Status and Plans of IMAS at ITER

S.D. Pinches
ITER Organization

Disclaimer: The views and opinions expressed herein do not necessarily reflect those of the ITER Organization

Supporting ITER design activities and performance assessments

HIGH-FIDELITY PLASMA SIMULATOR

High-Fidelity Plasma Simulator

- As part of an effort to build in-house experience, over the past year the IO has developed a Python workflow for Heating and Current Drive modelling
- Builds on long history of H&CD integration within EUROfusion WPCD Task Force by Thomas Jonsson
- Within IO, primarily supported by internship students:
  - Verena Mitterauer (6 months): development of the Python H&CD workflow and its graphical interface
  - Amanda Wei (3 months): comparison of the results between the Kepler and Python H&CD workflows
- Integrates contributions of others:
  - Adaptation of GENRAY to IMAS (Shinichiro Kojima), PION (Ignacio Lopez et al), OGRAY (Alexey Kuyanov)

Developer-friendly properties of Python H&CD workflow

- **Version control:**
  Can be developed by multiple users simultaneously with merge capabilities.

- **Auto-generated graphical interface and representation of workflow:**
  Changes to workflow are represented in main GUI and graphical representation. The overall interface / workflow is light (< 100 kB).

- **Workflow compatibility:**
  Easy to couple workflows using standard IDSs data representations. Workflow doesn't depend on specific Python versions and is not compiled.

- **Easy to add physics codes to workflow:**
  To add an additional IMAS actor, only its name has to be added to an xml file (I/O IDSs automatically detected). When a code is updated, the changes are automatically seen by the workflow (no compilation required).

User-friendly properties of Python H&CD workflow

- **Speed:**
  The Python graphical interface opens in ~10 ms (Tkinter).

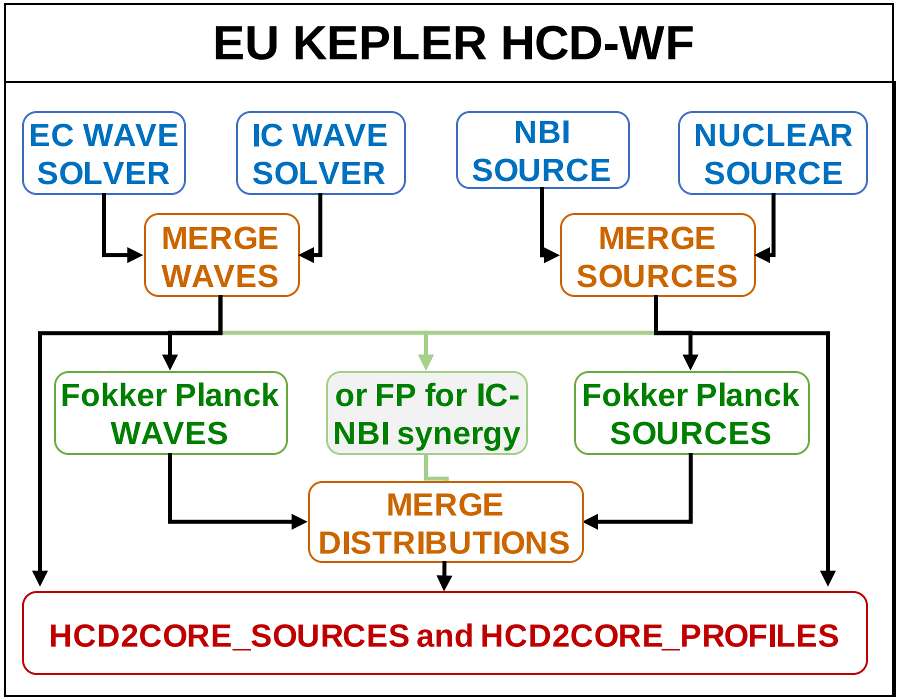
- **Actor dependencies:**
  The Python workflow does not need all its physics actors to be present / compiled to be executed (as long as the selected physics actors are there).

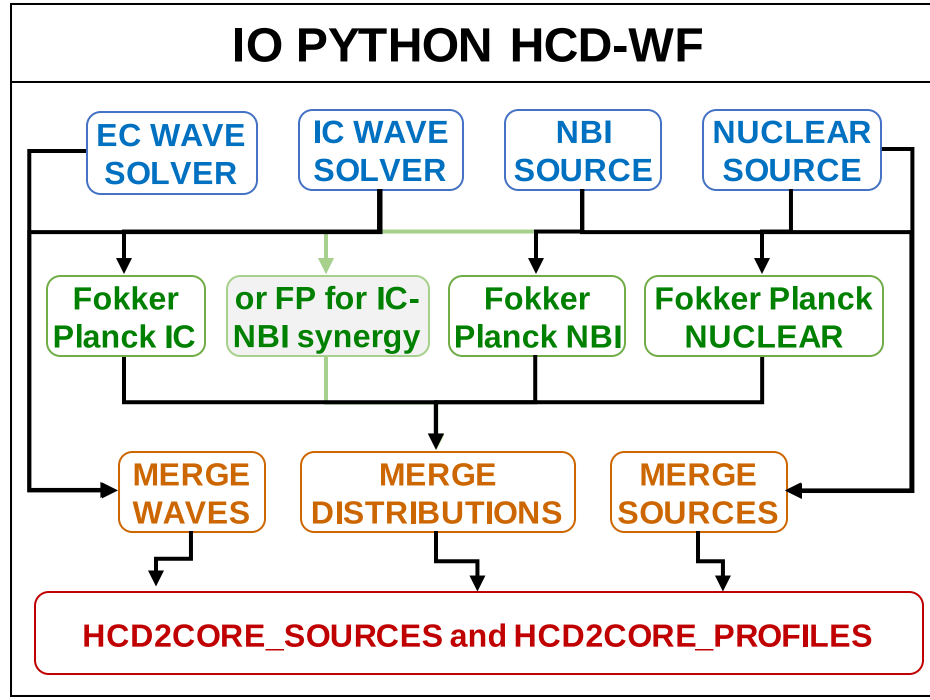
- **Simulation parameters:**
  They are all centralised and accessible from the main interface (parameters of the workflow and of the physics codes). They are saved independently of the workflow and can be re-loaded for another simulation.

- **Execution:**
  Several Python workflows can be run in parallel.

Conceptual differences between Kepler and Python algorithms

- Algorithm not strictly the same: less elegant but more flexible version





Results of wave and source codes are merged before Fokker-Planck calculation.
→Cleaner, but only one kind of FP solver can be selected for NBI, nuclear reactions (and ICRH in case of synergy).

The merge is done after the Fokker-Planck calculation.
→ Different FP solvers can be selected for all heating sources separately: more flexible for the user.

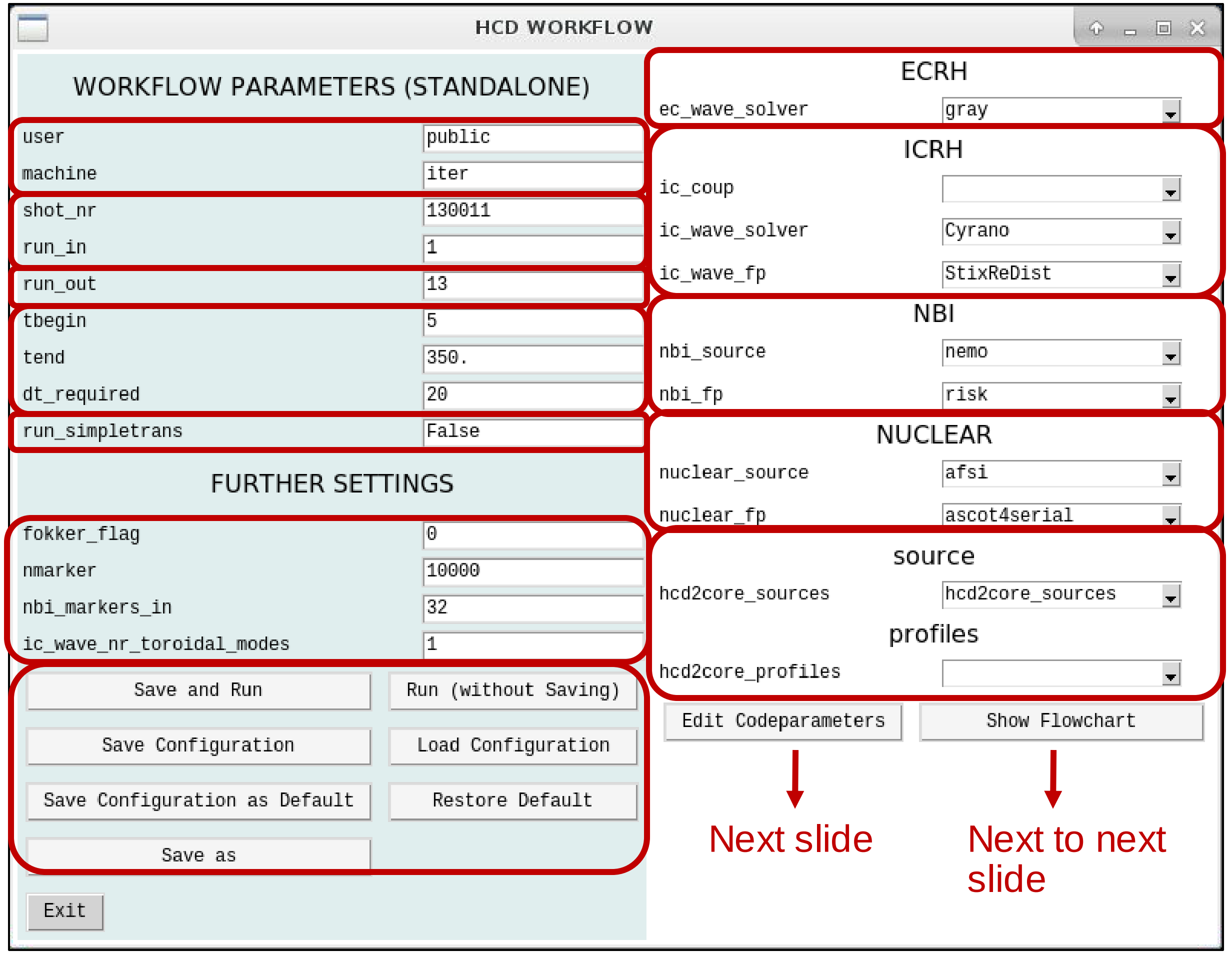
Reminder about H&CD codes currently adapted to IMAS

	Nuclear reactions	ECRH	ICRH	NBI	LH
Wave or source	AFSI SPOT (α)	GENRAY GRAY (OGRAY)	CYRANO LION PION TOMCAT	BBNBI NEMO	GENRAY
Fokker-Planck	ASCOT SPOT	(OGRAY)	ASCOT PION SPOT	ASCOT RISK SPOT	SPOT (ions)

Other codes used in the H&CD workflow:

- ICCOUP: for IC wave coupling calculation
- Mergers: merge IDSs together (waves, distribution_sources, distributions).
- hcd2core_sources: fill the core_sources IDS for transport solvers.
- hcd2core_profiles: fill the fast and thermal components of core_profiles IDS for transport solvers.

Description of the Python H&CD workflow



Input database

Input scenario

Output run

Time configuration

Run simple transport solver

Extra settings

Save and load configuration, execute the workflow

EC solver

IC wave and FP solvers

NBI source and FP solvers

Nuclear source and FP solvers

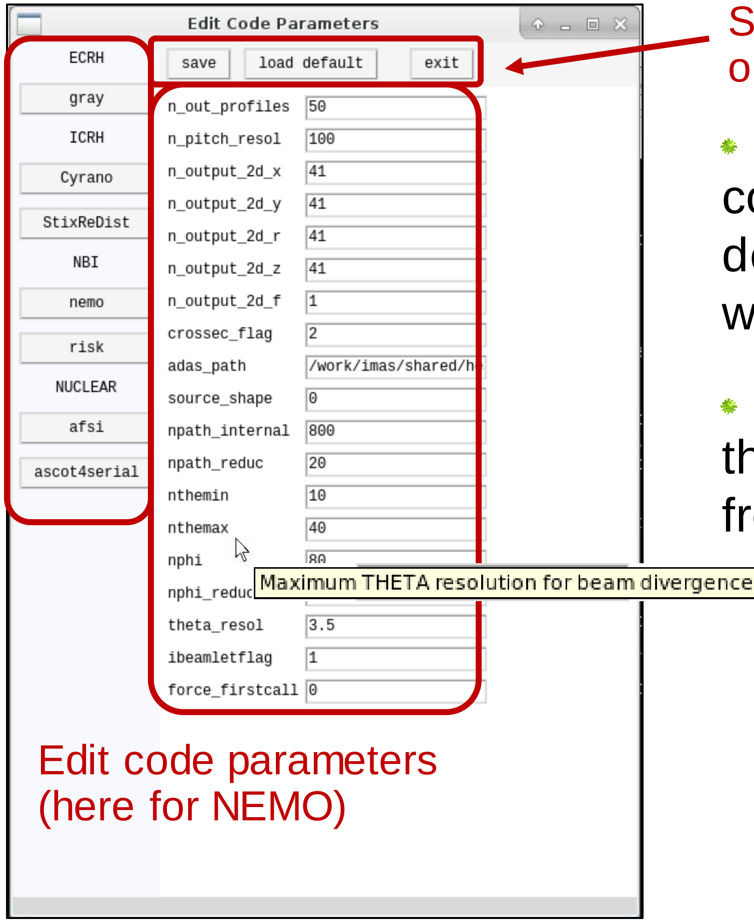
Fill core_sources and core_profiles (or not)

Next slide

Next to next slide

Edit Code Parameters using dynamically created GUI

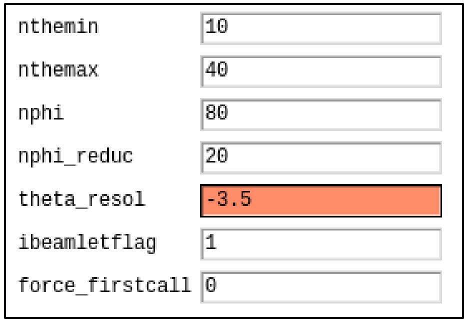
Edit parameters for each code selected from the main GUI



Edit code parameters (here for NEMO)

Save parameters or restore default

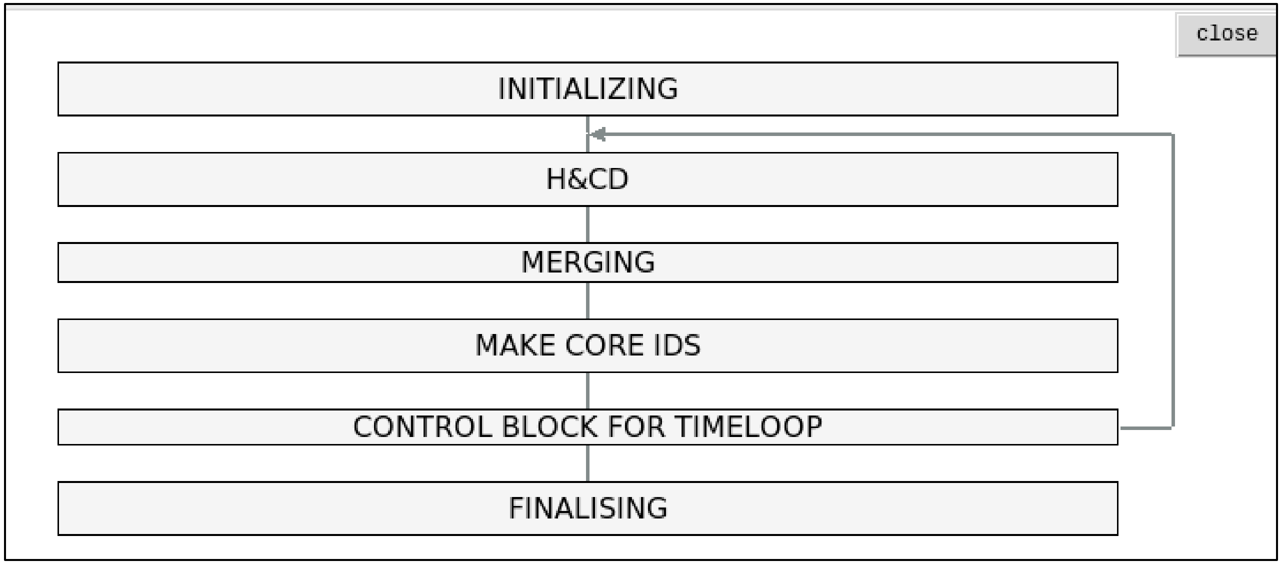
- Parameter files for each H&CD code are stored on disk in a dedicated folder along with the workflow simulation parameters.
- Rules and definitions are set through an xsd file to prevent the user from choosing wrong parameters.



Show flowchart

- General view of the flowchart with expandable boxes representing each main step of the H&CD workflow.

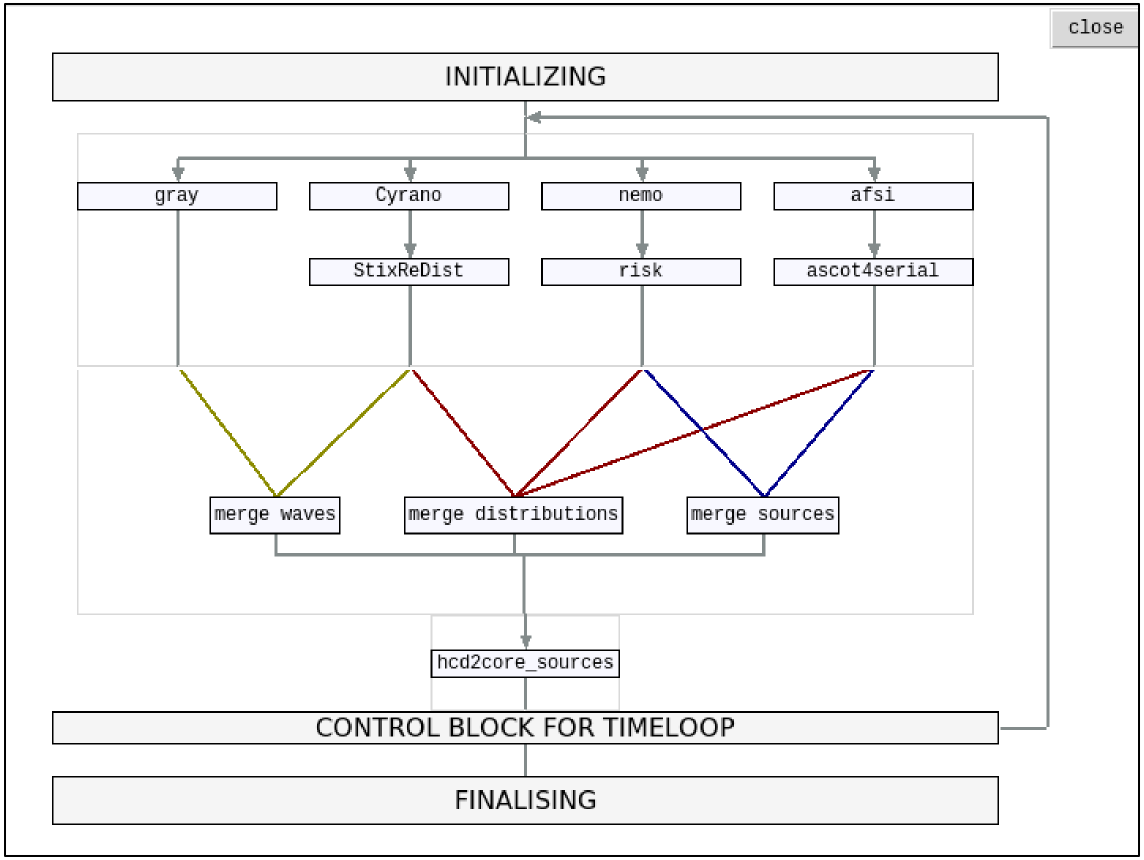
- Clicking on one of these boxes expands the view to more details.



Show flowchart

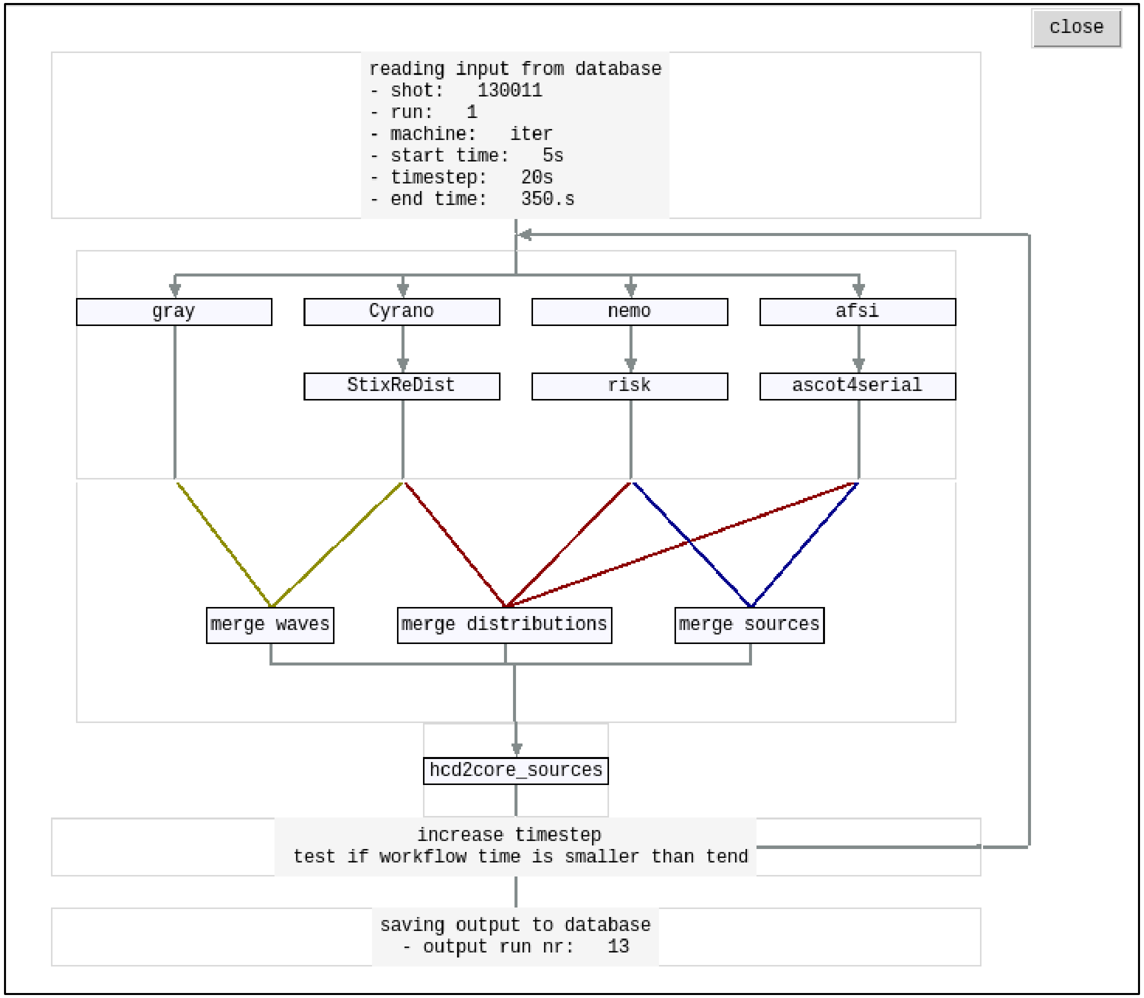
- The H&CD box expands to display the H&CD codes as selected from the main interface.

- The merging box shows the mergers for:
  - waves
  - distributions
  - distribution_sources



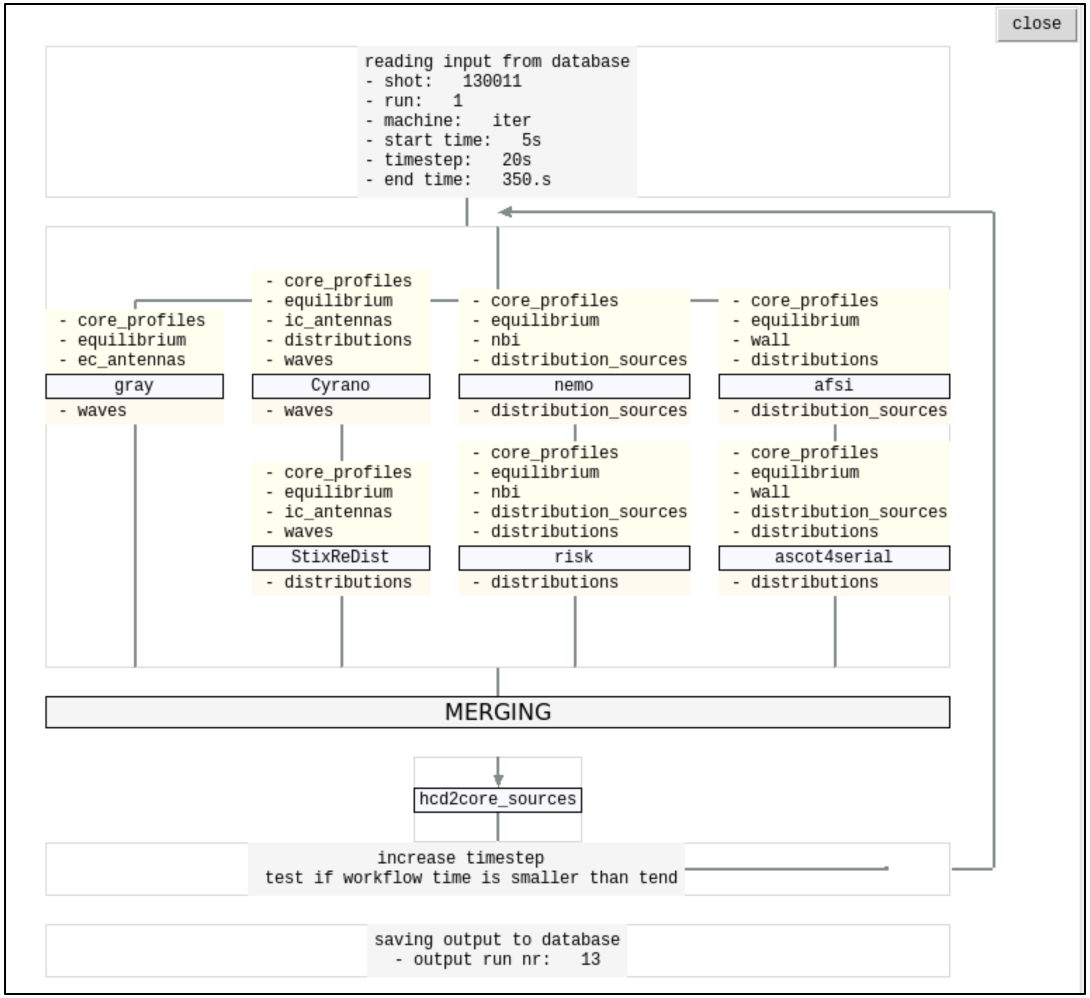
Show flowchart

- Expanding the initialization and finalization boxes give general details on the global simulation itself:
  - Input scenario.
  - Time management:
    - Starting time
    - End Time
    - Time step
  - Output run.

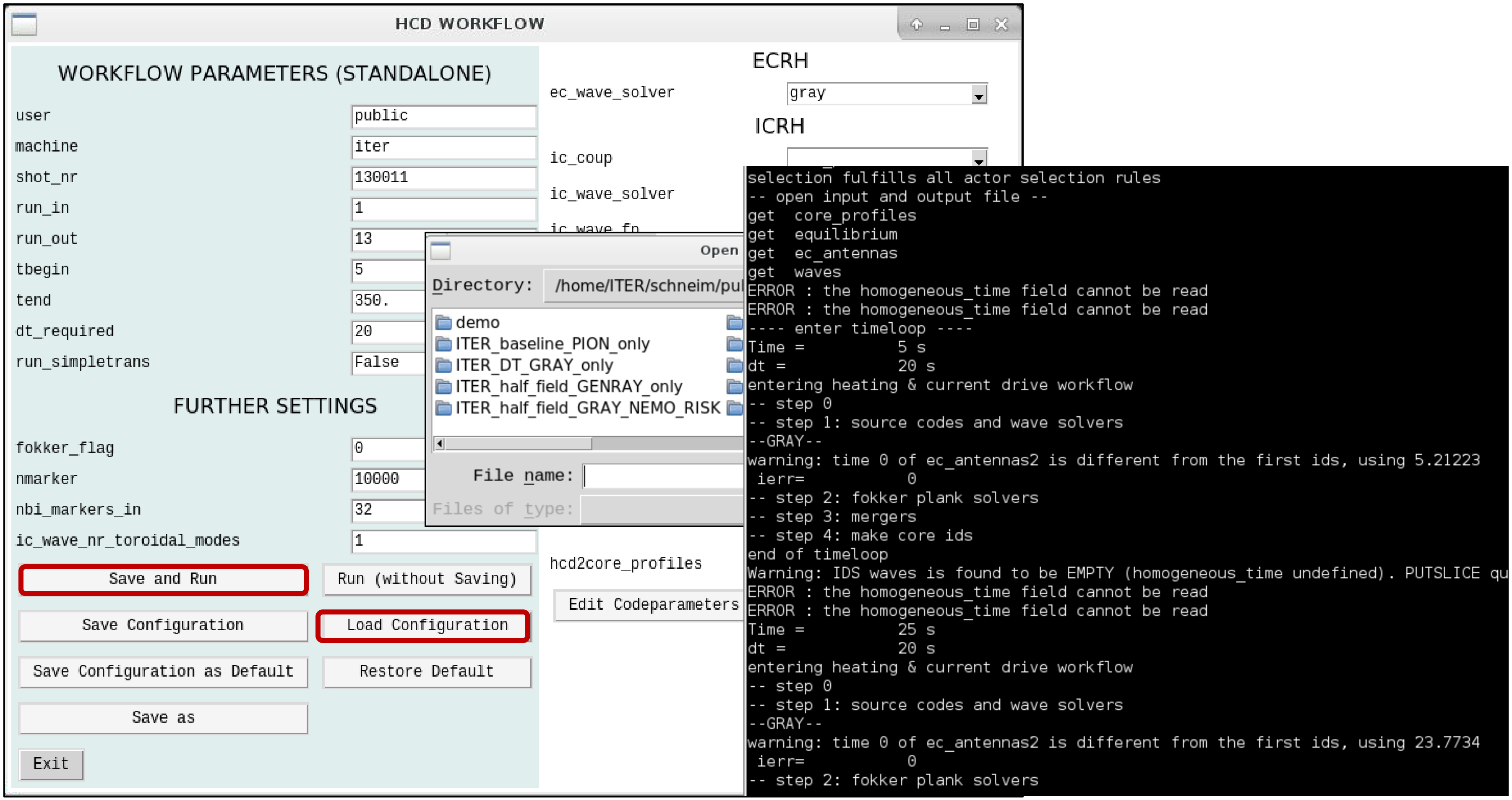


Show flowchart

- Clicking on each H&CD box leads to the display of their input and output IDSs.



Execute the workflow



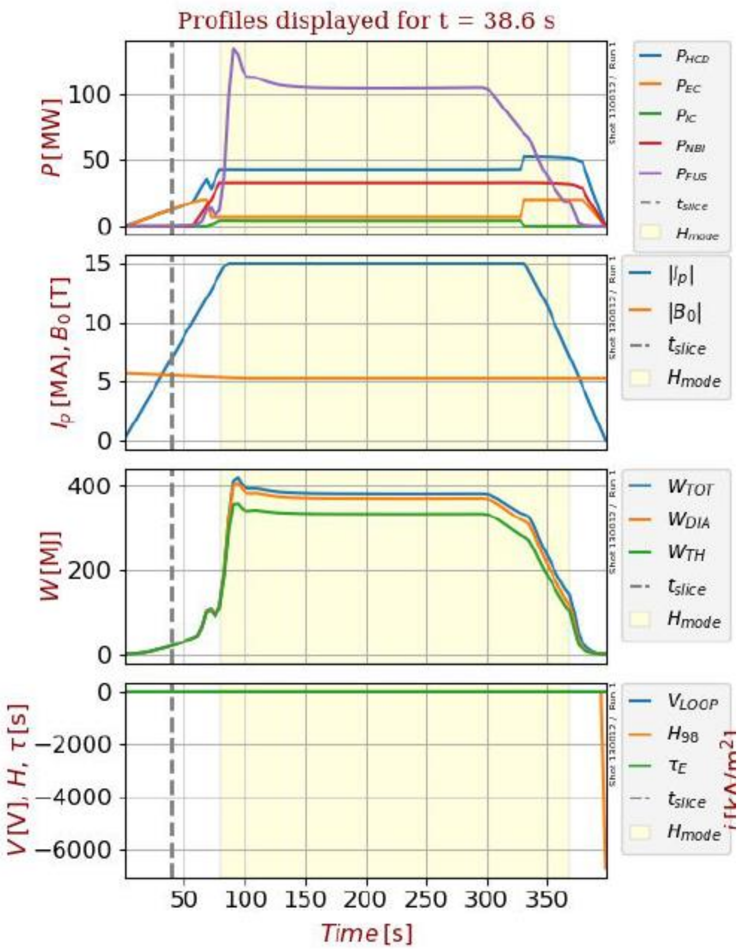
SOLPS-ITER

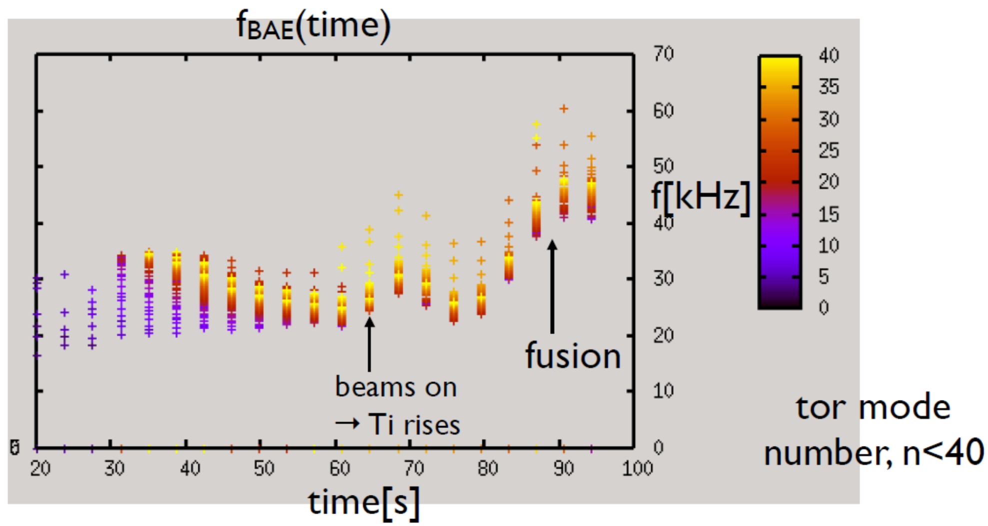
- Over 200 SOLPS edge simulations now available as IDSs
  - All SOLPS4.3 runs now in scenario database
- All cases include new radiation IDS that was added to support ITER diagnostic design activities

Pulse	Run	Machine	Reference	Ip[MA]	B0[T]	Fuelling	Confinement	Workflow
102274	1	iter	ITER#2274_(F57-20MW-H,_tau=0.25e-6,_ntime=50,_dt)	-15.0	-5.3	H	tbd	SOLPS
102277	1	iter	ITER#2277_(F57-40MW-H,_tau=0.25e-6,_ntime=50)	-15.0	-5.3	H	tbd	SOLPS
102278	1	iter	ITER#2278_(F57-H-40MW-NF-Be0,_tau=0.21e-6,_ntime=150)	-15.0	-5.3	H	tbd	SOLPS
102279	1	iter	ITER#2279_(F57-40MW-H,_tau=0.25e-6,_ntime=150)	-15.0	-5.3	H	tbd	SOLPS
102280	1	iter	ITER#2280_(F57-40MW-H,_tau=0.19e-6,_ntime=150)	-15.0	-5.3	H	tbd	SOLPS
102281	1	iter	ITER#2281_(F57-40MW-H,_tau=0.28e-6,_ntime=150)	-15.0	-5.3	H	tbd	SOLPS
102282	1	iter	ITER#2282_(F57-40MW-H,_tau=0.28e-6,_ntime=150)	-15.0	-5.3	H	tbd	SOLPS
102283	1	iter	ITER#2283_(F57-40MW-H,_tau=0.28e-6,_ntime=150)	-15.0	-5.3	H	tbd	SOLPS
102284	1	iter	ITER#2284_(F57-40MW-H,_tau=0.18e-6,_ntime=150)	-15.0	-5.3	H	tbd	SOLPS
102285	1	iter	ITER#2285_(F57-60MW-H,_tau=0.17e-6,_ntime=150)	-15.0	-5.3	H	tbd	SOLPS
102286	1	iter	ITER#2286_(F57-60MW-H,_tau=0.21e-6,_ntime=150)	-15.0	-5.3	H	tbd	SOLPS
102287	1	iter	ITER#2287_(F57-60MW-H,_tau=0.21e-6,_ntime=150)	-15.0	-5.3	H	tbd	SOLPS
102288	1	iter	ITER#2288_(F57-60MW-H,_tau=0.21e-6,_ntime=150)	-15.0	-5.3	H	tbd	SOLPS
102289	1	iter	ITER#2289_(F57-60MW-H,_tau=0.21e-6,_ntime=150)	-15.0	-5.3	H	tbd	SOLPS
102290	1	iter	ITER#2290_(F57-60MW-H,_tau=0.21e-6,_ntime=150)	-15.0	-5.3	H	tbd	SOLPS
102291	1	iter	ITER#2291_(F57-60MW-H,_tau=0.21e-6,_ntime=150)	-15.0	-5.3	H	tbd	SOLPS

`scenario_summary -s SOLPS`

Fast Ion Stability





fBAE ~ √Ti

all BAEs between s=0.2-0.6

- LIGKA/HAGIS Python workflow to assess energetic particle stability in ITER scenarios
  - Will use EP distribution from HCD WF
  - See Lauber *et al.*, IAEA FEC 2020

Plans for High-Fidelity Plasma Simulator

- Continue adaptation of DINA and JINTRAC to IMAS
- Enable calling HCD workflow from JINTRAC
  - Benchmark use of codes in HCD workflow with integrated version, e.g. PION in JINTRAC vs. calling HCD workflow from JINTRAC
- Look to building single Plasma Simulator workflow utilizing both DINA and JINTRAC

Preparing for ITER experimental data

DATA PROCESSING & ANALYSIS

Automated Processing of Data

- Envisaged that many data processing chains will run concurrently (on dedicated hardware) as soon as their input raw data dependencies (in form of IDSs) are satisfied during a pulse
  - Whilst some simple linear chains will have modest computational requirements, more complex statistical (Bayesian) inference chains may consume significant resources
  - Since these latter chains are envisaged to be highly parallelizable, computational capabilities should ensure delivery of processed data does not impact inter-shot time or delay next pulse

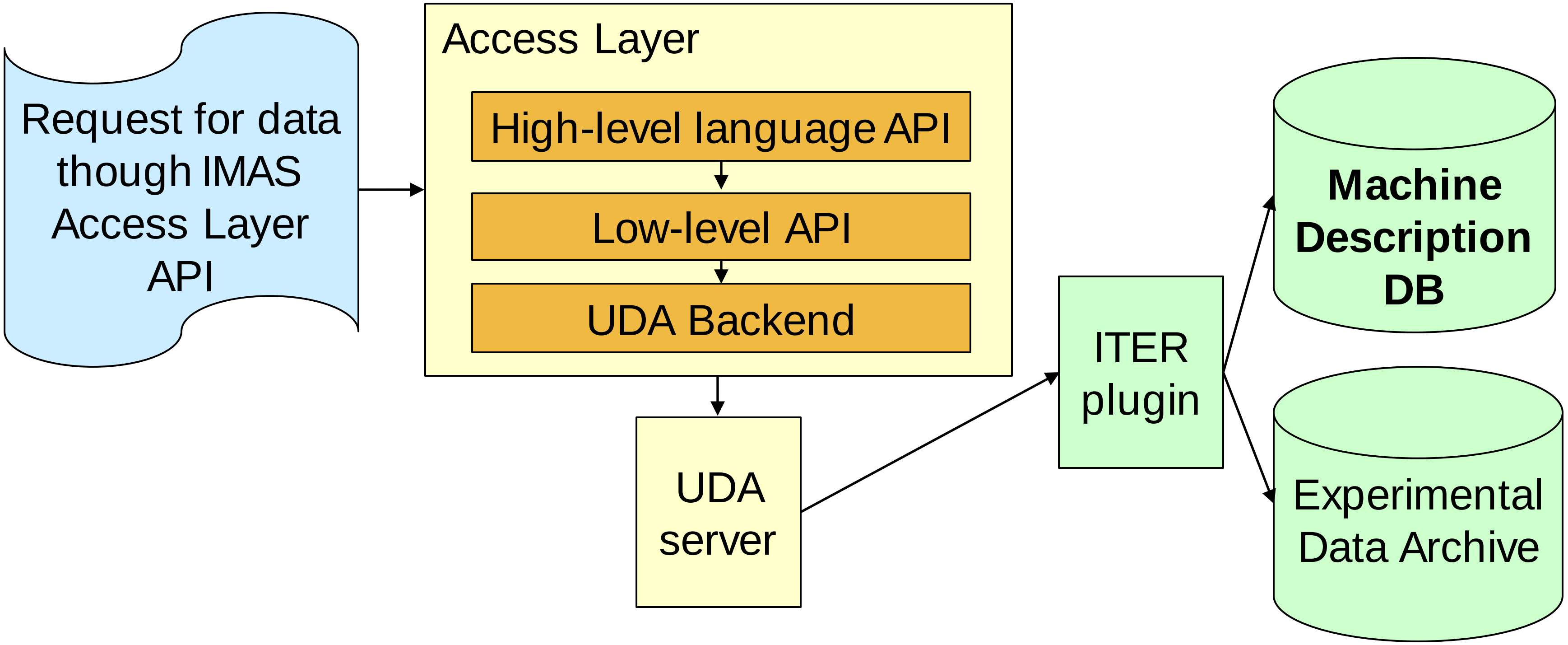
  → Scalable parallel computing infrastructure
  - Close collaboration with devices which can map and serve raw data and matching Machine Description data
  - First example magnetics-only equilibria

ITER Code Camp

- Proposal to hold dedicated Code Camp at ITER focused on generating IDS-based equilibrium reconstructions from raw experimental data that's been dynamically mapped into IDSs
- IMEG being consulted on possible dates / contributors

ITER Machine Description Data

- Centrally handled in dedicated SQL database
  - Created and managed by Controls Division
- UDA plug-in created to handle returning IDSs



Data Analysis and Interpretation Platform

- Call for initial development launched following discussion with ITPA Topical Group on Diagnostics and IAEA TM on Fusion Data Processing, Validation and Analysis (driven by IMEG)
  - ITPA Diagnostics provide voluntary support for evaluation and testing;
- Prototype to allow rigorous interpretation and analysis of experimental data and for a community of users (IO staff & external) to gain experience in its development and application
  - Interpret data from individual diagnostics;
  - Unified methodology for integrating data from multiple diagnostics to obtain improved results with derived uncertainties;
  - Support generation of realistic synthetic diagnostic data to assess performance of diagnostics and develop data interpretation techniques;
  - Estimate hardware requirements to support running automated interpretation workflows that combine a realistically achievable combination of diagnostics during ITER's PFPO and FPO phases;

ITER Computational Environment and IMAS Installation

INFRASTRUCTURE

Current ITER Computing Environment

- Current configuration has ~2500 cores and 90 TB of storage
  - Further upgrades planned towards end of year (increased CPUs)
- Shared by Science Division and other areas of Project
- At the end of 2019, the filesystem of the ITER GPC cluster was changed to GPFS with significantly improved performance
  - Up to limits of InfiniBand, 5 GB/s
- New GPU-equipped terminal servers being tested for access
  - Support remote visualization (scalable NoMachine-based solution)
- Environment modules provided by EasyBuild
  - Compiler toolchains: GCC/6.4.0, intel/2018a, PGI/18.4, NAGfor/6.2.14

  → Plan to move to latest versions in first half of 2020, e.g. GCC/9.2, etc.
  - Latest IMAS environment provided by the environment module: IMAS/3.26.0-4.5.0

Windows & MacOS

- IMAS is now routinely build for Windows on the ITER CI server
  - High-level Access Layer APIs in Python, Java and Matlab including remote data retrieval with UDA
- IMAS infrastructure can now also be build on MacOS
  - Imminent incorporation of Mac mini CI build agent will allow routine compilation
- Package managers for both Windows and Mac environments have been tested as a means to simplify distribution and installation

ITER Scenario Database

- Continues to expand, now >350 entries (through AL/v4)
- Will soon include copies of all scenarios used for ITER design / performance assessments
- Much data is available in both v3 and v4 flavours of Access Layer

  → In the future, data will only be available through v4

Plans for infrastructure at ITER

- Separate creation, installation and use of IMAS Python actors from Kepler
  - Gain flexibility with regard to central installations
- Improve standardization of actors/wfs to exclusively use IDSs for I/O and XML for actor/wf-specific parameters
  - Facilitated by new numerics IDS
  - Facilitates coupling and auto-generation of GUIs
- Establish central installations of released IMAS actors and workflows
  - Both actors and workflows have well defined interfaces
    - IDSs and optionally an actor/wf-specific XML file
  - Treat in same way as any other software installation
    - E.g. Environment module can handle paths and dependencies

Plans for infrastructure at ITER

- Provide remote access to ITER Scenario Database
  - Set up externally accessible UDA server
- Proceed with testing use of HDCs as identified at last IMAS Technical Review Meeting
  - Integrate option to use HDCs into IMAS installer
- Unify approach to treatment of IDS-based databases
  - ITER YAML-based prototype → SimDB, EUROfusion DBs, etc.