ASIPP

# Implementation of data integration toolkit for ITER Physics Data Model

Xiaojuan LIU ,Zhi YU, Nong XIANG

Institute of Plasma Physics ,Chinese Academy of Sciences

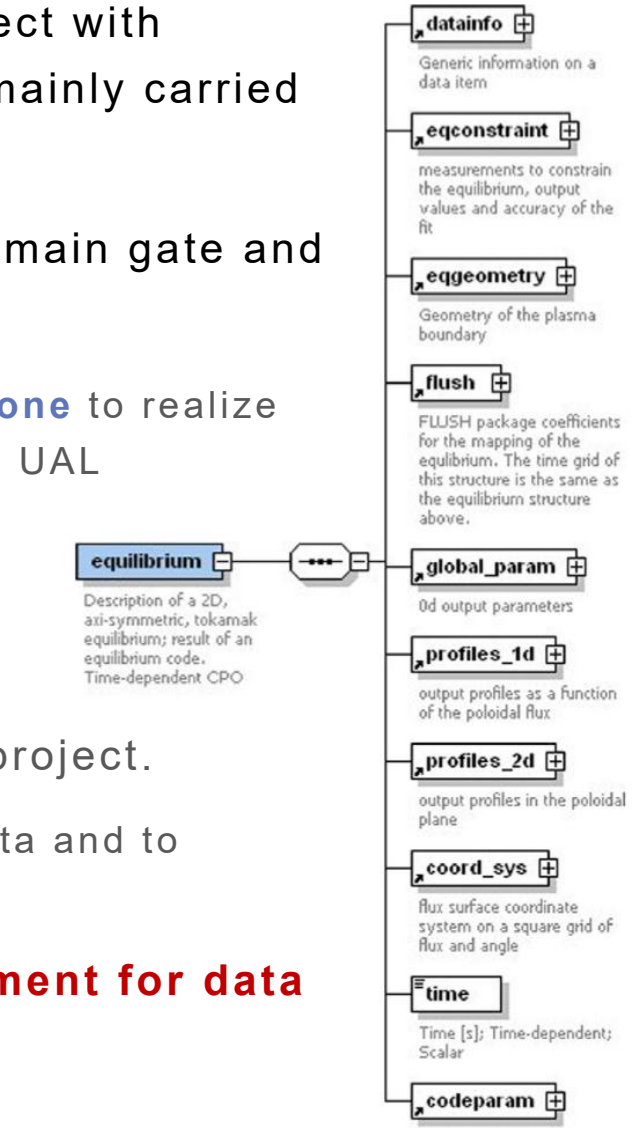
# Outline

- Background and Motivation
  - **—Why do we need a data integration toolkit in the fusion research ?**
- What is the **data integration ?**
  - **— Design concept and framework**
- Database mapping
  - **— Unified data access interface for experiment database.**
- Lightweight data description
  - **— Unified data exchange interface between physical programs.**
- Summary

ASIPP

# Motivation: Fusion research needs data integration tools.

- The magnetic confinement fusion is a complex large-scale scientific project with different time scales and space scales. **Fusion research activities** are mainly carried out around the **experimental data and simulation data.**
- ITER defined the **Physical Data Model 1 (PDM)** which aims at being the main gate and **general standard** to fusion data.
  - **IMAS**[2] being a particular case of application, choose **the PDM as the backbone** to realize the data interaction between physical modules and experimental by UDA and UAL
  - Integrate the data of different systems into the same system to realize the integrated analysis of the entire device. This process is the so-called **data integration process.**
- The European Fusion Community recently proposed the **FAIR4FUSION**[3] project.
  - One of the purposes is to improve the sharing and accessibility of fusion data and to achieve **data integration and standardization.**
- **We develop a data integration toolkit to provide a platform environment for data modeling and analysis in fusion research.**





1.Imbeaux, Frederic A generic data structure for integrated modelling of tokamak physics and subsystems 2010 Computer Physics Communications , Vol. 181, No. 6 Elsevier p. 987-998
2.Imbeaux, Frédéric Design and first applications of the ITER integrated modelling & analysis suite 2015 Nuclear Fusion , Vol. 55, No. 12 IOP Publishing p. 123006
3.https://www.fair4fusion.eu/

# Background: The de facto data standardization format and scheme

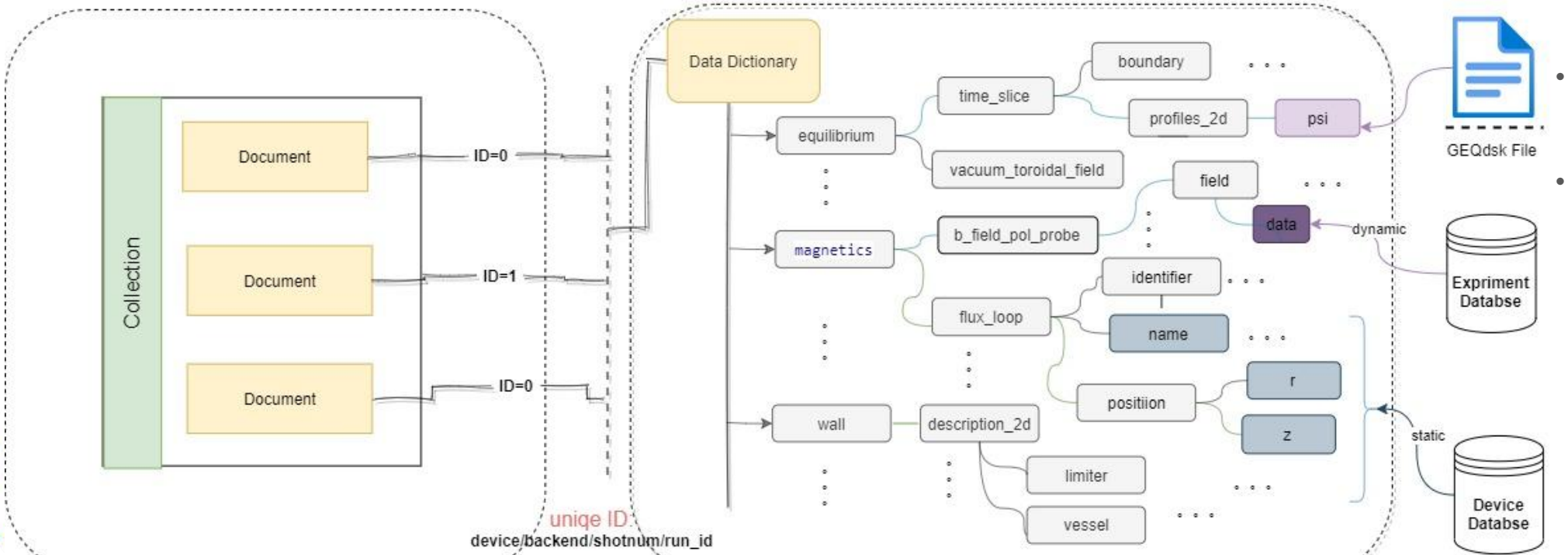
- **Proprietary, non-universal data format:**
  - SWIM[1] plasma state file (which is different from the ONETWO plasma state file)
  - Consistent Physical Object[2] is used by the ITM-TF,
  - BPSD which is a "in-house" standard data structure is used in TASK[3].
  - OMFIT[4] treats files, data and scripts as a uniform collection of objects organized into a single, self-descriptive, hierarchical structure (the OMFIT tree structure).
- **Universal, non-specific field data format:**
  - OMAS[5]: Ordered Multidimensional Array Structure
    - It provides a convenient Python API
    - it stores data with different file/database formats, mapping the physics codes I/O with OMFIT framework to the IMAS data model.
  - HDC[6]: Hierarchical Data Containers
    - HDC is a tiny library for exchanging hierarchical data (arrays of structures) in shared memory between multiple programming languages, currently supporting C, C++, Python, Fortran and MATLAB.
    - The hierarchical structure is organized as a tree.

ASIPP


1.W.R. Elwasif et al., in Parallel, Distributed and Network_x0002_Based Processing (PDP), 2010 18th Euromicro Interna_x0002_tional Conf. on, Institute of Electrical and Electronics En_x0002_gineers (IEEE), pages 419427 (2010)
2..A. Fukuyama et al., in Proc. 20th Fusion Energy Conf. Villamoura, Portugal (2004).
3.B. Guillerminet et al., Fusion Eng. Des. 83, 442 (2008).
4.Meneghini, O., Smith, S. P., Lao, L. L., Izacard, O., Ren, Q., Park, J. M., … Staebler, G. M. (2015). Integrated modeling applications for tokamak experiments with OMFIT. *Nuclear Fusion*, *55*(8), 083008.
5.Meneghini, Orso-MariaOMAS: A Python Library to Interface with the ITER Integrated Modeling and Analysis Suite (IMAS) 2020 APS Division of Plasma Physics Meeting Abstracts , Vol. 2020
6.JM10-0092.https://bitbucket.org/compass-tokamak/hdc/src/master/

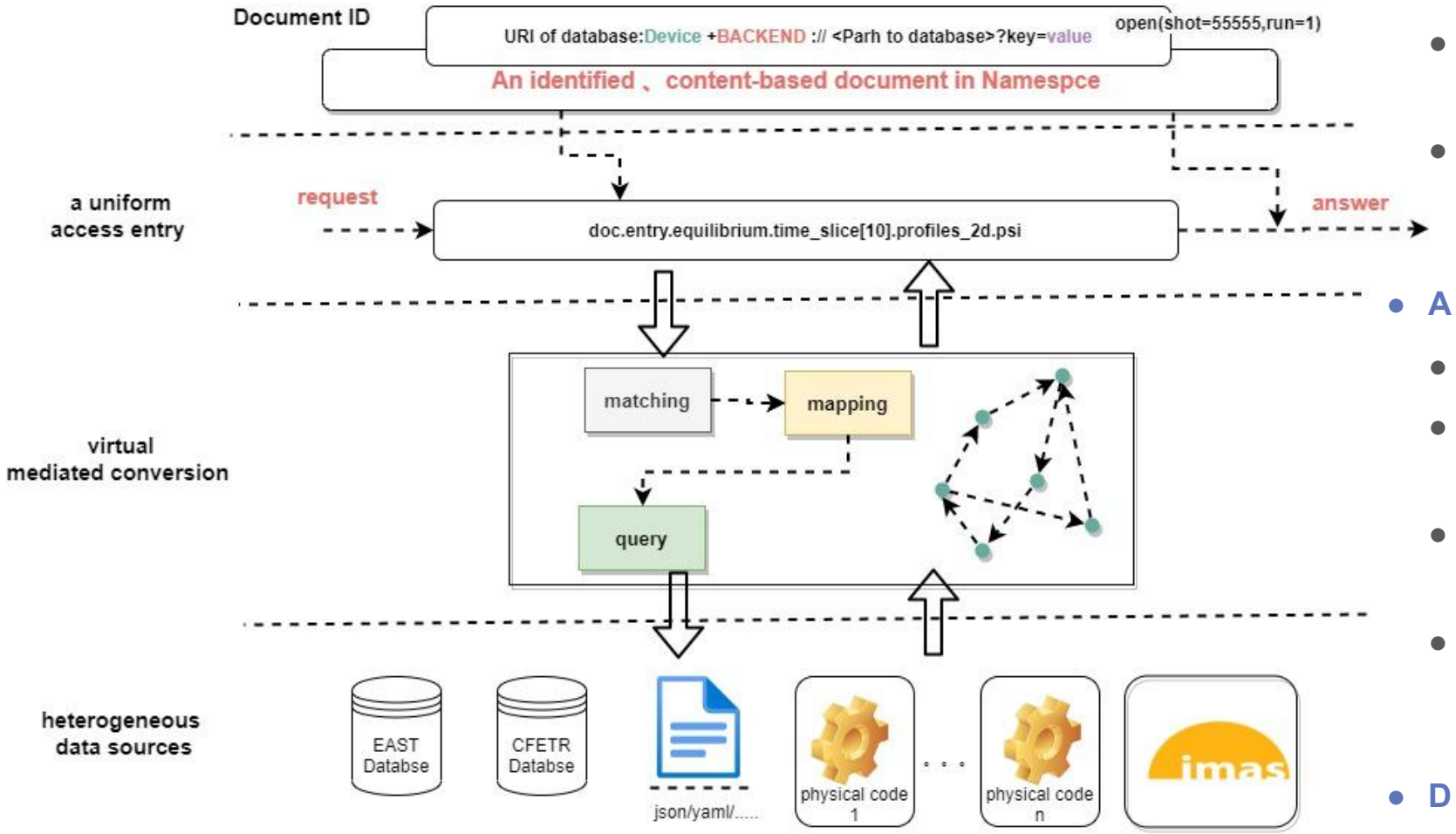
# Fundamental design concept ——
# "Database is a collection of documents"

- Purpose :The data interfaces of different data sources are unified into the ITER physical data model.
- Data Intergration: Integrate different data sources into a **unified namespace**, which can be accessed through **a URI.**
- **Namespace is a container that organizes and uniquely restricts content based on PDM**
  - The top container is a **collection of documents**.
  - The document is **a unified hierarchical tree structure** that integrates different data objects.
- **Data objects in Fusion** can be divided into three categories :




- **Configuration data** (readonly, identified by '**device**')
- **Experimental data** (readonly, identified by '**shot**')
- **Analysis data** readonly, identified by '**run**')

# Framwork

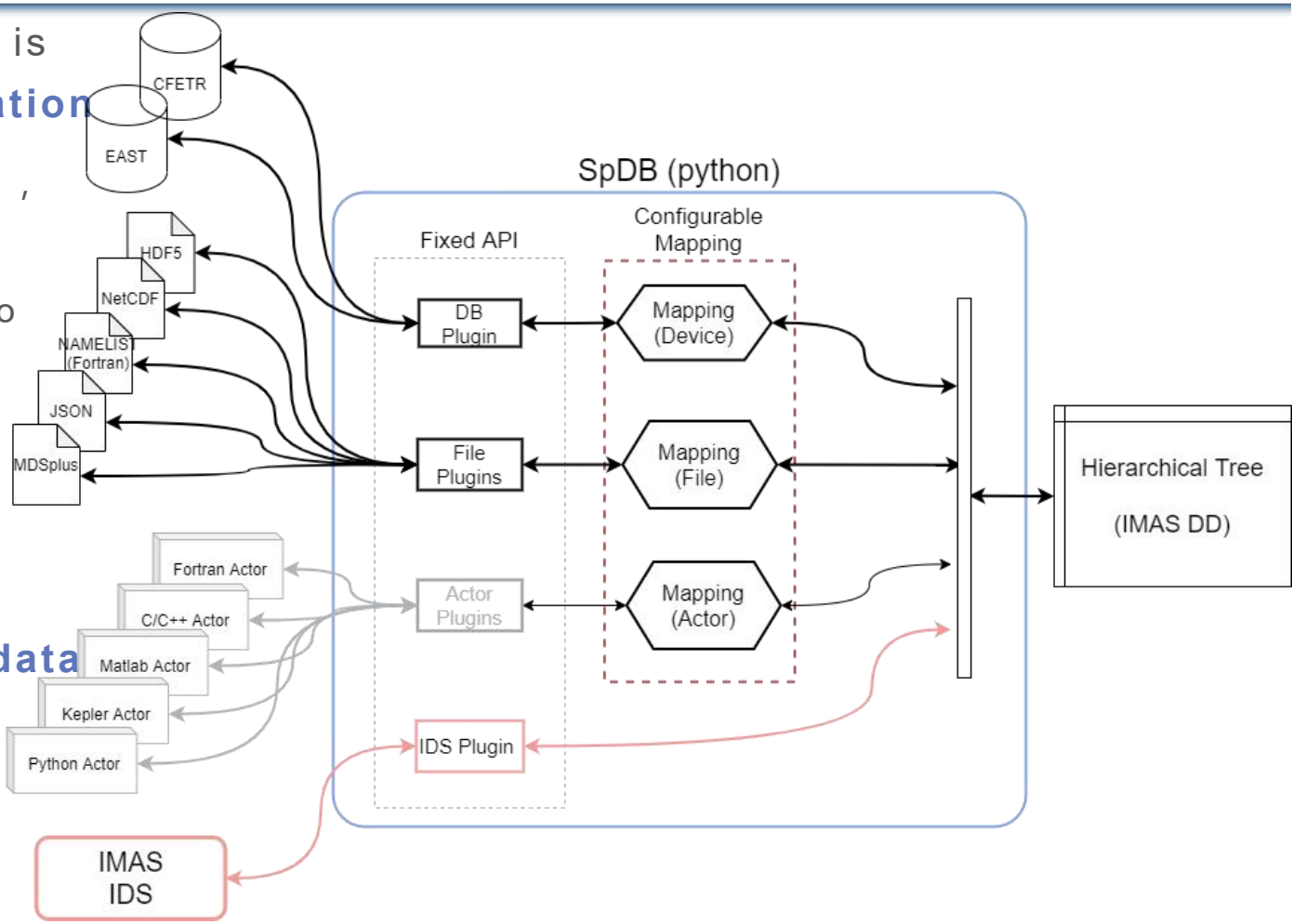



- **A uniform access entry:**
  - a uniform URI determined by a variable key-value:
  - a global unified request path which is defined by a hierarchical tree structure in the document
- **A virtual mediated conversion layer:**
  - **virtual:** not store any data
  - **matching** the key-value in URI to determine the source of the data
  - **mapping** the schema in request to the target schema in mapping file
  - **query reformulation** is to translate the user queries into a set of subqueries based on source schemas.
- **Different heterogeneous data sources:**
  - device database
  - physical modules IO
  - IMAS、OMAS

ASIPP

# Data model and Data representation

- Data is meaningful because of semantics; Data is defined by the **data model** and **data representation**
  - **Data model:** the semantics and structure of data ,
    - ITER Physics Data Model
  - **Data representation:** the concrete data format to store data
    - File Format: HDF5, MDSplus, JSON, XML...
    - Language-specified data type: Python(dict,list,numpy), Fortran(array,structure)....
- There are many types of **data representation(data format)**, and their APIs are always **fixed**.
- The **data model** has a unique definition, but the content changes **frequently.**

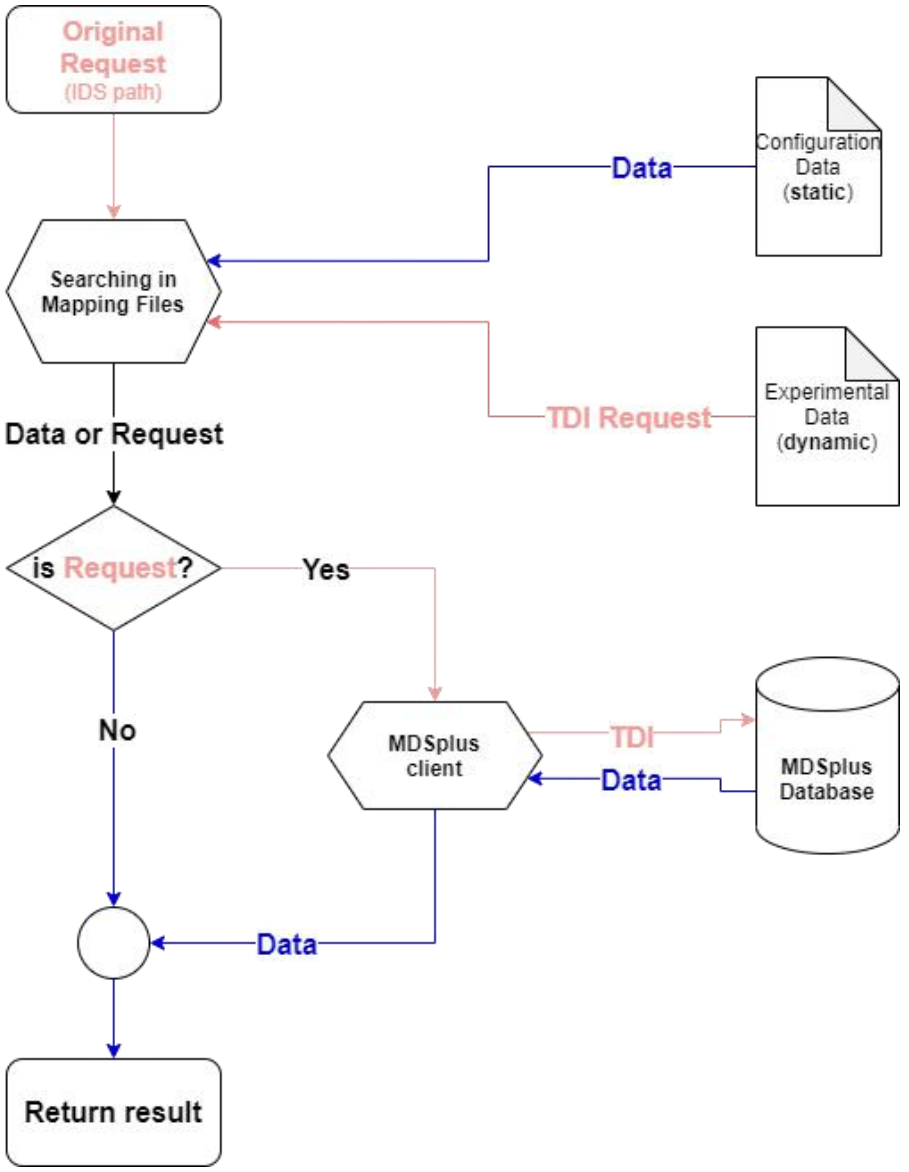



- **Data format conversion and data schema mapping are separated。 When the data model is changed, the content of the mapping file or configfile is modified without changing the API .**

# Database mapping

— Unified data access interface for experiment database.

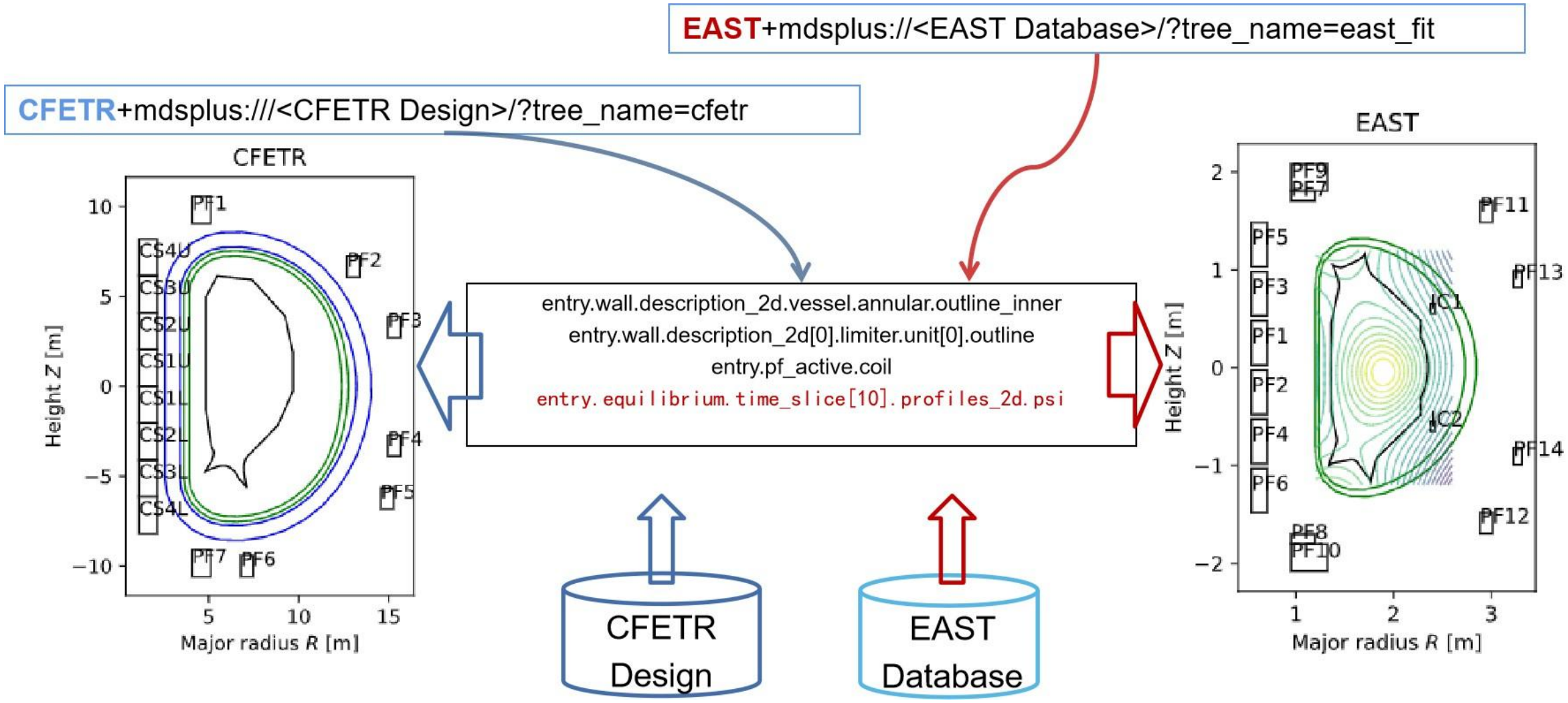
- **The mapping between data sources is the conversion of paths.**
- **Request is a PATH of data in a hierarchical tree-like structure;**
- A **configurable mapping rules** are defined in mapping files.
  - Support the **merging** of multiple data sources:static data and dynamic data
  - The position of the data element in a **hierarchical tree** is located by the path.
    - static data : return result
    - dynamic data :return TDI request
- **Data mapping is the conversion of paths between different data schema.**




ASIPP

# Unified access to experiment databases

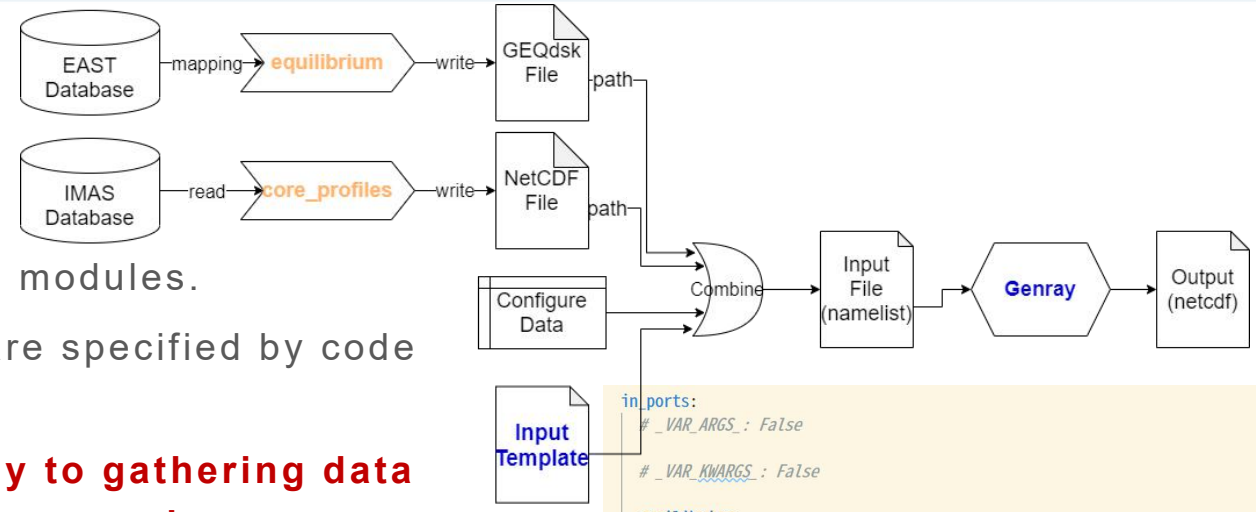
- **A uniform URI with different key-value** to identify two different databases.
- The **request to access data** from the two different databases is **unified** .
- The corresponding resultes are shown in the figure respectively .




ASIPP

# Lightweight data description mechanism
## — Unified data exchange interface between physical programs.

- Requirement for physics codes:
  - The input parameters are dynamically changing and multi-source gathered from experimental data, device data ,config file and the output of other physical modules.
  - Data format and semantics of input/output parameters are specified by code themselves
  - **The process of preparing input parameters is actually to gathering data from different sources into the same semantics and expression .**
  - It is essentially a **data integration process.**
- Implementation：
  - **One-to-one corresponding lightweight configuration description file ：**
    - The file defines the data semantics and data format of in/out parameters. when the module is called,the tool converts data into the format required by the code based on these descriptions.
- The data exchange between physics modules which are **called through a unified API** is achieved in the data semantic layer centered on PDM。




```
in_ports:
  # _VAR_ARGS_ : False

  # _VAR_KWARGS_ : False

equilibrium:
  $schema: geqdsk
  default:
    $class: file/geqdsk
    path: "{FY_MODULEFILE_DIR}/template/g063982.04800"

profile_in:
  $schema: "{metadata.annotation.name}/profile_in"
  default:
    $class: file/netcdf
    path: "{FY_MODULEFILE_DIR}/template/genray_profs_in.nc"

number_of_steps:
  $class: integer
  default: 1

dt:
  default: 0.1
  $class: float
  $comment: Inputs the data for all NAMELISTs used in the code.

config:
  $class: file/namelist
  template: "{FY_MODULEFILE_DIR}/template/genray.in"
  path: genray.dat
  default:
    genr:
      # dt:
      #   $class: "float"
      #   default:
      #     $ref: inputs.dt
```

ASIPP

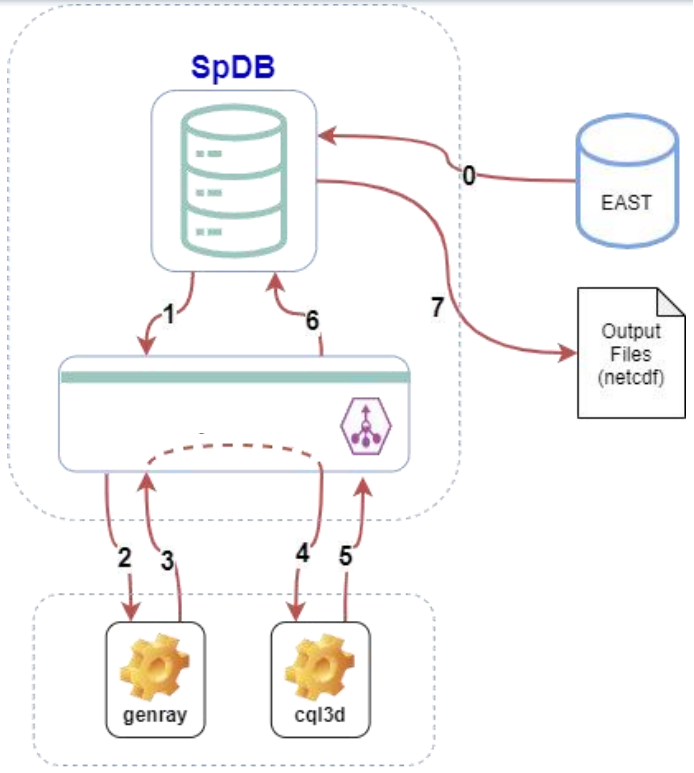
# Lightweight data integration: Genray+Cql3d

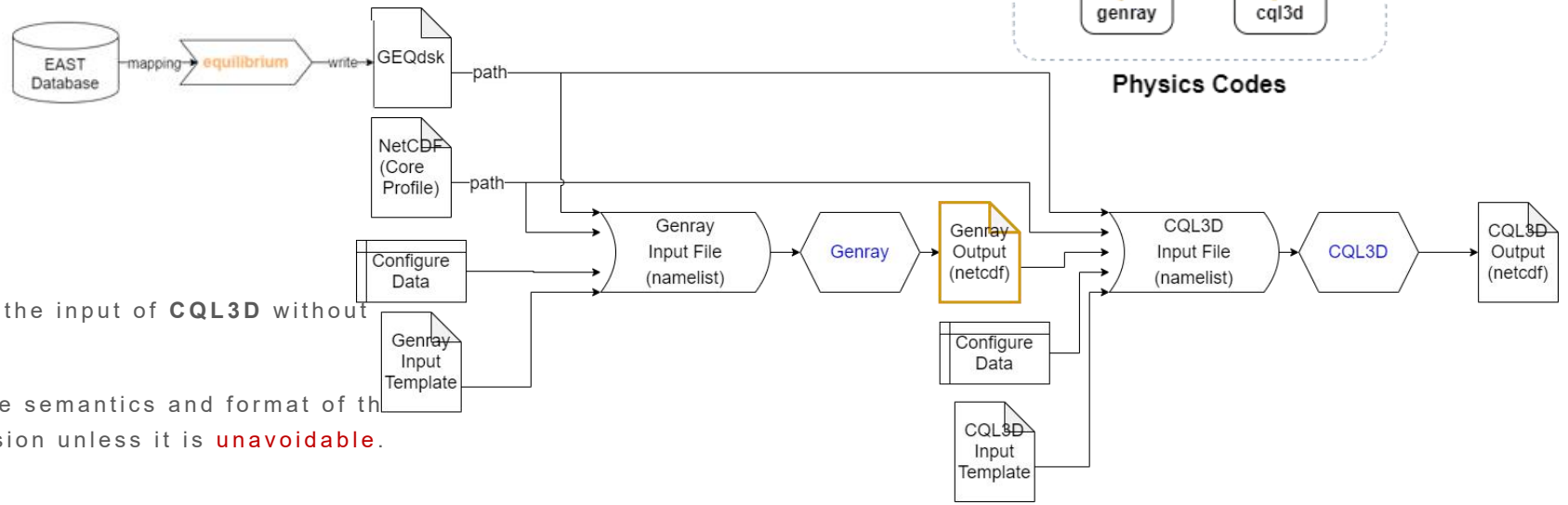
```
# open connection to EAST experiment database
doc = Collection(f"EAST+mdsplus://{DATA_DIR}/db/mdsplus/~t/?tree_name=efit_east",
                 mapping_data_path=DATA_DIR/"mapping") \
    .open(shot=55555, run=0, time_slice=100, mode="r") # open document

# load genray (Fortran version)
genray = spdm.module.physics.genray[{"version": "10.13_200117", "tag": "-gompi-2020a"}](
    profile_in=FILE_DIR/"genray/genray_profs_in.nc",
    equilibrium=FILE_DIR/"genray/g0.04800",     or from databas:equilibrium= doc.entry.get("equilibrium")
    config={"$class": "file/namelist",  "template": FILE_DIR/"genray/genray.in"}
)


# load cql3d (Fortran version)
cql3d = spdm.module.physics.cql3d(
    equilibrium=genray.outputs.equilibrium,
    genray_result=genray.outputs.result, # using the output of genery as input
    config={"$class": "file/namelist",
            "default": {
                "setup": {
                    "nstop": 1,
                    "nplot": 1,
                    "nplt3d": 0
                }
            },
            "template": FILE_DIR/"cql3d/cqlinput"})
```

The data integration of physical program IO is realized through standardized data description, which simplifies the data exchange between codes.




The output of **Genray** is directly used as the input of **CQL3D** without conversion throgh **SpDB** .

Sometimes it's better to be lazy! Keep the semantics and format of th original data, and do not perform conversion unless it is unavoidable.

ASIPP

# Summary

- A data integration toolkit was developed **for experiments, modeling and simulation** in fusion research.
- Using the **ITER Physical Data Model** as the global unified schema.
- The core design concept :
  - Hierarchically structured data can be organized into **collections of documents.**
  - Data is defined by **data model and data representation**, which should be separated in implementation.
- The Framework  is divided into three layers:
  - A uniform access entry,
  - A virtual mediated conversion layer
  - Plugins of different heterogeneous data sources
- A **lightweight data description** is defined to unify the data exchange interface between the physics module.

ASIPP

ASIPP

# Thank you for attention