# The modern Nuclear energy agency Data bank: an integrated hub for code and data development and validation

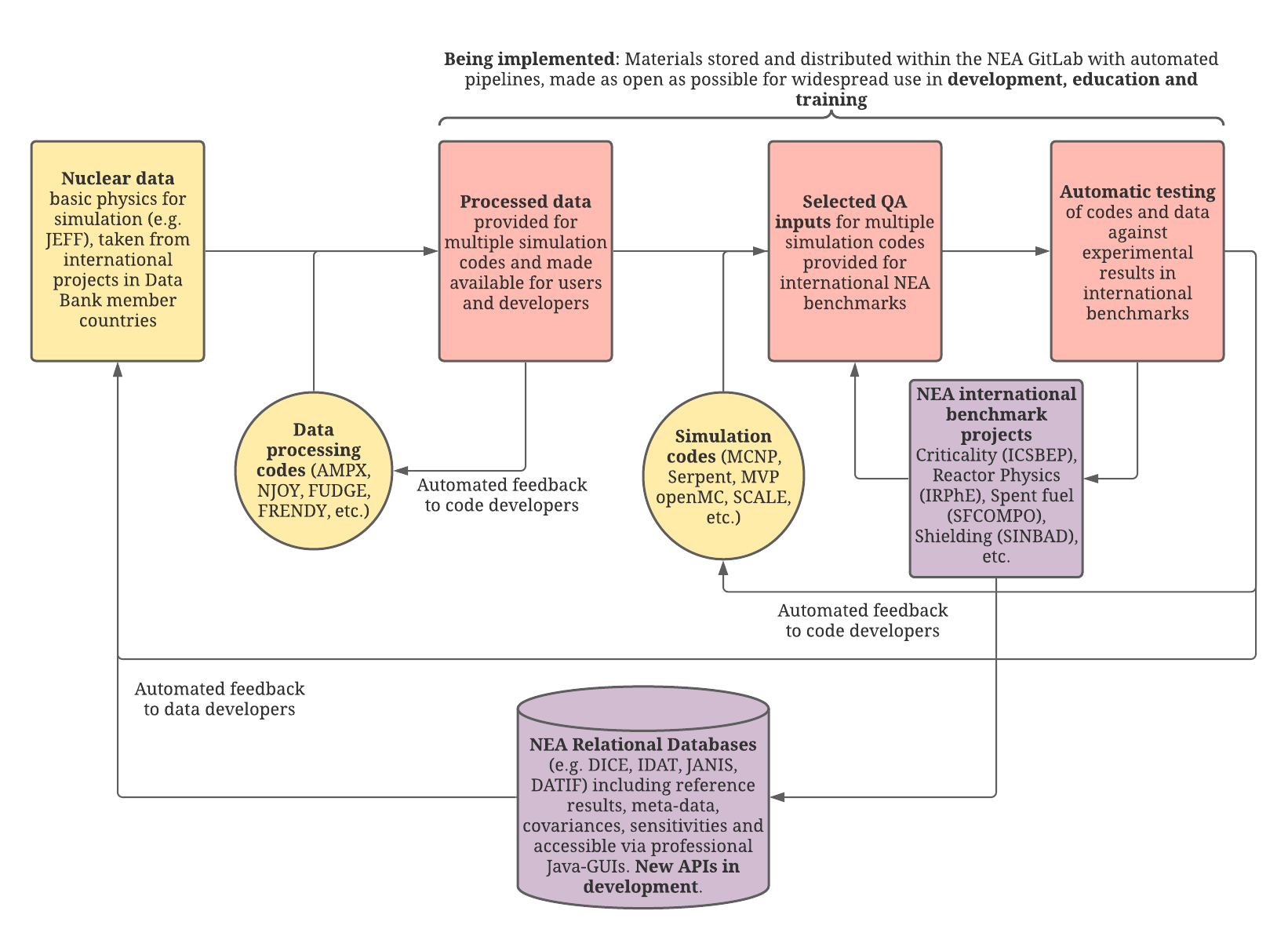
M. Fleming, A. Dufresne, I. Hill, D. Foligno, B. Adigun

OECD Nuclear Energy Agency, Boulogne-Billancourt, France

Email contact of corresponding author: michael.fleming@oecd-nea.org

The simulation of nuclear systems requires a combination of computer modelling codes, input physics data and integral experimental benchmarks to demonstrate the validity of the calculation results. The Nuclear Energy Agency (NEA) plays a central role in these activities by hosting projects such as the Joint European Fission and Fusion (JEFF) nuclear data library project, the International Criticality Safety Benchmark Evaluation Project (ICSBEP) and several other experimental benchmark projects for other application domains. The NEA Data Bank manages the outputs of these projects, as well as a catalogue of over 14000 computer codes, distributes them in compliance with any applicable restrictions, and organises training workshops.

The Data Bank is working on an integration of these products and services, including the full pipeline of input nuclear data, processing and verification codes and models for experimental benchmarks to create automated workflows that generate calculation results and insights for data and code developers. This is shown schematically in Figure 1.

*FIG. 1. Data Bank pipeline for the integration of data, codes, and benchmarks.*

These data and processes are stored within an NEA-hosted GitLab platform to provide version control, automate calculations through continuous integration and/or scheduled pipeline tasks, integrate projects, provide fora for feedback and discussion, and publish results for the public or delegates to the NEA bodies. Calculations are completely reproducible, using Docker containers to store the complete environment and dependencies required for all calculations. The first components of this pipeline – the data processing and verification - has already been developed and is being employed by the JEFF project to develop basic physics data. Input Evaluated Nuclear Data File (ENDF) data are tested using utility and processing codes including NJOY, FRENDY, FUDGE and OpenMC to generate a range of pointwise, multi-group, ACE, GNDS and HDF5 data. These data are made available as release candidate, development versions and ultimately released as official libraries alongside testing data from the automated processes.

The NEA Nuclear Science Committee’s Working Parties, which are responsible for the development of several integral experiment benchmark projects, have also generated suites of modelling code input decks for these peer-reviewed experiments. The Data Bank is collecting these to prepare reference calculation results with development nuclear data versions and codes from the Computer Program Service package catalogue and other open-source software. These already include input suites prepared as part of the Working Party on International Nuclear Data Evaluation Co-operation (WPEC) Subgroups 45 and 47 and other decks prepared as part of the JEFF project. By providing the inputs alongside version-controlled data files and simulation codes, calculation results are completely and practically reproducible, while the new NEA GitLab will provide users with an easily findable data source that is as open and accessible as possible, while respecting distribution restrictions as necessary.

The NEA relational databases for experimental benchmarks and nuclear data, which include the Database for ICSBEP (DICE) and International Reactor Physics Handbook Database and Analysis Tool (IDAT), supplement the handbooks with metadata, sensitivity profiles calculated with reference models and a variety of algorithms for established use cases. Coupled with the Java-based Nuclear Information System (JANIS) that provides a database of processed nuclear data, sensitivity-based perturbation and uncertainty calculations can be performed in a matter of CPU core-minutes on thousands of peer-reviewed experiments opening the possibility of direct data development insight and use in machine-learning algorithms. The NEA is developing application programming interfaces (APIs) for these products to integrate them into continuous integration workflows for data development and to provide users with access for a variety of advanced analyses.