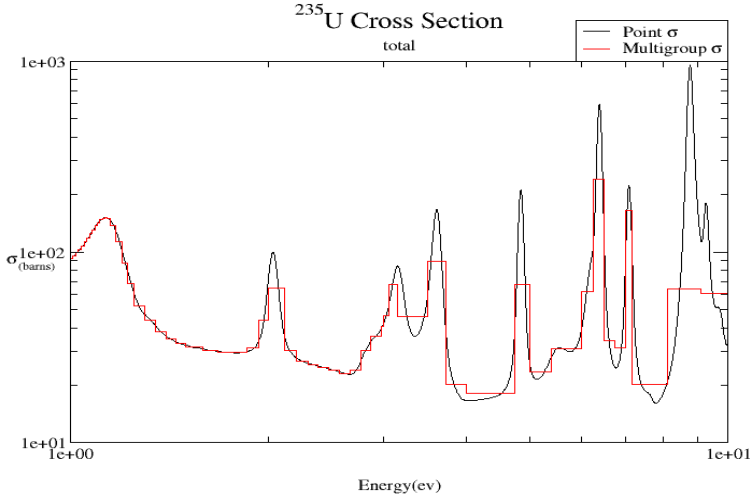
ENDF File uses in AMPX

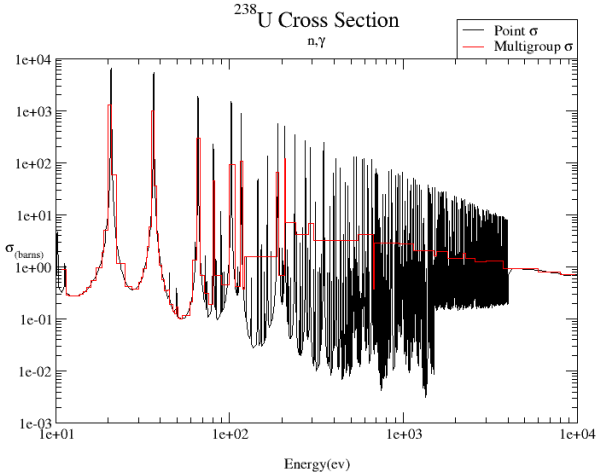
D. Wiarda
M. E. Dunn
L. C. Leal
Oak Ridge National Laboratory
Reactor & Nuclear Systems Division
Nuclear Data and Criticality Safety Group

AMPX Cross-Section Processing System



- SCALE relies on AMPX for data libraries
  - **MG and CE cross-section data**
  - **Data processing procedures including problem-dependent resonance self-shielding**
  - **Cross-section uncertainty data to support S/U methods in SCALE**

- AMPX Processes ENDF/B Formats
  - **Generate Temperature-Dependent Pointwise Cross Sections**
  - **Provide Resonance Self-Shielding for RRR and URR**
  - **Probability Table Generation for URR**
  - **Energy and Angle Distributions for Secondary Particles**
  - **Process S(α,β) Data for Thermal Moderators**
  - **Generate free-gas S(α,β) Data for Non-Thermal Moderators**
  - **Process Particle-Yield Data**
  - **Generate Pointwise Weighting Spectra**
  - **Multigroup Averaging Operations**
  - **Process Cross-Section Uncertainty Data for TSUNAMI**
  - **Automated Library Production—Process Multiple Nuclides**



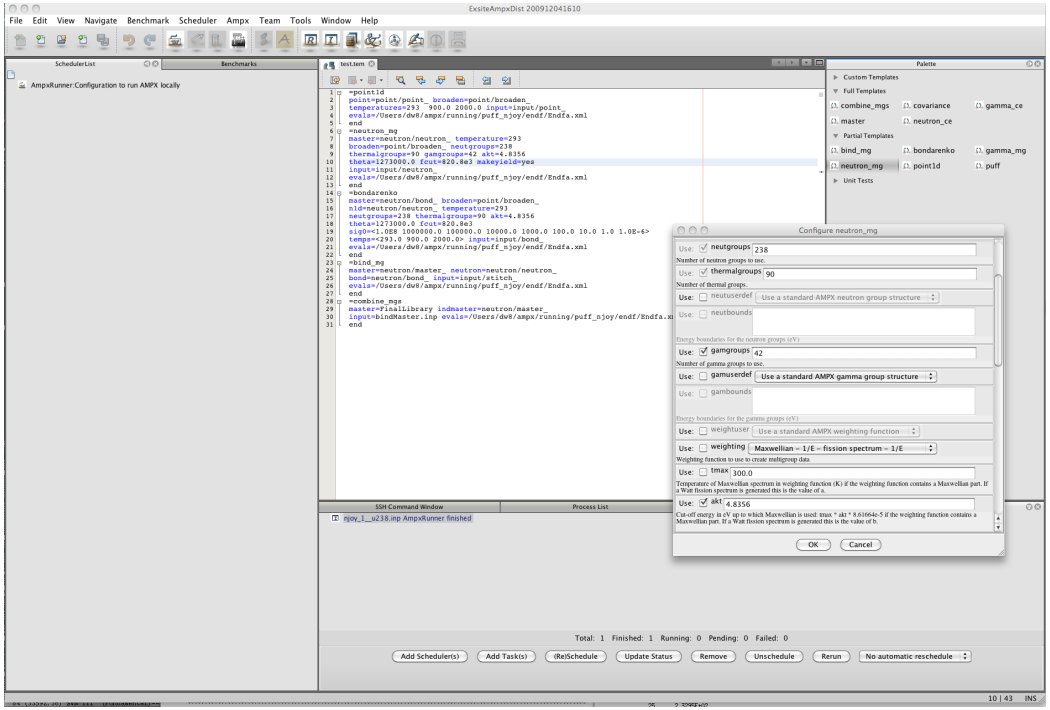
File formats used in AMPX

- Multi-group Master and Working library
- ENDF style TAB1 formatted files binary and ascii (binary can be single or double – AMPX automatically detects)
- Continuous-energy library format.
  Used in SCALE for self-shielding as well as Monte-Carlo codes.
- COVERX covariance library format, ASCII and binary

File conversion and comparison options

- AMPX has modules to compare
  - Master and working libraries
  - TAB1 formatted files
  - COVERX files
- AMPX has modules to convert
  - TAB1 files to and from NJOY PENDF (slight differences in unionization of energy points and how temperatures are handled)
  - NJOY GENDF to AMPX master
  - COVERX files to NJOY ERRORR format

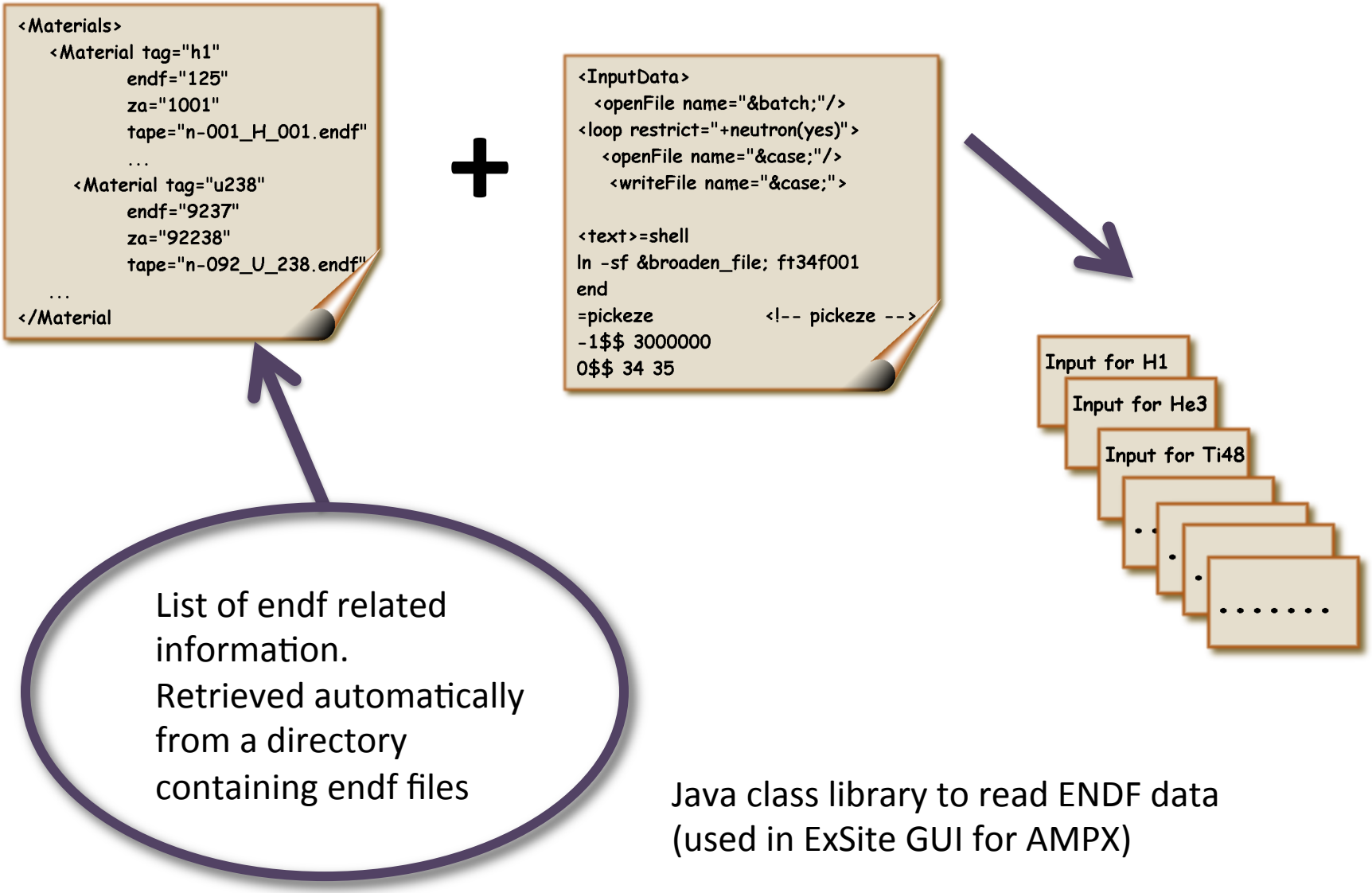
ExSite* is the Graphical User Interface for AMPX



- Create/Edit AMPX input files
- Automatically create AMPX input sequences for library creation.
- Input sequence generation uses abbreviated ENDF information and templates
- Abbreviated ENDF information is generated in ExSite from full ENDF file

*Extensible SCALE Intelligent Text Editor

Generate input files for processing



Java class library to read ENDF data
(used in ExSite GUI for AMPX)

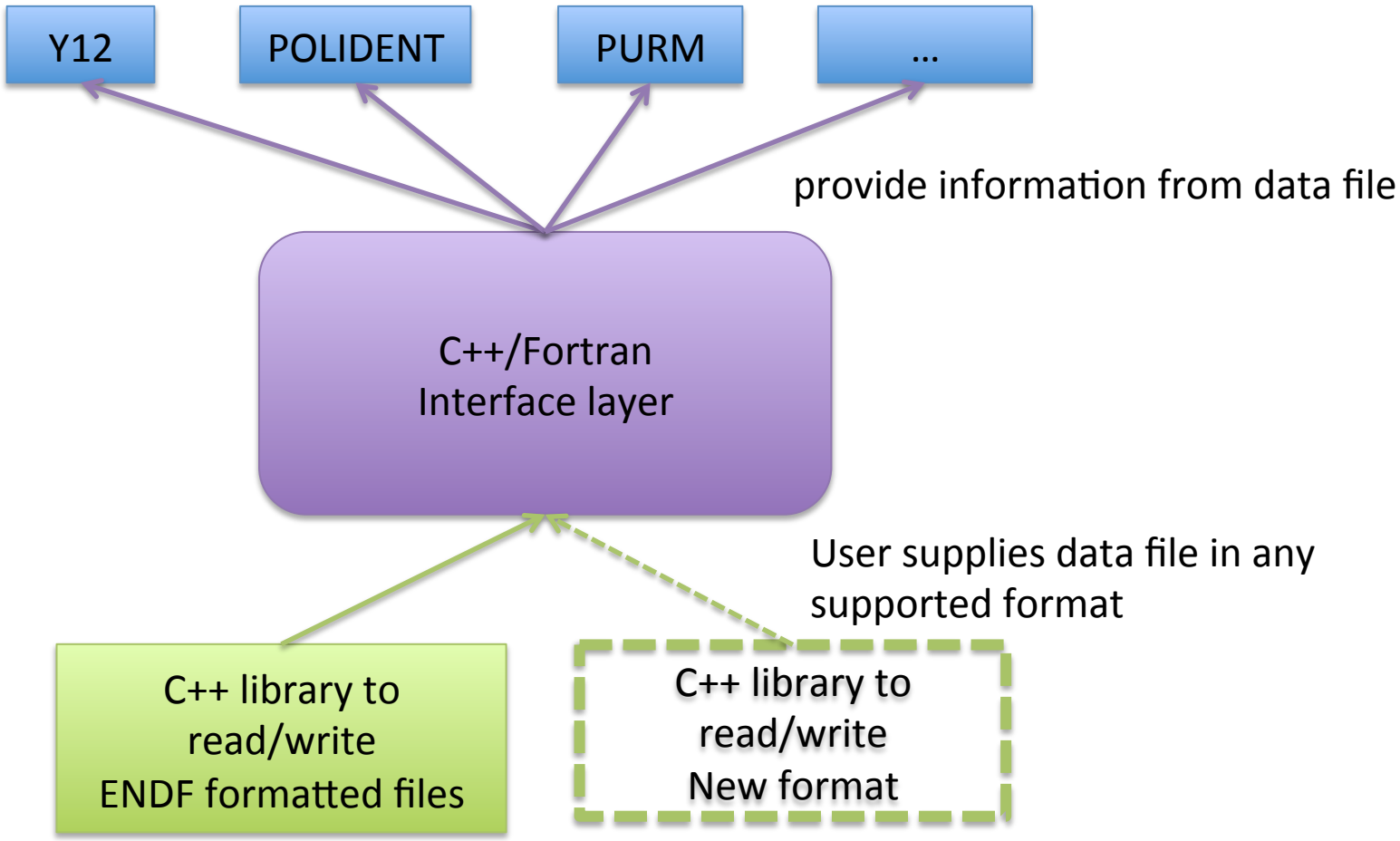
Currently in AMPX

- AMPX has a library of Fortran90 modules that read/write ENDF data
- All modules use the library instead of direct access to ENDF files

Planned

- Develop a C++ library to read/write ENDF files
- If a new file format becomes available develop a C++ library to access those data (or use existing functions)
- AMPX develop an intermediate layer that transforms the data into AMPX specific structures for further processing
- No AMPX module will access the low level ENDF read/write functionality, thus allowing the use of different file formats.

Planned structure for AMPX module access to nuclear data files



This will allow AMPX to use several different formats for the nuclear data files.

Observations

- Clear definition of on-disk format. Thus, XML is nice, as it is mostly ASCII (see note below on binary) and very structured and would allow users to easily peruse the data event without processing. Data management systems like HDF5 or SILO pose a problem as the underlying on disk file format can change. Who would be responsible for converting to new binary format and ensure all processing code follow suite and are linked with the newest HD5 library.
- While library functions and interfaces (API) in a language like Python or C++ are nice to have as a reference, they should not define the file format. Would we all still be using ENDF formats if we had to use FORTRAN-IV routines to read it ?
- Allow high precision for data that need it (covariance matrices, energy points), but allow lower precision for other data (cross section data).
- Binary format, if used, should be very clearly defined, so that a program can reconstruct from bits if need be. For example: IEEE double floating point number in network order.
- Allow for easy extension, i.e. if a new feature is added, processing codes that do not know about it can ignore it. For example an extra element or attribute in XML can define new data, but can be skipped by processing codes.