OAK RIDGE National Laboratory

AMPX and SAMMY Status Report

D. Wiarda

A. Holcomb

G. Arbanas

J. Brown

Oak Ridge National Laboratory

WPEC, Nov. 2020

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

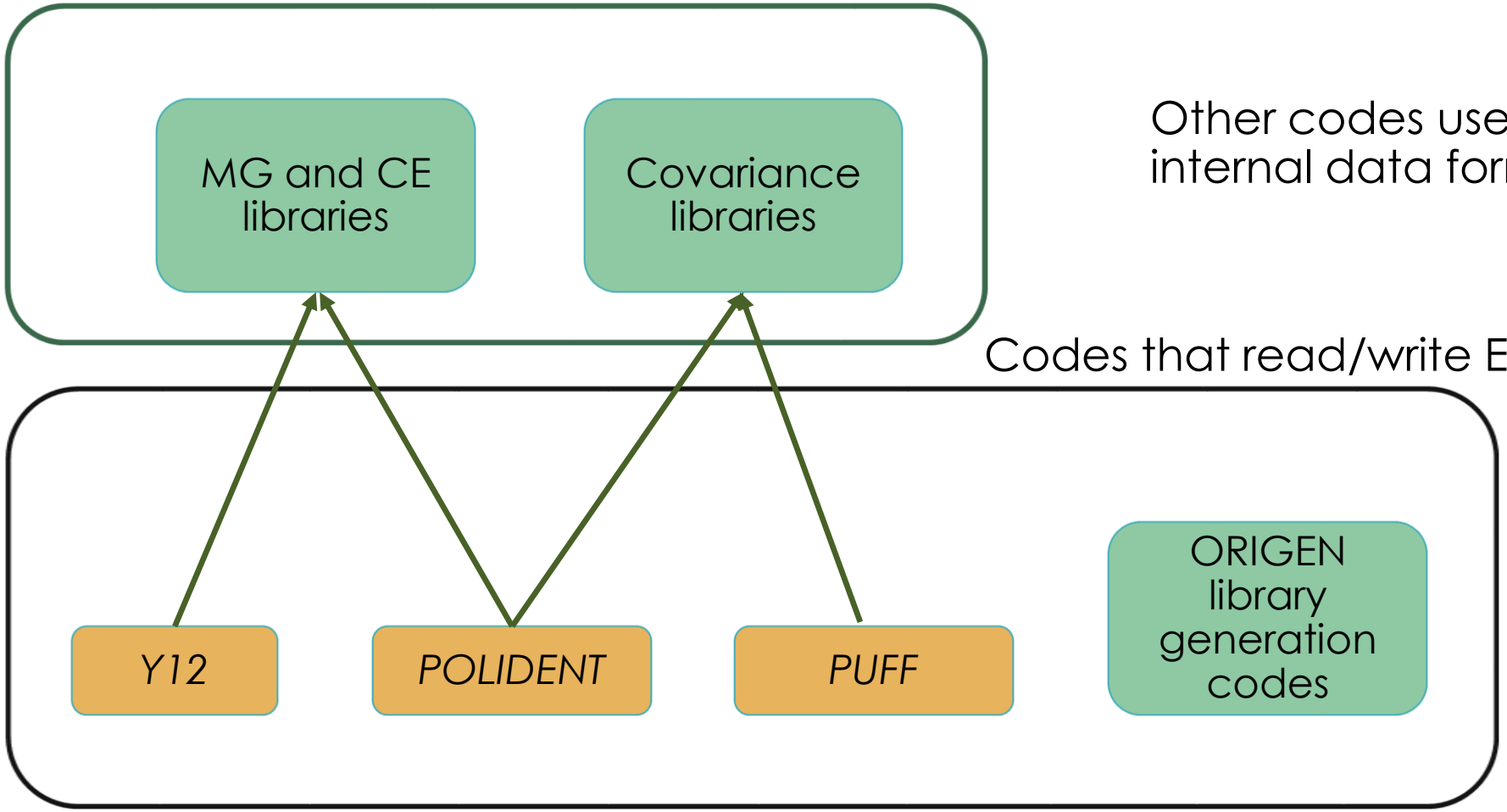
U.S. DEPARTMENT OF ENERGY

Summary

- AMPX and ENDF
- GNDS API level classes
- SAMMY
- SAMMY and GNDS
- Outlook

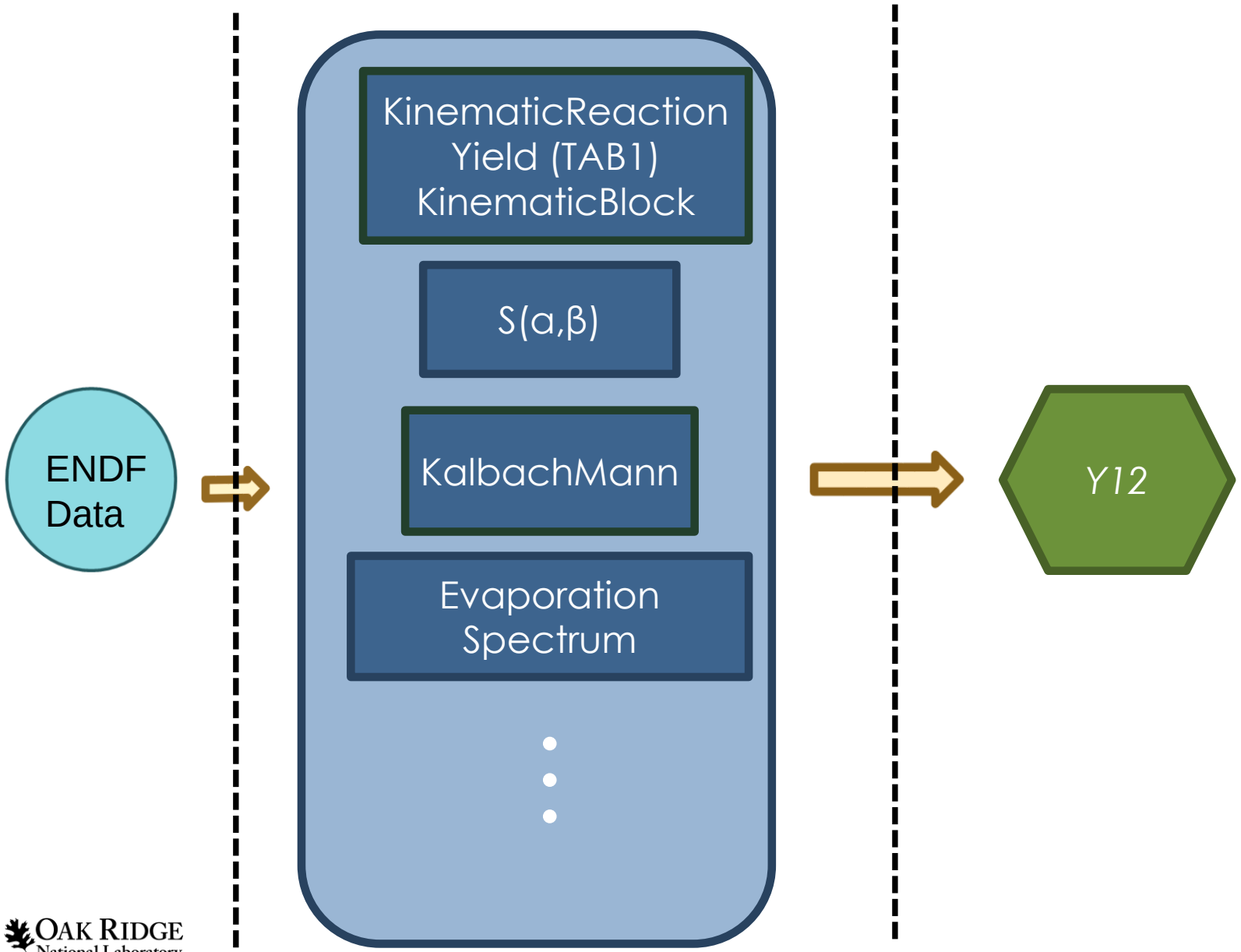
ENDF data in AMPX

Codes producing final libraries
using SCALE and AMPX in-memory formats



Other codes use AMPX/SCALE internal data formats

ENDF Processing



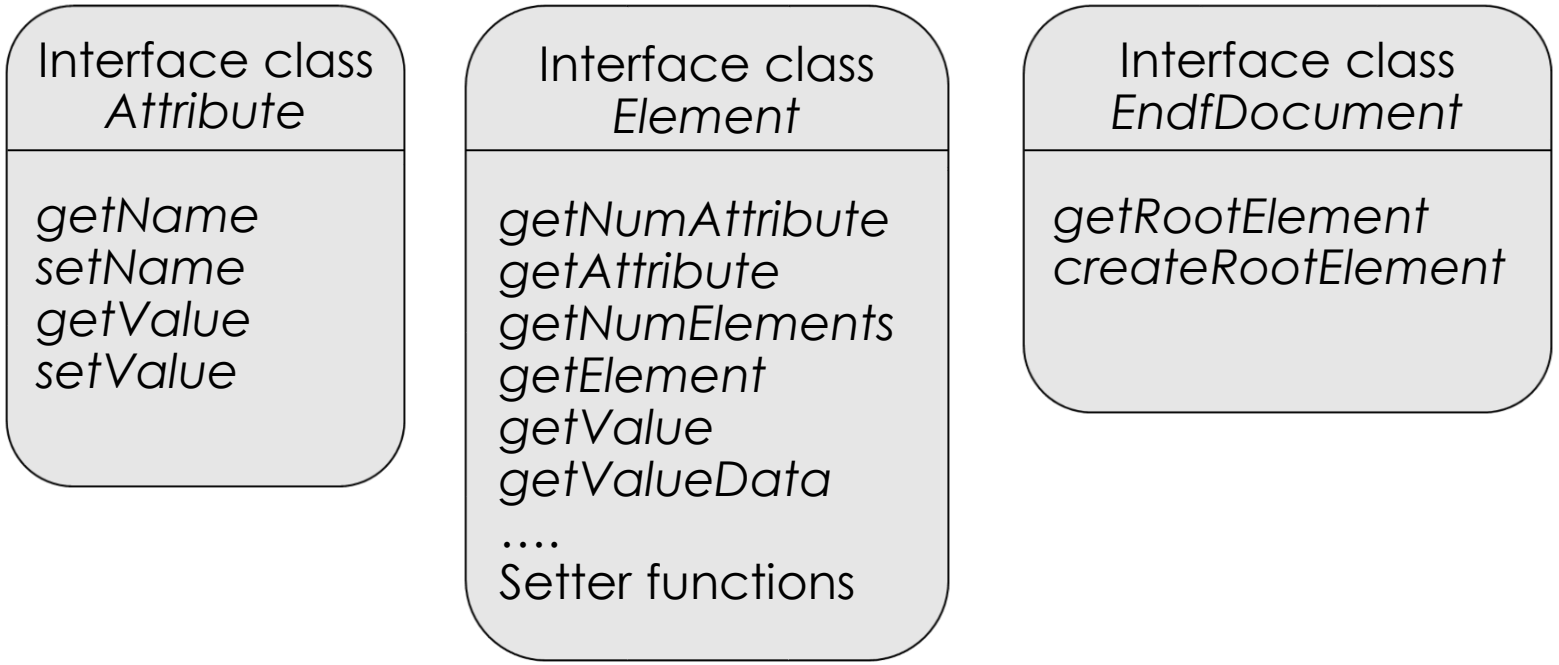
- A C++ library reads/writes ENDF data.
- Data are collected into an AMPX/SCALE internal in-memory structure without processing.
- Processing codes use the above in-memory structure.
- Therefore only the ENDF reading/writing part of the code needs to be changed if the ENDF format changes.
- We plan on using the ENDF reading/writing part in other codes, like SAMMY

How to support GNDS

- Internal AMPX C++ structures are the "API" that AMPX and SAMMY will access.
- Add a Reader/Writer that supports GNDS and fills AMPX internal C++ structures.
- Test by processing ENDF formatted and GNDS formatted files and compare results.
- Find an efficient way to support GNDS and also allow us to easily apply updates.
- Updates might not immediately be propagated to the internal AMPX C++ classes if not needed in AMPX.

GNDS access layers in AMPX: Part I

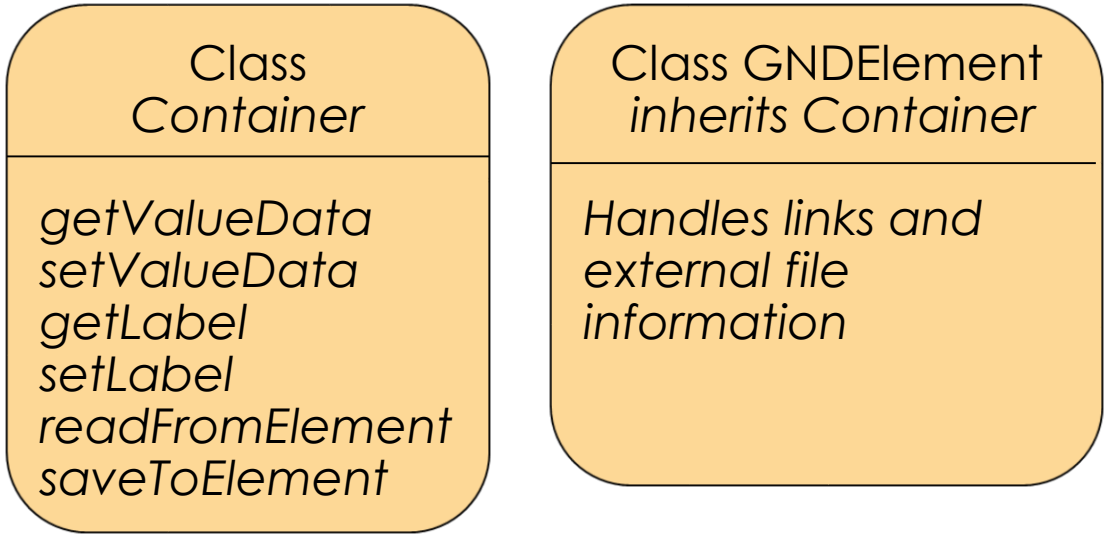
The direct access to the XML (or HDF5 or JSON) GNDS file is abstracted into access to elements and attributes:



Currently using QT XML but will change to pugixml as SCALE has started to use it.

GNDS access layers in AMPX: Part II

Two common base classes:



Classes for: Value, Array and Table objects in GNDS, inheriting from Container

GNDS access layers in AMPX: Part III

- Python generated C++ classes for all objects described in the GNDS specification. Generation is based on the JSON files of the GNDS standard. Approx. 290 classes are generated. All inherit from GNDElement.
- Special names are selected for GNDS objects such as Double, which have names not allowed in C++.
- Namespaces are handled as in the GNDS specification, thus the same name but in different namespaces is allowed.
- Some correction for errors in the specification are built into the Python generation code. If corrections are applied, they are reported.

These classes are a very low-level access API to the GNDS content, that mirror the specification directly.

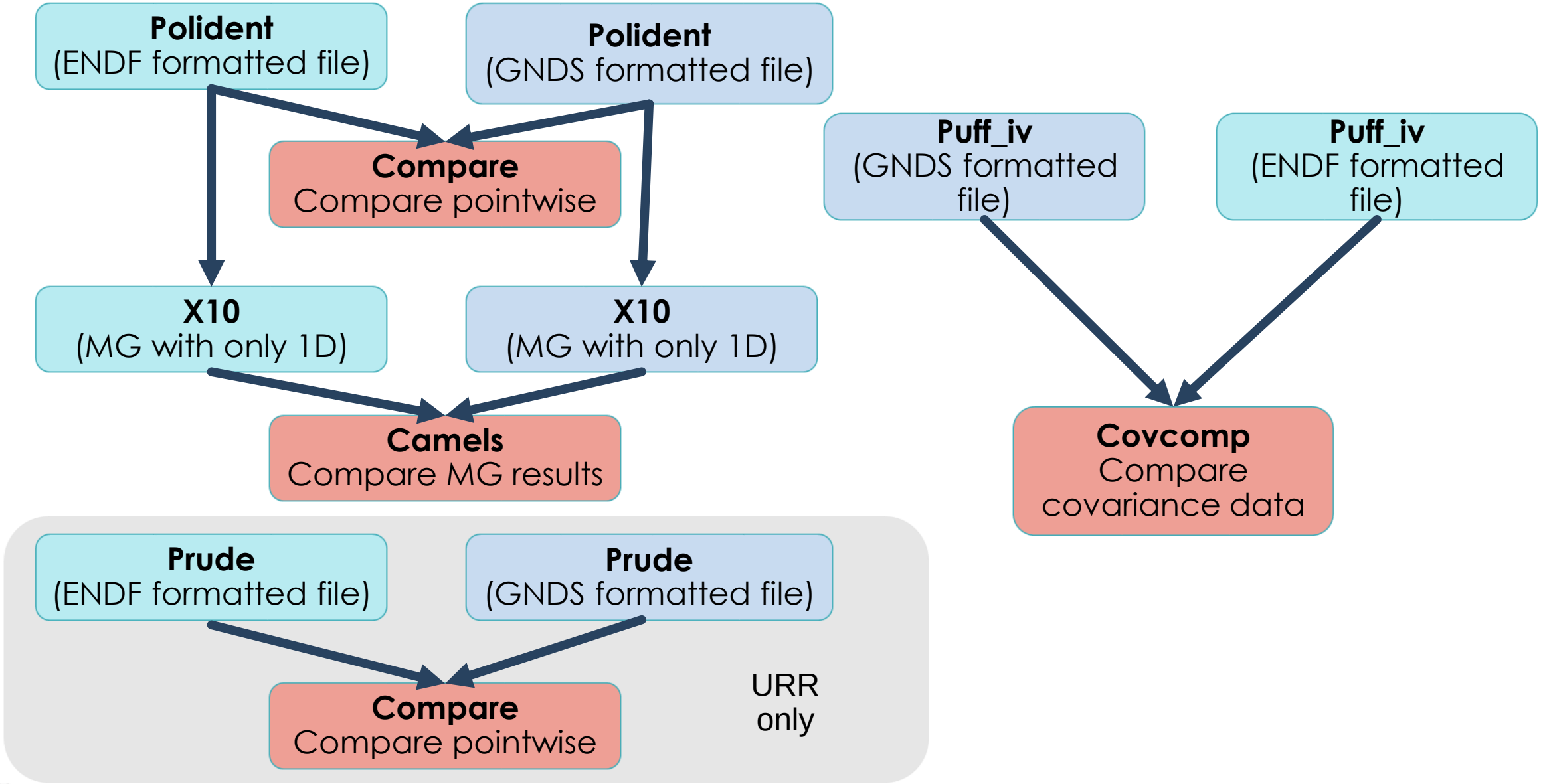
GNDS access layers in AMPX: Part IV

Classes that fill the AMPX C++ in-memory structures. These classes are needed as:

- To select the correct "style" of the data the user requested, which includes following the inheritance chain.
- Convert GNDS units to AMPX units
- Convert GNDS constructs into AMPX constructs.
- More user-friendly access methods to Particle data base

This layer is currently only reading data, but writing will be added as needed. The first implementation for reading will be for resonance parameters and corresponding covariance matrices for use in SAMMY.

Test the GNDS reading

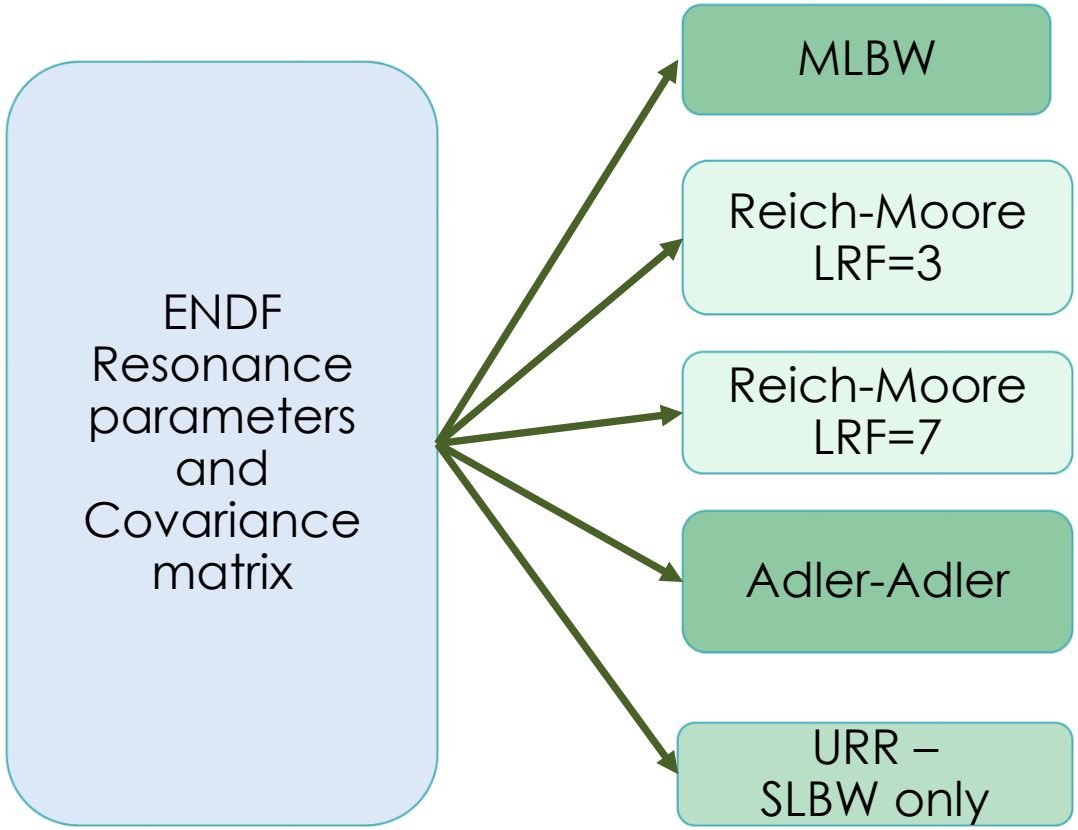


SAMMY

- Code to analyze experimental data in the Resolved Resonance Range (some capabilities in the unresolved range)
- Still mostly F77 (fixed length) code
- Using one big container array to allocate/deallocate data – restricts size of problem and error detection.
- SAMMY is used in production environments, so during the modernization its functionality cannot be lost. That means we can not simply rewrite it from scratch.
- Modernization is needed to allow us to add new features (for example new and improved algorithms in the resolved range).
- SAMMY has a large suite of regression tests that can be run and whose outputs can be checked automatically. During the modernization process, we will not change the results of these regression tests, except if errors in the original coding are found and corrected.

SAMMY and AMPX connection

C++ in-memory storage, read and write (LRF=7, parameters only, ENDF only) is available in AMPX.



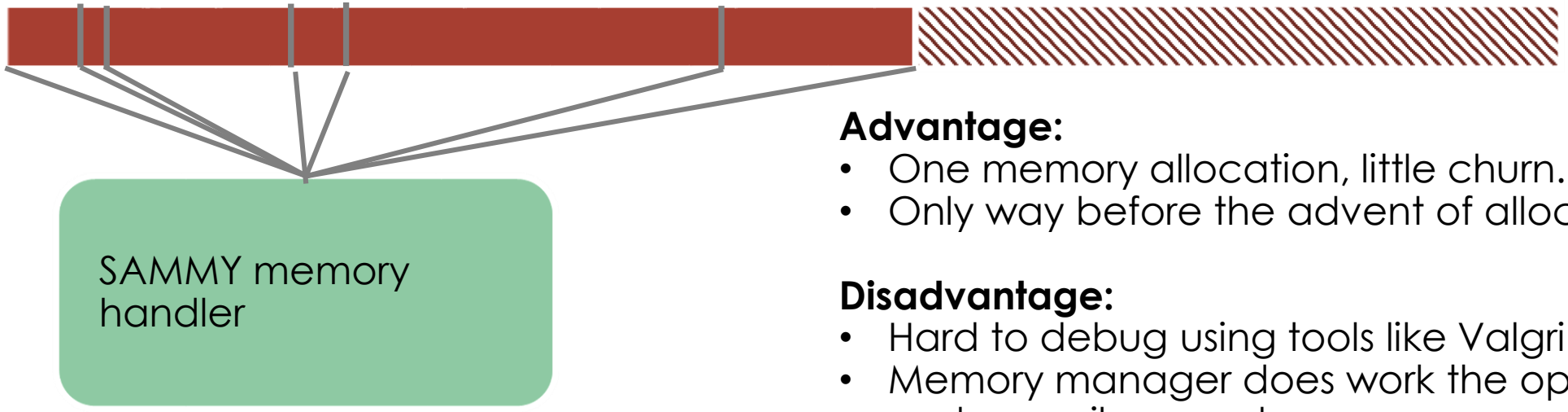
- SLBW and MLBW parameter are stored in the same class with a flag indicating which formalism to use
- Resonance parameters for Reich-Moore for LRF=3 are initially stored in a different class, but are converted to a LRF=7 class before calculation
- If derivatives are desired, all formalisms (except URR) are converted to LRF=7
- All resonance parameter classes can contain a covariance matrix. If converting to a different formalism, the covariance matrix is re-organized accordingly

Resonance parameters

- SAMMY and AMPX now use the same in-memory C++ storage for the resonance parameters and the covariance matrix.
- SAMMY still uses its own reading/writing of parameters in ENDF format. We intend to switch to the AMPX reading and writing routines. If we do, this will enable access to GNDS formatted files.
  - We have delayed to implement this as we concentrated on making the SAMMY code more modular.
- We also intend to share the same code to calculate cross section data and derivatives

This connection is the reasons that a SCALE license is currently needed to compile SAMMY.

Container array



Advantage:
- One memory allocation, little churn.
- Only way before the advent of allocate

Disadvantage:
- Hard to debug using tools like Valgrind.
- Memory manager does work the operating system and compiler can do
- Array is fixed and needs to be dimensioned for largest desired problems.

We have eliminated all use of the Container array in SAMMY and can now run modern error detection tools.
Added advantage: The code is a bit more modular and additional changes are easier to implement.

Energy grids in SAMMY

Energy grid C++ class

E_1	Exp. Data 1	Theory 1	$\frac{\partial \sigma}{\partial p_1}$	…
…	…	…	…	…
E_N	Exp. Data 2	Theory 2	$\frac{\partial \sigma}{\partial p_1}$	…

implemented

In progress

This is in preparation to separate fitting and model calculation in SAMMY

- SAMMY uses two energy grids:
  - Experimental data grid
  - Auxiliary grid used for resolution broadening
- Energy points for both grids now use the C++ class
- Switch to use derivatives and other data in C++ class is ongoing.
- Switch to data covariance matrix is ongoing

Future updates to SAMMY

- Introduce imaginary component to *R*-matrix channel radii to parameterize eliminated *direct* capture channels in Reich-Moore approximation
- Simultaneous fitting of multiple targets and compound systems
- New parameter fitting methods would quantify the effects of assuming linear models and normal PDFs.
- Any new GNDS formats will be implemented in SAMMY, as needed, and then proposed to the EG-GNDS group.

AMPX Release

- An older implementation of the GNDS reading is available in the current SCALE beta. The current implementation will be available in an upcoming beta.
- We have permission to distribute AMPX as open source (https://code.ornl.gov/RNSD/AMPX). But since AMPX is developed within SCALE we need to solve logistic problems to make sure only open-source code gets distributed. This has been held up be the release cycle of SCALE.
- To compile SAMMY, a source code release is necessary.

SAMMY Release

- SAMMY source code is available from https://code.ornl.gov/RNSD/SAMMY
- The code currently needs SCALE 6.3 beta 9 and up to compile, instructions are provided.
- Stable versions will be tagged and distributed as before