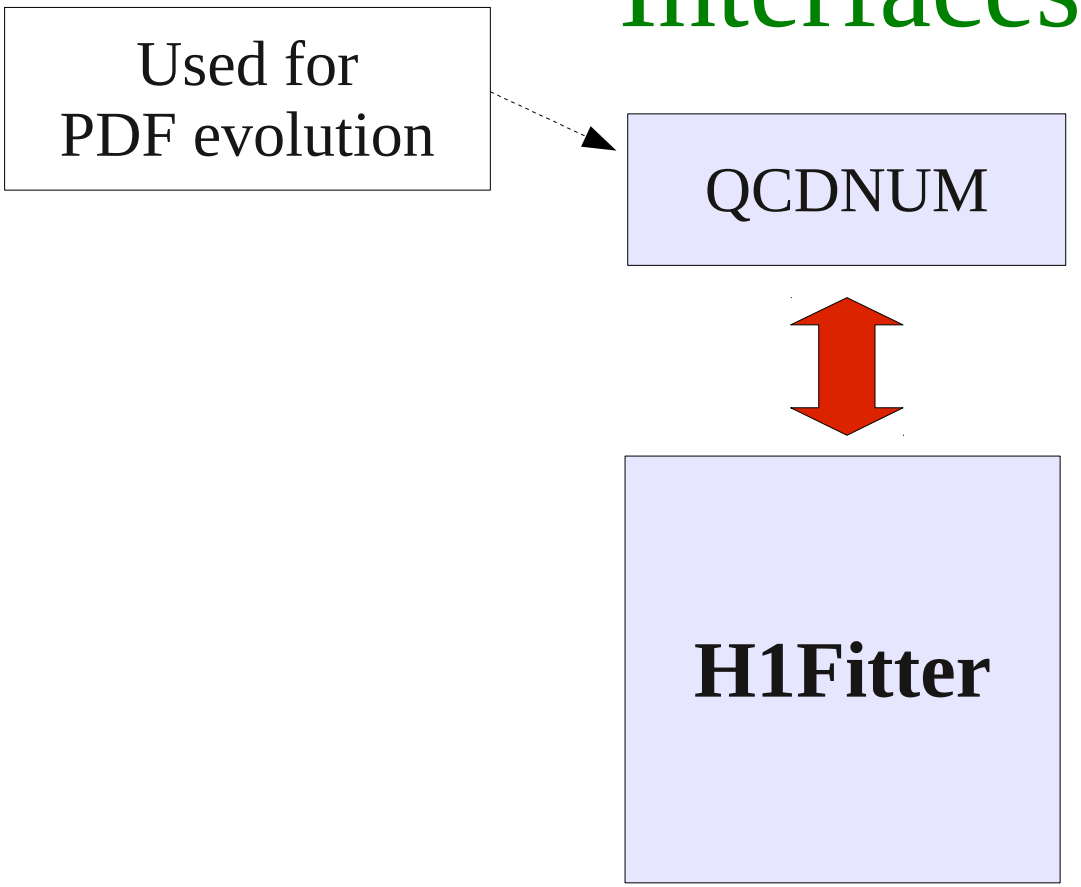
HERA Fitter: Theory Interfaces

Krzysztof Nowak

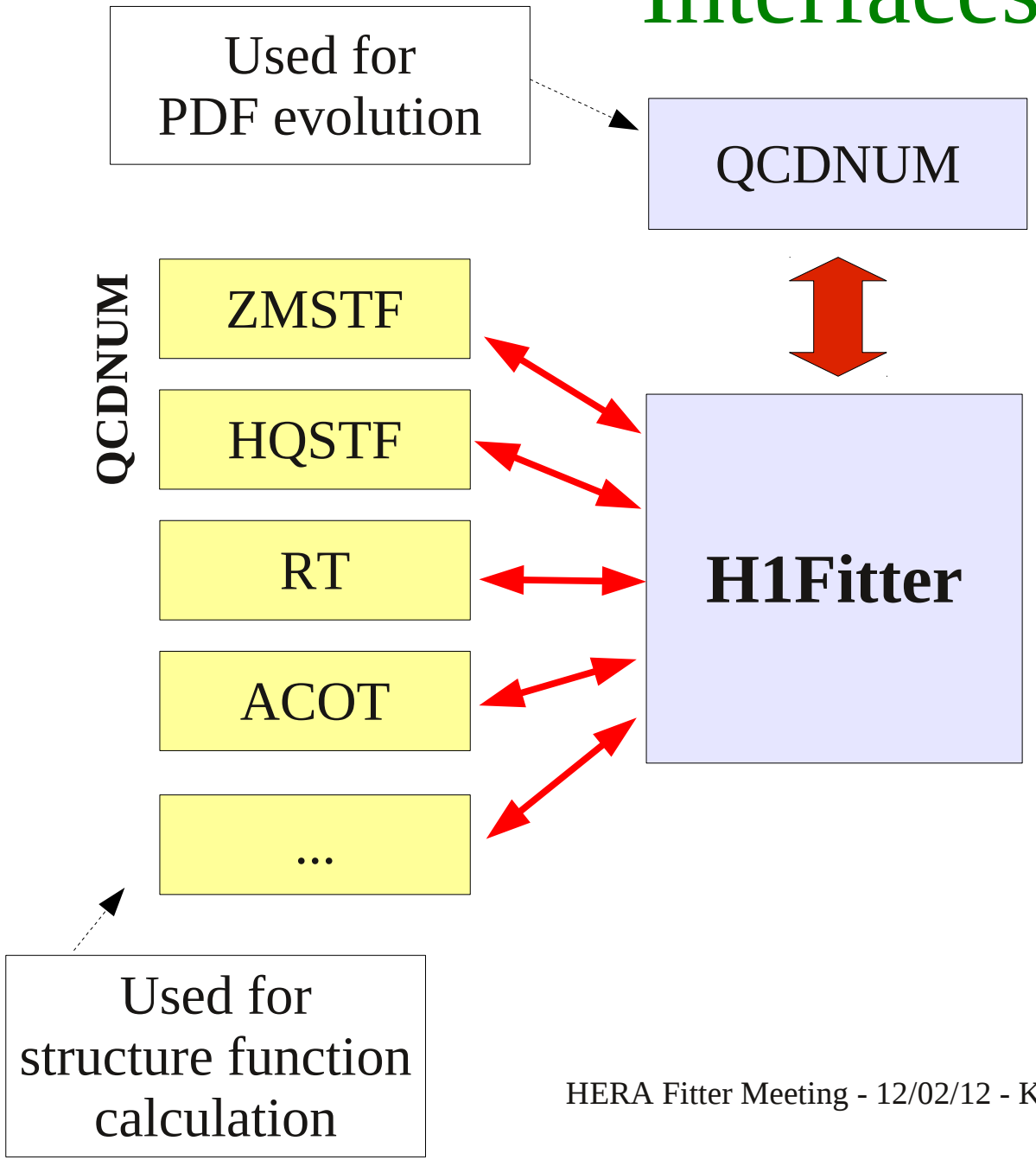
Introduction

- H1Fitter – unique hep project combining theory developments in many sometimes relatively distinct areas
- Great opportunity for a profitable experience exchange, forum for fruitful comparisons

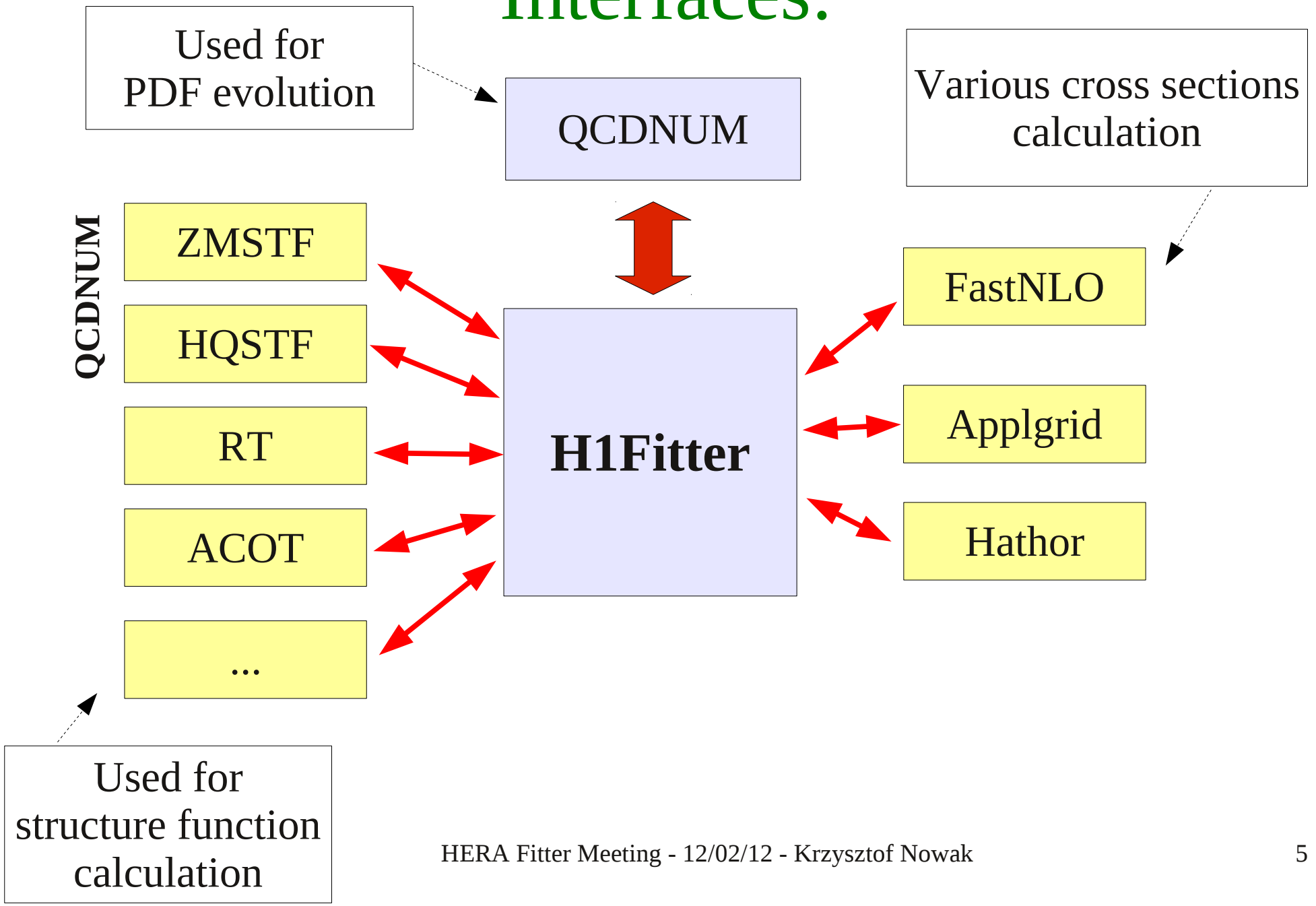
Interfaces:



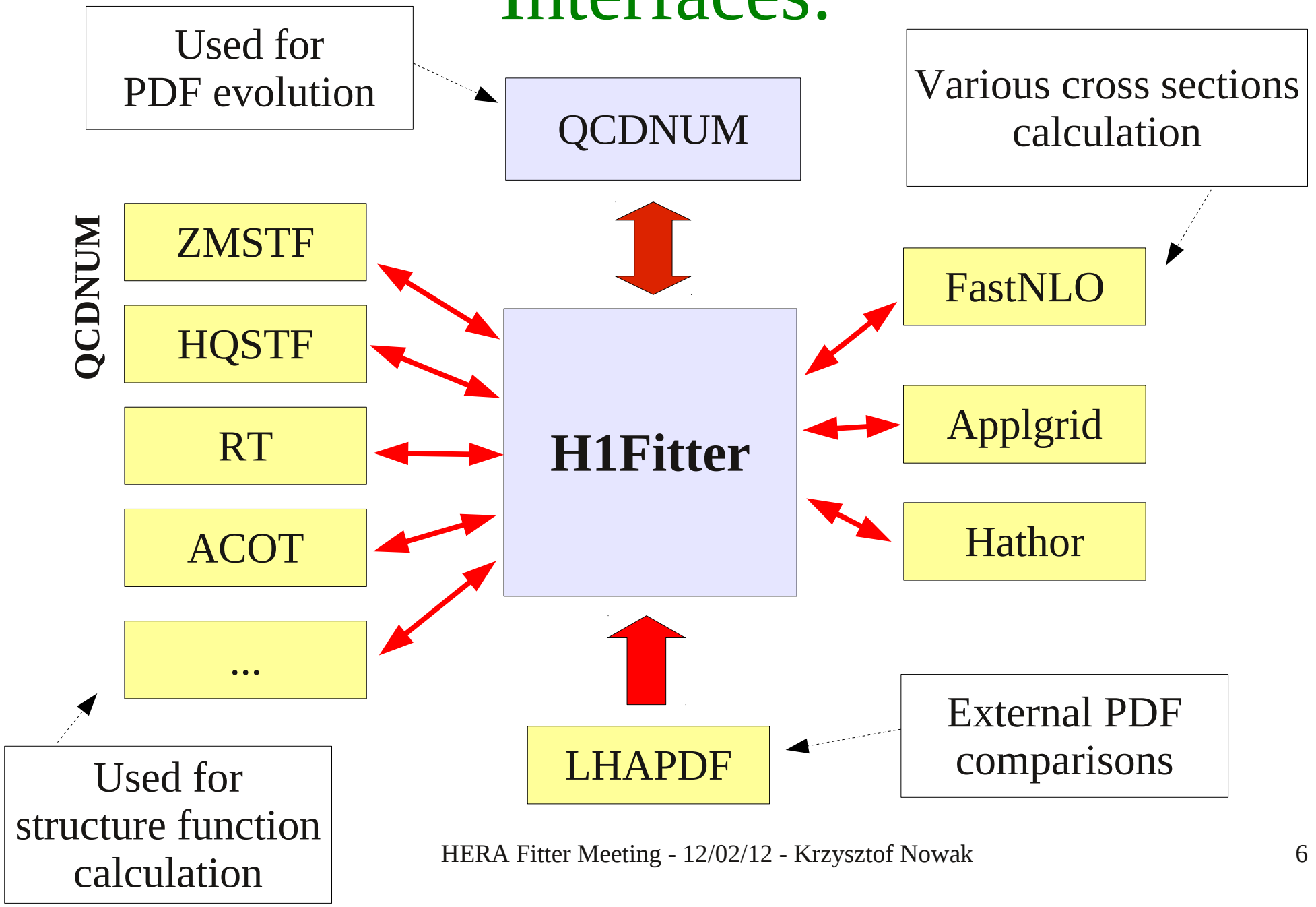
Interfaces:


Used for
PDF evolution
QCDNUM
QCDNUM
ZMSTF
HQSTF
RT
ACOT
...
H1Fitter
Used for
structure function
calculation

Interfaces:


Used for
PDF evolution
QCDNUM
Various cross sections
calculation
QCDNUM
ZMSTF
HQSTF
RT
ACOT
...
H1Fitter
FastNLO
Applgrid
Hathor
Used for
structure function
calculation

Interfaces:


Used for
PDF evolution
QCDNUM
Various cross sections
calculation
QCDNUM
ZMSTF
HQSTF
RT
ACOT
...
H1Fitter
FastNLO
Applgrid
Hathor
Used for
structure function
calculation
LHAPDF
External PDF
comparisons

Interfaces strategy

- **Modular**:
  - keep all the different schemes possibly decoupled
  - possibly no modification to original codes
- **Steerable:** minimize number of hardcoded settings
- **Nonpartial**: all schemes equally treated
- **Flexible**: open for different requirements of different schemes
- **Expandable**: Additional schemes easily accomodable
- Open for changes, might change as needed

Steering

```
*
* Main steering cards
*
  ITheory = 0   ! =0 use collinear factorisation with QCDNUM

  IOrder  = 2   ! For itheory =0
    (collinear factorisation) : LO fit (1) or NLO (2) or NNLO (3)

  Q02     = 1.9 ! Evolution starting scale

 ! --- Scheme for heavy flavors :
 ! ---  HF_SCHEME = 'ZMVFNS'    : ZM-VFNS (massless),
 ! ---  HF_SCHEME = 'RT'        : Thorne-Roberts VFNS (massive)
 ! ---  HF_SCHEME = 'RT FAST'   : Fast approximate TR VFNS scheme
 ! ---  HF_SCHEME = 'ACOT Full' : ACOT - F.Olness Ver. (massive)
 ! ---  HF_SCHEME = 'ACOT ZM'   : ACOT - F.Olness Ver. (massless)
 ! ---  HF_SCHEME = 'FF' : Fixed Flavour Number Scheme from qcdnum

  HF_SCHEME = 'RT'
```

Structure Function Calculation

```
C Follow a qcdnum formalism: calculate structure functions array
C                                          for a grid in (q2, x)
      Double precision q2(npts), x(npts)
      Double precision F2(npts), FL(npts), xF3(npts)
      Double precision F2c(npts), FLc(npts), F2b(npts), FLb(npts)

C Calculate structure functions in zero mass approximation
      call UseZmvnsScheme    (F2, FL, xF3,
$        q2, x, npts)

C Use requested massive schemes
      if (ACOT) then
         call UseAcotScheme (F2, FL, xF3, F2c, FLc, F2b, FLb,
$           x, q2, npts)
      elseif (RT) then
         call UseRtScheme    (F2, FL, xF3, F2c, FLc, F2b, FLb,
$           x, q2, npts)
      elseif (HQSTF) then
         call UseHqstfScheme(F2, FL, xF3, F2c, FLc, F2b, FLb,
$           x, q2, npts)
      endif

C Calculate reduced cross sections
      Xsec = (1+polarity)* 0.5 * (yplus*F2 - yminus*xF3 - y*y*FL)
```

Structure Function Calculation

```
C Follow a qcdnum formalism: calculate structure functions array
C                                          for a grid in (q2, x)
      Double precision q2(npts), x(npts)
      Double precision F2(npts), FL(npts), xF3(npts)
      Double precision F2c(npts), FLc(npts), F2b(npts), FLb(npts)

C Calculate structure functions in zero mass approximation
      call UseZmvnsScheme   (F2, FL, xF3, Z0corrF2, Z0corrFL,
$       q2, x, npts, ...)

C Use requested massive schemes
      if (ACOT) then
         call UseAcotScheme (F2, FL, xF3, F2c, FLc, F2b, FLb,
$          x, q2, npts, ...)
      elseif (RT) then
         call UseRtScheme   (F2, FL, xF3, F2c, FLc, F2b, FLb,
$          x, q2, npts, Z0corrF2, Z0corrFL, ...)
      elseif (HQSTF) then
         call UseHqstfScheme(F2, FL, xF3, F2c, FLc, F2b, FLb,
$          x, q2, npts, ...)
      endif

C Calculate reduced cross sections
      Xsec =(yplus*F2 - yminus*xF3 - y*y*FL) * …
```

Theory wrappers

- Used to bring all theory modules to a common ground:

  - ACOT ($F_3 \rightarrow xF_3$)
  - RT ($F_2 = F_2^{RT} * Z_0^{corrF2}$, $F_L = F_L^{RT} * Z_0^{corrFL}$)
  - HQSTF ($F_2 = F_2^{HQSRF} + F_2^c + F_2^b$)

- Chosing selected version/options/kfactors etc. of the given scheme

ACOT Full | ACOT ZM | ACOT Chi

k-factors usage

- Used to fasten complicated calculations
- ACOT:

First iteration

$$k_2 = F_2^{NLO} / F_2^{LO}$$
$$k_L = F_L^{NLO} / F_L^{LO}$$
$$k_{2c} = F_{2c}^{NLO} / F_{2c}^{LO}$$
$$k_{Lc} = F_{Lc}^{NLO} / F_{Lc}^{LO}$$
$$k_{2b} = F_{2b}^{NLO} / F_{2b}^{LO}$$
$$k_{Lb} = F_{Lb}^{NLO} / F_{Lb}^{LO}$$

Following iterations

$$F_2^{NLO} = F_2^{LO} * k_2$$
$$F_L^{NLO} = F_L^{LO} * k_L$$
$$F_{2c}^{NLO} = F_{2c}^{LO} * k_{2c}$$
$$F_{Lc}^{NLO} = F_{Lc}^{LO} * k_{Lc}$$
$$F_{2b}^{NLO} = F_{2b}^{LO} * k_{2b}$$
$$F_{Lb}^{NLO} = F_{Lb}^{LO} * k_{Lb}$$

- RT:

$$k_2 = F_2^{RT} / F_2^{QCDNUM}$$
$$k_L = F_L^{RT} / F_L^{QCDNUM}$$
$$\ldots$$

$$F_2^{RT} = F_2^{QCDNUM} * k_2$$
$$F_L^{RT} = F_L^{QCDNUM} * k_L$$
$$\ldots$$

- Time efficiency improved, but approximate calculations

Summary

- H1Fitter:
  - a package that collects developments from different theory groups
  - providing forum for a comparisons on possibly equal footing
  - encouraging share of experience
- Hoping to accomodate new theory modules soon
- Feedback is highly welcomed