New developments in IMAS and Muscle3 for integrated modelling and digital twins

Daan van Vugt

IGNITIONCOMPUTING

iter
china eu india japan korea russia usa

ignitioncomputing.com
dvanvugt@ignitioncomputing.com

About Us

Building the software powering science

- Since 2020, in Eindhoven, NL
- Extensive track record:
  - Simulation software
  - Numerical methods
  - Machine Learning
  - High Performance Computing
  - Visualization
  - Performance optimization
- Team of 10 with MSc/PhD in Fusion, Applied Physics, CS or Mathematics

iter
china eu india japan korea russia usa

PACIFIC FUSION

EUROfusion

DIFFER

Plasma Matters.

CelSian
Glass & Solar

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

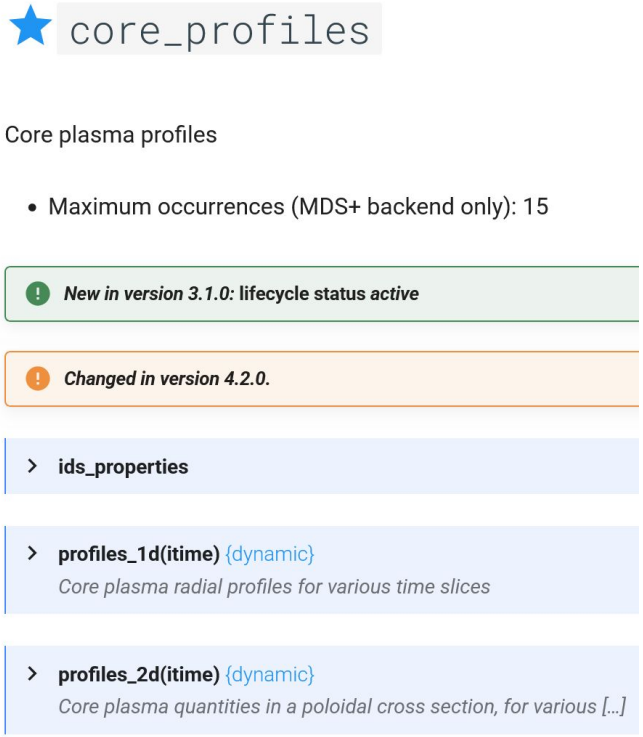
ProJoules

Diabatix

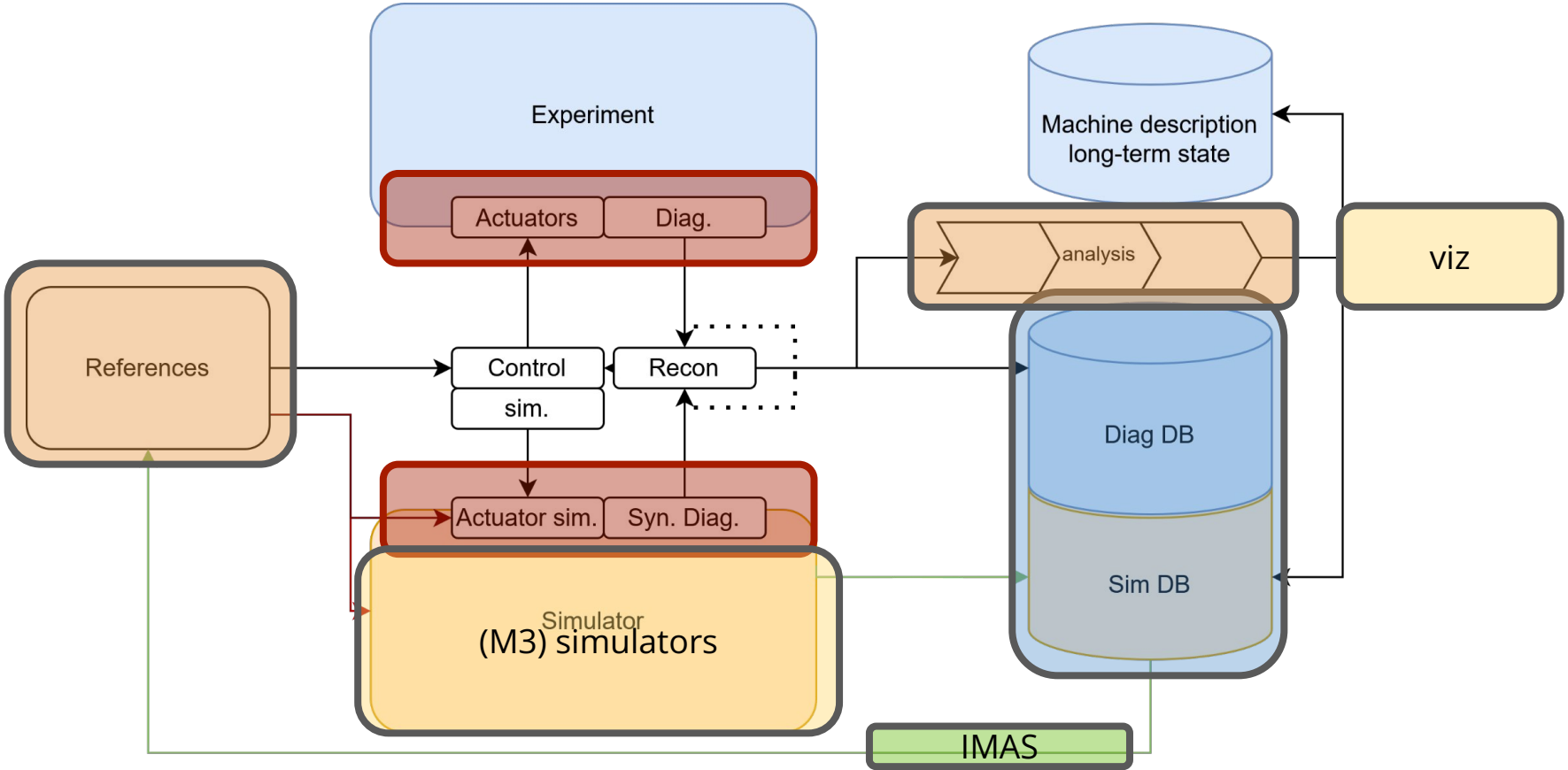
https://imas-data-dictionary.readthedocs.io/en/latest/

IMAS

- Standardized data ontology
- In-memory data structures
- File formats
- Software tools
- Physics applications & interoperability



What's in a digital twin

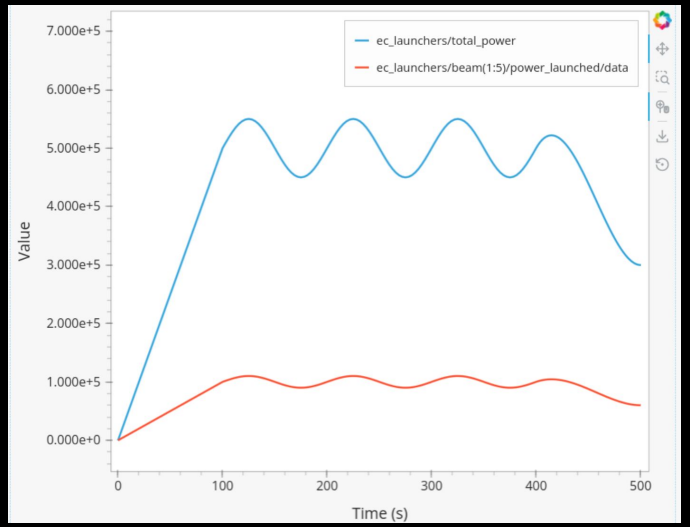


https://github.com/iterorganization/Waveform-Editor

Waveform Editor

- YAML format for specifying waveforms
  - Machine & human-readable descriptions
    - e.g. `linear`, `piecewise`, `repeat`, `sine`
  - Easy to share, alter, review
  - LLM compatible
- With expressions
  - `waveform1 = waveform2 + 10 * waveform3`
- Output IMAS variables
  - Export to PCSSP, M3 actor, more

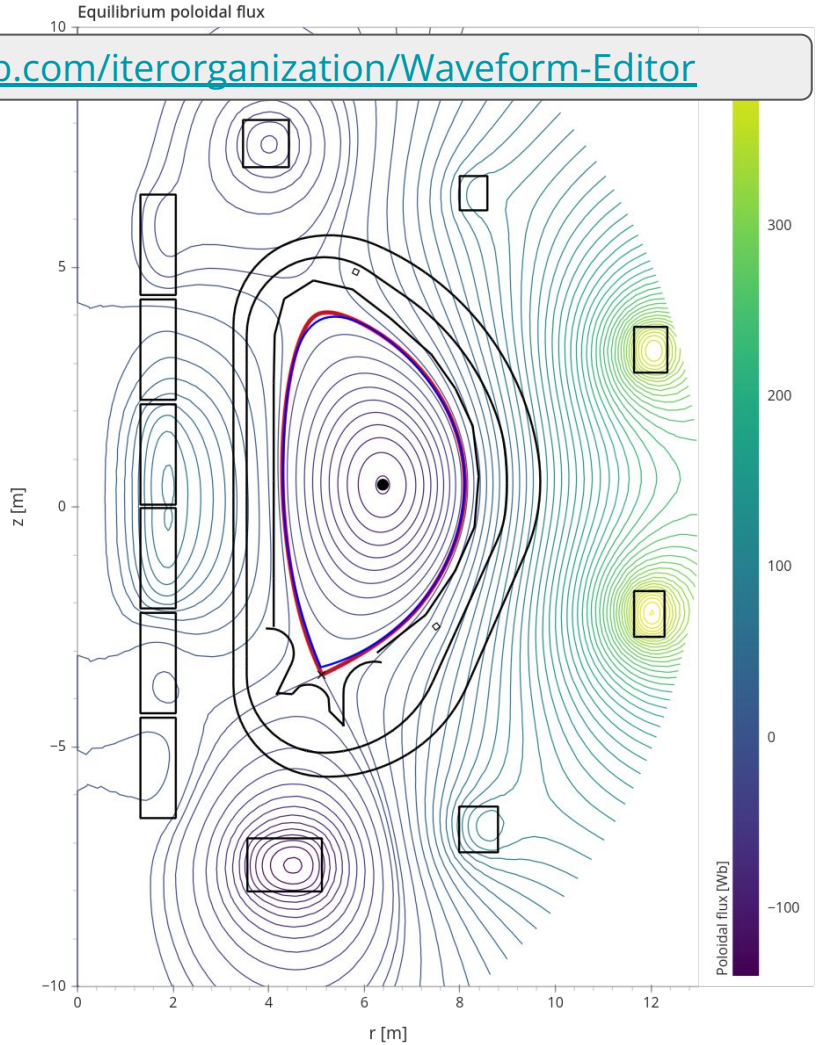
```
ec_launchers:
  ec_launchers/total_power:
    - { type: linear, duration: 100 }
    - { type: sine, base: 5e+5, amplitude: 5e4,
          frequency: 0.01, duration: 300 }
    - { type: smooth  to: 3e+5, duration: 100 }
  ec_launchers/beam(1:5)/power_launched/data: |
    'ec_launchers/total_power' / 5
```



https://github.com/iterorganization/Waveform-Editor

Plasma Shape Editor

- Define shape targets, test equilibrium
- Uses NICE FBE code
  - Desired coil currents → plasma shape
  - Desired plasma shape → coil currents
- Link to waveform editor
- Machine-agnostic
  - Demonstrate on WEST



Planned prototype

(IMAS?) descriptions of actuators

- (Control system) commands to actuators form a ‘natural’ interface
  - Could send to simulator instead
- Available actuators and configuration varies per machine
  - Limits application of simulators with controllers between machines
- Actuator simulators to translate commands into physics IDS
  - Example: ECRH steering and power commands -> ec_launchers IDS (t)

Planned prototype

(IMAS?) descriptions of diagnostics

- Several open diagnostics frameworks!
  - https://github.com/ToFuProject/tofu
  - https://github.com/ProjectTorreyPines/FusionSyntheticDiagnostics.jl
  - https://github.com/hoppe93/SOFT2
  - https://tokalab.github.io/SynDiag/Overview/
  - ...
- How to also describe which diagnostics exist, LoS etc?
  - Example:https://imas-data-dictionary.readthedocs.io/en/latest/generated/ids/bolometer.html#bolometer-camera-channel-line_of_sight
  - Link to machine description databases
  - Configure diagnostics framework, instead of program it?
- Build M3 actors based on this, and link to simulators

https://github.com/iterorganization/imas-standard-interfaces

IMAS interfaces

- Sometimes too many options to store data in IMAS
- Encode some necessary conditions for code/data interoperability
  - IDS variable requiredness, optionality
  - Use IMAS-validator for stricter checks
  - Whole interface far too complex. Use LLM to read interface code instead?
- Check incompatibility upfront
  - In M3 workflow documentation pages
  - Checked reads
- Data reduction before sending

```
schema_version: 0.0.1

doi_url: https://doi.org/10.5072/zenodo.511119

dd_version_range: [3.42.2, 4.0.0]

paths:
- equilibrium/ids_properties/version_put/data_dictionary
- equilibrium/time_slice/boundary/outline
```

https://imas-muscle3.readthedocs.io/en/latest/actor_olc.html

IMAS M3 Operating Limits & Conditions checking

```python
@validator("pf_active")
def validate_force_limits_pf(ids):
    """Validate forces in poloidal field coils are within operational limits"""
    for coil in ids.coil:
        for key in pf_force_limits.keys():
            if key in coil.name:
                if coil.force_vertical.data.value:
                    assert coil.force_vertical.data.value < pf_force_limits[key][0]
                    assert coil.force_vertical.data.value > pf_force_limits[key][1]
```

- **halt_on_error**: (bool) Whether or not the simulation should be forcibly stopped when a validation test fails. Defaults to False.
- **extra_rule_dirs**: (str) The rule directories in which to look for IMAS-Validator rulesets. If inserting multiple, split them with a ';'. Defaults to ''.
- **rulesets**: (str) The names of rulesets to run in the found rule directories. If inserting multiple, split them with a ';'. Defaults to 'PDS-OLC'.
- **apply-generic**: (bool) Whether or not to apply the generic bundled validation tests. Defaults to True.

https://github.com/iterorganization/IMAS-Validator

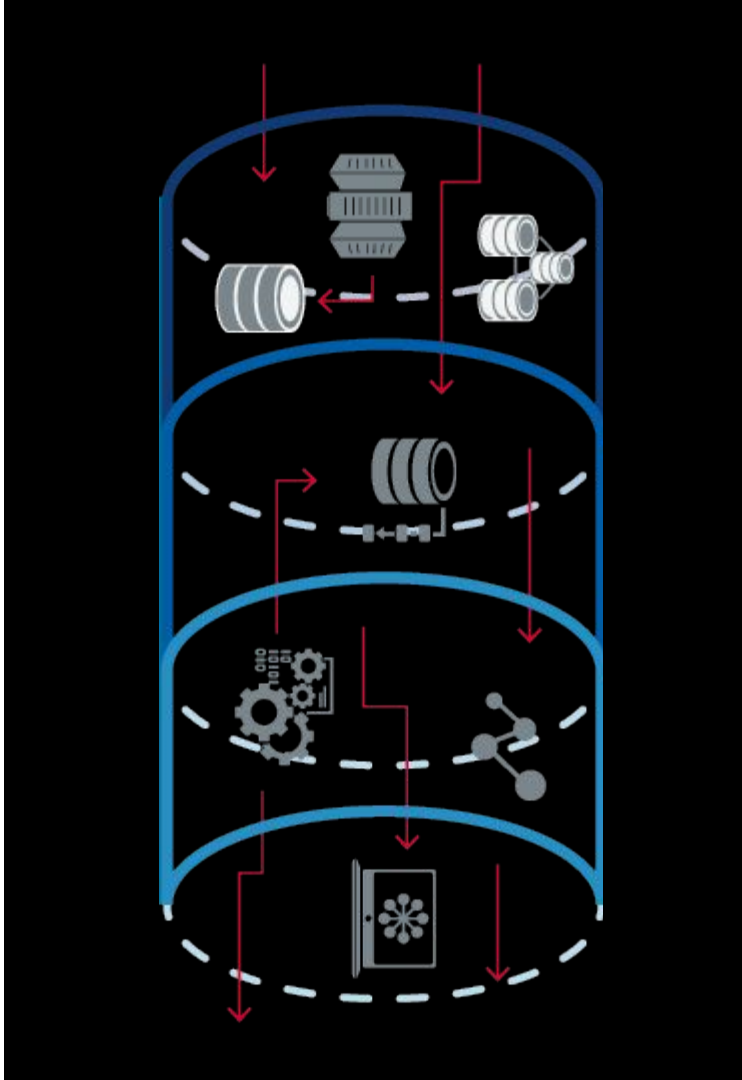
https://imas-muscle3.readthedocs.io/en/latest/

IMAS M3 Operating Limits & Conditions checking

```
Summary Report :
Tested URI : imas:hdf5?path=ids-validator-course/bad
Number of tests carried out : 16
Number of successful tests : 14
Number of failed tests : 2

PASSED IDSs:

FAILED IDSs:
- IDS core_profiles occurrence 0
        RULE: generic/generic.py:validate_increasing_time
                MESSAGE:
                TRACEBACK: FrameSummary file /work/imas/opt/EasyBuild/software/IDS-Validator/0.4.0-intel-2
023b/lib/python3.11/site-packages/ids_validator/assets/rulesets/generic/generic.py, line 43 in validate_in
creasing_time
                NODES COUNT: 1
                NODES: ['time']

- IDS waves occurrence 0
        RULE: generic/generic.py:validate_increasing_time
                MESSAGE:
                TRACEBACK: FrameSummary file /work/imas/opt/EasyBuild/software/IDS-Validator/0.4.0-intel-2
023b/lib/python3.11/site-packages/ids_validator/assets/rulesets/generic/generic.py, line 43 in validate_in
creasing_time
                NODES COUNT: 1
                NODES: ['time']
```

Analysis pipelines

- Streaming data during pulses
  - Mapped to IMAS data standard
  - Stream data frames with Kafka
- Near-live processing of physics data
  - Run interpretive simulations in tandem with experiments
- Browser-based visualisation
  - Standard & custom plots
  - Compatible with PDS



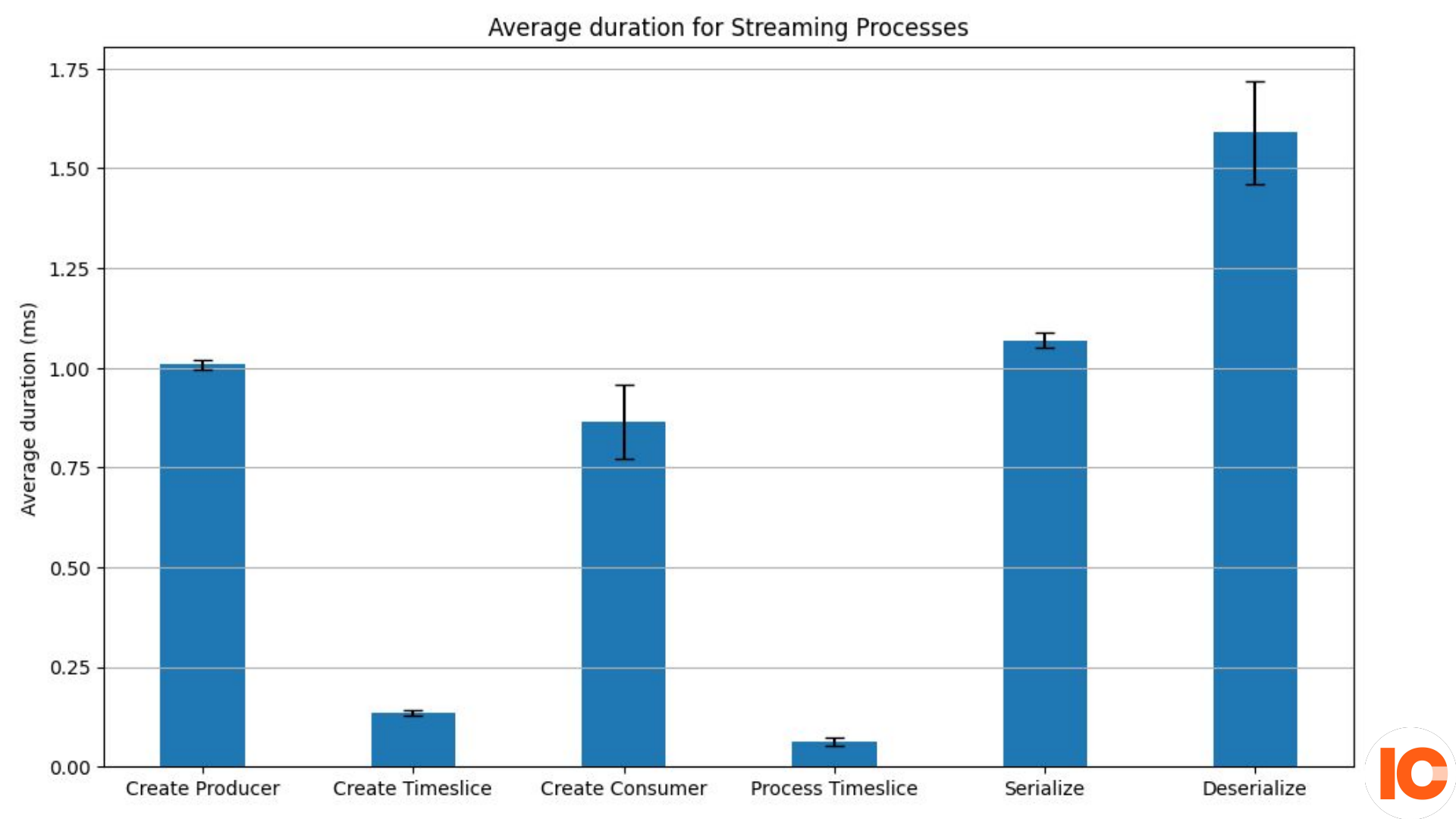
https://github.com/iterorganization/IMAS-Streams

IMAS-Streams

- Much data in an IMAS time slice is static
- Serialize/deserialize unnecessarily repeats

- Split static data and dynamic data
- Only send data arrays for each time slice

- Use case: streaming IMAS data for in-pulse data processing at ITER
- Limitation: array sizes fixed!


Average duration for Streaming Processes
Average duration (ms)
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Create Producer
Create Timeslice
Create Consumer
Process Timeslice
Serialize
Deserialize

https://github.com/iterorganization/IMAS-ITER-Mapping

IMAS-ITER-Mapping

Basic mapping

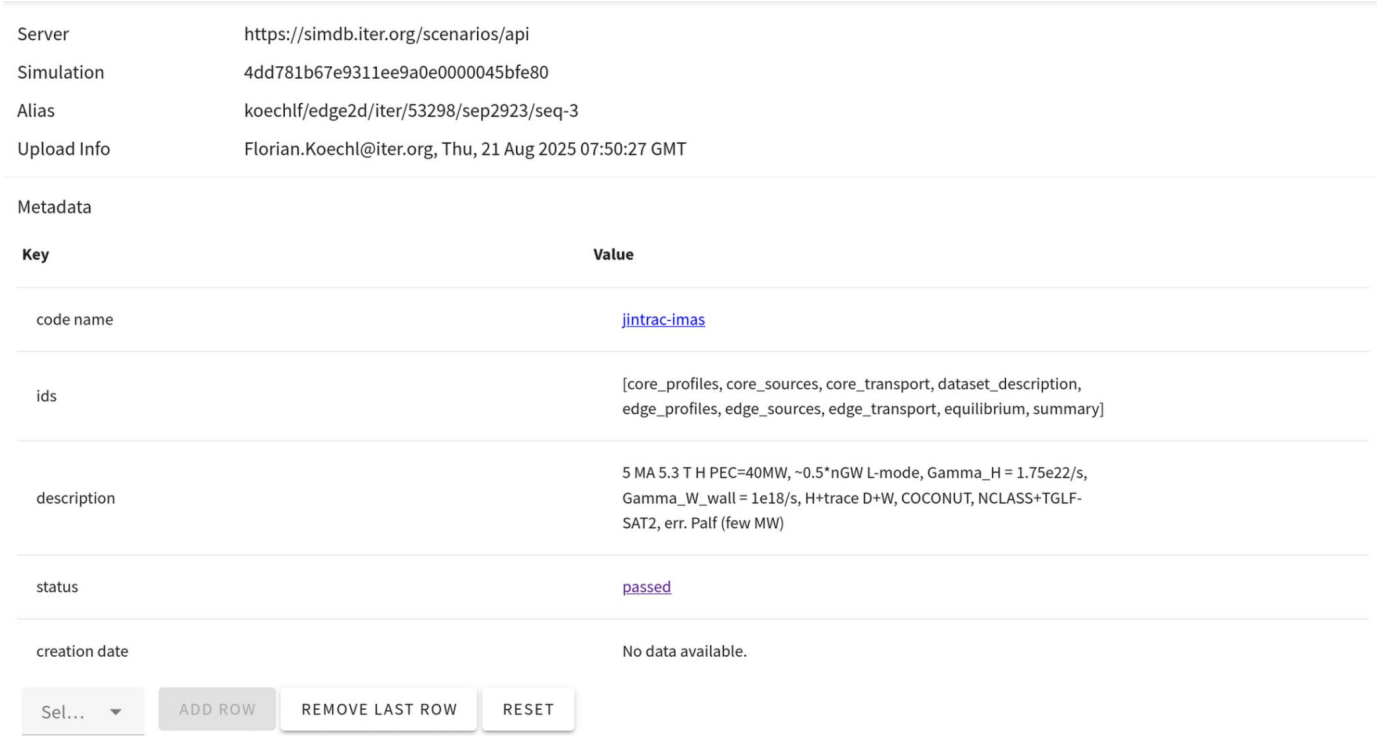
Complex stuff in later steps of pipeline

```yaml
description: Example mapping file for the README
data_dictionary_version: 4.0.0
machine_description_uri: iter_md_magnetics_150100_5.nc
target_ids: magnetics
signals:
  flux_loop:
  - name: 55.AD.00-MSA-1001
    flux/data: AAA-BBBB-CCCC-DDD-EEEE:FF1001-XI [Wb]
    voltage/data: AAA-BBBB-CCCC-DDD-EEEE:FF1001-XP [V]
  - name: 55.AD.00-MSA-1002
    flux/data: AAA-BBBB-CCCC-DDD-EEEE:FF1101-XI [Wb]
    voltage/data: AAA-BBBB-CCCC-DDD-EEEE:FF1101-XP [V]
```

https://github.com/iterorganization/SimDB

SimDB

- Metadata, storage, search for IMAS data
- Validation
- Provenance
- simdb.iter.org



Server	https://simdb.iter.org/scenarios/api
Simulation	4dd781b67e9311ee9a0e0000045bfe80
Alias	koechlf/edge2d/iter/53298/sep2923/seq-3
Upload Info	Florian.Koechl@iter.org, Thu, 21 Aug 2025 07:50:27 GMT

Metadata

Key	Value
code name	jintrac-imas
ids	[core_profiles, core_sources, core_transport, dataset_description, edge_profiles, edge_sources, edge_transport, equilibrium, summary]
description	5 MA 5.3 T H PEC=40MW, ~0.5*nGW L-mode, Gamma_H = 1.75e22/s, Gamma_W_wall = 1e18/s, H+trace D+W, COCONUT, NCLASS+TGLF-SAT2, err. Palf (few MW)
status	passed
creation date	No data available.

Sel... ADD ROW REMOVE LAST ROW RESET

https://imas-python.readthedocs.io/en/stable/netcdf.html

IMAS NetCDF file format

- Tensorized, annotated, standardized
  - S3-compatible
  - Conventions here: https://imas-python.readthedocs.io/en/stable/netcdf/conventions.html#imas-conventions-for-the-netcdf-data-format
- Write, read with IMAS-Python, xarray, fsspec
  - No IMAS-Core writing, slice writing support yet
- (WIP) read natively (without IMAS-Core!) from Fortran, C++, MATLAB

```
import xarray
ds: Dataset = xarray.load_dataset(filename_or_obj="iter_disruption_113112_1.nc", group="equilibrium/0")
```

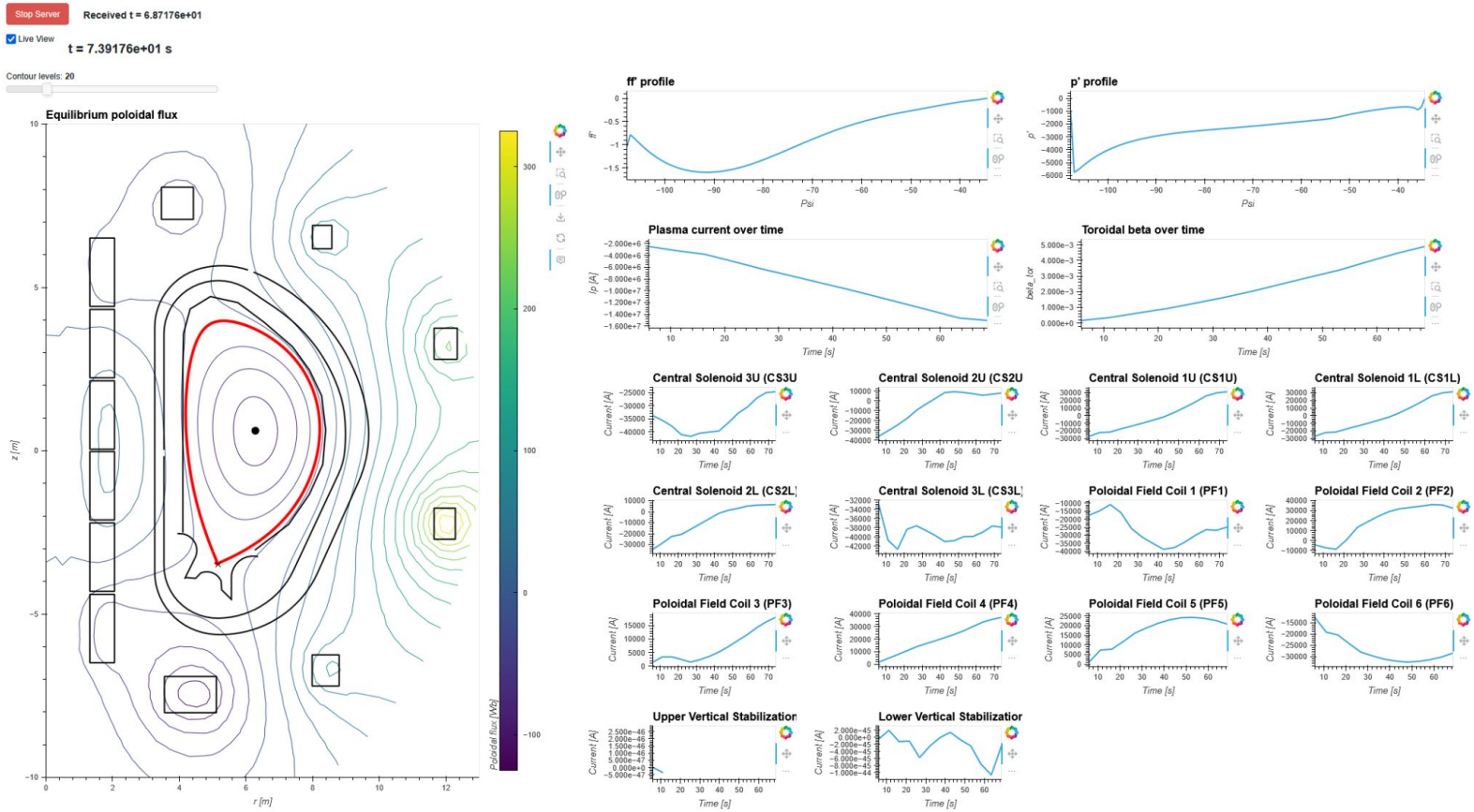
✓ 1.1s

```
ds["time_slice.profiles_2d.psi"]
```

✓ 0.0s

https://imas-muscle3.readthedocs.io/en/latest/

IMAS M3 Visualization actor

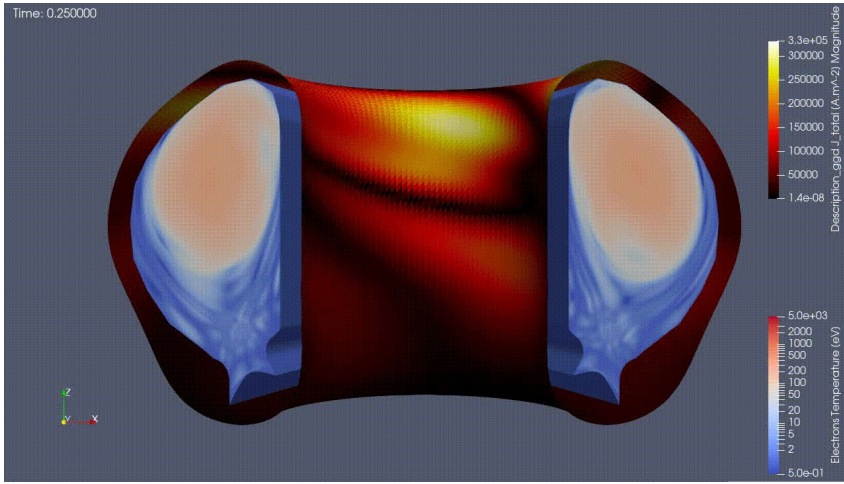


https://imas-paraview.readthedocs.io/en/latest/

IMAS-ParaView

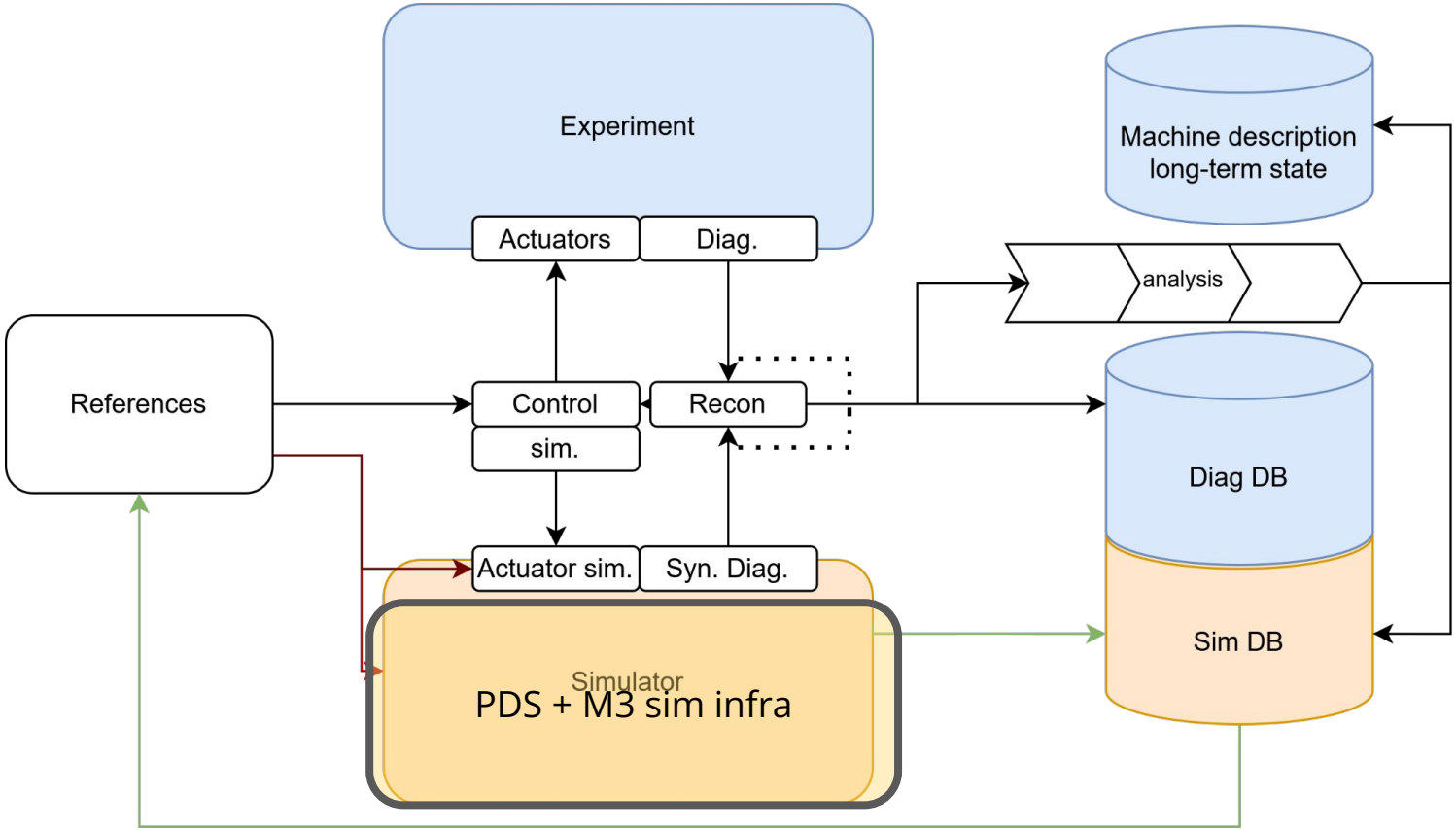
- ParaView plugins for visualizing various IMAS data
  - Generalized Grid Descriptions (GGDs)
  - 1D/2D Profiles
  - Many other Machine Description structures
  - Camera sightlines

- CLI tool available
  - GGD → VTK
  - VTK → GGD



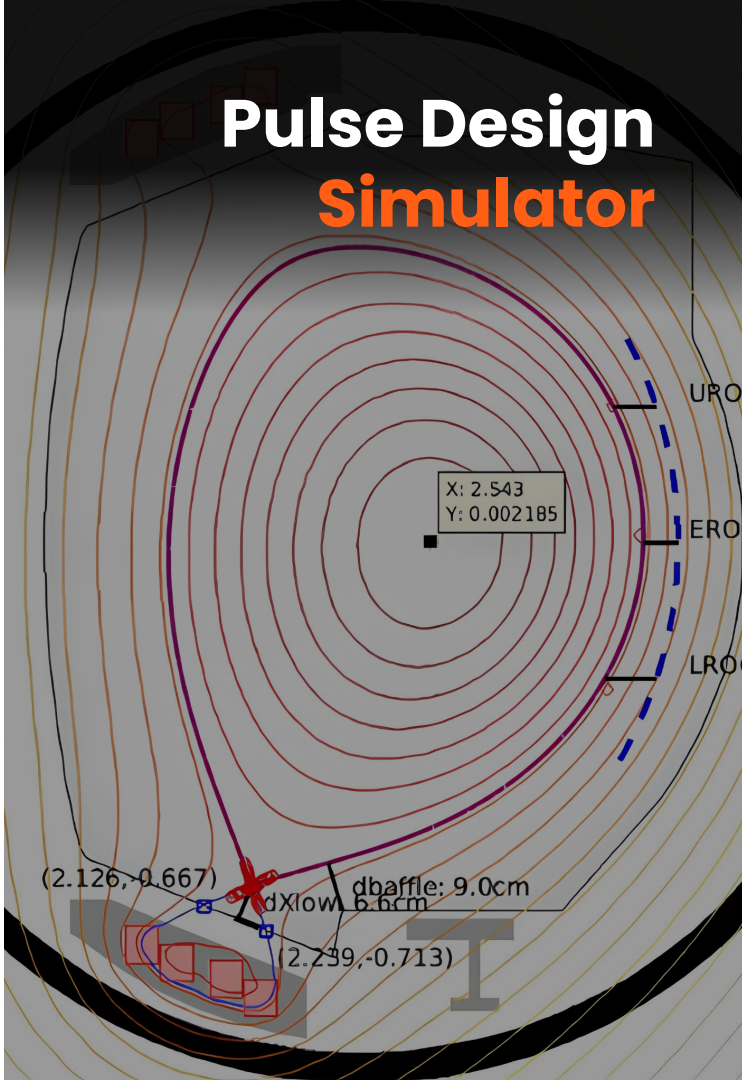
Current in vacuum vessel wall, and electron temperature of a JOREK simulation visualized with IMAS-ParaView

Simulators


Experiment
Actuators
Diag.
Machine description
long-term state
analysis
References
Control
sim.
Recon
Diag DB
Actuator sim.
Syn. Diag.
Sim DB
Simulator
PDS + M3 sim infra

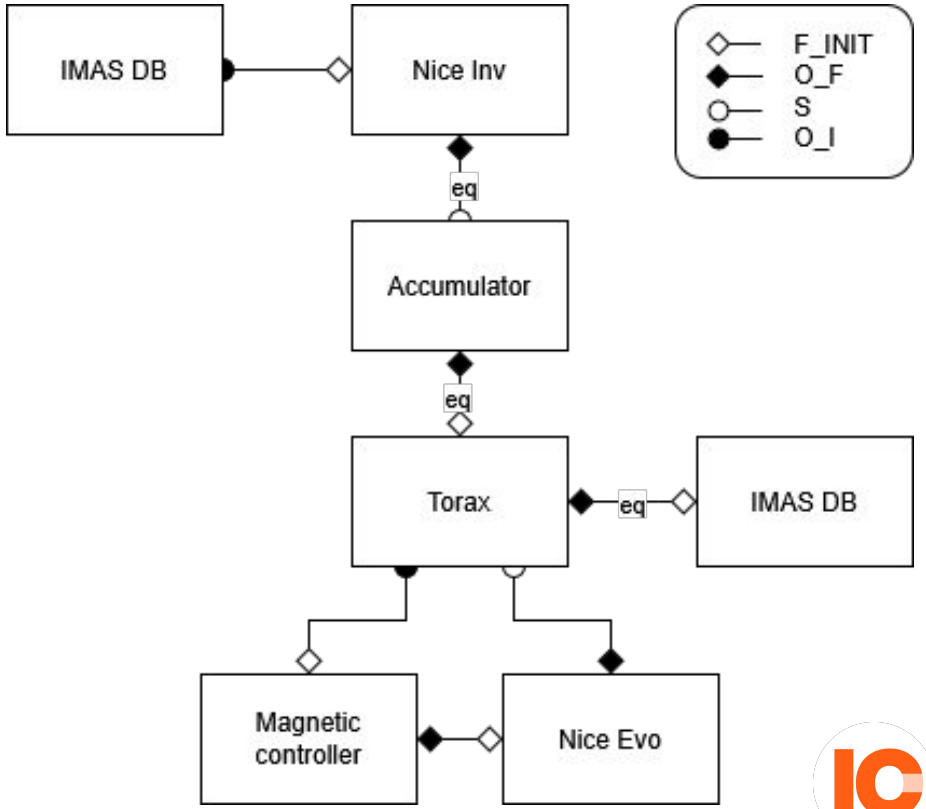
Simulator development

- Combine fast, simple **physics simulators** with **control system**
- Send IMAS data structures between codes
- Design pulse schedule & control targets
- Transport (METIS or TORAX)
- Forward & inverse 2D equilibrium (NICE)
- ITER control system simulator integration



Pulse Design Simulator example workflow

- Inverse + Forward simulation
  - Freeboundary
- Separate models, running individually as actors
  - Sending IMAS messages
- Several configurations relevant



https://muscle3.readthedocs.io/en/latest/

Distributed, multiscale simulation

MUSCLE 3

- Communication library for heterogeneous programs
  - C++, Fortran, C, Python, Matlab/Simulink
  - MPI compatible
- Communication scheme for multiscale modelling
  - Local micro-models (ex: turbulent transport) & more
  - Timescale bridges
- Distributed, globally consistent checkpointing

```yaml
components:
  macro: my.macro_model
  meso:
    ports:
      f_init: boundary_in
      o_i: state_out
      s: state_in
      o_f: boundary_out
    implementation: my.meso_model
    multiplicity: 5
  micro:
    implementation: my.micro_model
    multiplicity: [5, 10]
```

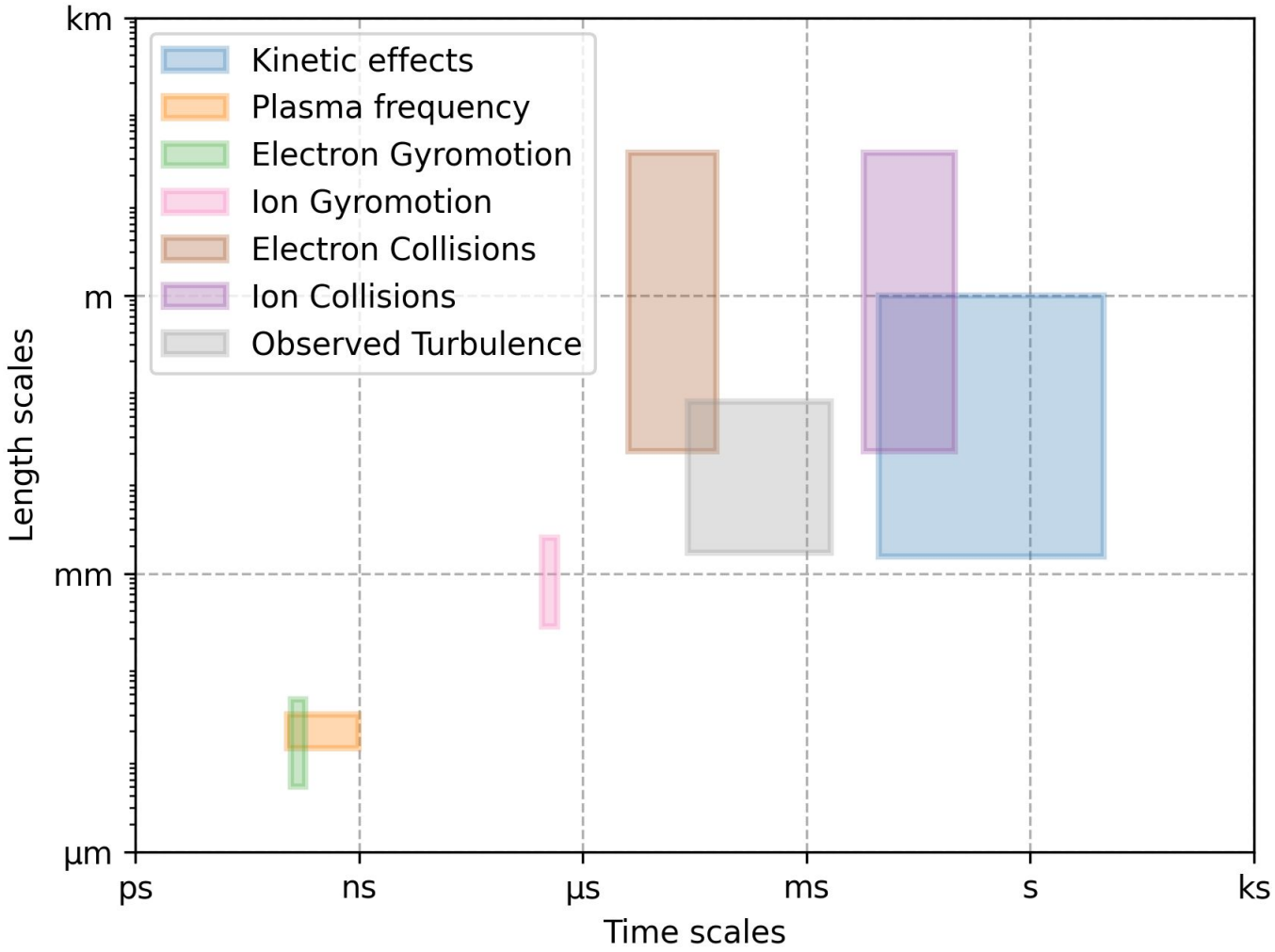
Computational Science Lab

netherlands eScience center

Multiscale

Physical process scales (kinetics only, simplified)

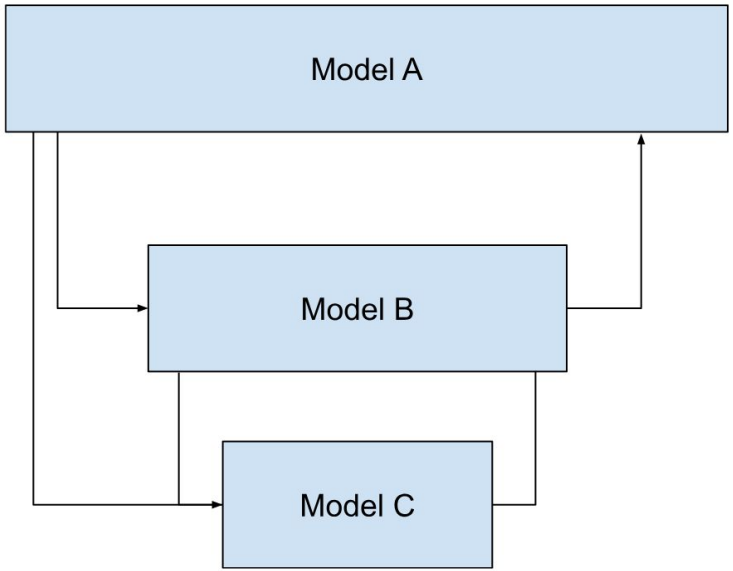
Avoid computing ‘too much’; use appropriate resolution in time and space



https://muscle3.readthedocs.io/en/latest/

MUSCLE3 developments

- Subworkflows
  - Treat an entire workflow like a model
- Timelines
  - Annotate ports with a timeline
  - Repeat messages as needed
  - Avoid pass-through f_init messages
- General time bridge



https://muscle3.readthedocs.io/en/latest/

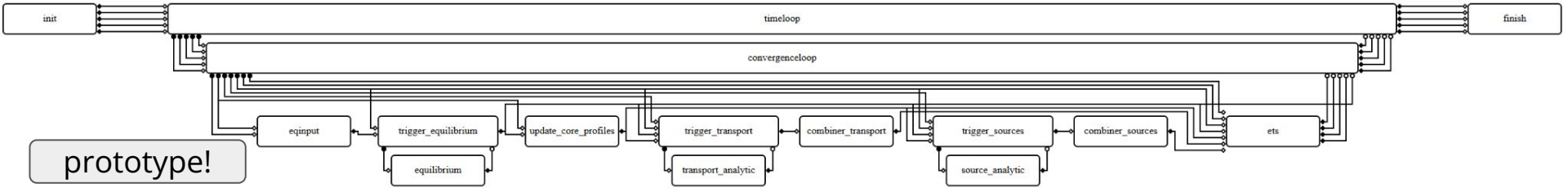
MUSCLE3 developments

- Optional components
  - Write connections as usual
  - Only start if implementation defined
- Pytest helpers
- Dynamic ports
  - Useful for generic components

https://muscle3.readthedocs.io/en/latest/

MUSCLE3 developments

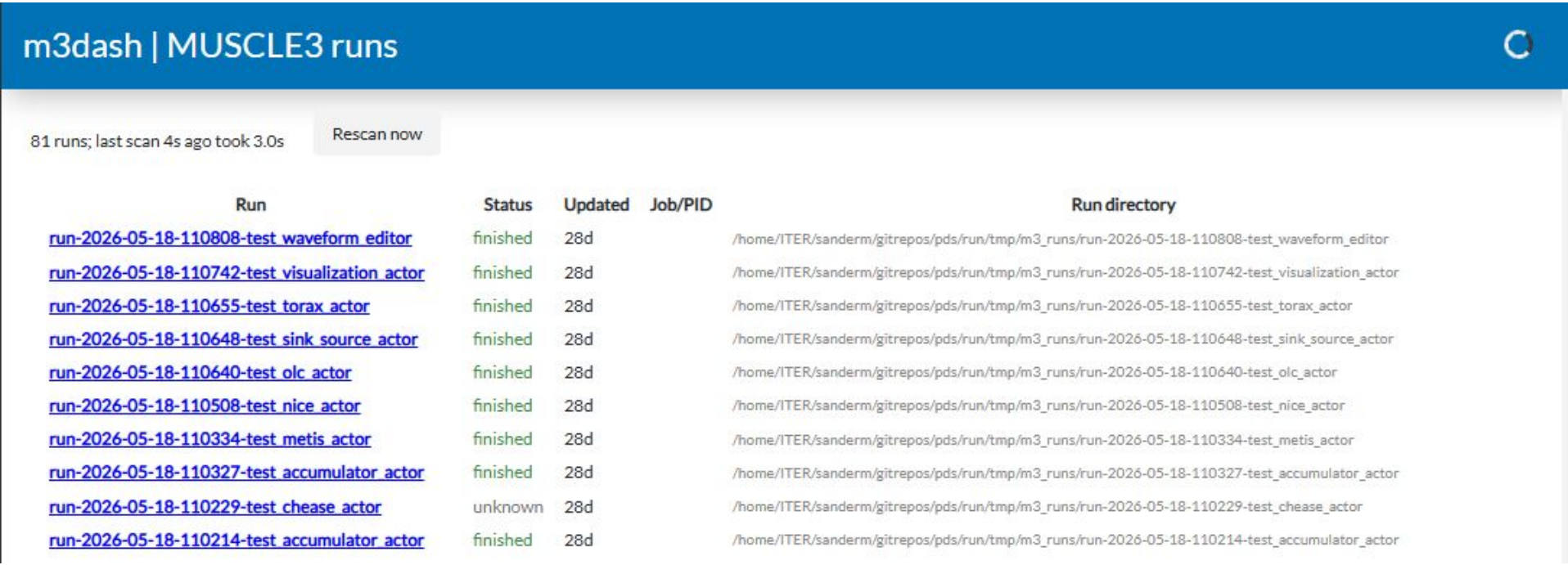
- Workflow documentation
  - Rules-based workflow graph rendering
- Model discovery
  - Python plugin mechanism
  - YMMSL_PATH autodiscovery
  - Useful with env modules



prototype!

https://github.com/daanvanvugt/muscle3-dashboard

MUSCLE3 dashboard



Run on HPC system, access through port forward or ssh (exec) bridge

MUSCLE3 dashboard

Component status

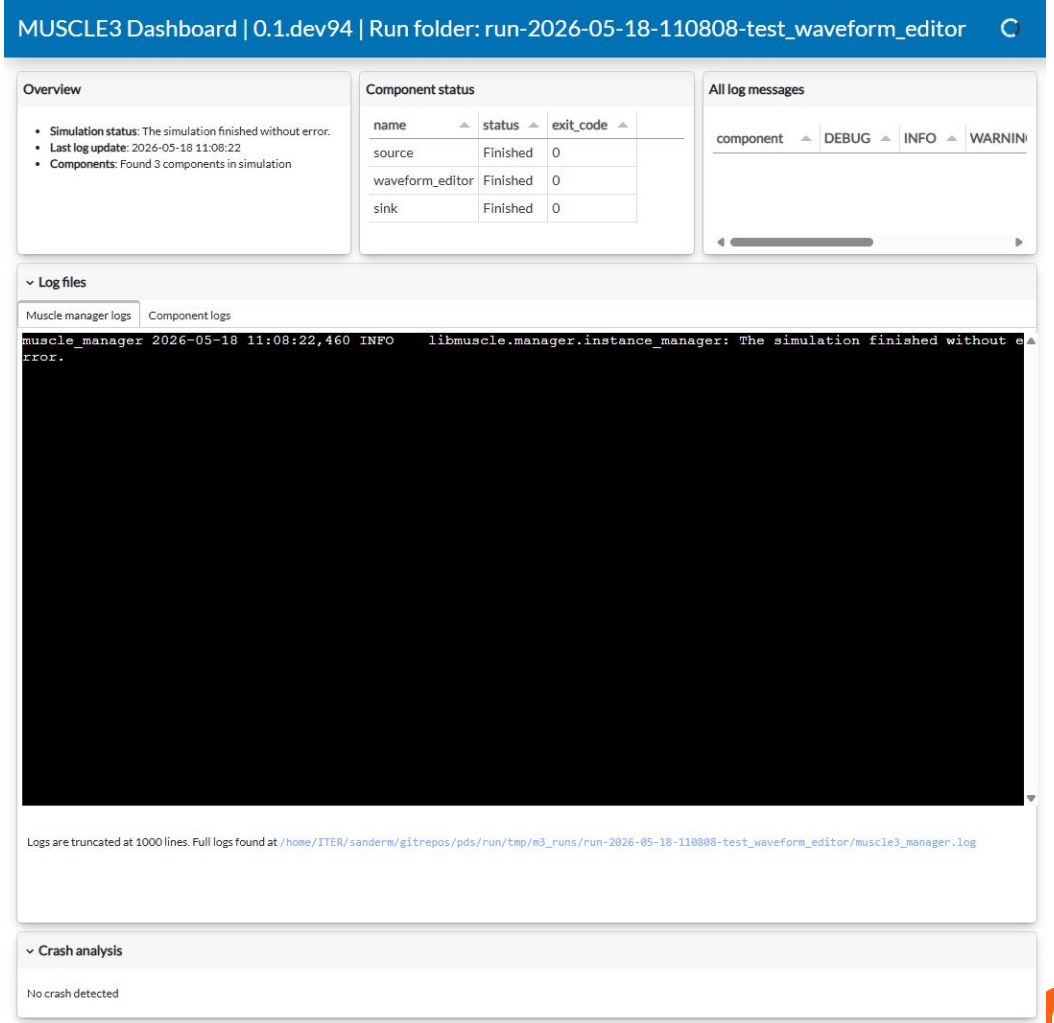
Proxy to analysis pages (http)

Crash analysis

Log viewing

Diagram to be included

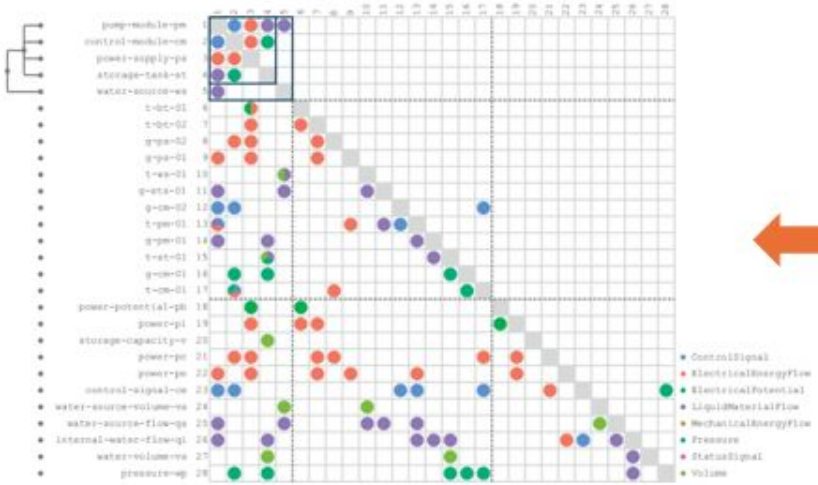
What would you like to see?



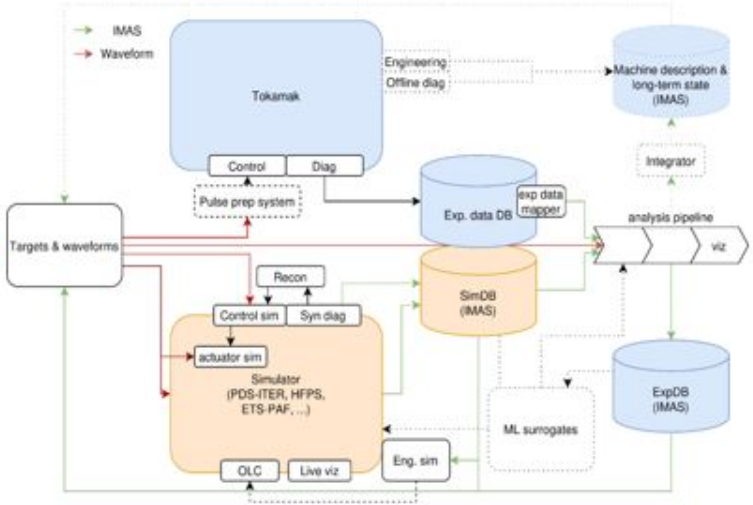
Systems Engineering for Physics Intensive Systems

The goal: Structured development of multi-physics simulations for early system-level requirement verification

Systems Engineering



Physics Modeling



ratioCASE IGNITION COMPUTING DIFFER TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Thanks to Olivier Hoenen, Simon McIntosh, Mireille Schneider, Francesca Poli, Paulo Abreu, Simon Pinches, Richard Pitts, ... @ ITER

Thanks to IC team (on the right)

ℹ️ **Would you like to collaborate? Let's talk!**

ignitioncomputing.com
dvanvugt@ignitioncomputing.com