Lawrence Livermore National Laboratory

# LLNL Nuclear Data: Processing Codes Update and Proposed New Format

Caleb Mattoon

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

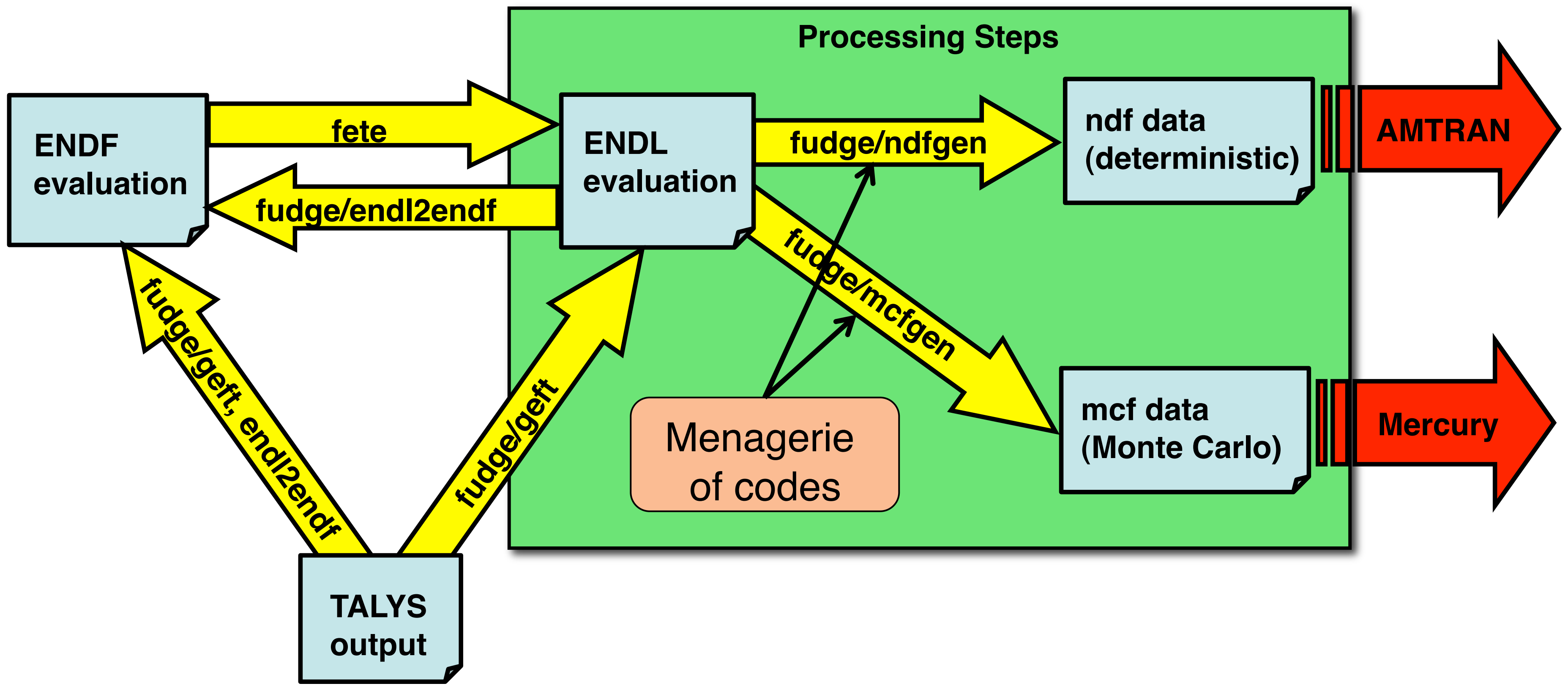
This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344

## Outline

- The old way: multiple processing codes and formats
- The new way: unified processing within FUDGE
- Introduction of the Nuclear Reaction Format ‘GND’

Goal: All for one (GND) 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

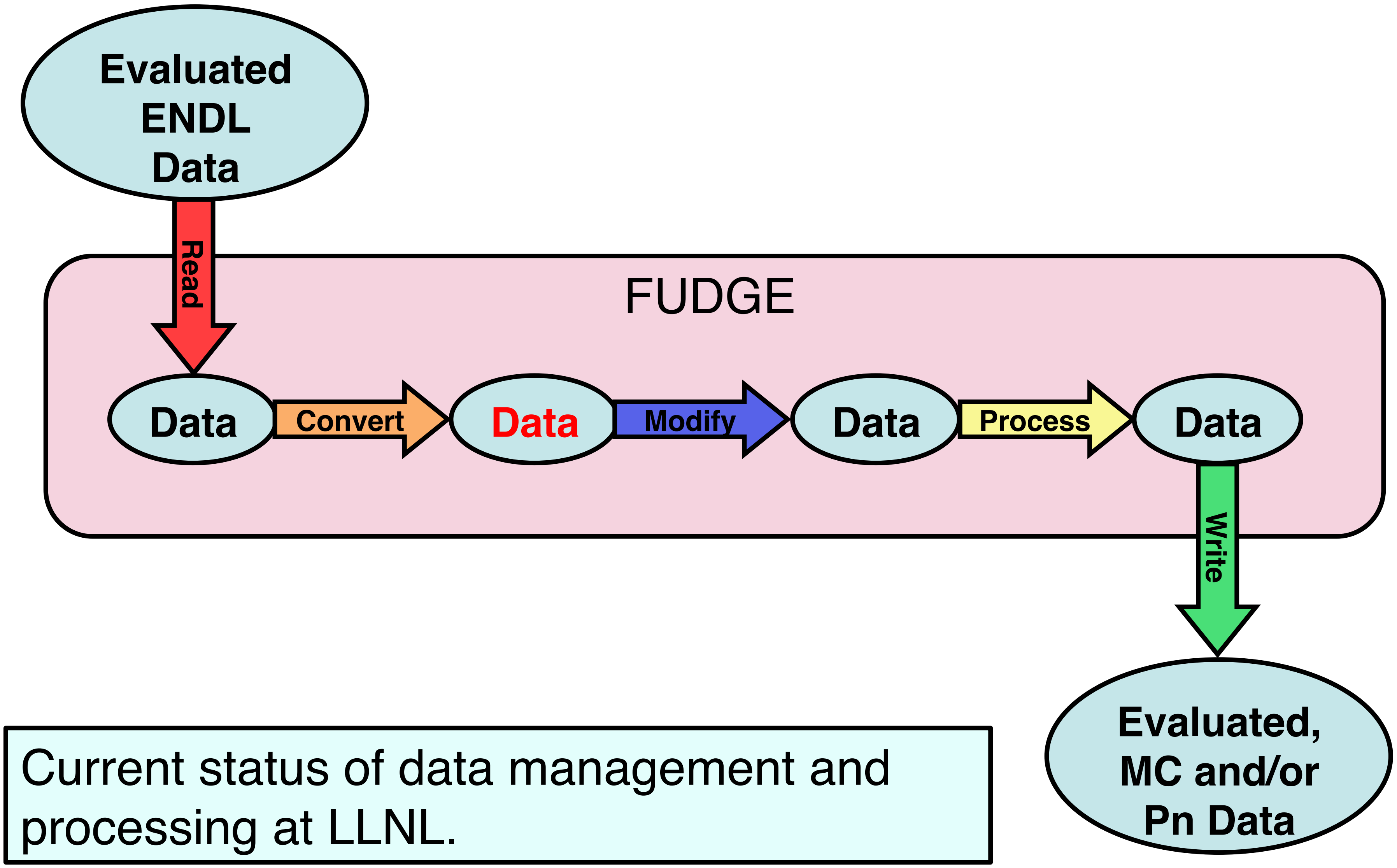
C, C++ and FORTRAN codes for processing, etc.

```
ENDLUURtoPDB                          endepC++
bdflsFile.so                          endlret
bdfls_info                            fudge2dThin.so
cendlret                              fudgeConvolutions.so
checkMCF_PDBFile4residualZA           getInfoFromMCFCrayFile
checkNDFFile4residualZA               getInfoFromMCFPDBFile
cmcf_pdbupdate                        getInfoFromNDFCrayFile
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
```

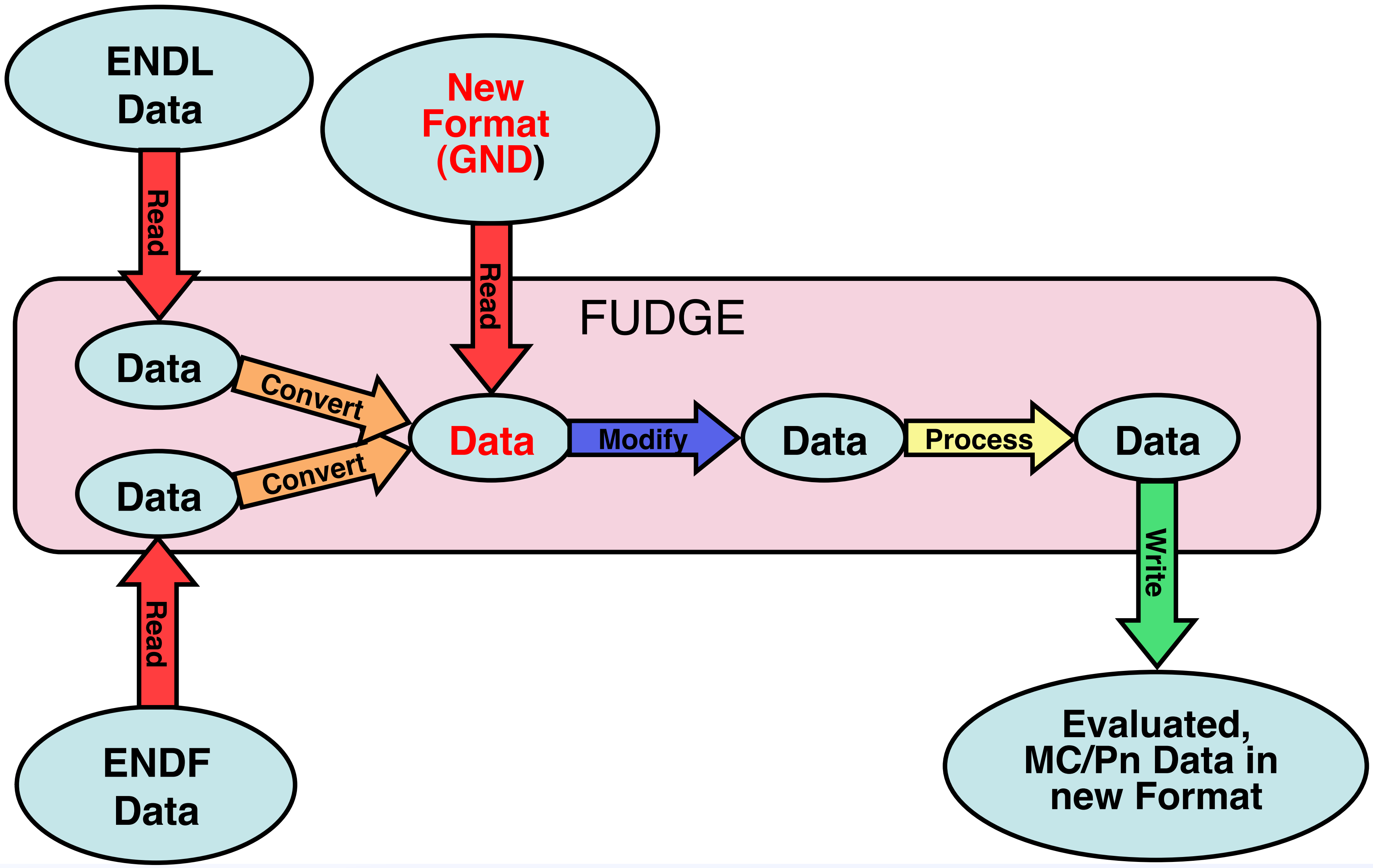
Goal: Convert most of this coding to FUDGE/python

## Simplified work-flow using FUDGE



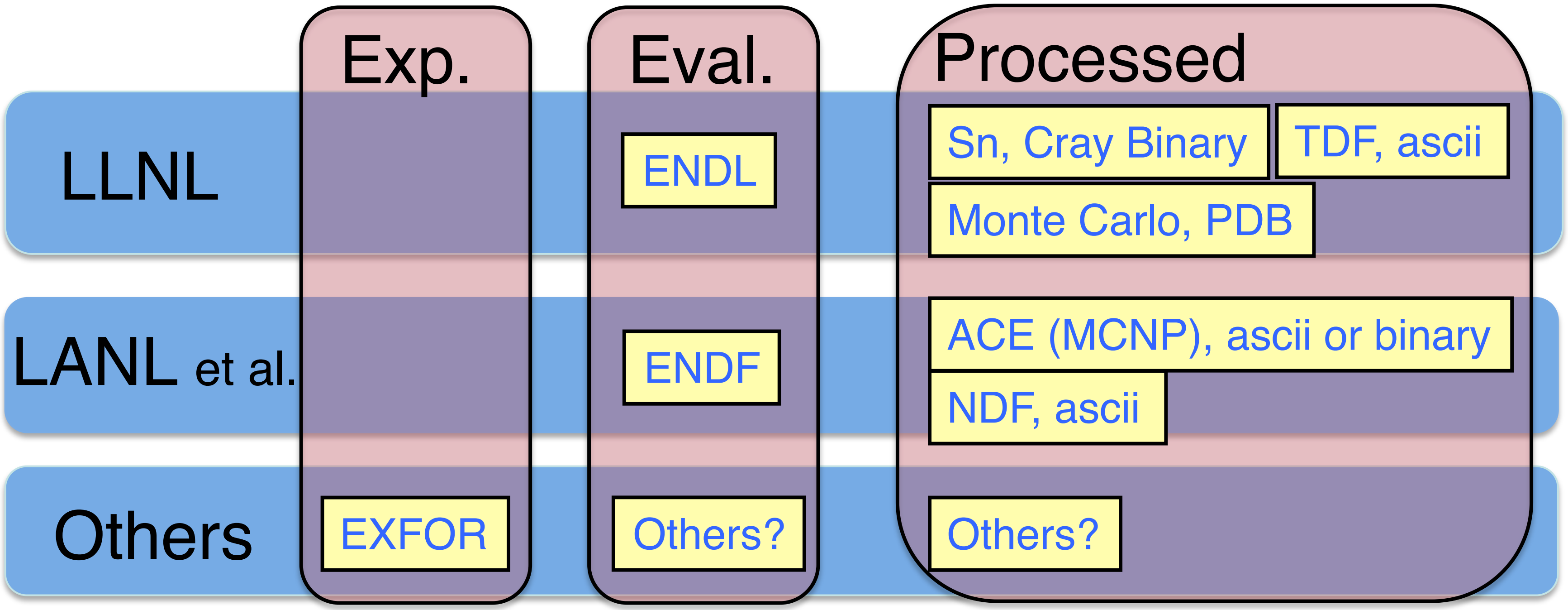
Current status of data management and processing at LLNL.

## Simplified work-flow using FUDGE + new format:



## Need for a new format:

- Currently have too many competing formats:



- Complicated formats and access routines
- New format should take advantage of OO (object-oriented) tools

## Need for new format, continued:

- Goals:
  - Replace the ‘menagerie’ of formats with one unified format for evaluated, MC, deterministic and unevaluated experimental data
  - Use a structured hierarchy (easily expressed in xml, HDF5 and in object-oriented languages) to store Nuclear Data
  - Data should be easy to understand, and representative of underlying physics

## GND (Generalized Nuclear Data) Format:

- Beta version released today (or soon)! Available at nuclear.llnl.gov  Release includes:
  - Converting ENDF-6 to python classes
    - Supports writing out to xml or ENDF-6 format
  - XML 'schema' (i.e., xml rules) defining the format
  - Conversion from XML to HDF5
  - Currently supported: cross sections, energy and angular distributions, multiplicities (corresponding to MF 1,3,4-6,8-10).
  - Next step: add support for resonances and emitted photons (corresponding to MF 2, 12-15)
- Please take a look and give us feedback!

## GND format:

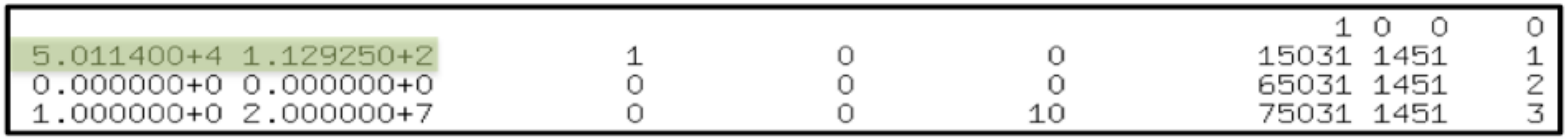
- GND format for nuclear data features an extensible, hierarchic structure:

```
<heatedTarget>                      // one target per file
   <styles>…</styles>               // for now, style=“evaluated”
   <documentation>…</documentation>
   <particles>…</particles>  // list all particles produced in all channels
   <channel>                        // one ‘channel’ per reaction
        <crossSection>…</crossSection>
        <product label=“n1”>
                <distribution>…</distribution></product></channel>
   <channel>…</channel>
</heatedTarget>
```

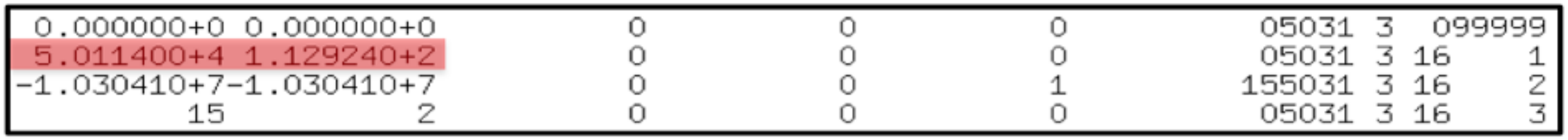
- This defines the **structure** of the new format, which is intended to be portable across file formats and programming languages.

## A few issues:

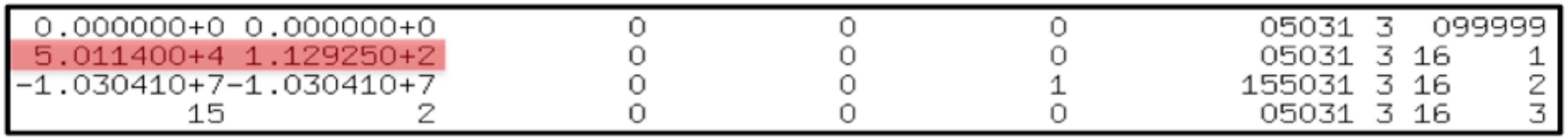
- New release implements conversion from ENDF to GND and back.
- Most of resulting ENDF file is identical to original
- Some exceptions:
  - New format only stores mass of the target once. If ENDF contains several different AWT values, the first value encountered will be used:



```
                                                                   1 0  0    0
 5.011400+4 1.129250+2          1          0          0          15031 1451    1
 0.000000+0 0.000000+0          0          0          0          65031 1451    2
 1.000000+0 2.000000+7          0          0         10          75031 1451    3
```



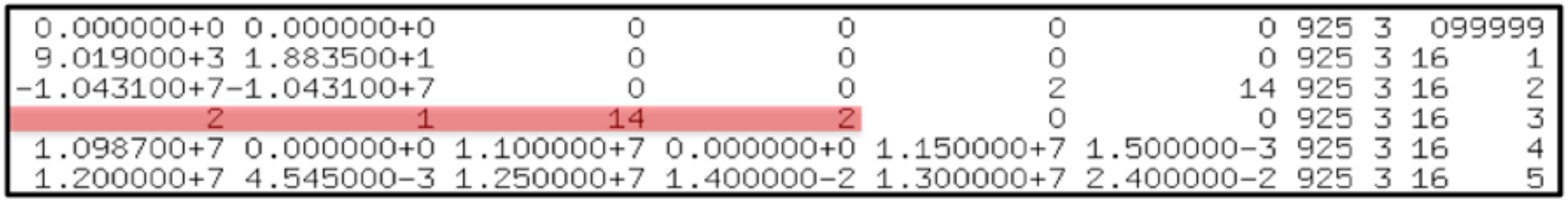
```
 0.000000+0 0.000000+0          0          0          0          05031 3  099999
 5.011400+4 1.129240+2          0          0          0          05031 3 16    1
-1.030410+7-1.030410+7          0          0          1         155031 3 16    2
         15          2          0          0          0          05031 3 16    3
```



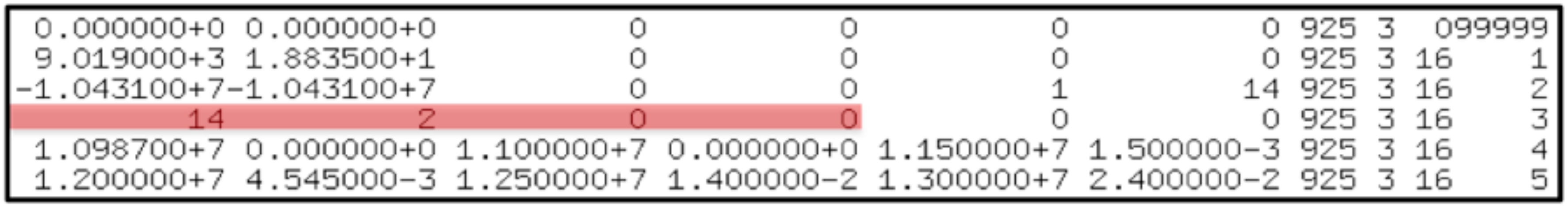
```
 0.000000+0 0.000000+0          0          0          0          05031 3  099999
 5.011400+4 1.129250+2          0          0          0          05031 3 16    1
-1.030410+7-1.030410+7          0          0          1         155031 3 16    2
         15          2          0          0          0          05031 3 16    3
```

## A few issues:

- New release implements conversion from ENDF to GND and back.
- Most of resulting ENDF file is identical to original
- Some exceptions:
  - ENDF may specify extra interpolation regions. New format combines these to one region:



```
 0.000000+0 0.000000+0          0          0          0          0 925 3  099999
 9.019000+3 1.883500+1          0          0          0          0 925 3 16     1
-1.043100+7-1.043100+7          0          0          2         14 925 3 16     2
          2          1         14          2          0          0 925 3 16     3
 1.098700+7 0.000000+0 1.100000+7 0.000000+0 1.150000+7 1.500000-3 925 3 16     4
 1.200000+7 4.545000-3 1.250000+7 1.400000-2 1.300000+7 2.400000-2 925 3 16     5
```



```
 0.000000+0 0.000000+0          0          0          0          0 925 3  099999
 9.019000+3 1.883500+1          0          0          0          0 925 3 16     1
-1.043100+7-1.043100+7          0          0          1         14 925 3 16     2
         14          2          0          0          0          0 925 3 16     3
 1.098700+7 0.000000+0 1.100000+7 0.000000+0 1.150000+7 1.500000-3 925 3 16     4
 1.200000+7 4.545000-3 1.250000+7 1.400000-2 1.300000+7 2.400000-2 925 3 16     5
```

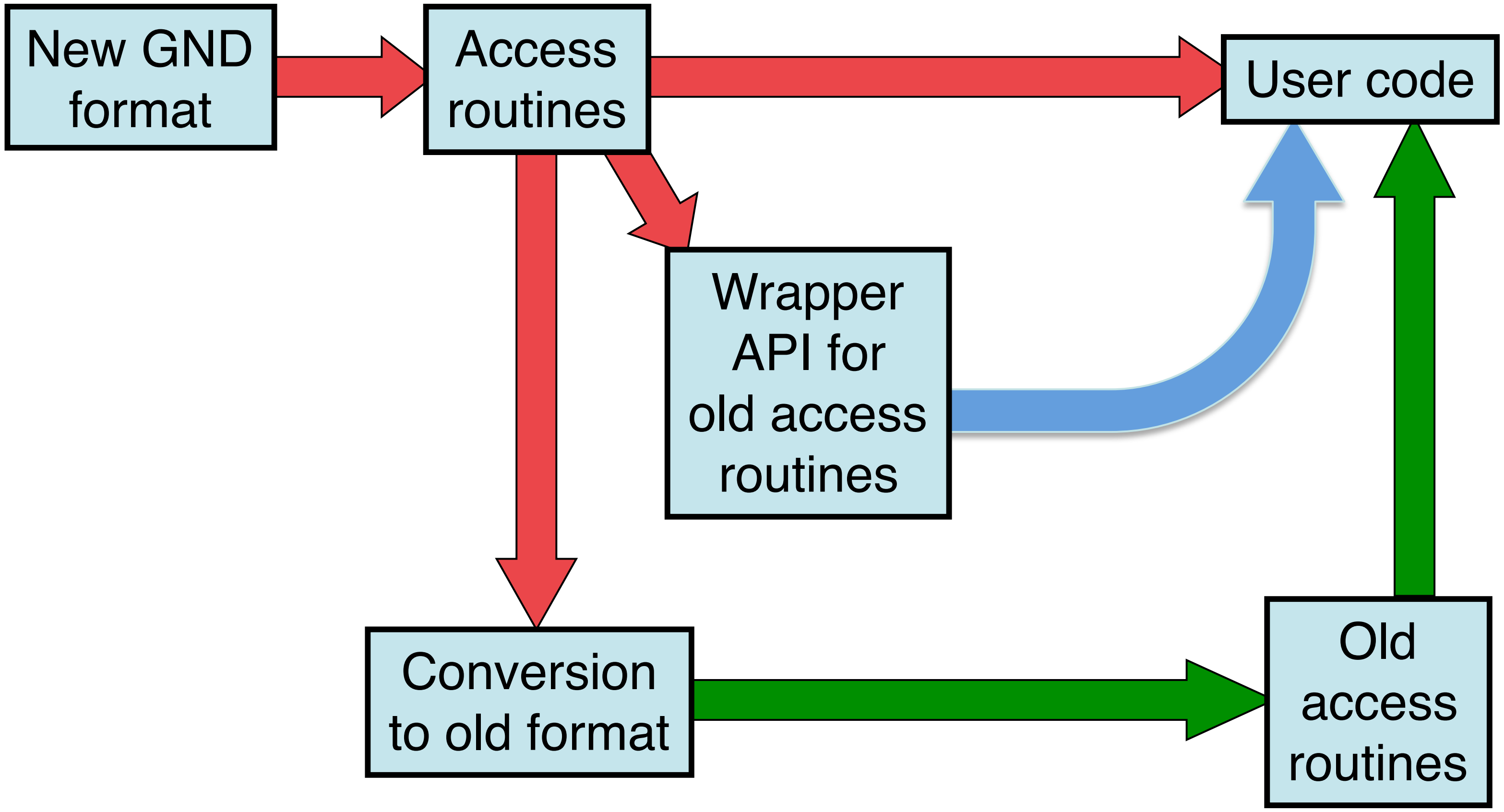
## A few issues:

- New release implements conversion from ENDF to GND and back.
- Most of resulting ENDF file is identical to original
- Some exceptions:
  - Energy-dependent multiplicities are sometimes used in ENDF when the multiplicity is fixed: (n,2n) for example. Only one multiplicity is stored for these reactions in the new format.
  - Some ENDF files have errors, and can't currently be converted to the new format! Common example: negative excitation energies
  - More detail is available in the GND documentation.

## Summary of New Format Release

- Release includes:
  - ‘Src’: python source code containing classes that represent new format.
  - ‘Data’: sample ENDF files for conversion plus miscellaneous data file
  - ‘schema.xsd’: XML schema for the new format
  - ‘Doc’: documentation on new format and tools
  - ‘Bin’: convenience tools for converting ENDF to the new format (and back)
    - rePrint.py
    - rePrintSample.py
    - toHDF5.py

## Three steps to nirvana



## Future work

- Processing:
  - For deterministic finish processing ENDF outgoing particle data types (evaporation models, etc.)
  - Deterministic multi-temperature data
    - Currently done off line with legacy codes, very fragile
  - Longer term: May put most MC processing into access routines
    - Allow user to pick group boundaries on the fly
- Format:
  - Continue expanding format (resonances, covariances, etc)
  - Develop XML schema (Done!)
  - Convert to HDF5 (Done!) and compare performance with xml
  - Collaboration with SLAC to implement in GEANT
  - Add particle database
    - Mass, spin, parity, level structure (Neil Summers), etc.
  - add web-based visualization

- Extra Slides:

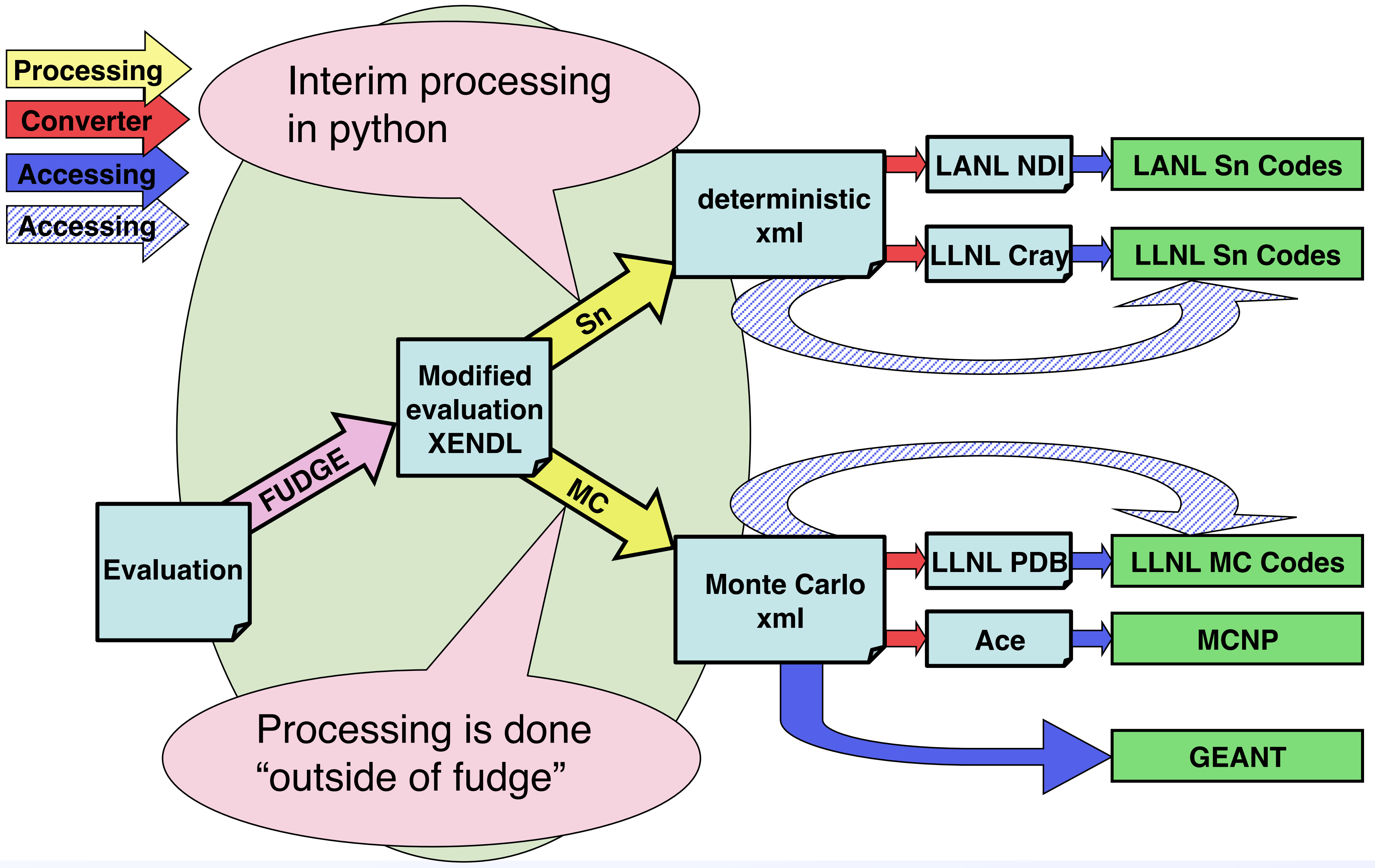
```
<?xml version="1.0" encoding="UTF-8"?>
<heatedTarget projectile="n1" target="Pu239" version="xendl version 0.1" temperature="0. K">

  <styles>
    <style name="evaluated" version="ENDF/B-VII"></style></styles>

  <documentation name="endfDoc"><![CDATA[
 94-Pu-239 LANL       EVAL-SEP06 Young,Chadwick,MacFarlane,Derrien
                      DIST-DEC06
----ENDF/B-VII        MATERIAL 9437
-----INCIDENT NEUTRON DATA
------ENDF-6 FORMAT
*****************************************************************
                  ENDF/B-VII EVALUATION
                        ...
 rest of documentation here. Documentation may also be included in each reaction channel
  ]]></documentation>


  <!-- next list all particles produced by all reaction channels
     notice that particle information, including target mass, only appears ONCE per file -->
  <particles>
    <particle token="Pu236" name="Pu236" genre="nuclearParticle" version="unknown" mass="236.046057964 amu"/>
    <particle token="Pu237" name="Pu237" genre="nuclearParticle" version="unknown" mass="237.048409658 amu"/>
    <particle token="Pu238" name="Pu238" genre="nuclearParticle" version="unknown" mass="238.049559894 amu"/>
    <particle token="Pu239" name="Pu239" genre="nuclearParticle" version="unknown" mass="239.052172899498 amu"/>
    ...
    <particle token="gamma" name="gamma" genre="photonParticle" transportable="true" version="unknown" mass="0. amu"/>
    <particle token="n1" name="n1" genre="nuclearParticle" transportable="true" version="unknown" mass="1.00866491574
      amu"/></particles>


  <!-- now list each reaction channel, including cross section, products, and energy/angular distributions
   The channel is identified by the outgoing products, but for now, MT numbers are also listed -->
  <channel projectile="n1" target="Pu239" label="0" outputChannel="n1 + Pu239" temperature="0. K"
      ENDL_CS="10,0" ENDF_MT="2" Q="0. eV" date="YYYYMMDD=20060901" genre="twoBody">
    <crossSection>
      <piecewise><variable index="0" frame="lab" name="energy_in" unit="eV" interpolation="byRegion"/>
        <variable index="1" frame="lab" name="crossSection" unit="barn" interpolation="byRegion"/>
        <region index="0" type="2d.xy" interpolation="linear,linear">1.e-05 0. 1000. ...
               1.98e+07 3.00669 2.e+07 3.015507</region>
      </piecewise></crossSection>
    <!-- each reaction product listed separately -->
    <product name="n1" label="n1" nativeData="angularTwoBody" multiplicity="1">
      <distributions nativeData="angular">
        <angular nativeData="LegendrePiecewise">
          <LegendrePiecewise><variable index="0" frame="centerOfMass" name="energy_in"
               unit="eV" interpolation="byRegion"/>
            <variable index="1" frame="centerOfMass" name="C_l" unit="" interpolation="byRegion"/>
            <region index="0" type="3d.xlc" interpolation="linear,linear">
              <!-- list angular distribution Legendre coefficients (MF 4) for each incident energy -->
              <energy value="1.e-05" index="0"><xData type="1d.x" length="3"> 0.5 0.0 0.0</xData></energy>
              ...
              <energy value="2.e+07" index="32"><xData type="1d.x" length="21"> 0.5 0.9421 ...</xData></energy>
            </region></LegendrePiecewise></angular></distributions></product>
    <product name="Pu239" label="Pu239" nativeData="unknown" multiplicity="1">
      <distributions nativeData="none"></distributions></product></channel>


  <!-- next channel: (n,n') to continuum, equivalent to MT 91.  Includes energy-angular distributions (from MF 6) -->
  <channel projectile="n1" target="Pu239" label="41" outputChannel="n1 + Pu239_u" temperature="0. K"
      ENDL_CS="11,0" ENDF_MT="91" Q="0. eV" date="YYYYMMDD=20060901" genre="NBody">
    <crossSection>...</crossSection>
    <product name="n1" label="n1" nativeData="NBody" multiplicity="1">
      <distributions nativeData="energyAngular">
        <energyAngular nativeData="KalbachMann">
          <KalbachMann><variable index="0" frame="centerOfMass" name="energy_in" unit="eV" interpolation="flat"/>
            <variable index="1" frame="centerOfMass" name="energy_out" unit="eV" interpolation="flat"/>
            <variable index="2" frame="centerOfMass" name="f" unit="1/eV" interpolation="linear"/>
            <variable index="3" frame="centerOfMass" name="r" unit="" interpolation="linear"/>
            <!-- energy-angular distributions in Kalbach-Mann form for each incident energy -->
            <energy value="636680." index="0"><xData type="1d.x" length="6"> 0.0 0.0001338959 0.0 7468.487
               0.0 0.0</xData></energy>
            ...
            <energy value="2.e+07" index="39"><xData type="1d.x" length="276"> 0.0 5.06662e-12 0.0005567875
               ...</xData></energy>
          </KalbachMann></energyAngular></distributions></product></channel>
```

```
<!-- Total fission channel, MT 18.  Includes energy distribution example (fission neutron spectrum, from MF 5 ) -->
<channel projectile="n1" target="Pu239" label="45" outputChannel="n1[multiplicity:'energyDependent',
    emissionMode:'prompt'] + n1[multiplicity:'energyDependent', emissionMode:'delayed', decayRate:'1.24811000e-02'] +
    ..."
    temperature="0. K" ENDL_CS="15,0" ENDF_MT="18" Q="198843800. eV" date="YYYYMMDD=20060901"
    genre="NBody" fissionGenre="total">
  <crossSection> ... </crossSection>
  <fissionEnergyReleased>
    <polynomial order="0" energyUnit="eV" hasUncertainties="True">
      <promptProductKE> 175550000.0 100000.0</promptProductKE>
      <promptNeutronKE> 6070000.0 100000.0</promptNeutronKE>
      ...
    </polynomial></fissionEnergyReleased>
  <product name="n1" label="n1" nativeData="NBody" multiplicity="energyDependent" emissionMode="prompt">
    <distributions nativeData="uncorrelatedAngularEnergy">
      <angular nativeData="constant">
        <constant/></angular>
      <!-- fission neutron spectrum: -->
      <energy nativeData="piecewise">
        <piecewise><variable index="0" frame="lab" name="energy_in" unit="eV" interpolation="linear,linear"/>
          <variable index="1" frame="lab" name="energy_out" unit="eV" interpolation="byRange"/>
          <variable index="2" frame="lab" name="P(energy_out|energy_in)" unit="1/eV" interpolation="byRange"/>
          <!-- list outgoing energy spectrum for each incident energy -->
          <energy value="1.e-05" index="0">
            <piecewise><variable index="0" frame="lab" name="energy_out" unit="eV" interpolation="byRegion"/>
              <variable index="1" frame="lab" name="P(energy_out|energy_in)" unit="1/eV" interpolation="byRegion"/>
              <region index="0" type="2d.xy" interpolation="linear,linear">0. 0. 10. 1.765009e-09 ...
                    3.1e+07 0.</region></piecewise></energy>
          ...
          <energy value="2.e+07" index="20">
            <piecewise>
              <variable index="0" frame="lab" name="energy_out" unit="eV" interpolation="byRegion"/>
              <variable index="1" frame="lab" name="P(energy_out|energy_in)" unit="1/eV" interpolation="byRegion"/>
              <region index="0" type="2d.xy" interpolation="linear,linear">0. 0. 10. 1.558453e-09 ...
                    3.e+07 2.68673e-15 3.1e+07 0.</region>
            </piecewise></energy></piecewise></energy>
      <uncorrelatedAngularEnergy nativeData="angular=constant : energy=piecewise">
       </uncorrelatedAngularEnergy></distributions>
    <multiplicity nativeData="pointwise">
      <pointwise><variable index="0" frame="lab" name="energy_in" unit="eV" interpolation="linear"/>
        <variable index="1" frame="lab" name="multiplicity" unit="" interpolation="linear"/>
        <xData type="2d.xy" length="2726">1e-05 2.874262 3e-05 2.87426 ...
             19500000.0 5.637402 20000000.0 5.696949</xData>
      </pointwise></multiplicity></product>
  <product name="n1" label="n1__a" nativeData="NBody" multiplicity="energyDependent"
             emissionMode="delayed" decayRate="1.24811000e-02">
    ...</product>
  <product name="">  <!-- Continue until each reaction product has been listed -->  </product></channel>


<!-- example of a 'referredData' section, for data computed by weighting a 'composite' channel (MF=9) -->
<channel projectile="n1" target="Pu239" label="50" outputChannel="...">
  <crossSection>
    <weightedPiecewise referredDataKey="0">
      <piecewise><variable index="0" frame="lab" name="energy_in" unit="eV" interpolation="byRegion"/>
        <variable index="1" frame="lab" name="crossSection" unit="" interpolation="byRegion"/>
        <region index="0" type="2d.xy" interpolation="linear,linear">1.295e+07 1. 2.e+07 1.</region>
      </piecewise></weightedPiecewise></crossSection>
  <product name="">...</product></channel>


<!-- here's the section pointed to by 'referredDataKey': -->
<referredData>
  <referredDatum key="0">
    <crossSection>
      <piecewise>
        <variable index="0" frame="lab" name="energy_in" unit="eV" interpolation="byRegion"/>
        <variable index="1" frame="lab" name="crossSection" unit="barn" interpolation="byRegion"/>
        <region index="0" type="2d.xy" interpolation="linear,linear">1.295e+07 0. 1.3e+07 0.0016
             ... 2.e+07 0.194</region>
      </piecewise></crossSection></referredDatum>
</referredData>
</heatedTarget>
```

## 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

## Prior rewriting of processing codes


Processing
Converter
Accessing
Accessing
Interim processing in python
Evaluation
FUDGE
Modified evaluation XENDL
Sn
MC
deterministic xml
Monte Carlo xml
LANL NDI
LLNL Cray
LLNL PDB
Ace
LANL Sn Codes
LLNL Sn Codes
LLNL MC Codes
MCNP
GEANT
Processing is done "outside of fudge"