

**Lawrence Livermore National Laboratory**

## **LLNL Nuclear Data Processing Codes**



**Bret Beck, Dave Brown, Frank Daffin, Chris  
Hagmann, Sofia Quaglioni and Neil Summers**

This work performed under the auspices of the U.S. Department of Energy by Lawrence  
Livermore National Laboratory under Contract DE-AC52-07NA27344.

# Outline

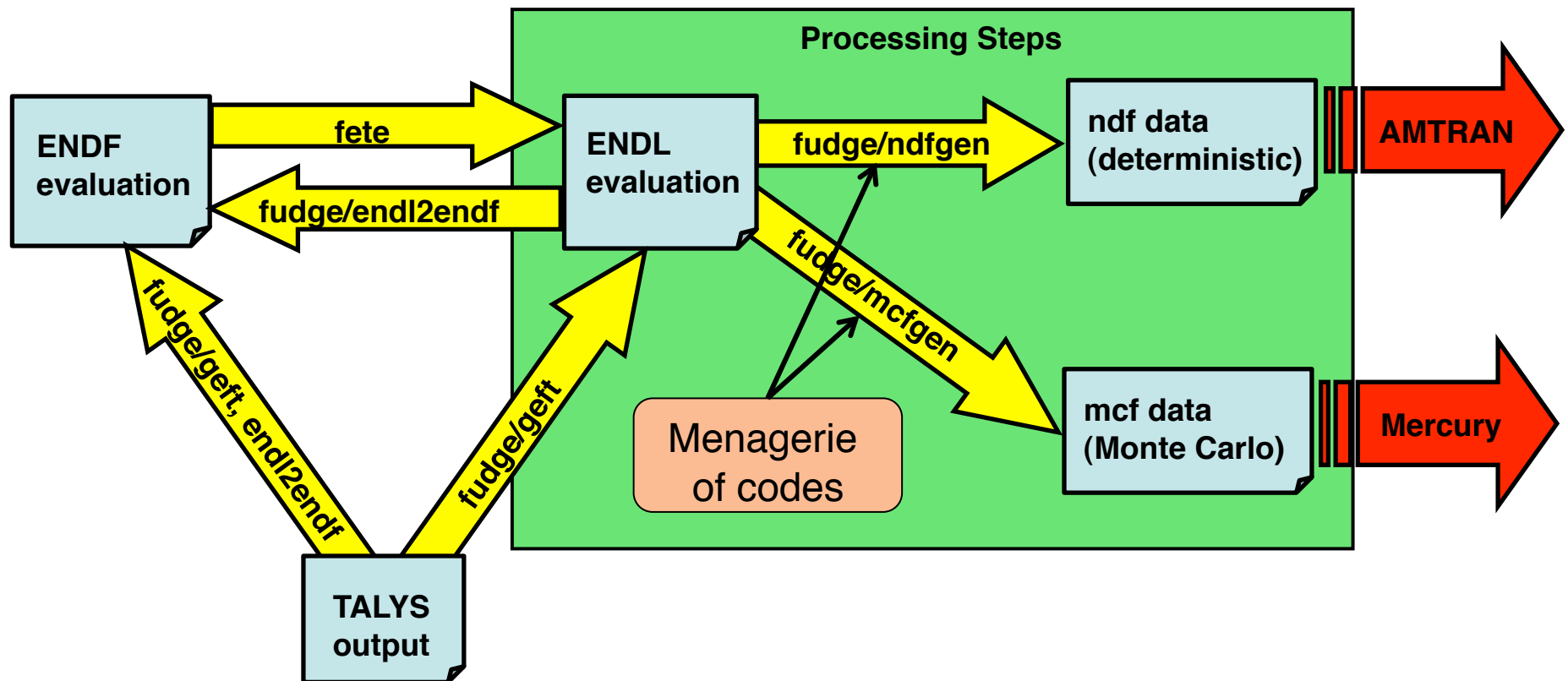
---

- The menageries
- Refactoring
- Future work
- FUDGE (For Updating Data and Generating Endl)
  - Data management
    - Modifying
    - Plotting
    - Processing

Motto: All for one (format) and one for all (FUDGE)



# Interim LLNL Processing



Recently moved processing into FUDGE, in part to remove the menagerie of codes

# The code menagerie – not as portable as python

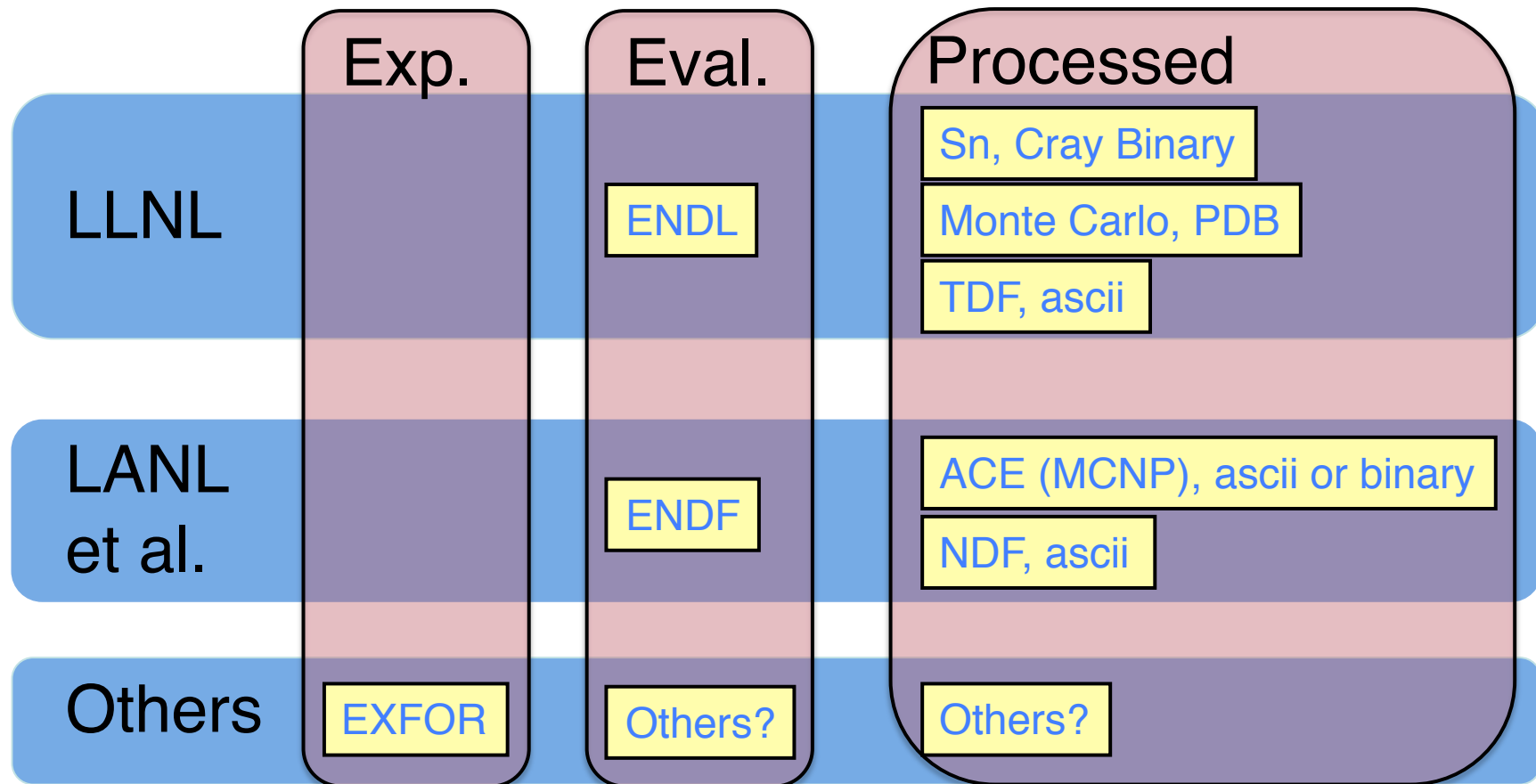
C, C++ and FORTRAN codes for processing, etc.

ENDLUURtoPDB	endepC++
bdfIsFile.so	endlret
bdfIs_info	fudge2dThin.so
cendlret	fudgeConvolutions.so
checkMCF_PDBFile4residualZA	getInfoFromMCF_CrayFile
checkNDFFile4residualZA	getInfoFromMCF_PDBFile
cmcf_pdbupdate	getInfoFromNDF_CrayFile
cmcfbin	mcf_GetDates
cmcfupdate	mcf_IsCrayOrPDB
cndfbin	mcf_add_zalist
cndfexplode	mcfgen
cndfgen	ndfFile.so
cndfupdate	ndf_GetDates
create	ndf_table
crossSectionAdjustForHeatedTarget.so	ndfgen
cross_ChangeDate	nuclearLLNLMisc.so
endep	tart_ChangeDates

endep.com	endlmod.com	mcf_IsCray.com	mcf_IsPDB.com
mcfmod.com	ndfmod.com	egdlmcf.pl	zacis.pl

Goal: Convert most of this coding to FUDGE/python

## The format menagerie – not portable between labs, etc.



Each format is different and has different access routines

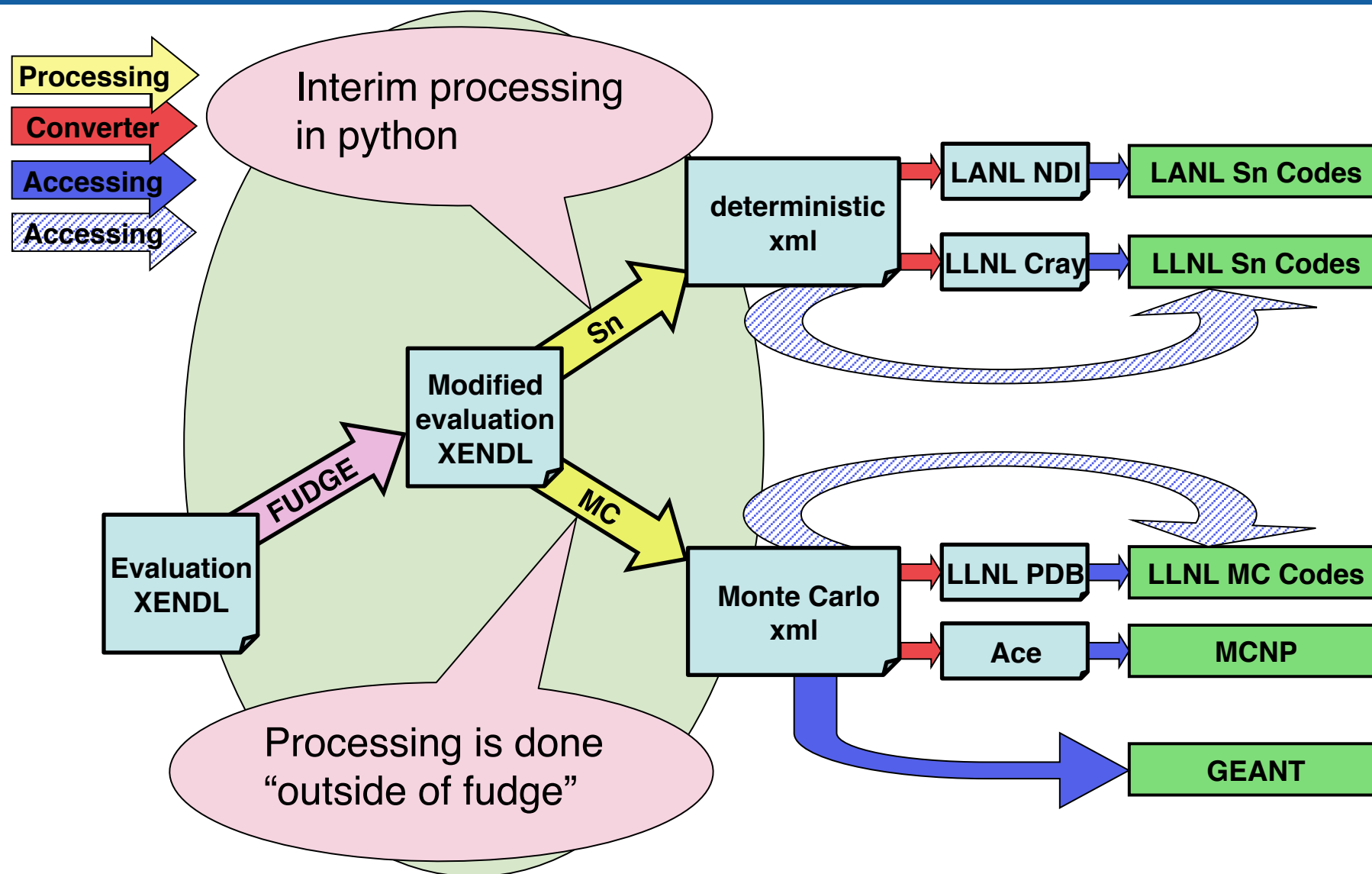
Goal: have one format for all

---

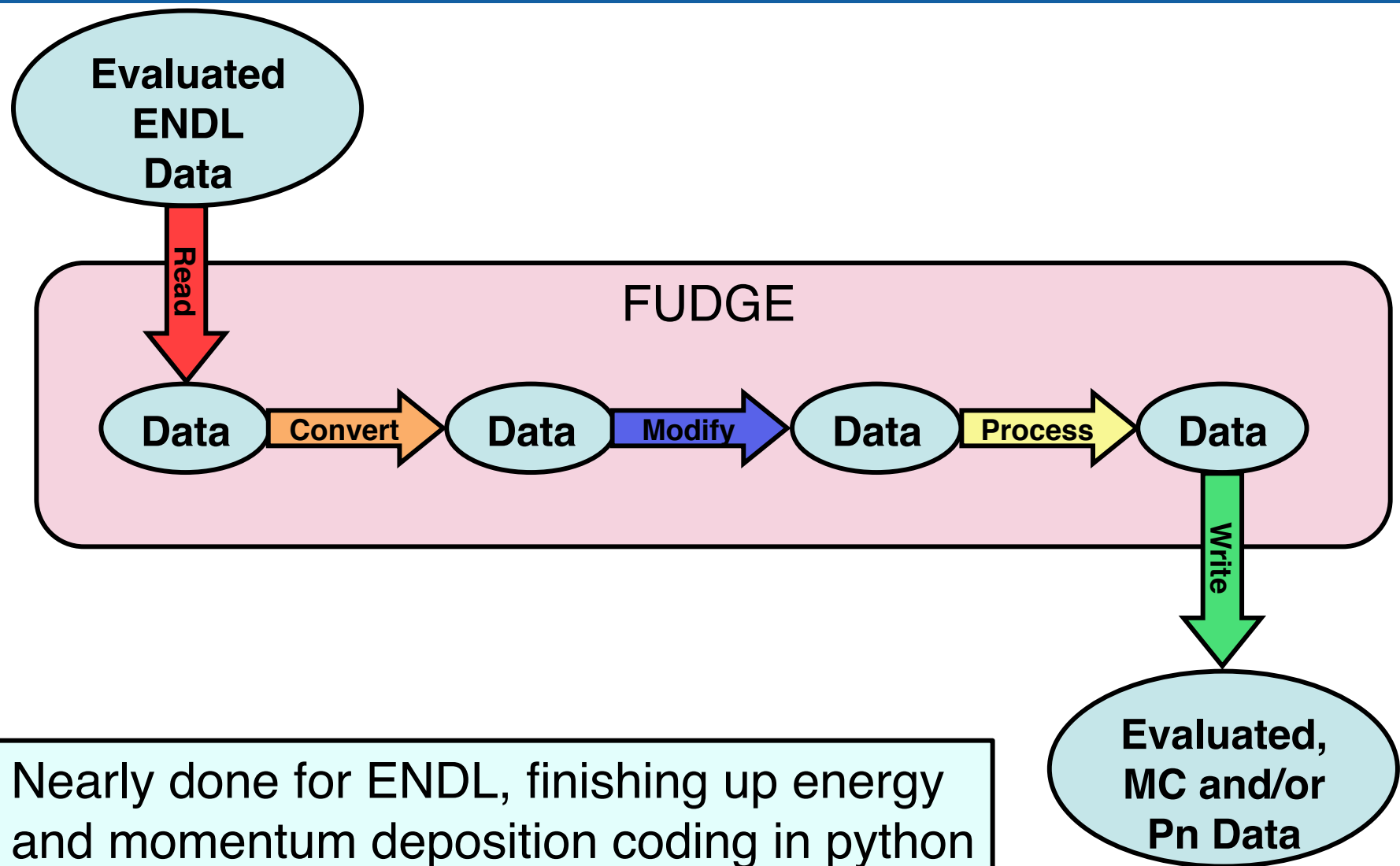
## What we are doing to remove the menageries



## Prior rewriting of processing codes



## Refactoring fudge and the processing





## Code refactoring summary

---

- Have FUDGE handle most of the processing directly
  - Use python when speed is not an issue
    - Fast code development
    - Simpler code with well designed classes
  - Use C or C++ for computationally intensive tasks
    - Heating cross sections: completed
    - Calculating transfer matrices
      - Completed for ENDL
      - ~3/4 done for extra ENDF data types
    - Cross sections, resonance region parameters to point-wise
    - URR probability tables
      - Currently use NJOY

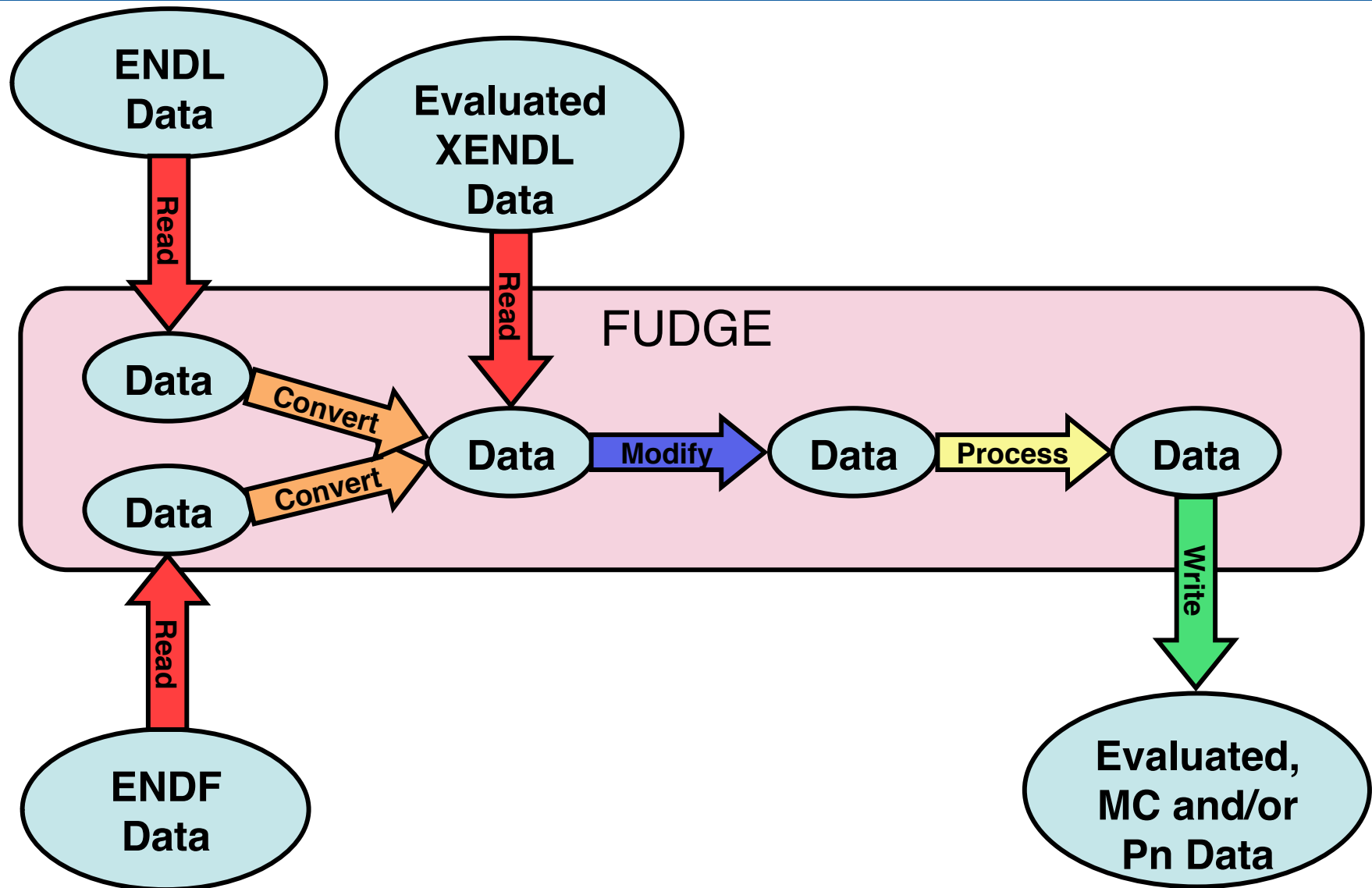


## Data format refactoring summary

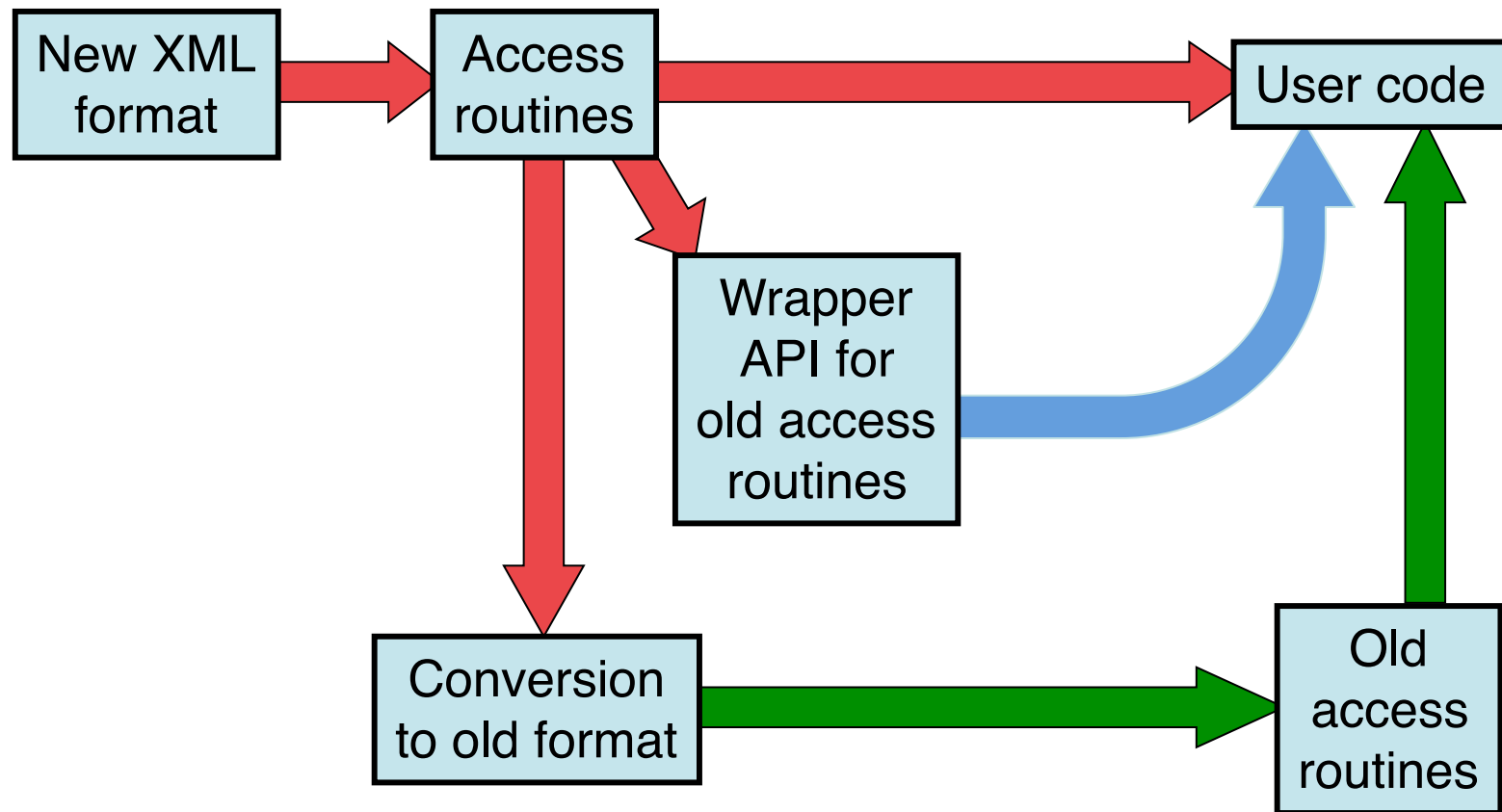
---

- Have “one” format for evaluated, MC and deterministic
  - About six months ago I realized that evaluated, MC and deterministic can be put into one format with some simple changes to the formats I was developing
  - xml vs HDF5
    - For python, best format is probably xml
    - For speed, may also want an HDF5 format for MC and deterministic data
      - Converting: xml  $\Leftrightarrow$  HDF5
- Should also include experimental (i.e., exfor)

## Post-Doc expanding to support reading/converting of ENDF data



# Three steps to nirvana



## Future work

---

- For deterministic finish processing ENDF outgoing particle data types (evaporation models, etc.)
- Add particle database
  - Mass, spin, parity, level structure (Neil Summers), etc.
- We have stimulus money (5 years worth) for a Post-Doc to
  - expand format to support ENDF evaluated data types
  - add web-based visualization
- Deterministic multi-temperature data
  - Currently done off line by me with legacy codes, very fragile
- Evaluate HDF5 option, mainly for speed
- Develop XML schema
- Continue collaborating with SLAC people to implement in GEANT
- Longer term: May put most MC processing into access routines
  - Allow user to pick group boundaries on the fly

