Lawrence Livermore National Laboratory

Lessons from first LLNL attempt at defining a new nuclear data structure

Caleb Mattoon

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

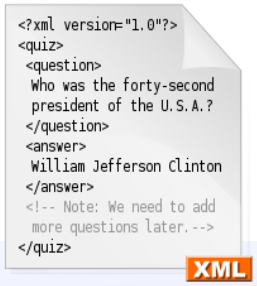
LLNL already began developing a new nuclear data structure for our own use, and encountered some surprises and lessons along the way.

- Goals for Generalized Nuclear Data (GND):
  - Replace legacy databases with a new **structure** for evaluated nuclear reaction data, that mirrors reaction physics
  - Support ENDF data types + translating ENDF ⟺ GND
  - Design flexible data containers for use in GND and beyond
  - More extensible, less redundant than legacy databases

- What have we achieved so far?
  - GND-v1.1 freely available, v1.2 coming soon! Infrastructure for translating, plotting, modifying and processing nuclear data
    - https://ndclx4.bnl.gov/gf/project/gnd
  - Several advantages already apparent
  - Some design questions still need to be resolved

Lesson #1: define a *structure* that can be represented in multiple formats.

- XML is convenient for viewing and transmitting data, but GND should be store-able in any ‘meta-language’ that allows nested elements and data
- Translating XML to HDF5 (hierarchical, binary format) was accomplished with only one day’s work, about 150 lines of code
- May be represented in other meta-languages (JSON, Root TTree, ...) depending on needs of users



Lesson #2: ENDF has served for 50+ years for good reason. Take advantage of its strengths!

- Flexible, general-purpose data containers
- Parameterized data forms (for resonance region, outgoing spectra, etc)
- Equivalent ascii and binary representations
- Quality Assurance tools that check for and repair both format and physics content problems
- Well-documented and actively maintained

Other advantages of GND are already apparent. Some were expected, but some came as surprises:

- Data become easily searchable
- GND/XML is great for web integration!
- Robust XML toolkit includes schema, stylesheets for transforming into html, checking and validating tools, etc
- ENDF data problems revealed during ENDF ⟺ GND translation
- Database size roughly comparable to ENDF

Descriptive labels make GND much easier to search than ENDF

- ENDF searchable by MF/MT, but more detailed search is difficult
- Example: search for one product in W184 MF6/MT5 (22656 lines long):

```
...
1.311280+8 5.46877-14 1.326890+8 1.77179-14 1.342500+8 5.65655-157437 6   5 1143
1.358110+8 2.37056-15 1.373720+8 6.41685-16 1.389340+8 5.78888-187437 6   5 1144
1.404950+8 3.02755-20 1.404950+8 0.000000+0                    7437 6   5 1145
2.004000+3 3.967130+0          0          1          1          457437 6   5 1146
        45          2                                          7437 6   5 1147
1.000000-5 0.000000+0 6.000000+6 0.000000+0 6.500000+6 9.999770-17437 6   5 1148
7.000000+6 9.999920-1 7.500000+6 9.999360-1 8.000000+6 9.997420-17437 6   5 1149
8.200000+6 1.000110+0 8.400000+6 1.000010+0 8.600000+6 9.999530-17437 6   5 1150
```

Descriptive labels make GND much easier to search than ENDF

- XML <elements> permit more detailed searches:

```
...
<reaction outputChannel="sumOfRemainingOutputChannels" ENDF_MT="5">
  <crossSection>...</crossSection>
  <outputChannel Q="8.508e6 eV">
    <product name="n" label="n" multiplicity="energyDependent">...</product>
    <product name="H1" label="H1" multiplicity="energyDependent">...</product>
    <product name="He4" label="He4" multiplicity="energyDependent">
      <distributions nativeData="uncorrelated"> ... </distributions></product>
    <product ...>
```

- Easy to search, both with familiar and more advanced tools:

```
>grep -l KalbachMann *.xml

>>>xdoc.xpath( '/reactionSuite/*[not(@outputChannel=
  "sumOfRemainingOutputChannels")]//KalbachMann' )
```

XML merges seamlessly with web applications, and new tools allow easy visualization of GND data

- Tools for XML include XSLT and AJAX for use on the web
  - XSLT ‘stylesheets’ used to transform any xml document, may be used to write html
  - Javascript may be embedded in XSLT template: result only requires GND file and the template
  - Example:

Another advantage: ENDF to GND conversion has uncovered data problems in ENDF-6 libraries

- During preparation for ENDF-VII.1 release, neutron and gamma libraries were translated.
- Errors discovered in the translator **and** in the ENDF data: 118 out of 411 files in ENDF-VII.1 β-0 neutrons had errors, all of which were fixed before official VII.1 release.
- Expertise for maintaining F77 ENDF checking codes is now scarce, but GND + modern languages make replacing these checking codes simple
- Some common problems:
  - energies not in ascending order
  - non-integer values for multiplicity in (n,2n)
  - internal inconsistencies: mass, level energy, isomer index, etc

GND has many strengths, but also reveals some issues that need more careful treatment and design. Example: links and references

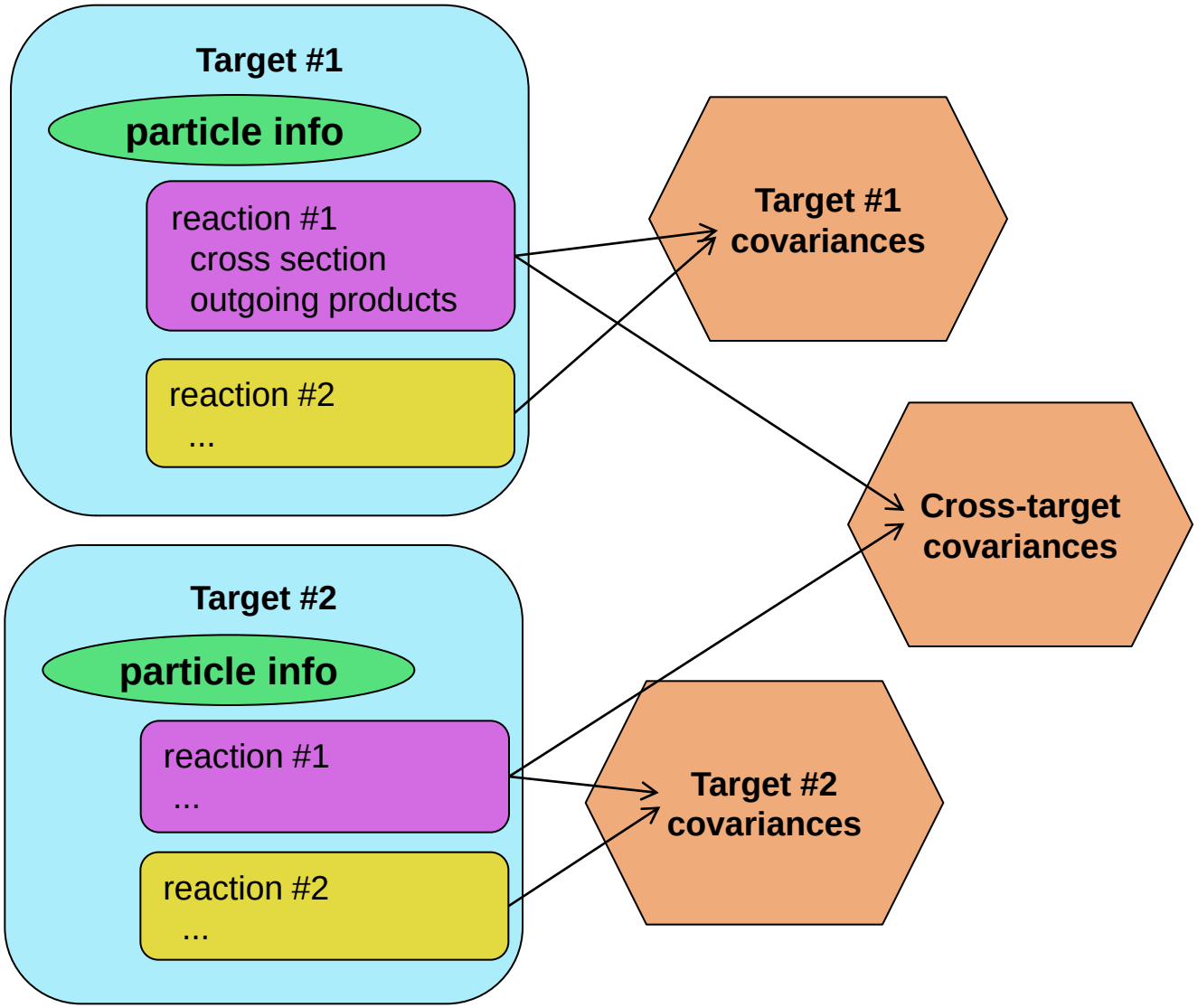
- Use links and references to avoid storing redundant (and possibly discrepant data), and to connect different databases.

- ENDF uses some references (MF8, for example), but GND can use many more. Use XML xPath as standard linking syntax:

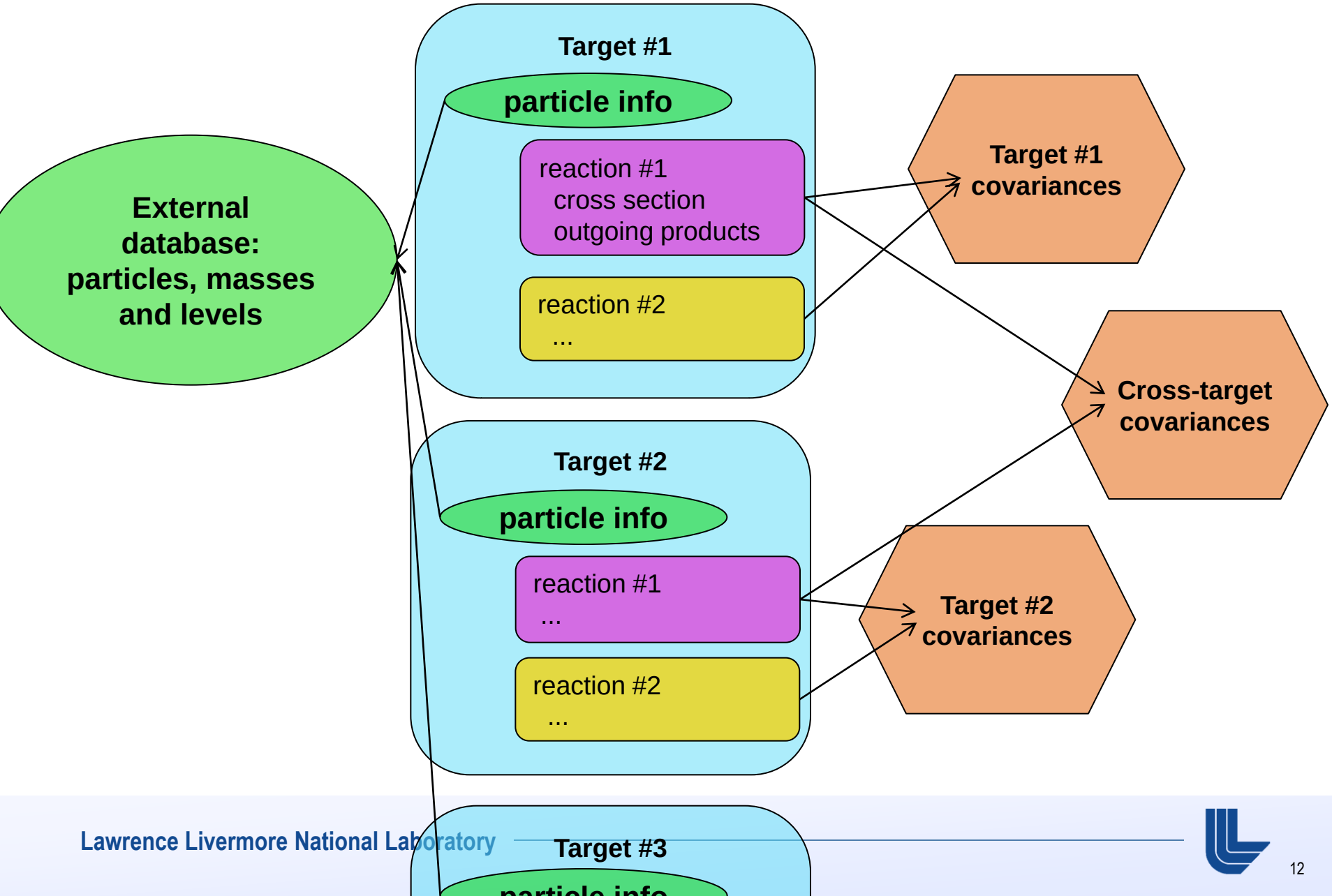
```
<link xlink:type="simple" xlink:href="http://url/file.xml#/path/within/file"/>
```

- Too much linking could become unwieldy if not properly designed, however! Requires forethought, especially when linking to external files

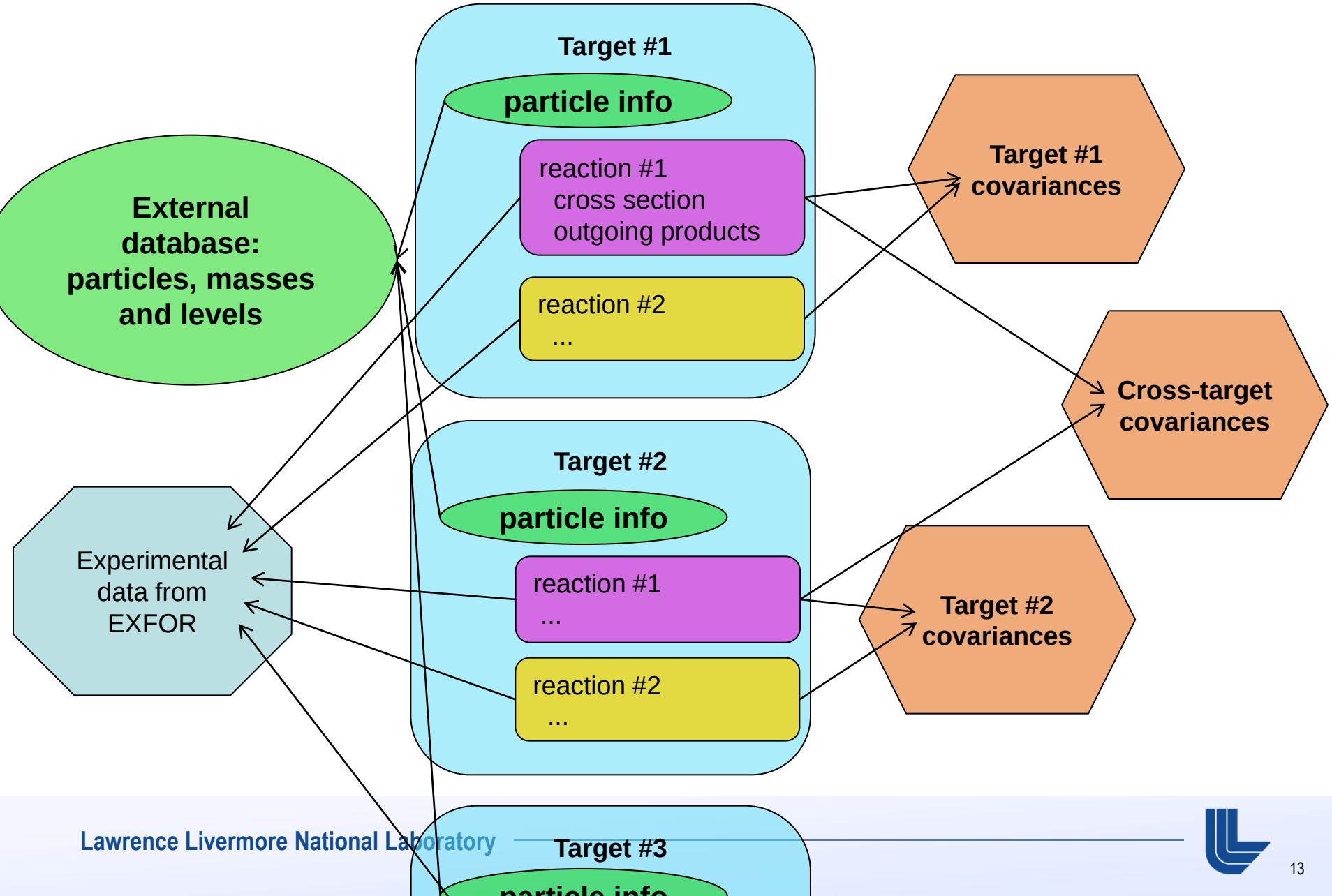
So far GND makes limited use of links. More options are possible, but create more complexity



So far GND makes limited use of links. More options are possible, but create more complexity



So far GND makes limited use of links. More options are possible, but create more complexity



GND demonstrates the feasibility of modernizing nuclear data storage.

- GND is freely available. Already starting to be used both internally at LLNL and externally (GEANT)
- Use of modern tools boosts productivity: plotting, checking, web display all made much simpler using computer tech of today
- Ability to translate between ENDF and GND means that the two can co-exist for some time (until new options introduced that have no ENDF equivalent)

Goals established during design of GND, that might be adopted by SG-38:

- The new format should
  - Store data in a structured hierarchy that mirrors the underlying nuclear reaction physics
  - Support parameterized data types, including all ENDF-6 options
  - Support any reaction with two incident particles, and any number of outgoing channels
  - Store data only once, use a key or link to refer to the data from elsewhere
  - Design flexible, general-purpose data containers

Status of automatic translation:

ENDF-VII.1*

neutrons	gammas	charged particles
423 / 423	163 / 163	52 / 56

JEFF-3.1.1

neutrons	gammas	charged particles
?/ ?	? / ?	? / ?

JENDL-4

neutrons	charged particles
377 / 406 394 / 406 (forgiving)	? / ?

EAF

default translation:
630 / 816

with 'skipBadData' option:
816 / 816

TENDL-2011

neutrons:
? / ?

maybe some others?

*Many problems in ENDF-VII were patched prior to VII.1 release. Same can be done for other libs!

Sharing processed data

- GND could enable better sharing of data
- Common processed format, can be translated by each group into their own processed formats?
- Sample GND transfer matrix:

```
<grouped xData="matrix" size="87,87">
        <axes>
          <axis index="0" label="energy_in" unit="eV" interpolation="linear,flat" frame="lab"/>
          <axis index="1" label="energy_out" unit="eV" interpolation="linear,flat" frame="lab"/>
          <axis index="2" label="C_l(energy_in,energy_out)" unit="b" frame="lab"/></axes>
        <l value="0">
          <matrix rows="87" columns="87" form="sparse_asymmetric">
            0 0 1 8.04765353
            1 0 2 0.0128888641 7.59496411
            2 1 2 0.0385045077 8.95273144
            3 2 2 0.0592808172 11.3596173
            4 3 2 0.109015267 9.60703412
            5 4 2 0.0470416286 8.87828991
            ...
```