The Open Standard for Particle-Mesh Data

Franz Poeschel (CASUS/HZDR)

9th Annual MT Meeting @KIT Karlsruhe
Data Management and Analysis Session

On behalf of the openPMD Community incl. content from
Axel Huebl (LBNL), Lipeng Wan (GSU), Remi Lehe (LBNL)
Norbert Podhorszki (ORNL), Junmin Gu (LBNL),
Maxence Thévenet (DESY), Erik Schnetter (PITP),

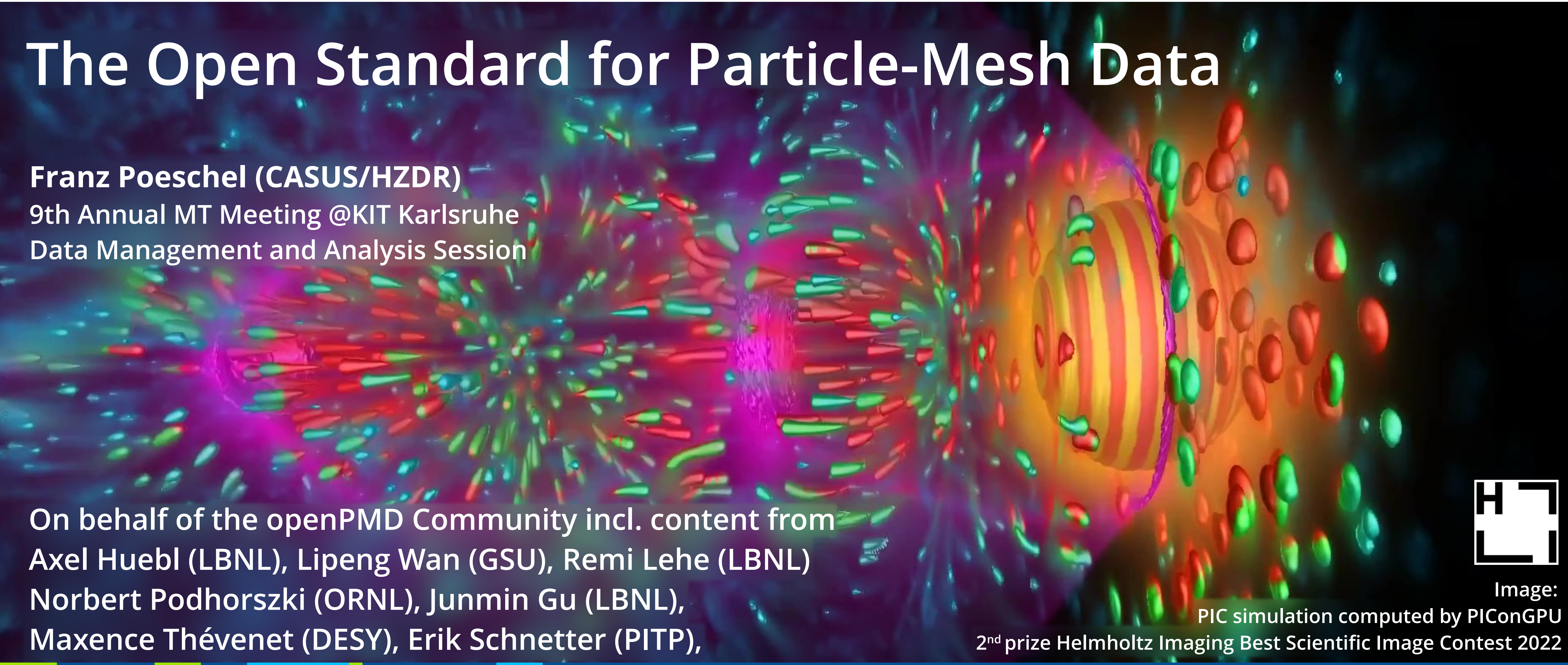
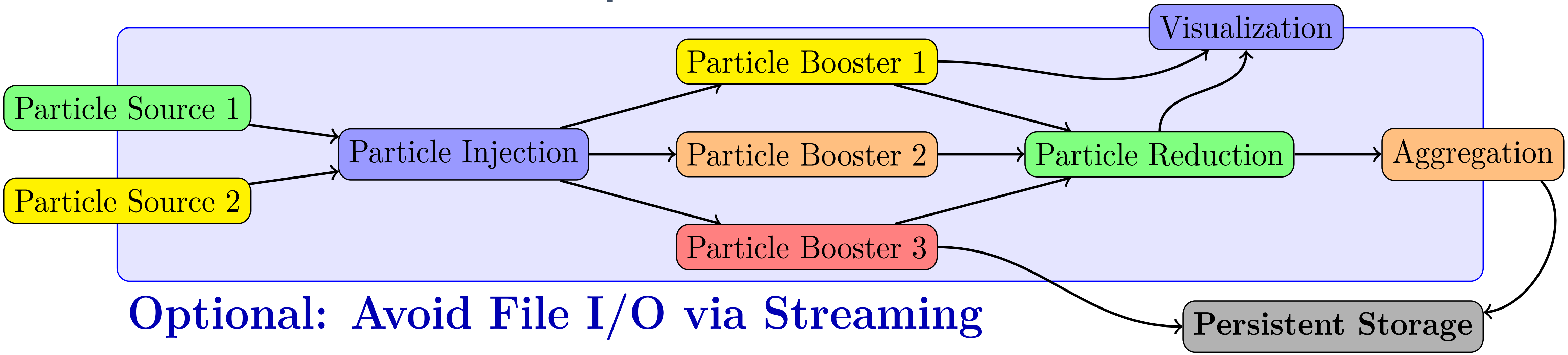


Image:
PIC simulation computed by PIConGPU
2nd prize Helmholtz Imaging Best Scientific Image Contest 2022

Heterogeneity through Standardized Data

Scientific workflows are complex:

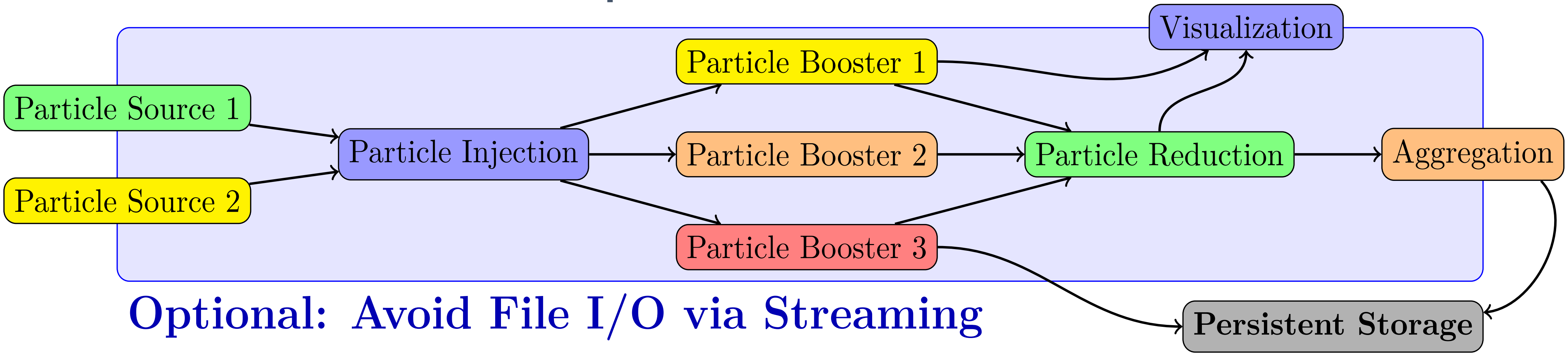


- need to span different **time** and **length scales**
- scientific modeling requires **multiple codes**, collaborating in a **data processing pipeline**
- **bridge heterogeneous models** by standardization of data

Axel Huebl et al. “openPMD: A meta data standard for particle and mesh based data”. 2015. doi: 10.5281/zenodo.591699. url: https://openPMD.org
Franz Poeschel et al. “Transitioning from file-based HPC workflows to streaming data pipelines with openPMD and ADIOS2“. 2021. doi:10.1007/978-3-030-96498-6_6

Heterogeneity through Standardized Data

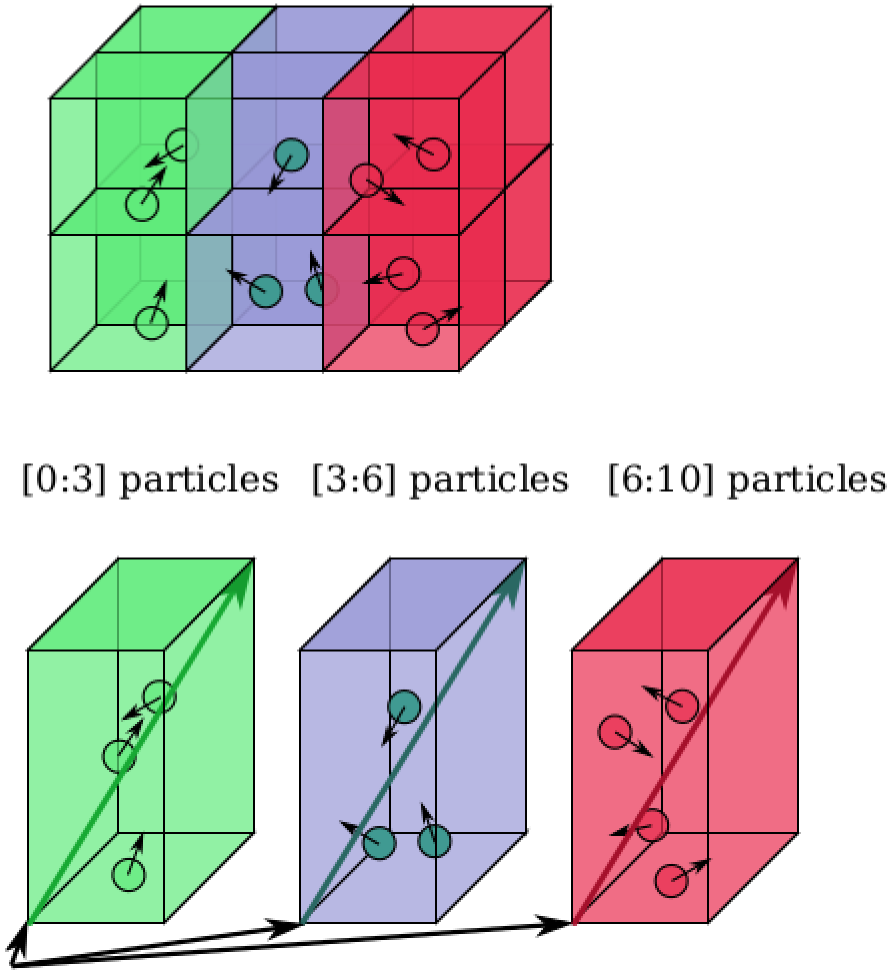
Scientific workflows are complex:



→ openPMD standard
for **p**article-**m**esh **d**ata
as communication layer

Axel Huebl et al. "openPMD: A meta data standard for particle and mesh based data". 2015. doi: 10.5281/zenodo.591699. url: https://openPMD.org
Franz Poeschel et al. "Transitioning from file-based HPC workflows to streaming data pipelines with openPMD and ADIOS2". 2021. doi:10.1007/978-3-030-96498-6_6

What is particle-mesh data?



Mesh

n-dimensional space,
divided into discrete cells

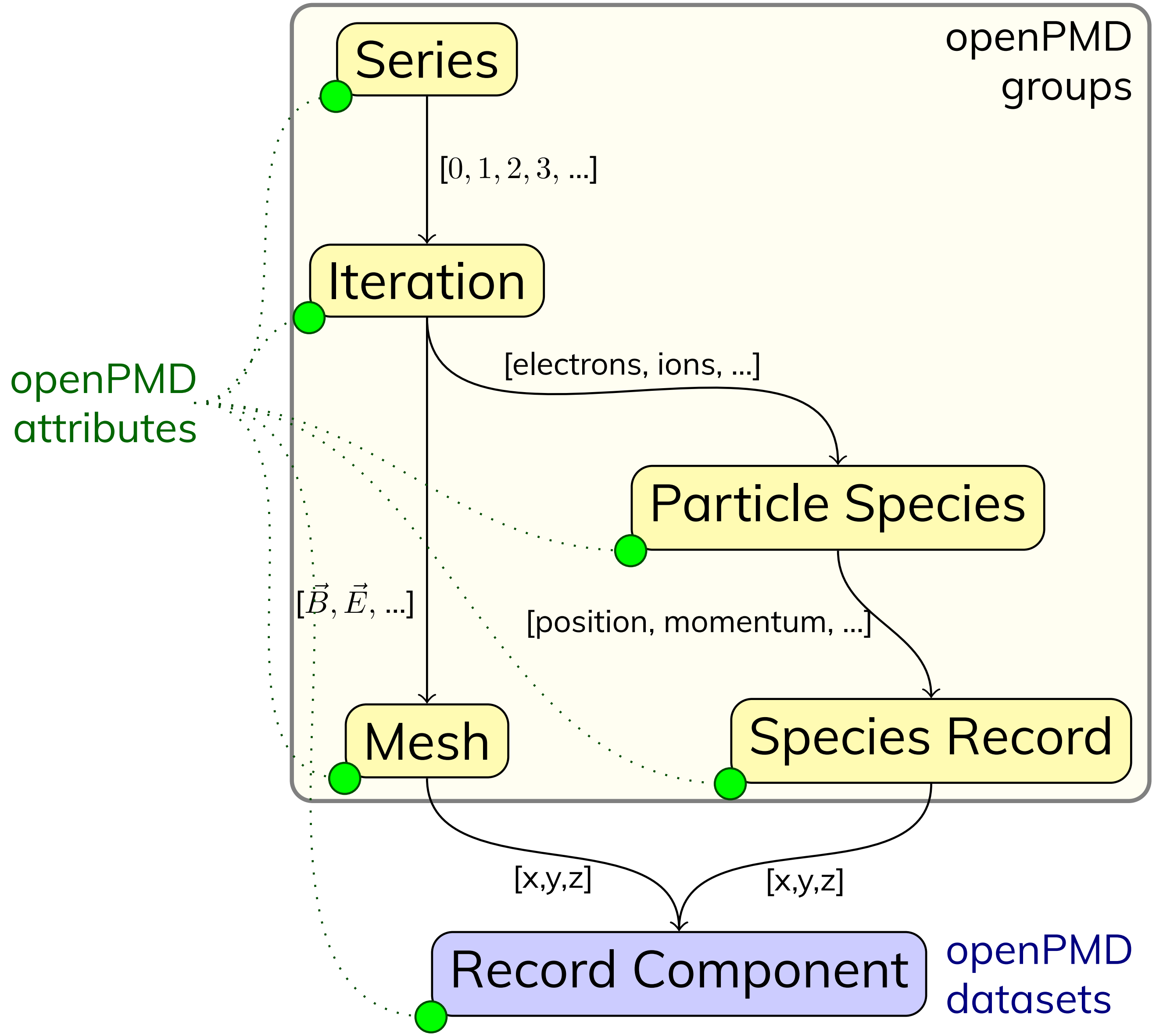
- e.g. temperature:
  store a scalar number per cell
- e.g. electrical fields:
  store a 3D vector per cell

Particles

A list of discrete objects,
located on the mesh

- for each particle: list its position
- optionally: list charge, weight, …

openPMD hierarchy

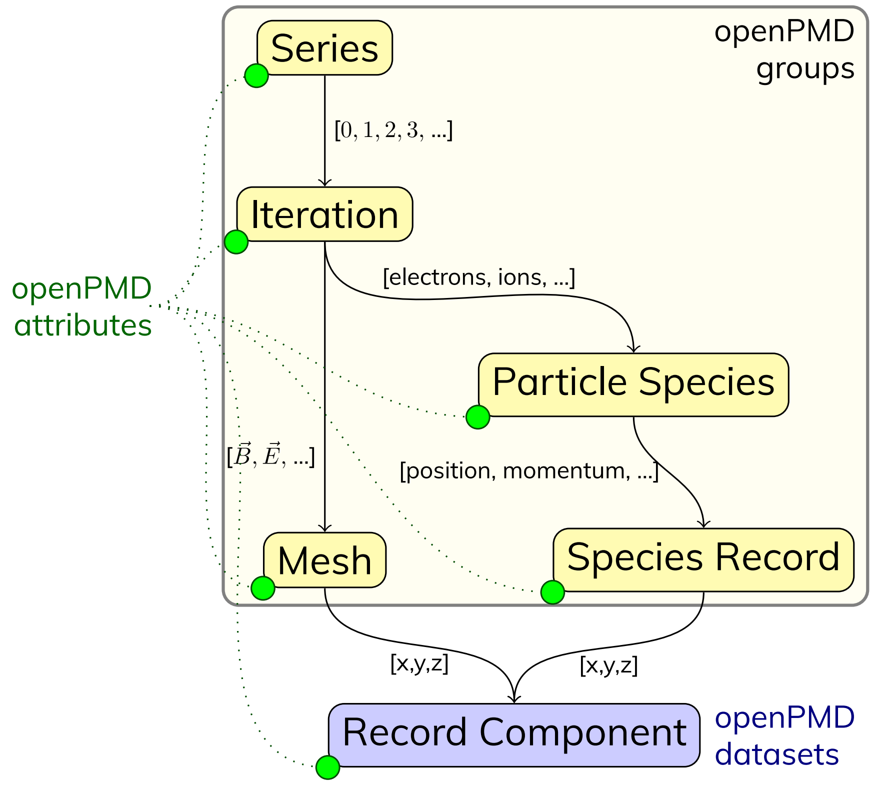


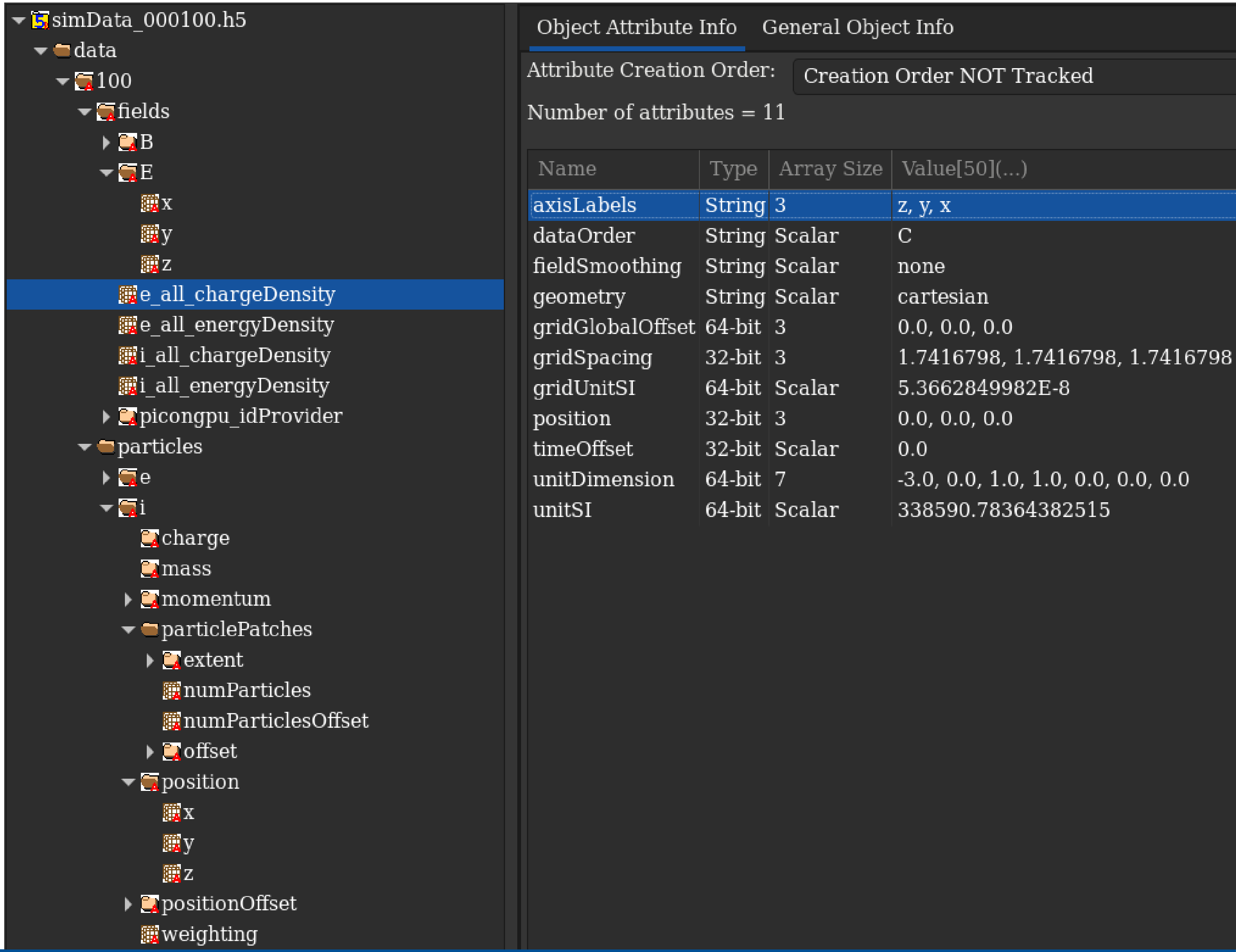
- **Structure** for series & snapshots encoded as either:
  - **files** (one file per iteration)
  - **groups** (reuse files)
  - **variables** (reuse files & variables in ADIOS2)

- Records for **physical observables**
  constants, mixed precision, complex numbers

- **Attributes:** unit conversion, description, relations, mesh geometry, authors, env. info, ...

Example dataset: HDF5 backend





Sample data
created with PIConGPU

Example dataset: ADIOS2 backend

```
float     /data/50/fields/E/x                           {128, 128, 128}
float     /data/50/fields/E/y                           {128, 128, 128}
float     /data/50/fields/E/z                           {128, 128, 128}
float     /data/50/particles/e/position/x               {50053105}
float     /data/50/particles/e/position/y               {50053105}
float     /data/50/particles/e/position/z               {50053105}
int32_t   /data/50/particles/e/positionOffset/x         {50053105}
int32_t   /data/50/particles/e/positionOffset/y         {50053105}
int32_t   /data/50/particles/e/positionOffset/z         {50053105}
```

Hierarchical data organization

n-dim. datasets for heavyweight data

```
string    /data/50/fields/E/axisLabels                  attr   = {"z", "y", "x"}
string    /data/50/fields/E/dataOrder                   attr   = "C"
string    /data/50/fields/E/fieldSmoothing              attr   = "none"
string    /data/50/fields/E/geometry                    attr   = "cartesian"
double    /data/50/fields/E/gridGlobalOffset            attr   = {0, 0, 0}
float     /data/50/fields/E/gridSpacing                 attr   = {1.74168, 1.74168, 1.74168}
double    /data/50/fields/E/gridUnitSI                  attr   = 5.36628e-08
float     /data/50/fields/E/timeOffset                  attr   = 0
double    /data/50/fields/E/unitDimension               attr   = {1, 1, -3, -1, 0, 0, 0}
float     /data/50/fields/E/x/position                  attr   = {0.5, 0, 0}
double    /data/50/fields/E/x/unitSI                    attr   = 9.5224e+12
float     /data/50/fields/E/y/position                  attr   = {0, 0.5, 0}
double    /data/50/fields/E/y/unitSI                    attr   = 9.5224e+12
float     /data/50/fields/E/z/position                  attr   = {0, 0, 0.5}
double    /data/50/fields/E/z/unitSI                    attr   = 9.5224e+12
```

Attributes for self-description

openPMD – a FAIR standard

Findable: Standardized metadata to identify the data producer

```
string    /author              attr   = "franz"
string    /software            attr   = "PIConGPU"
string    /softwareVersion     attr   = "0.5.0-dev"
```

Accessible: Open standard, implementable in various formats

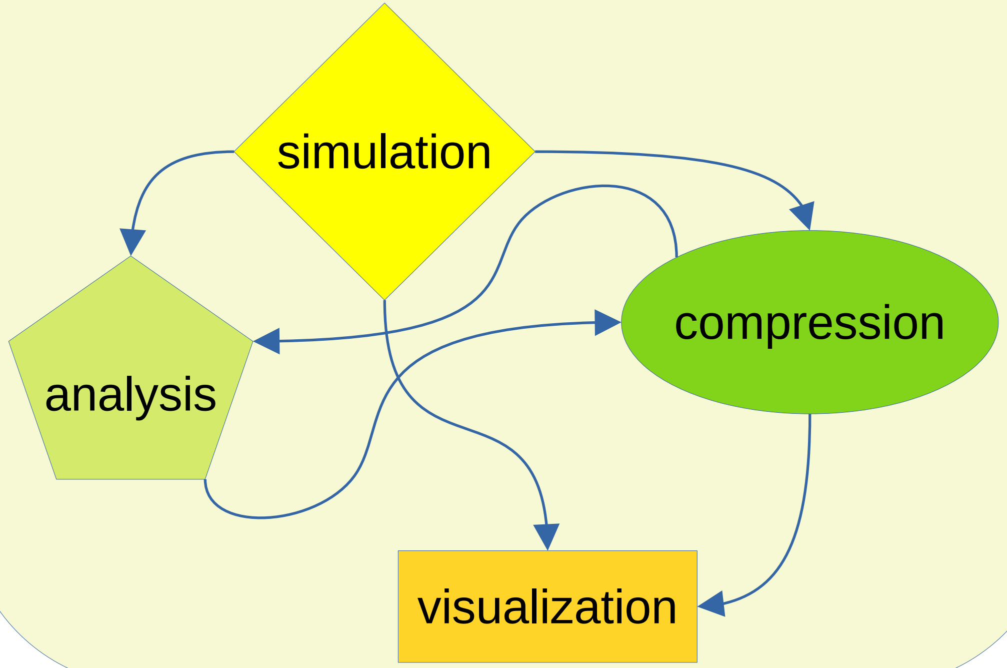
*currently implemented,
but not limited to

"The FAIR Guiding Principles for scientific data management and stewardship" (Mark D. Wilkinson et al.)

openPMD – a FAIR standard

Interoperable:

Data exchange spans applications, platforms and teams



Reusable:

Rich and standardized description for physical quantities

Name	Value
axisLabels	[b'z' b'y' b'x']
dataOrder	b'C'
fieldSmoothing	b'none'
geometry	b'cartesian'
gridGlobalOffset	[0. 0. 0.]
gridSpacing	[4.252342 1.0630856 4.252342]
gridUnitSI	4.1671151662e-08
position	[0. 0. 0.]
timeOffset	0.0
unitDimension	[-3. 0. 1. 1. 0. 0. 0.]
unitSI	15399437.98944343

“The FAIR Guiding Principles for scientific data management and stewardship” (Mark D. Wilkinson et al.)

Ecosystem & Community

openPMD powered Projects and Users

Documents:

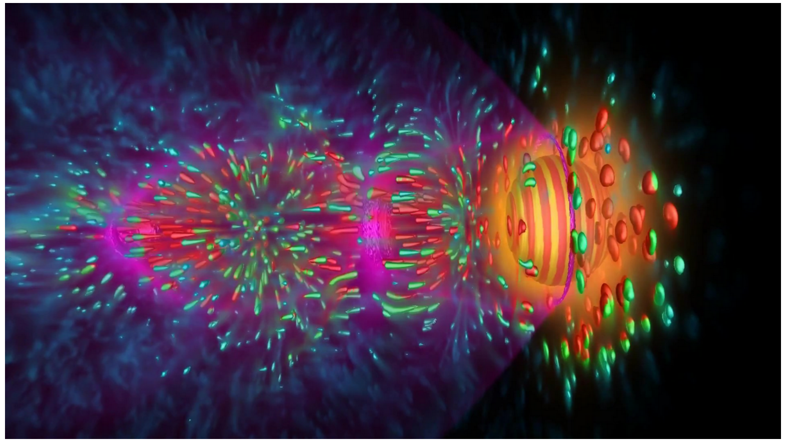
- **openPMD standard** (1.0.0, 1.0.1, 1.1.0)
  the underlying file markup and definition
  A Huebl et al., doi: 10.5281/zenodo.33624

Language Binding:

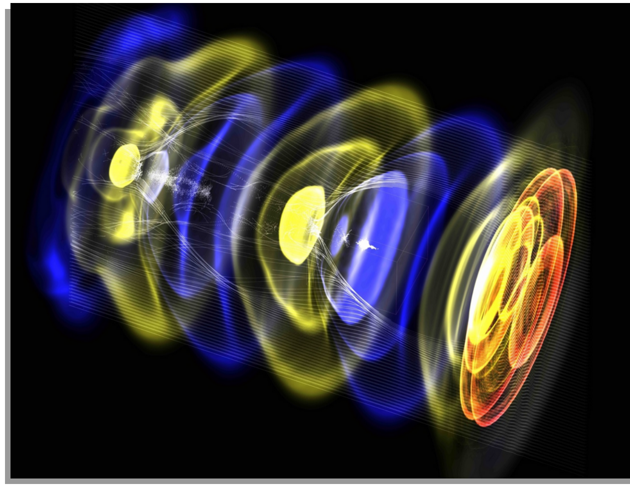
- **openPMD-api** (HZDR, CASUS, LBNL)
  reference API for openPMD data handling
  maintainers: A Huebl, J Gu, F Poeschel et al.

Scientific Simulations:

- **PIConGPU** (HZDR)
  electro-dynamic particle-in-cell code
  maintainers: R Widera, S Bastrakov, A Debus et al.
- **WarpX** (LBNL, LLNL)
  electro-dynamic/static particle-in-cell code
  maintainers: JL Vay, D Grote, R Lehe, A Huebl et al.
- **FBPIC** (LBNL, DESY)
  spectral, fourier-bessel particle-in-cell code
  maintainers: R Lehe, M Kirchen et al.
- **SimEx Platform** (EUCALL, European XFEL)
  simulation of advanced photon experiments
  maintainer: C Fortmann-Grote



PIConGPU+ISAAC on Summit
2nd prize Helmholtz Imaging
Best Scientific Image Contest 2022
Image credit: Felix Meyer/HZDR



WarpX
PI: Jean-Luc Vay/LBNL

see also: https://github.com/openPMD/openPMD-projects

openPMD powered Projects and Users

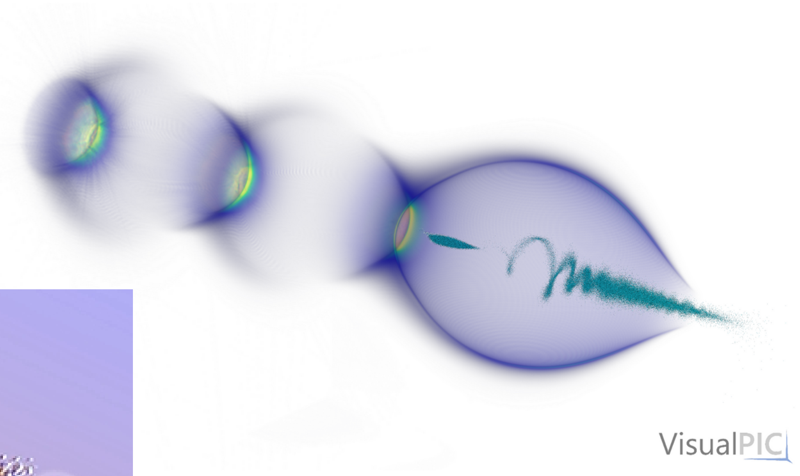
Documents:

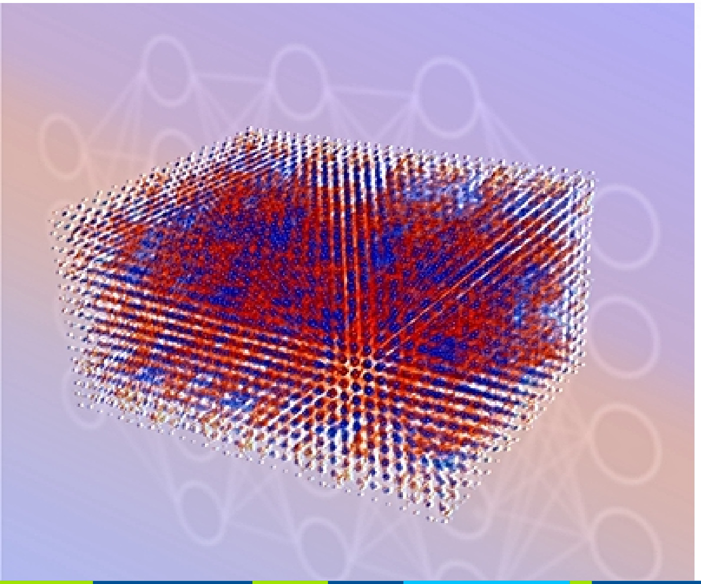
- **openPMD standard** (1.0.0, 1.0.1, 1.1.0)
  the underlying file markup and definition
  A Huebl et al., doi: 10.5281/zenodo.33624

Language Binding:

- **openPMD-api** (HZDR, CASUS, LBNL)
  reference API for openPMD data handling
  maintainers: A Huebl, J Gu, F Poeschel et al.

HiPACE++ → VisualPIC
Credit: M.Thévenet
& A. Ferran Pousa (DESY)





MALA → ParaView
Credit: A. Cangi (CASUS)

- **Wake-T** (DESY)
  fast particle-tracking code for plasma-based accelerators
  maintainer: A Ferran Pousa
- **HiPACE++** (DESY, LBNL)
  3D GPU-capable quasi-static PIC code for plasma accel.
  maintainers: M Thevenet, S Diederichs, A Huebl
- **Bmad** (Cornell)
  library for charged-particle dynamics simulations
  maintainers: D Sagan et al.
- **MALA** (CASUS, SNL)
  ML models that replace DFT calculations in materials science
  maintainers: Attila Cangi & Sivasankaran Rajamanickam
- and more...

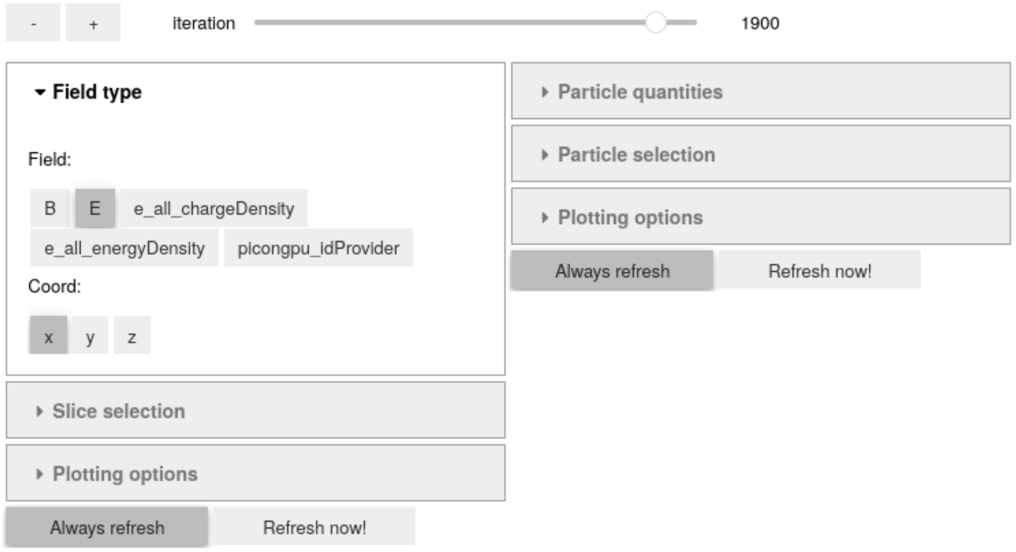
see also: https://github.com/openPMD/openPMD-projects

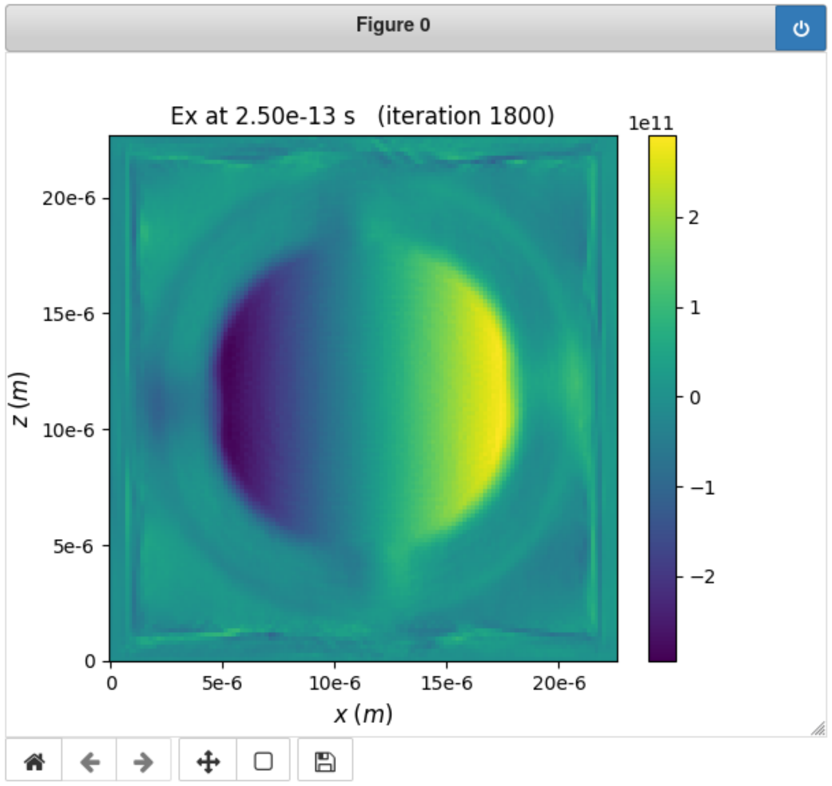
Analysis and Visualization

In [1]:
```python
import numpy as np
%matplotlib notebook
# or `%matplotlib inline` for non-interactive plots
# or `%matplotlib widget` when using JupyterLab (github.com/matplotlib/jupyter-matplotlib)
import matplotlib.pyplot as plt
from openpmd_viewer import OpenPMDTimeSeries
```

In [2]:
```python
# Replace the string below, to point to your data
ts = OpenPMDTimeSeries('/home/franzpoeschel/singularity_build/pic_run/openPMD')
```

In [3]:
```python
# Interactive GUI
ts.slider()
```

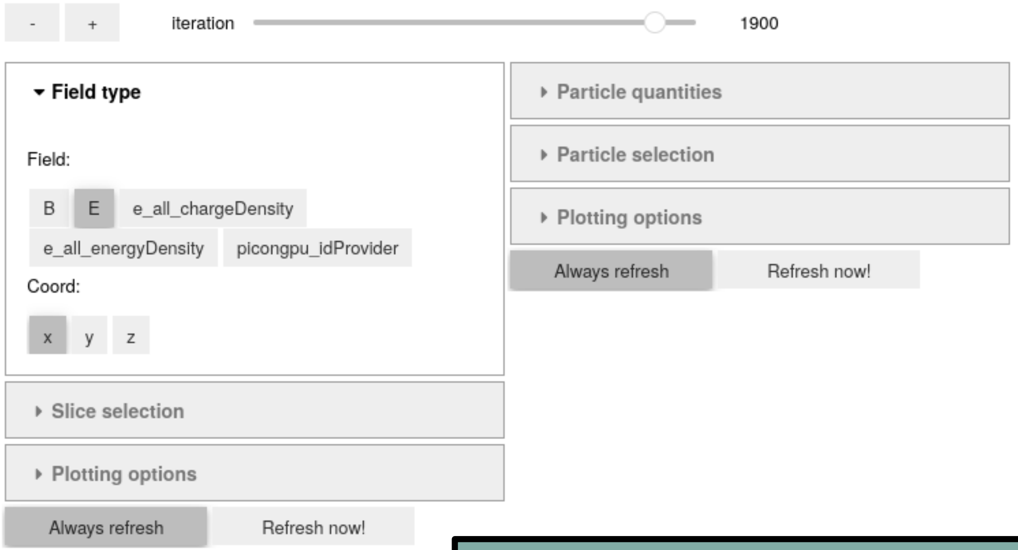


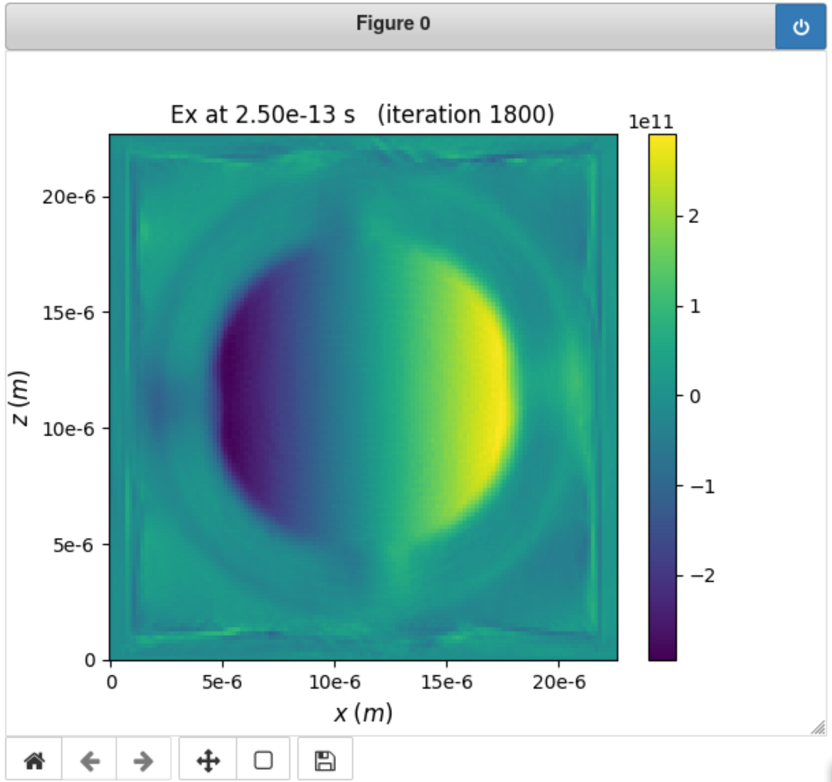


openPMD/openPMD-viewer

Analysis and Visualization

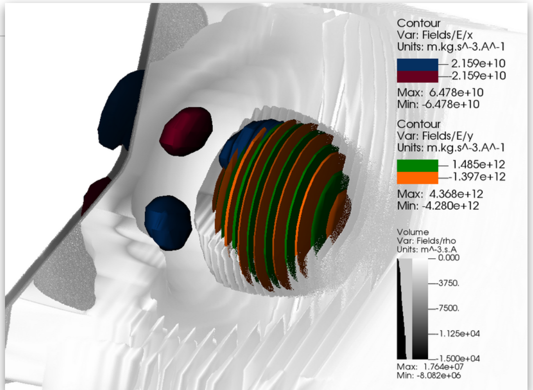
```
In [1]: import numpy as np
        %matplotlib notebook
        # or `%matplotlib inline` for non-interactive plots
        # or `%matplotlib widget` when using JupyterLab (github.com/matplotlib/jupyter-matplotlib)
        import matplotlib.pyplot as plt
        from openpmd_viewer import OpenPMDTimeSeries

In [2]: # Replace the string below, to point to your data
        ts = OpenPMDTimeSeries('/home/franzpoeschel/singularity_build/pic_run/openPMD')

In [3]: # Interactive GUI
        ts.slider()
```





openPMD/openPMD-viewer

Standardization of data
→ integration into modern scientific compute workflows

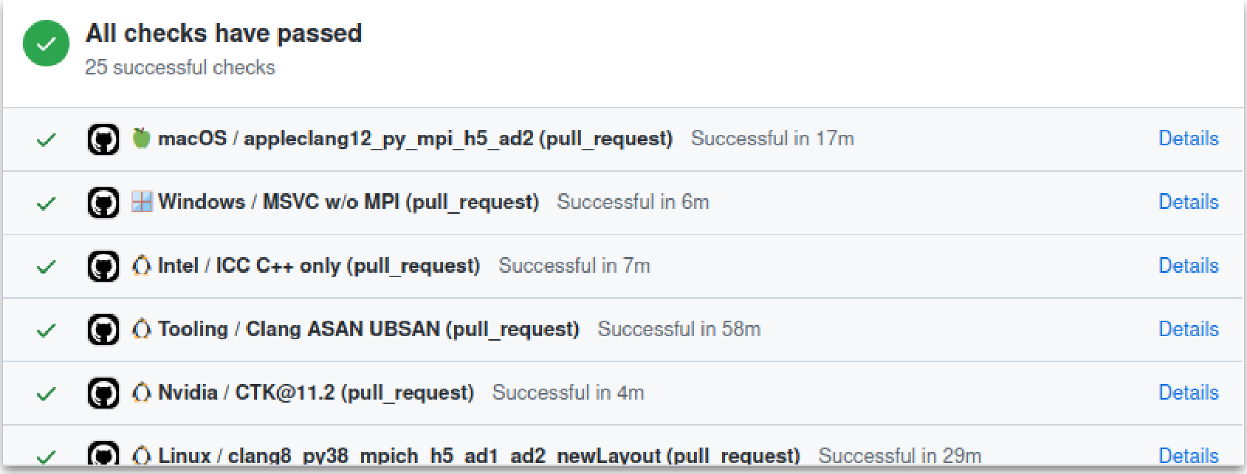


Reference Implementation in C++ & Bindings: Python and Julia

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING

Online Documentation:
openpmd-api.readthedocs.io

Open-Source Development & Tests:
github.com/openPMD/openPMD-api



Rapid and easy installation on any platform:

```
python3 -m pip install
        openpmd-api
```

```
brew tap openpmd/openpmd
brew install openpmd-api
```

```
cmake -S . -B build
cmake --build build
      --target install
```

```
conda install
    -c conda-forge
    openpmd-api
```

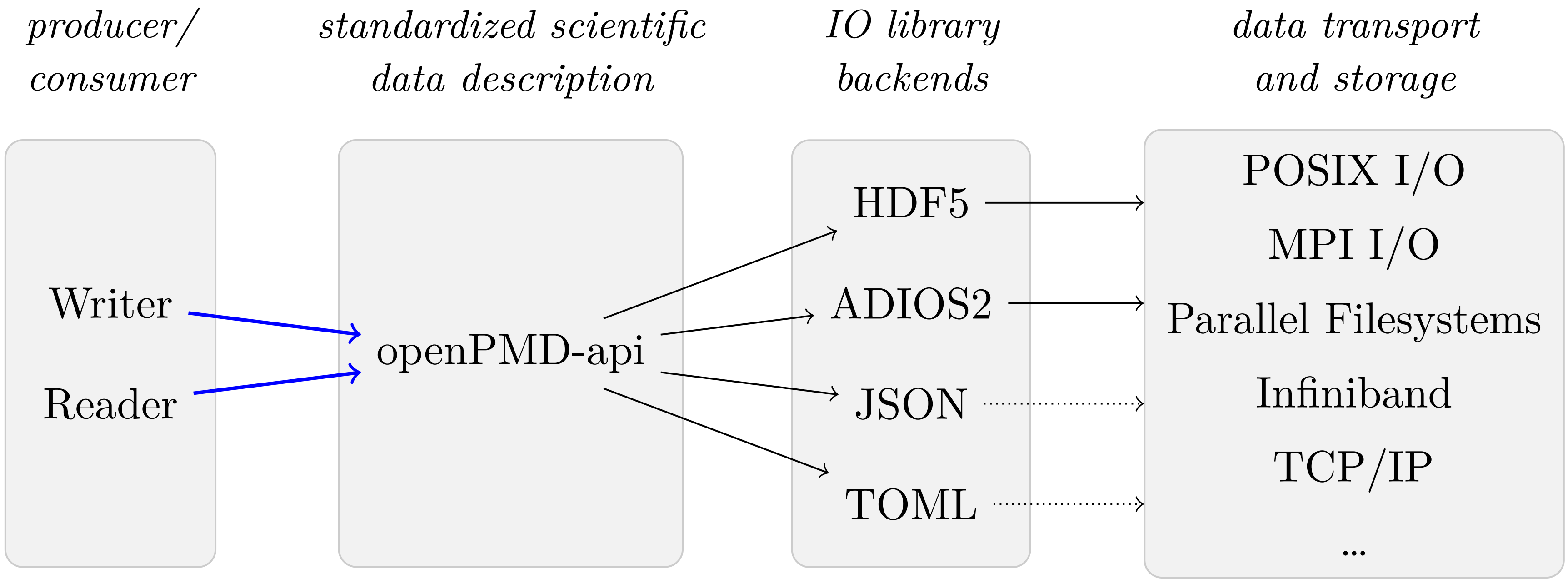
```
spack install
      openpmd-api
```

E4S

```
module load openpmd-api
```

A Huebl, F Poeschel, F Koller, J Gu, et al.
"openPMD-api: C++ & Python API for Scientific I/O with openPMD" (2018) DOI:10.14278/rodare.27

openPMD-api – open stack for scientific I/O




```python
import openpmd_api as io

# pick backend by filename extension
series = io.Series("simOutput.h5",   io.Access.create)
series = io.Series("simOutput.bp",   io.Access.create)
series = io.Series("simOutput.sst",  io.Access.create)
series = io.Series("simOutput.json", io.Access.create)
```

- MPI support at all levels
- Implemented in C++17
- Bindings in C++17, Python and (dev version only) Julia
- Specify backend at runtime:
  I/O library, transport, compression, streaming, aggregation, ...

The **openPMD standard** is co-authored by Axel Huebl, Rémi Lehe, Jean-Luc Vay, David P. Grote, Ivo F. Sbalzarini, Stephan Kuschel, David Sagan, Frédéric Pérez, Fabian Koller, Franz Poeschel, Carsten Fortmann-Grote, Ángel Ferran Pousa, Juncheng E, Maxence Thévenet, and Michael Bussmann.

The authors are thankful for the **community contributions** to libraries, software ecosystem, user support, review and integrations. Particularly, thank you to Yaser Afshar, Lígia Diana Amorim, James Amundson, Weiming An, Igor Andriyash, Ksenia Bastrakova, Jean Luca Bez, Richard Briggs, Heiko Burau, Jong Choi, Ray Donnelly, Dmitry Ganyushin, Marco Garten, Lixin Ge, Berk Geveci, Daniel Grassinger, Alexander Grund, Junmin Gu, Marc W. Guetg, Ulrik Günther, Sören Jalas, Manuel Kirchen, John Kirkham, Scott Klasky, Noah Klemm, Fabian Koller, Mathieu Lobet, Christopher Mayes, Ritiek Malhotra, Paweł Ordyna, Richard Pausch, Norbert Podhorszki, David Pugmire, Felix Schmitt, Erik Schnetter, Dominik Stańczak, Klaus Steiniger, Michael Sippel, Frank Tsung, Lipeng Wan, René Widera, and Erik Zenker!

Ongoing projects

Bridge the gap between openPMD and NeXus?

The HELPMI project

openPMD	NeXus
focus on simulations	focus on experiments
background: laser-plasma physics	background: photon and neutron physics

Bridge the gap between openPMD and NeXus?
The HELPMI project

openPMD		NeXus
focus on simulations	→ ? ←	focus on experiments
background: laser-plasma physics		background: photon and neutron physics

and in between?

Bridge the gap between openPMD and NeXus?

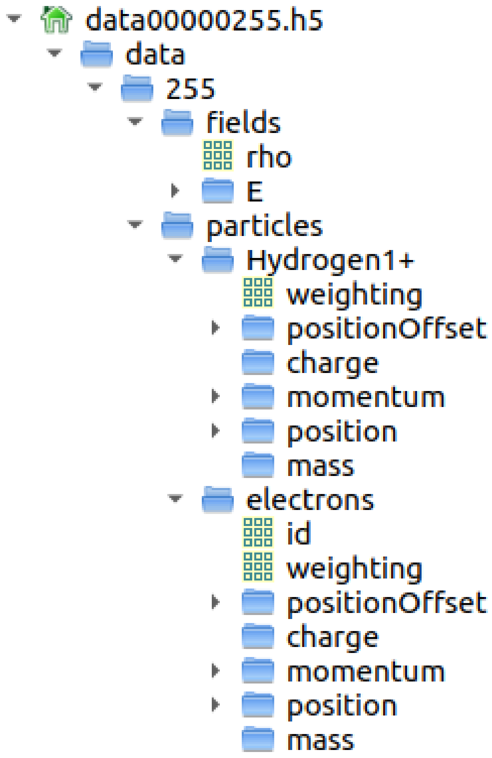
The HELPMI project

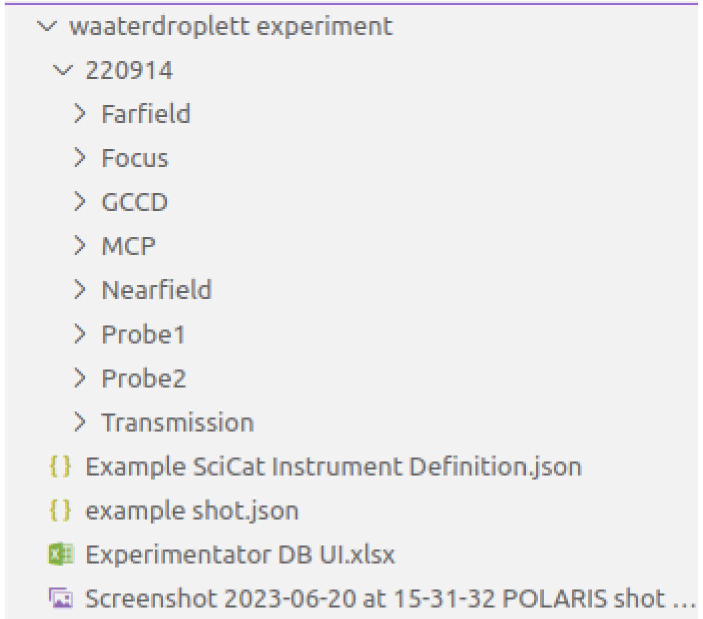
openPMD	HELPMI	NeXus
focus on simulations	→ focus on experiments →	focus on experiments
background: laser-plasma physics	background: ← laser-plasma physics ←	background: photon and neutron physics

Bridge the gap between openPMD and NeXus?

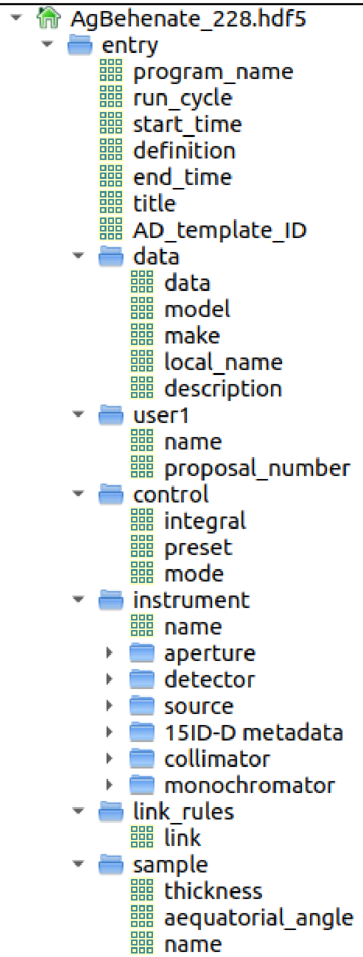
The HELPMI project

openPMD	HELPMI	NeXus



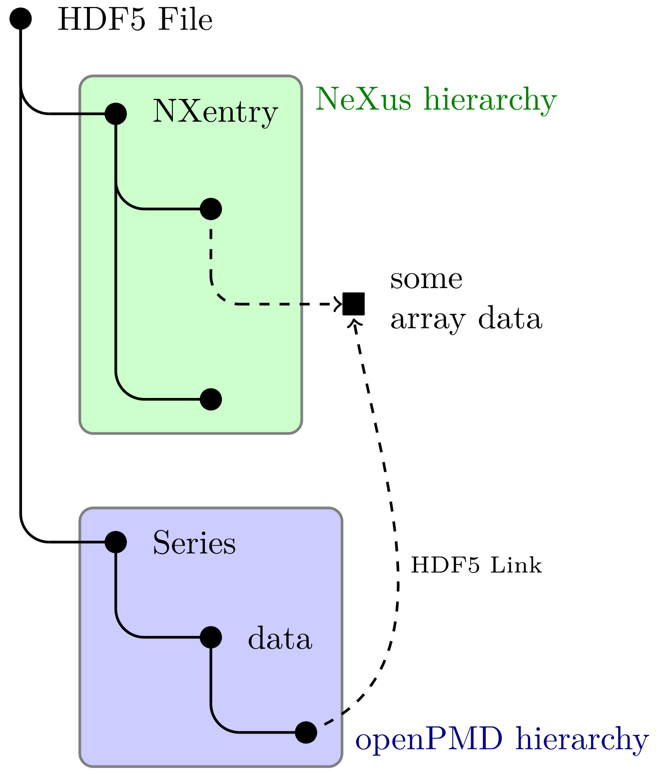


Metadata?
Setup Models?
Calibration?



HELPMI – a Helmholtz Metadata Collaboration project

- Primary goal: Develop a user-driven NeXus extension proposal for laser-plasma experiments
- Develop a glossary for LPA experiment data and infer the ontology for automated validation and processing
- openPMD is an existing standard for laser-plasma *simulations:*
  - aim for interoperability between both standards
  - example: openPMD "view" into NeXus data to compare experimental and simulation data



& project observers at **LBNL, LMU, ELI and STFC**

HELPMI – a Helmholtz Metadata Collaboration project

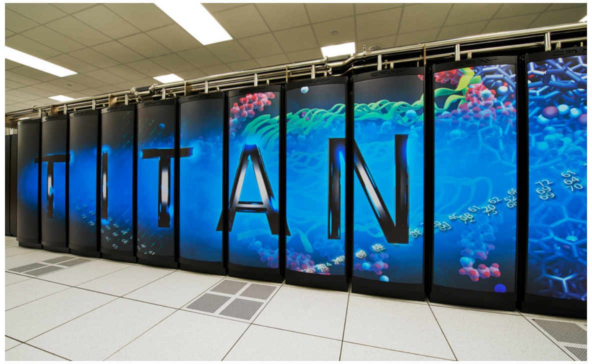
- Primary goal: Develop a user-driven
  NeXus extension proposal for laser-plasma experiments
- Develop a glossary for LPA experiment data
  and infer the ontology
  for automated validation and processing
- openPMD is an existing standard for
  laser-plasma *simulations:*
  - aim for interoperability between both standards
  - example: openPMD “view” into NeXus data
    to compare experimental and simulation data

Contact and time frame

- Project from April 2023 to April 2025
- Upcoming: Helpmi Workshop
  at GSI (Darmstadt) Nov 13-14
- helpmi@hzdr.de

& project observers at
LBNL, LMU,
ELI and STFC

I/O Performance lags behind Compute Performance



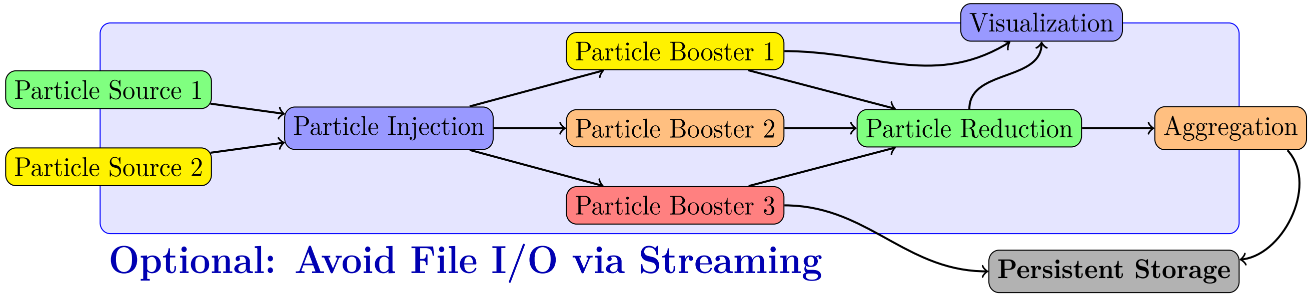
	Titan	Summit	Frontier	*Growth Factor*
Peak Performance:	27 Pflop/s	200 Pflop/s	1.6 Eflop/s	~60
FS Throughput:	1 TiByte/s	2.5 TiByte/s	5~10 TiByte/s	5~10
FS Capacity:	27 PiByte	250 PiByte	500~1000 PiByte	18~37

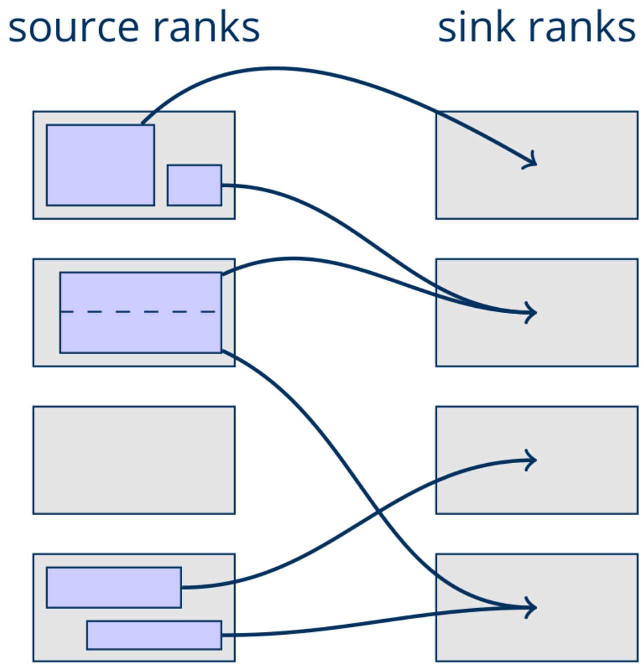
→ **parallel bandwidth** insufficient for HPC at full scale
→ **filesystem capacity** insufficient for HPC at full scale

Same trend in **experiments**?
→ Increasing **camera resolutions and data rates**

Franz Poeschel et al. "Transitioning from file-based HPC workflows to streaming data pipelines with openPMD and ADIOS2". 2022. doi: 10.1007/978-3-030-96498-6_6.

Streaming: Don't touch the Filesystem at all



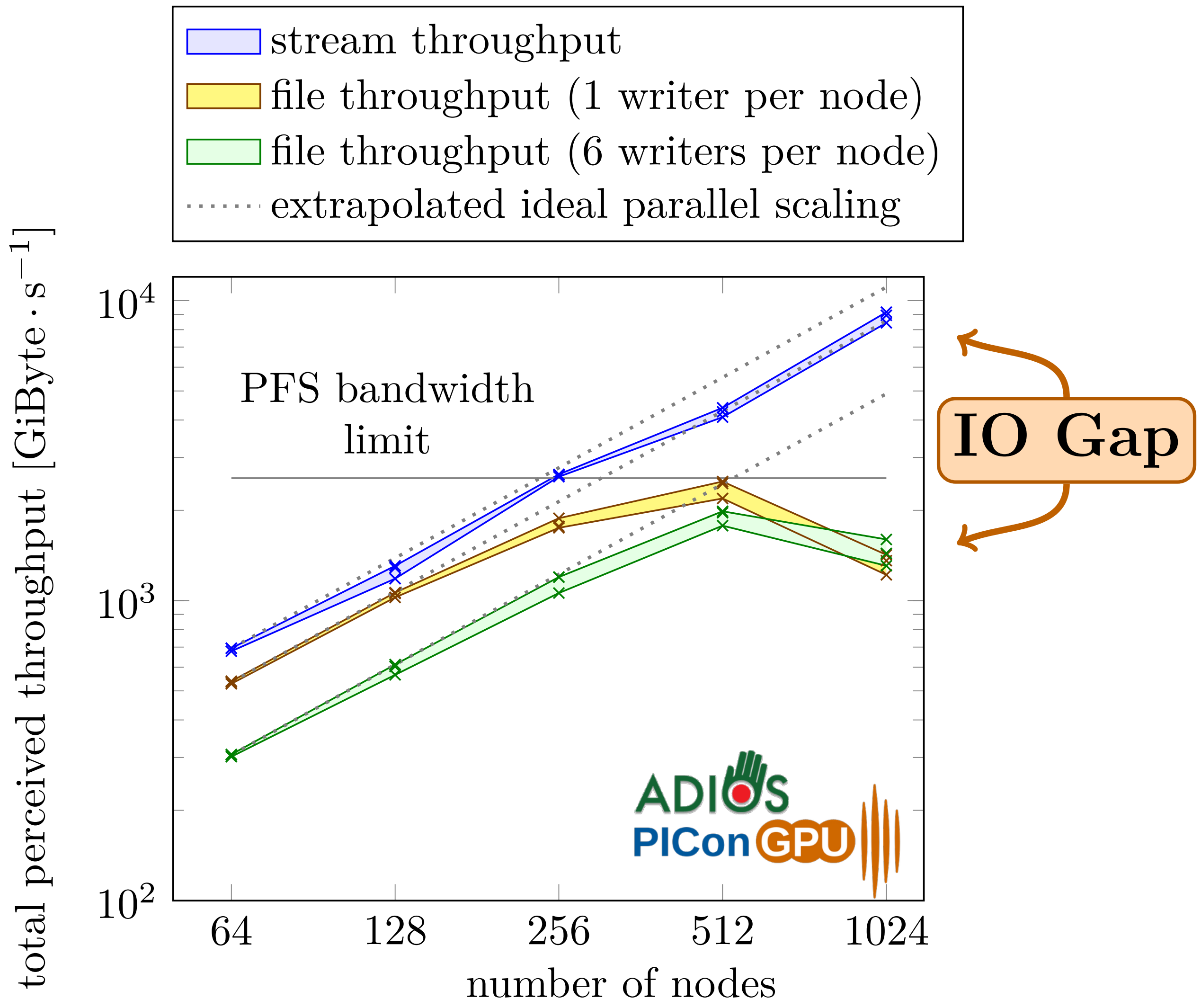


- Data processing pipelines and increasingly experiments setups have large I/O usage
- Scalable alternative: Streaming
  e.g. via Infiniband (on HPC systems)
  or wide area networks (in lab settings)

Challenge:
Compute a balanced, aligned, local mapping between two applications that remains useful in the problem domain

Franz Poeschel et al. "Transitioning from file-based HPC workflows to streaming data pipelines with openPMD and ADIOS2". 2022. doi: 10.1007/978-3-030-96498-6_6.

Break through Filesystem Bandwidth with Streaming

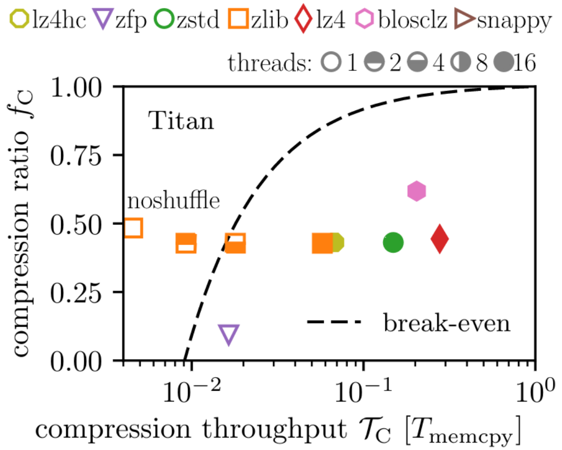


Memory-bound simulations reach the I/O system limits at a fraction of full scale

- Summit FS bandwidth (2.5TiByte/s) reached at 512 nodes (~11% of system size)
- Streaming workflows unaffected by filesystem bandwidth, use Infiniband hardware to scale beyond it

(benchmarks at 1024 nodes done after Summit system upgrade)

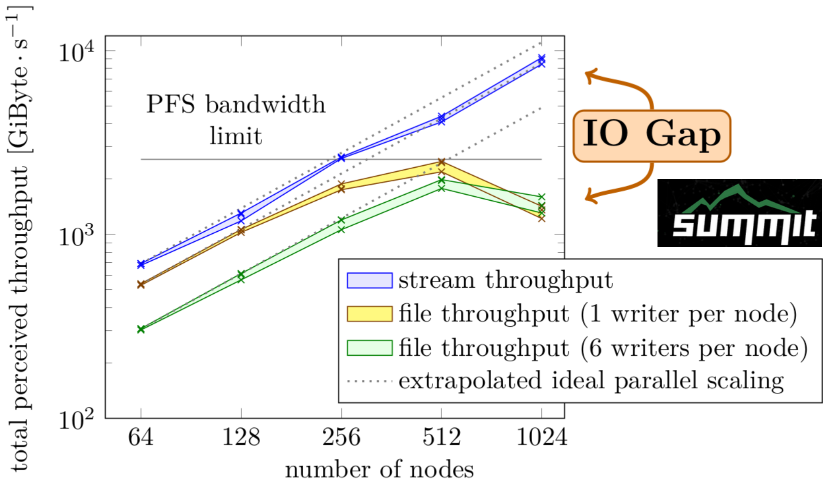
Performance: Data Layouts and no-file I/O



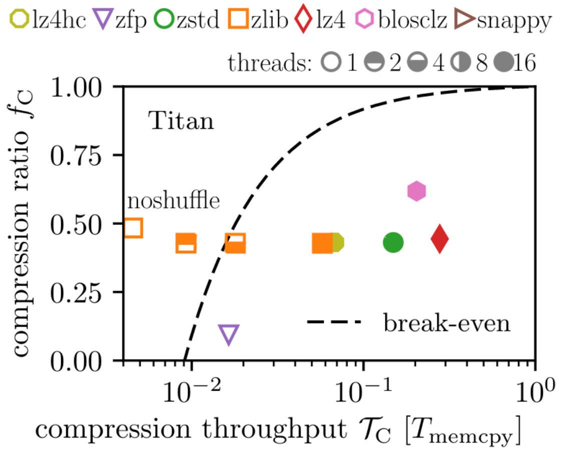
Fast Compressors Needed:
DOI:10.1007/978-3-319-67630-2_2
by A Huebl et al., ISC DRBSD-1 (2017)

$$\frac{\mathcal{T}_R \times (1 - f_R)}{1 - \mathcal{T}_R} > \mathcal{T}_{out}$$

Performance: Data Layouts and no-file I/O



Streaming Data Pipelines:
[DOI:10.1007/978-3-030-96498-6_6](https://doi.org/10.1007/978-3-030-96498-6_6)
by F Poeschel, A Huebl et al., SMC21 (2022)

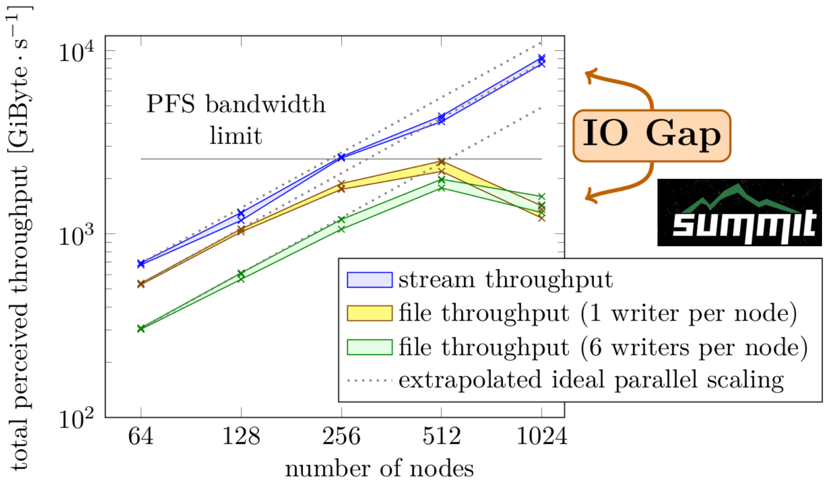


Fast Compressors Needed:
[DOI:10.1007/978-3-319-67630-2_2](https://doi.org/10.1007/978-3-319-67630-2_2)
by A Huebl et al., ISC DRBSD-1 (2017)

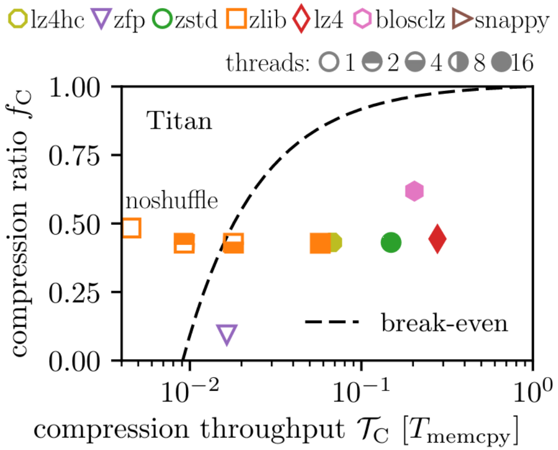
$$\frac{\mathcal{T}_R \times (1 - f_R)}{1 - \mathcal{T}_R} > \mathcal{T}_{out}$$

Performance: Data Layouts and no-file I/O

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING

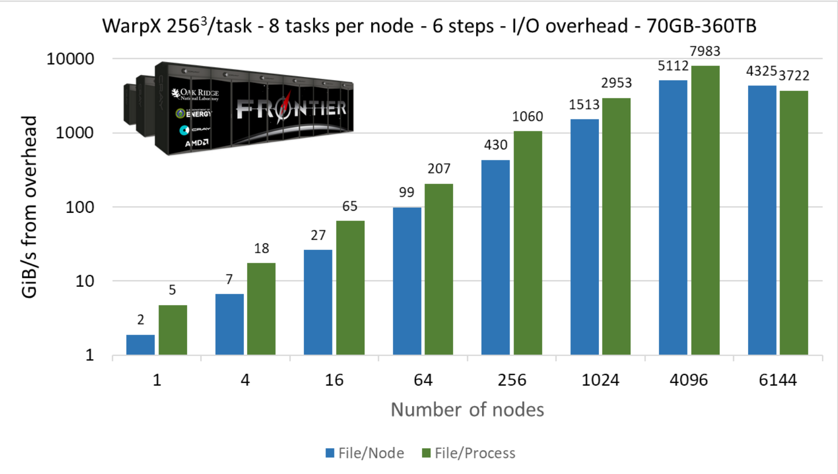


Streaming Data Pipelines:
DOI:10.1007/978-3-030-96498-6_6
by F Poeschel, A Huebl et al., SMC21 (2022)



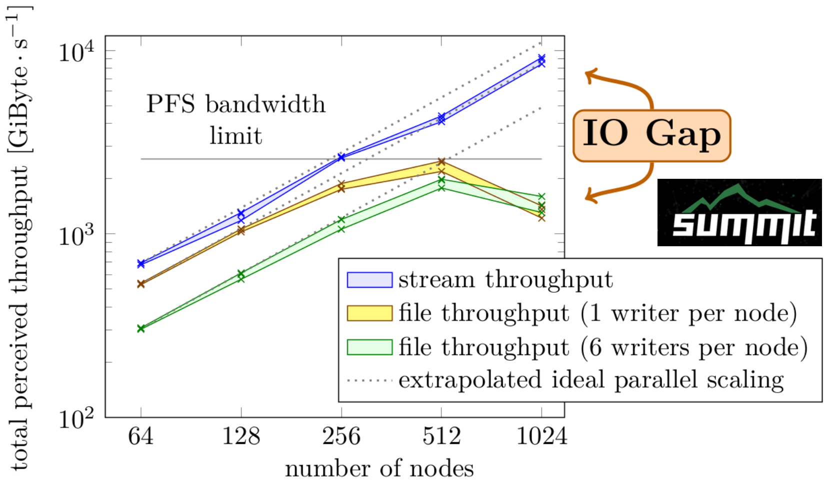
Fast Compressors Needed:
DOI:10.1007/978-3-319-67630-2_2
by A Huebl et al., ISC DRBSD-1 (2017)

ADIOS **openPMD-api w/ WarpX**

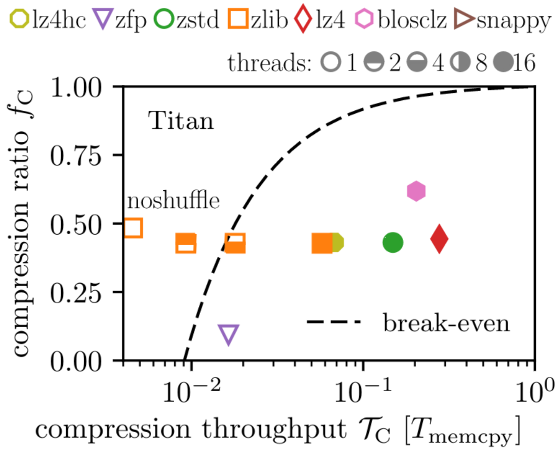


>5.5TB/s FS BW: two-tier lustre w/ high-performance storage & progressive files

Performance: Data Layouts and no-file I/O

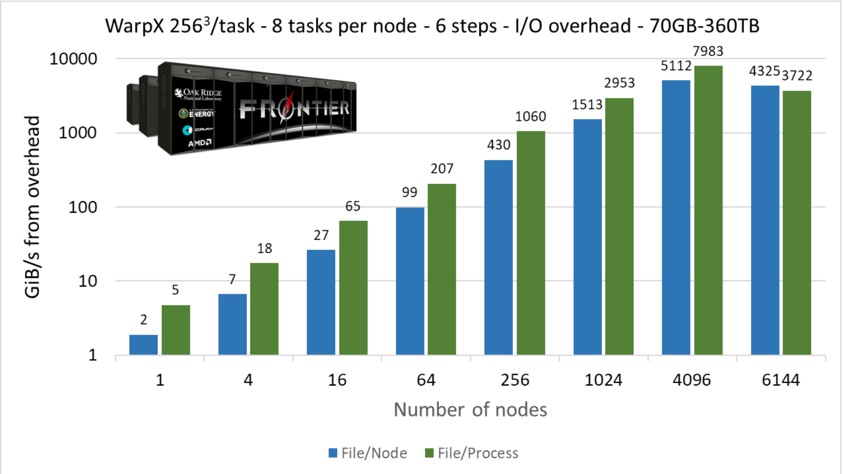


Streaming Data Pipelines:
DOI:10.1007/978-3-030-96498-6_6
by F Poeschel, A Huebl et al., SMC21 (2022)

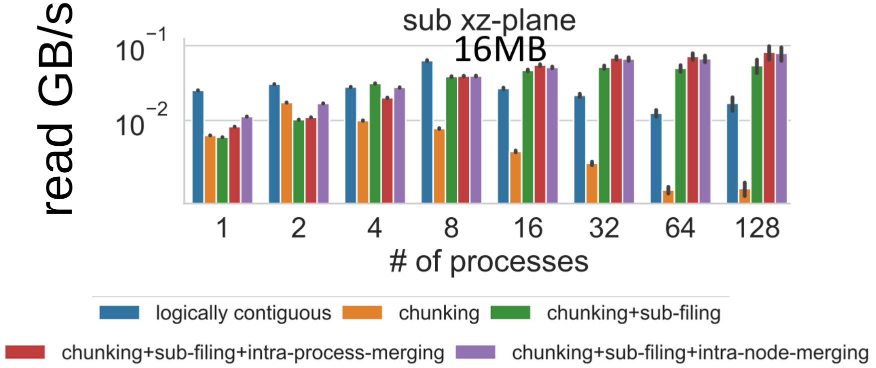


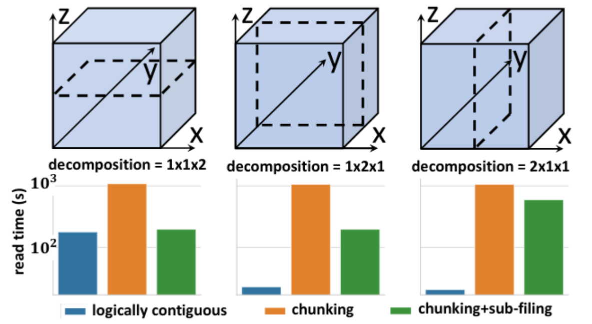
Fast Compressors Needed:
DOI:10.1007/978-3-319-67630-2_2
by A Huebl et al., ISC DRBSD-1 (2017)

ADIOS openPMD-api w/ WarpX



>5.5TB/s FS BW: two-tier lustre w/ high-performance storage & progressive files





Impact of decomposition schemes when reading

Online Data Layout Reorganization:
DOI:10.1109/TPDS.2021.3100784
by L Wan, A Huebl et al., TPDS (2021)

First Steps

→ head to https://github.com/openPMD/

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING

openPMD

Open Standard for Particle-Mesh Data Files

https://www.openPMD.org axelhuebl@lbl.gov

Repositories 17 | Packages | People 50 | Teams 5 | Projects

Pinned repositories

openPMD-standard
Open Standard for Particle-Mesh Data Files
41 · 17

openPMD-projects
Overview on Projects around openPMD
4 · 4

openPMD-viewer
Python visualization tools for openPMD files
Jupyter Notebook · 35 · 26

openPMD-api
C++ & Python API for Scientific I/O
C++ · 55 · 30

openPMD-visit-plugin
Plugin allowing VisIt to read openPMD files
C · 8 · 3

openPMD-example-datasets
HDF5 Example Files
Python · 5 · 1

...and of course https://openpmd-api.readthedocs.io/

https://github.com/openPMD/
https://openpmd-api.readthedocs.io/

Acknowledgements

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. Supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration). Supported by EC through Laserlab-Europe, H2020 EC-GA 871124. Supported by the Consortium for Advanced Modeling of Particles Accelerators (CAMPA), funded by the U.S. DOE Office of Science under Contract No. DE-AC02-05CH11231. This work was partially funded by the Center of Advanced Systems Understanding (CASUS), which is financed by Germany's Federal Ministry of Education and Research (BMBF) and by the Saxon Ministry for Science, Culture and Tourism (SMWK) with tax funds on the basis of the budget approved by the Saxon State Parliament. The HElmholtz Laser Plasma Metadata Initiative (HELPMI) project (ZT-I-PF-3-066) was funded by the "Initiative and Networking Fund" of the Helmholtz Association in the framework of the "Helmholtz Metadata Collaboration" project call 2022.

SACHSEN This project is co-financed by the Saxon State government out of the State budget approved by the Saxon State Parliament

Summary and Outlook

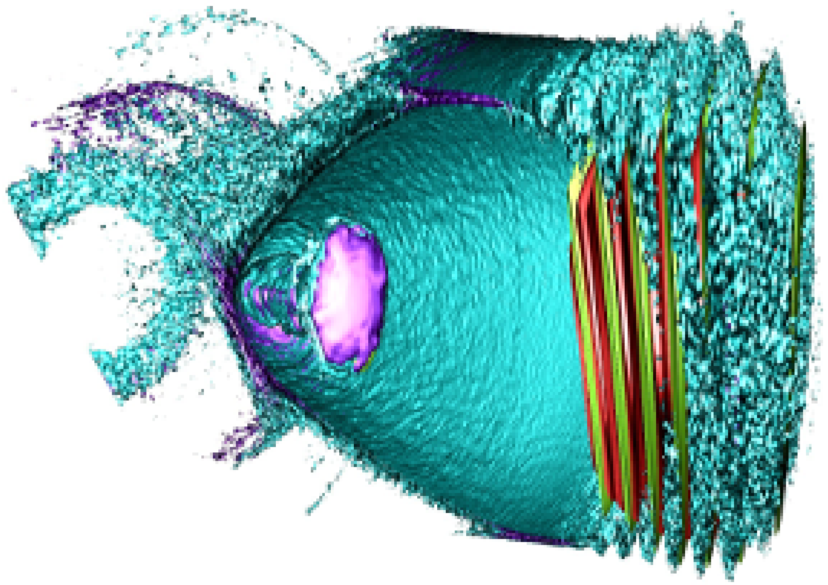
CASUS
CENTER FOR ADVANCED
SYSTEMS UNDERSTANDING

- openPMD is a **F.A.I.R. standard for scientific metadata**
  - bridge scientific models and domains
    by **common markup language**
  - Large **open-source ecosystem**: documentation, example data, validation, scripts, integration via plugins and converters, reference libraries
- Reference **implementation**:
  - Easy to use I/O for scientific data
  - **Scalable I/O** at the Exascale and Pbyte-scale
  - Scalable from small workstation via parallel in-transport data processing to file-less RDMA workflows
- **Outlook**
  - Complex data layouts such as **mesh refinement**
  - Bridge towards **experimental data acquisition systems**
  - Transfer HPC solutions to experiments challenges

openPMD

openpmd.slack.com

openpmd.org



Picture: LWFA simulation in PIConGPU