UK Atomic Energy Authority

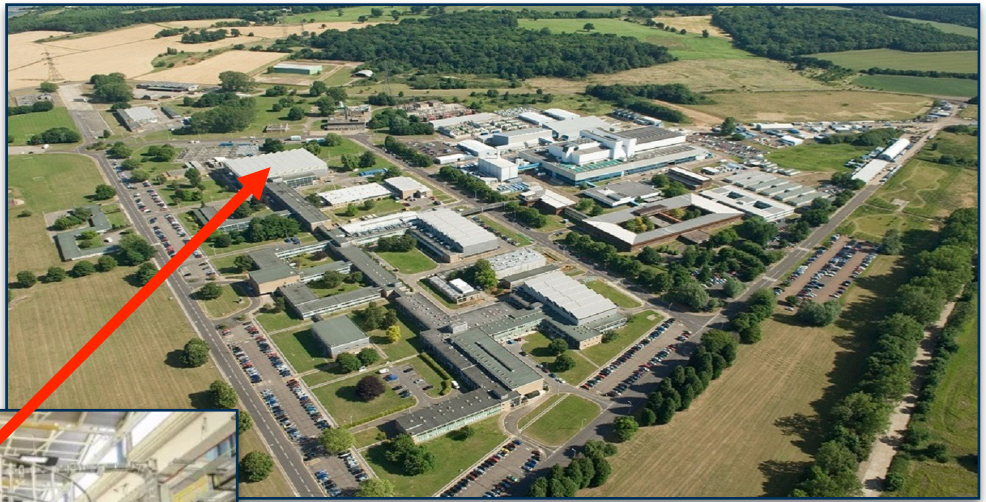
Towards an Analysis-Ready, Cloud-Optimised service for FAIR fusion data

Samuel Jackson et al, UKAEA

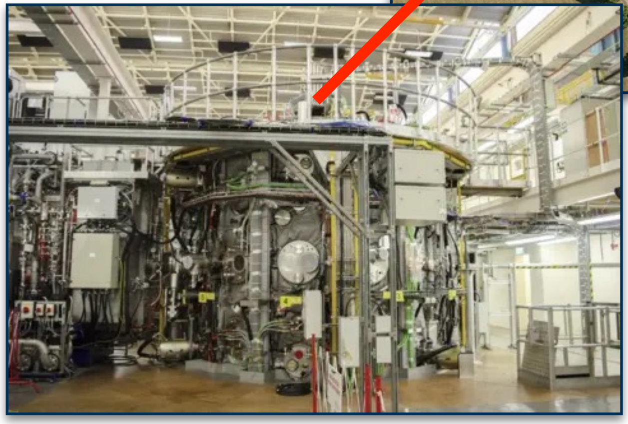
Overview & Motivation

MAST

- MAST (Mega Amp Spherical Tokamak)
- Spherical tokamak design commissioned by EURATOM/UKAEA
- Built at Culham Centre for Fusion Energy, Oxfordshire, UK
- Experiments ran from 1999 through to 2013
- Produced ~30,000 shots over its history
- Succeeded by MAST Upgrade (MAST-U) in 2020



Culham Centre for Fusion Energy, UK



MAST Tokamak

Motivation

We want to:

- Have software tools that are robust and can scale
- Gain expertise from complementary domains
- Collaborate with the wider world
    - Fusion energy, Data, and AI/ML communities

We need:

- Open access with minimal barriers.
- Integrate with data analysis & reduction tools that scale.
- Integrate with domain agnostic tools.
    - We cannot afford to build everything ourselves.
- Perform search, retrieval, and analysis across the historical record

Motivation

UK Atomic Energy Authority

Because our funders tell us too…

UKRI Open Research Data Taskforce:

- that published scientific results should be open access - digital, online, free of charge, and free of most copyright and licensing restrictions; and
- that the data acquired by individual scientists and scientific groups should be subject to a default position whereby it is made findable, accessible, interoperable and re-useable (FAIR);

EPSRC Research Data Policy:

1. EPSRC-funded research data is a public good produced in the public interest, and should be made freely and openly available with as few restrictions as possible in a timely and responsible manner.

Final Report of the UKRI Open Research Task Force
EPSRC Data Policy

Project Objectives

UK Atomic Energy Authority

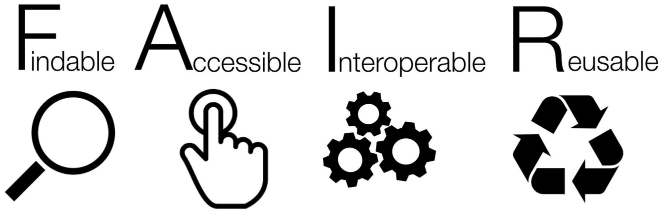
Goal: "*To produce a framework for public access to MAST data in a FAIR (Findable, Accessible, Interoperable, and Reusable) manner*".

- Data must be easily **findable** through the metadata
- Data must be in exposed in an **interoperable** format
- Prioritise **performance optimisation** for artificial intelligence (AI) and machine learning (ML) workflows
- Minimise **loading** and **transferring** data (lazy loading)
- Support **data analysis** and **ML/AI** frameworks
- Support **larger-than-memory** & **parallel** computation
- Be **publicly** accessible

The Wider Picture

FAIR Data

- **Findable -** Metadata and data should be easy to find for both humans and computers
- **Accessible** - It should be clear how to access the data once found.
- **Interoperable -** Data can be integrated with other data and interoperate with applications or workflows for analysis, storage, and processing.
- **Reusable -** Metadata and data should be well-described so that they can be replicated and/or combined in different settings.

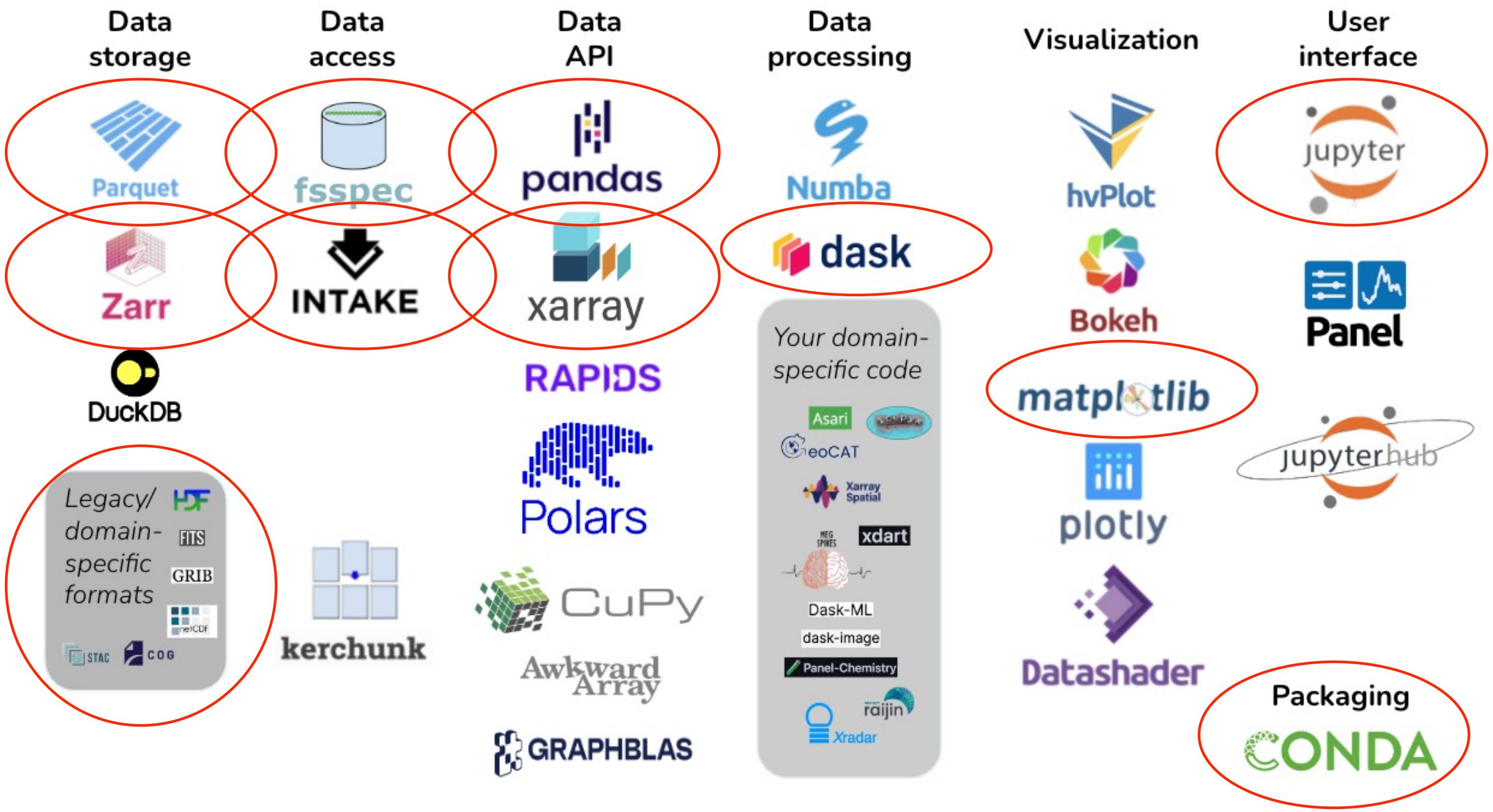


GO FAIR: https://www.go-fair.org/fair-principles/
Wilkinson, M. D. et al. The FAIR Guiding Principles for scientific data management and stewardship. (2016).
Strand, P. et al. A FAIR based approach to data sharing in Europe. (2022).

Pandata Stack

UK Atomic Energy Authority

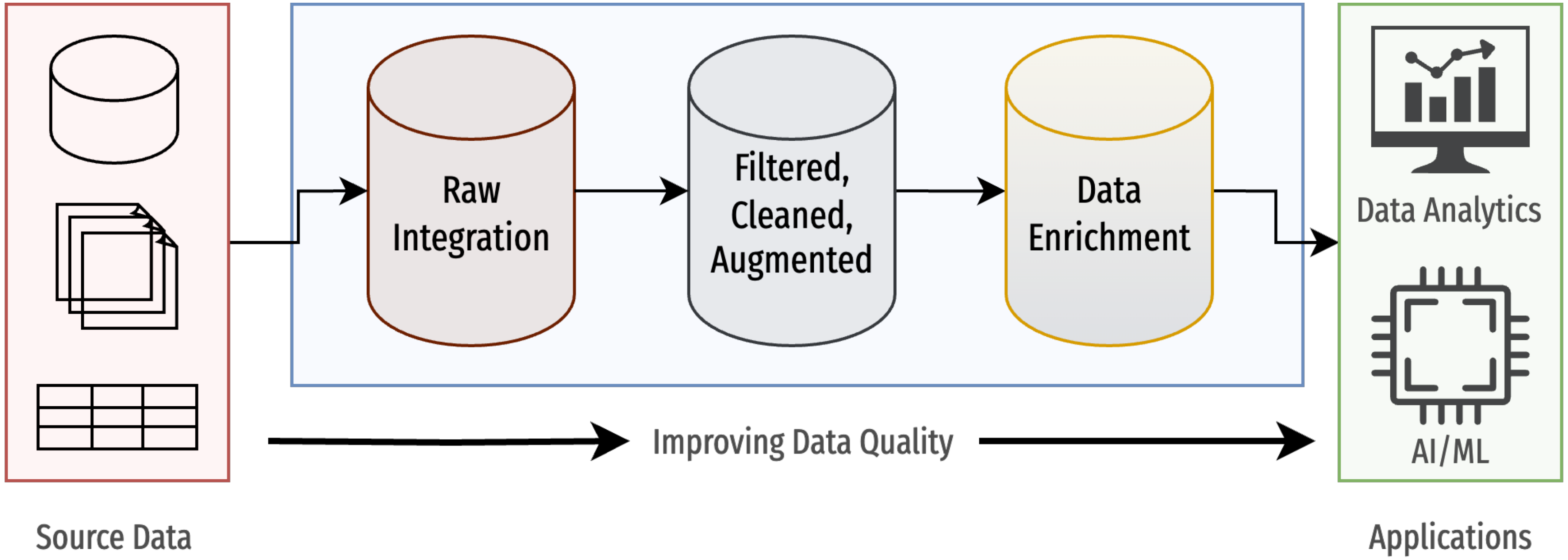
Pandata stack is an **open-source** set of **interoperable, composable**, and **domain agnostic** software technologies for data analysis and scientific computation.



Bednar J.A., Durant M. The Pandata Scalable Open-Source Analysis Stack.

Medallion Architecture

UK Atomic Energy Authority



Medallion architecture of data management design pattern aims to improve ***reliability***, ***scalability***, and ***performance*** of data processing systems.

- **Raw data integration**: data gathered in one place.
- **Filtered, Cleaned, Augmented**: common, standardised view of the data
- **Data Enrichment**: optimised project specific views of the data

Databricks: Medallion Architecture

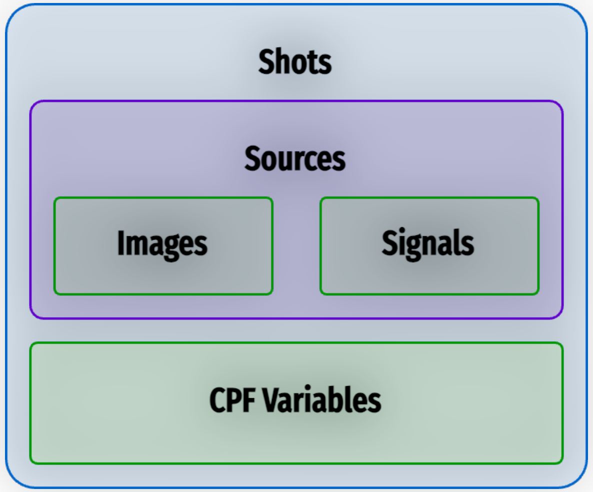
MAST Data

UK Atomic Energy Authority

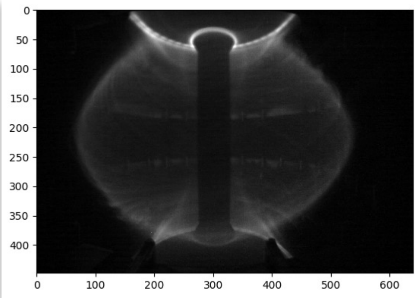
MAST Diagnostic Data

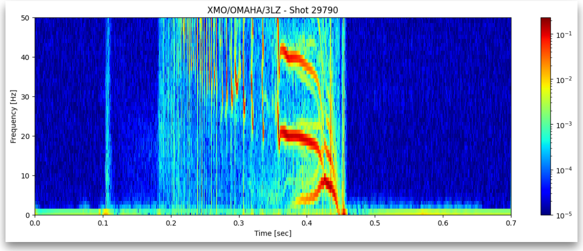
MAST Data can be thought of in terms of:

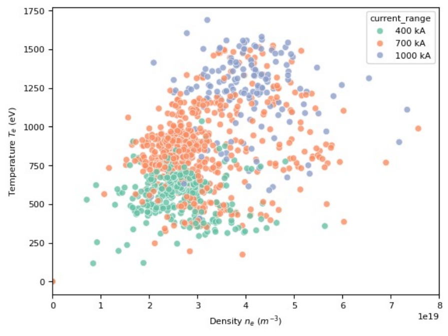
- **Shots:** A single experimental shot taken by the machine.
- **Sources:** Each shot contains multiple diagnostic sources.
  - Examples include: Mirnov Coils, Thompson scattering, EFIT output etc.
- **Signals:** Each source contains multiple recorded quantities.
  - In MAST these were conceptually split into “signals” and “images”.
- **Summary Physics Variables:** Additional summary statistics documenting a shot.
  - e.g. max plasma current, beta, confinement time

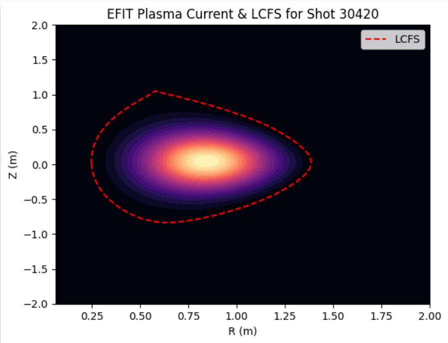


Conceptual overview of different types of data from MAST



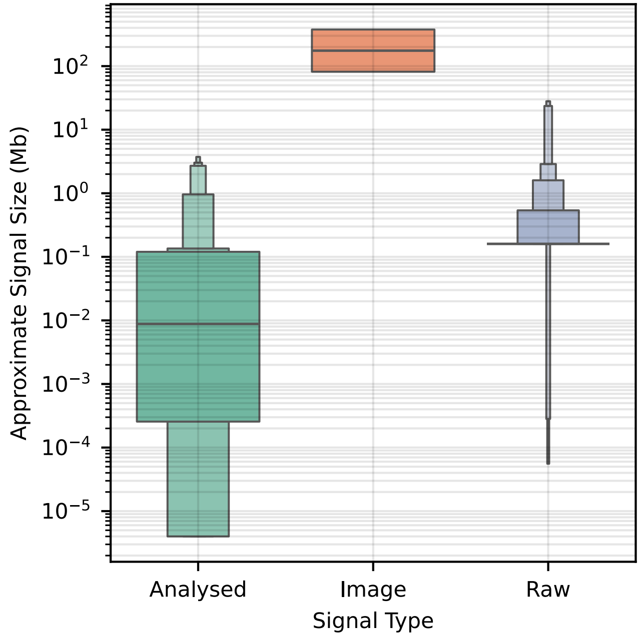


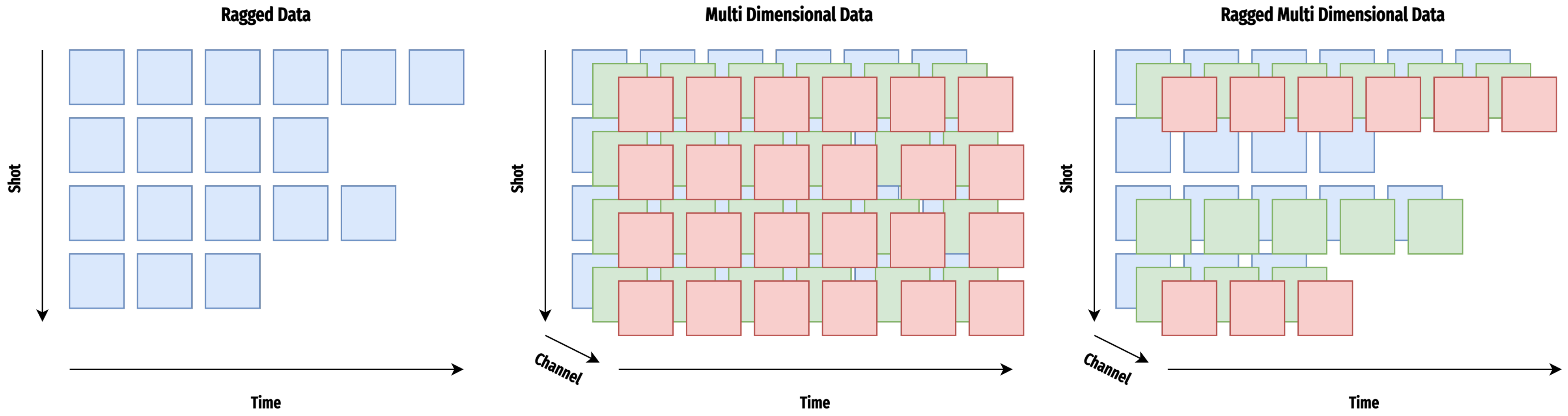




Data Challenges

- Data is multi-dimensional and ragged
- E.g. `time`, `channel`, `psi`, `radial_index`
- Data varies in size from very small (few kb) to large (1GB)
- Data comes from scattered sources/formats
- Data has inconsistent naming, units, dimensions name etc.

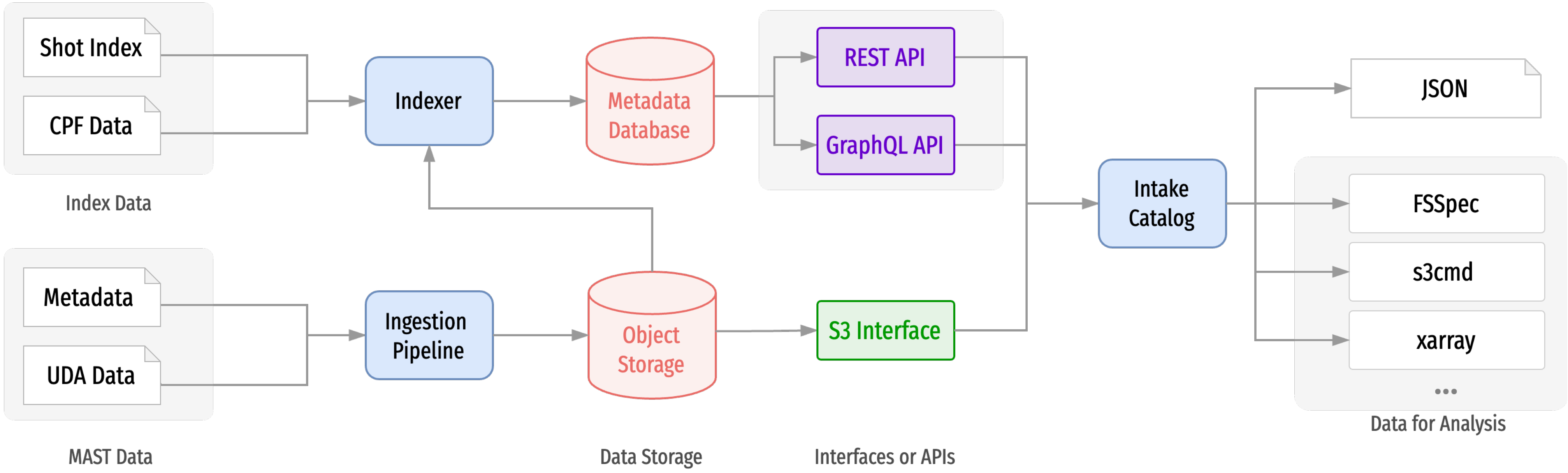




Architecture

System Architecture

UK Atomic Energy Authority



- **Object storage**
  - Holding shot, source, and signal data in a self-describing, cloud optimised file format.
  - Accessible by S3 protocol.

- **Metadata database** indexing data in the object storage
  - For searching and finding data in the object storage
  - Accessible by web APIs

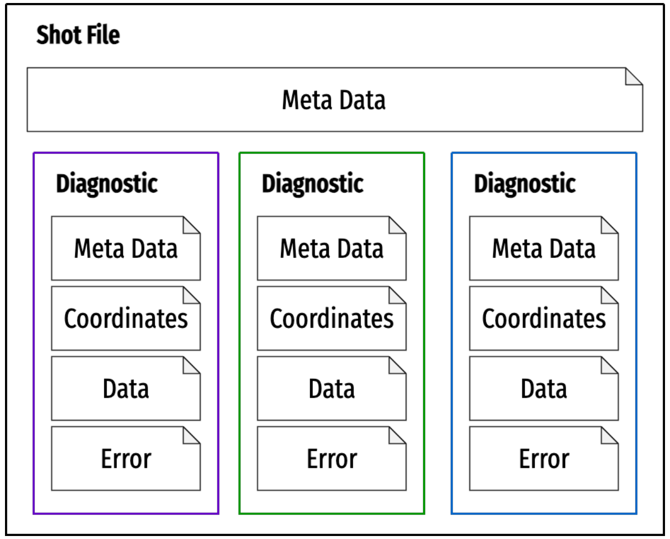
File Format

We choose to use a hierarchical self-describing file format.

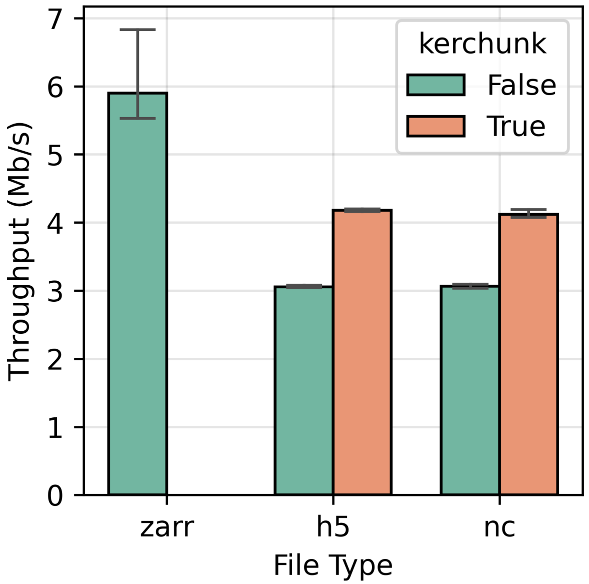
- Group data by shot
- Group signals by diagnostic
- Each group may contain metadata
- Coordinate axes are also defined

For our implementation we choose Zarr format

- Hierarchical format
- HDF-like interface
- Consolidated metadata
- Parallel read/write
- Cloud optimised
- Interoperable with different languages
- Lazy loading



Above: File format structure

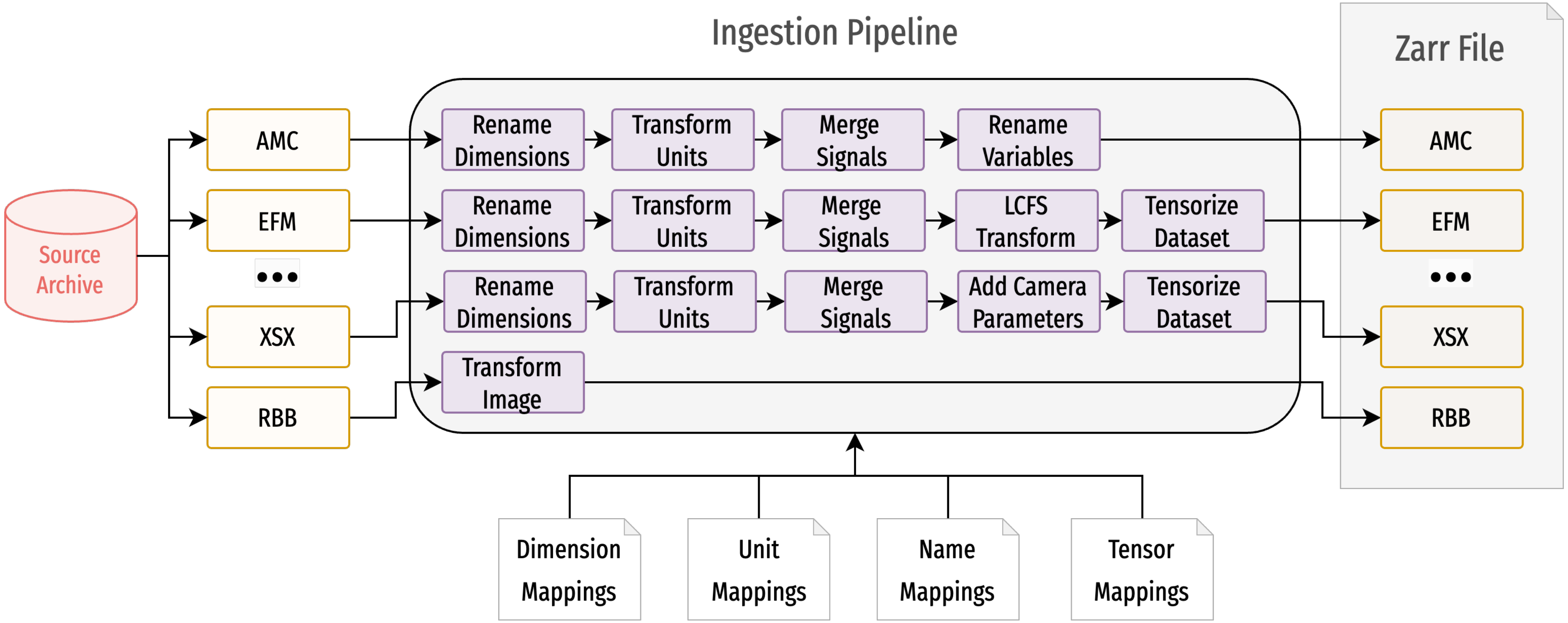


Above: Performance comparison of Zarr/NetCDF/HDF with and without Kerchunk RBB camera data.

UK Atomic Energy Authority

Zarr File Format

Ingestion Pipeline



- We start from our internal archive of historical data.
- Each source is transformed through a specific pipeline
- Normalising names, dimension names, units, and grouping channels.
- Source specific transformations.
- Written to Zarr & synchronised to S3

Indexing

Our metadatabase indexes the data records within each file.

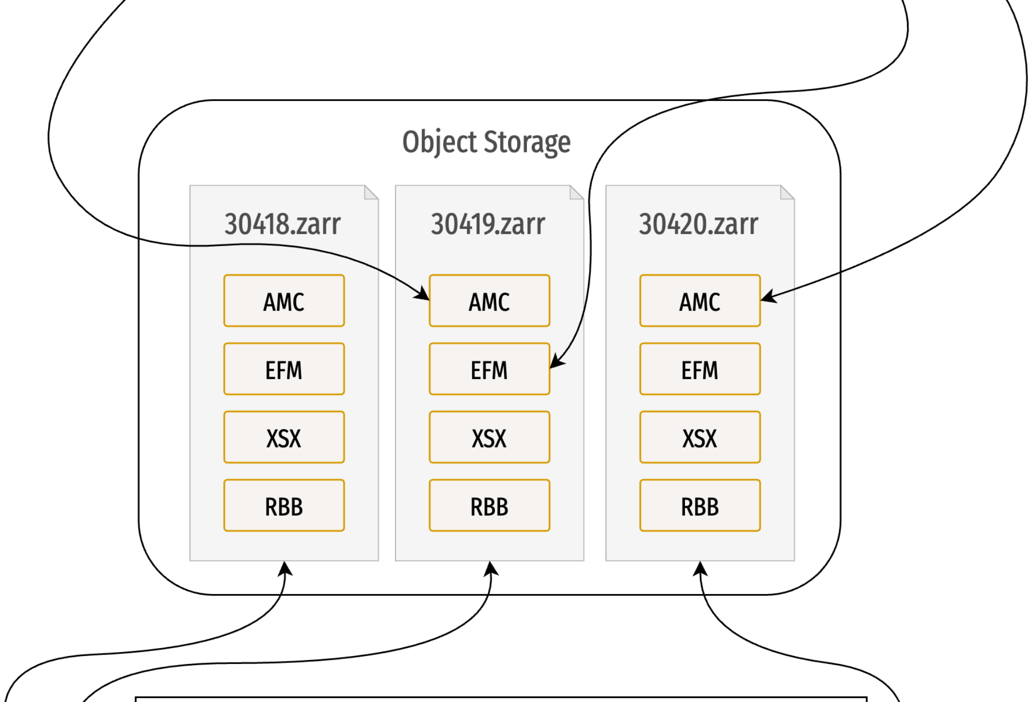
We index on three levels:
- Shots
- Signals
- Sources

Each item has a UUID assigned to it and references a URL which links to the object storage.

Database implemented with PostgreSQL

Source Metadata Index

Name	Shot ID	URL
amc	30420	s3://mast/shots/30420.zarr/amc
efm	30419	s3://mast/shots/30419.zarr/efm
amc	30419	s3://mast/shots/30419.zarr/amc



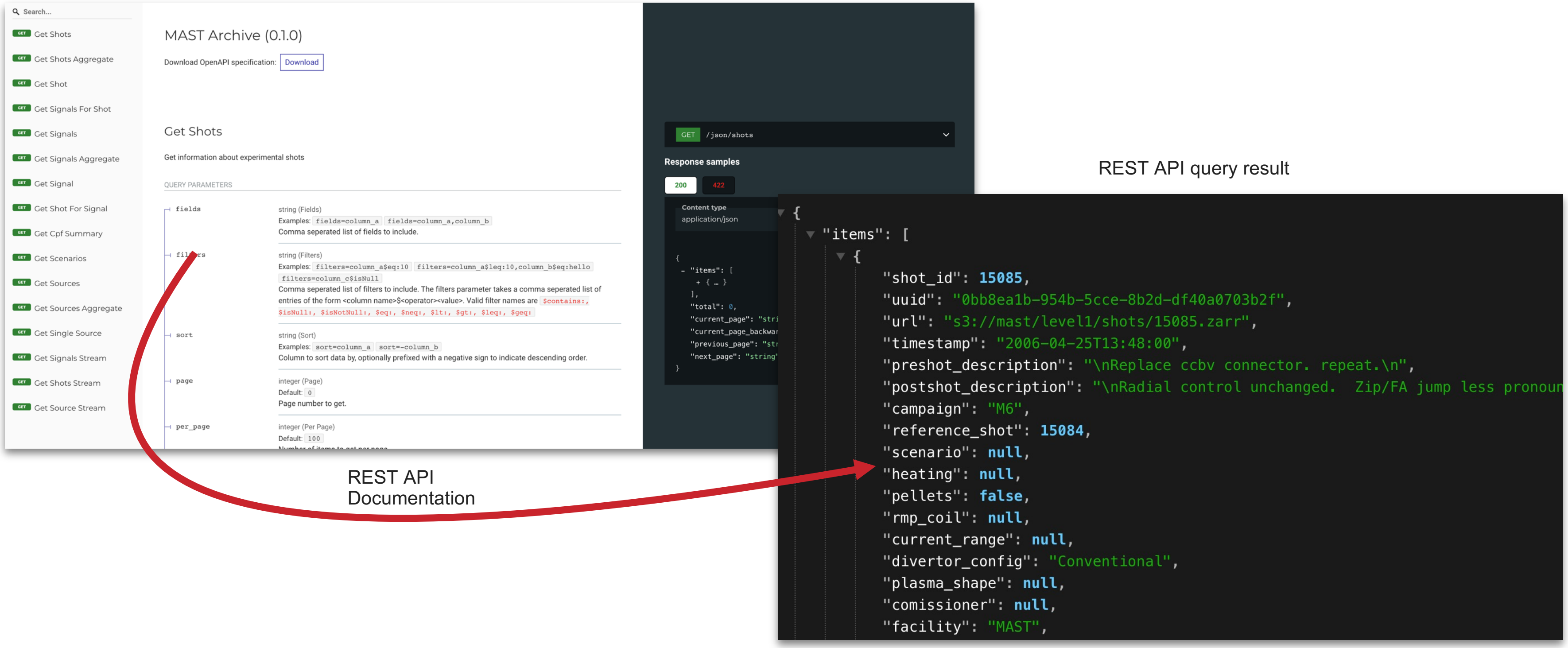
Shot Metadata Index

Shot ID	URL	Campaign
30420	s3://mast/shots/30420.zarr	M9
30419	s3://mast/shots/30419.zarr	M9
30419	s3://mast/shots/30419.zarr	M9

FAIR Principles
[F4. (Meta)data are registered or indexed in a searchable resource]
[A2. Metadata are accessible, even when the data are no longer available]

Usage

Metadata APIs: REST

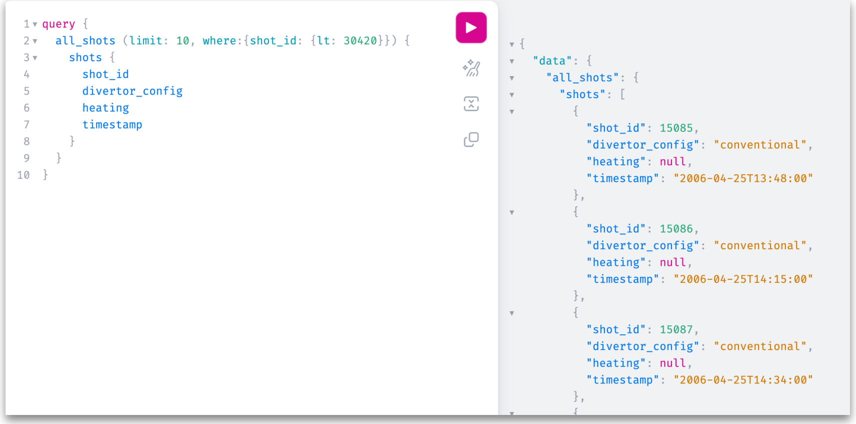


REST API query result

REST API Documentation

REST API implemented with `fastapi`, `sqlmodel`, and `sqlalchemy`

Experimented with GraphQL written on top with `strawberry`



GraphQL query explorer

User Access: Xarray, Dask, S3

UK Atomic Energy Authority

Loading MAST data in **2 lines of code**:

```
import xarray as xr
dataset = xr.open_zarr("https://s3.echo.stfc.ac.uk/mast/level1/shots/30420.zarr/amc")
```

A more explicit example with S3:

```
import s3fs
import xarray as xr
import matplotlib.pyplot as plt

# s3 storage location
endpoint_url = 'https://s3.echo.stfc.ac.uk'
# URL of data we want to load
url = 's3://mast/level1/shots/30420.zarr/amc'

# fsspec handle to remote file system
s3 = s3fs.S3FileSystem(anon=True, endpoint_url=endpoint_url)

# open the dataset
dataset = xr.open_zarr(s3.get_mapper(url))

# data only loaded at this point!
plt.plot(dataset['time'], dataset['plasma_current'])
```

User Access: Intake Catalogs

UK Atomic Energy Authority

- A python package describing, loading, and processing data.
- Intake Catalogs can be ***thin and flexible*** access layers.
- Same example as before, but now agnostic to data specifics:

INTAKE

```
import intake
import matplotlib.pyplot as plt

catalog  = intake.open_catalog('https://mastapp.site/intake/catalog.yml')
url = 's3://mast/level1/shots/30420.zarr/amc'

dataset = catalog.level1.shots(url=url)
dataset = dataset.to_dask()

# data only loaded at this point!
plt.plot(dataset['time'], dataset['plasma_current'])
```

This also enables us to insert a **caching** between the user and the data!
Second time reading is much faster!

Writing custom intake catalog is also completely possible. It's just a YAML file.

Intake: ReadTheDocs

User Access: Intake Catalogs

- Same access pattern for metadata index
- Can load metadata straight into a pandas dataframe

```
import intake
import matplotlib.pyplot as plt
catalog = intake.open_catalog('https://mastapp.site/intake/catalog.yml')
shots_df = catalog.index.level1.shots().read()
```

	url	preshot_description	postshot_description	campaign	current_range	divertor_config	pla
0	s3://mast/level1/shots/11695.zarr	\n0.1T TF SHOT\n	\nOK\n	M5	None	Conventional	
1	s3://mast/level1/shots/11696.zarr	\nSTANDARD 0.3T TF SHOT\n	\nOK\n	M5	None	Conventional	
2	s3://mast/level1/shots/11697.zarr	\nRAISE TO 0.5T\n	\nOK, ALARMS ARE LOWER\n	M5	None	Conventional	
3	s3://mast/level1/shots/11698.zarr	\nRAISE TO .56T\n	\nSTILL ALARMS BUT LOWER AGAIN\n	M5	None	Conventional	
4	s3://mast/level1/shots/11699.zarr	\nRAISE TO .58T\n	\nOK\n	M5	None	Conventional	
...	...	...	...	...	...	...	
15548	s3://mast/level1/shots/30467.zarr	\nRepeat with new neutron camera position.\ncH...	\nTwo times lower DD neutron rate than referen...	M9	700 kA	Conventional	

User Access: Bulk Download

UK Atomic Energy Authority

Bulk download of data can be done using your favourite S3 command line tool.
For example, `s5cmd` is a fast parallel transfer tool.

Download one whole shot

```
s5cmd --no-sign-request -–endpoint-url https://s3.echo.stfc.ac.uk \
  cp "s3://mast/level1/shots/30420.zarr/*" ./data/30420.zarr
```

Download a single source for all shots

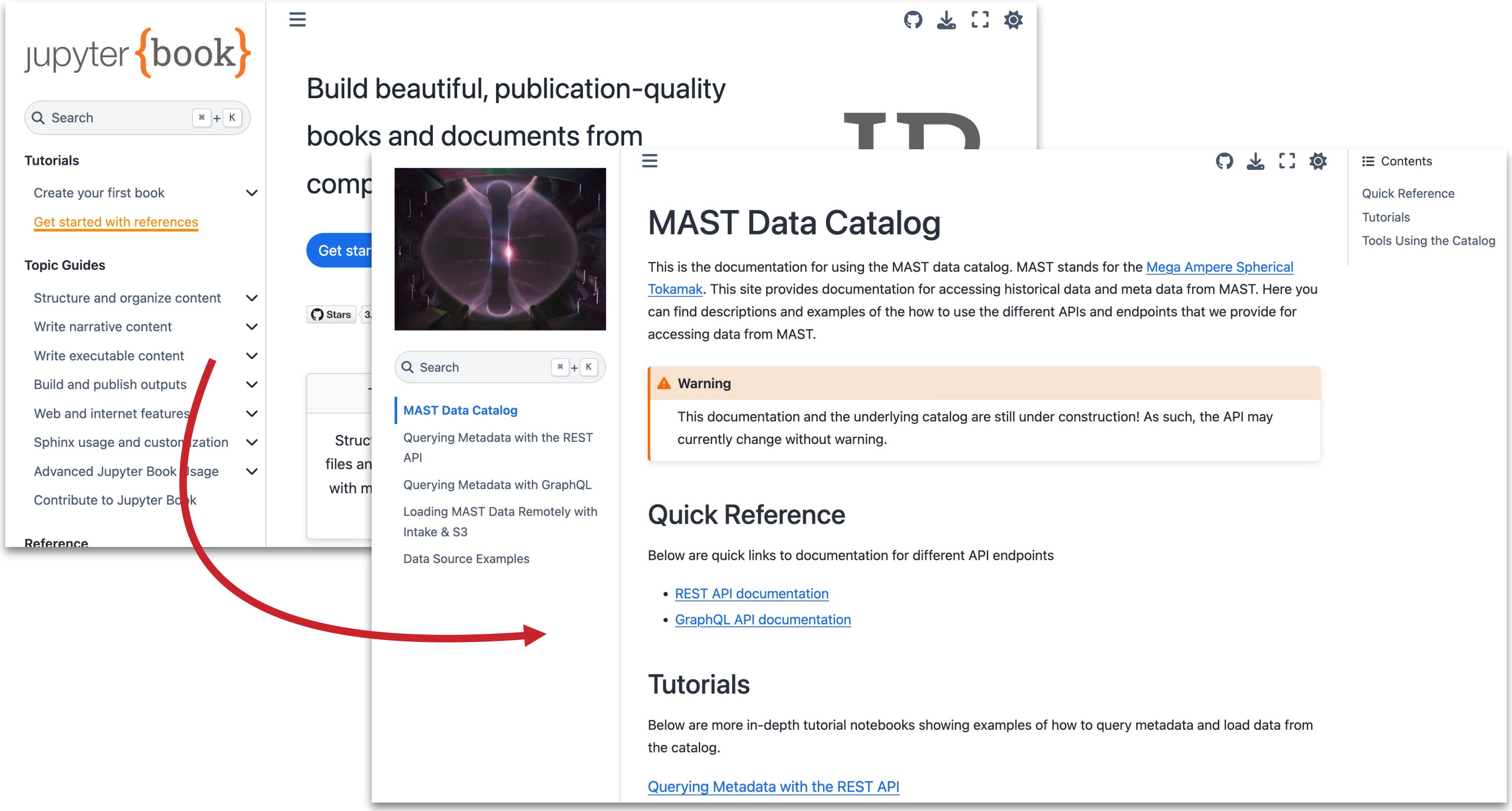
```
s5cmd –no-sign-request -–endpoint-url https://s3.echo.stfc.ac.uk \
  cp "s3://mast/level1/shots/*.zarr/rbb/*" ./data
```


s5cmd github

User Documentation

UK Atomic Energy Authority

Using Jupyter book to build documentation that is also executable



Jupyter Book Project

Future Directions

UKAEA & IMAS Schema

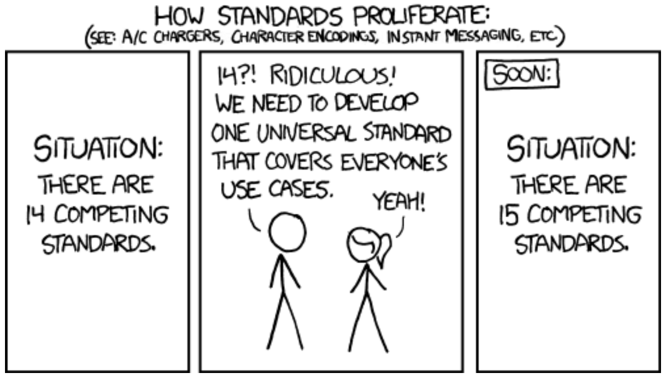
Ongoing work within UKAEA to create schemas for different experimental facilities.

Adam Parker/Jonathan Hollocombe’s work on mappings

See Jonathan’s talk at 10:10!

- Community Standards like DCAT, QUDT
- UKAEA Metadata Mappings
- IMAS Mappings

XKCD #927



MAST-U Schema -> IMAS Mappings


```
{
  "ids_properties/homogeneous_time": {
    "MAP_TYPE": "VALUE",
    "VALUE": 0
  },
  "flux_loop": {
    "MAP_TYPE": "VALUE",
    "VALUE": "{{ length(FLUX_LOOPS) }}"
  },
  "flux_loop[#]/name": {
    "MAP_TYPE": "VALUE",
    "VALUE": "{{ FL_NAME }}"
  },
  "flux_loop[#]/identifier": {
    "MAP_TYPE": "VALUE",
    "VALUE": "FLUX_LOOP_{{ indices.0 + 1 }}"
  },
  "flux_loop[#]/type/name": {
    "MAP_TYPE": "VALUE",
    "VALUE": "{{ FL_TYPE }}"
  },
  "flux_loop[#]/type/index": {
    "MAP_TYPE": "VALUE",
    "VALUE": "{{ at(FL_TYPE_MAP, at(FLUX_LOOPS, indices.0).TYPE) }}"
  },
  "flux_loop[#]/position": {
    "MAP_TYPE": "DIMENSION",
    "DIM_PROBE": "_flux_loop[#]/position_size"
  },
  "_flux_loop[#]/position_size": {
    "MAP_TYPE": "PLUGIN",
    "PLUGIN": "GEOMETRY",
    "ARGS": {
      "signal": "{{ FL_NAME }}",
      "key": "{{ at(FL_POSITION_MAP, at(FLUX_LOOPS, indices.0).TYPE) }}.r"
    }
  },
```


IMAS Schema

magnetics

Path	Dimensions	Type	Units	Description
magnetics				Magnetic diagnostics for equilibrium identification and plasma shape control.
magnetics.b_field_pol_probe	[1...N]	STRUCT_ARRAY		Poloidal field probes
magnetics.b_field_pol_probe[:].area		FLT_0D (uncertain)	m^2	Area of each turn of the sensor; becomes effective area when multiplied by the turns
magnetics.b_field_pol_probe[:].bandwidth_3db	[1...2]	FLT_1D (uncertain)	Hz	3dB bandwidth (first index : lower frequency bound, second index : upper frequency bound)
magnetics.b_field_pol_probe[:].field		STRUCTURE	T	Magnetic field component in direction of sensor normal axis (n) averaged over sensor volume defined by area and length, where n = cos(poloidal_angle)*cos(toroidal_angle)*grad(R) - sin(poloidal_angle)*grad(Z) + cos(poloidal_angle)*sin(toroidal_angle)*grad(Phi)/norm(grad(Phi))
magnetics.b_field_pol_probe[:].field.data	[magnetics.b_field_pol_probe[:].field.time]	FLT_1D (uncertain)	T	Data
magnetics.b_field_pol_probe[:].field.time	[1...N]	FLT_1D_TYPE	s	Time
magnetics.b_field_pol_probe[:].field.validity		INT_0D		Indicator of the validity of the data for the whole acquisition period. 0: valid from automated processing, 1: valid and certified by the diagnostic RO; - 1 means problem identified in the data processing (request verification by the diagnostic RO), -2: invalid data, should not be used (values lower than -2 have a code-specific meaning detailing the origin of their invalidity)
				Indicator of the validity of the data for each time slice. 0: valid

Future Directions

IMAS Compliance

Data versioning

- Ongoing work by James Hodson

Integration with DEFUSE for event tagging

- Collaboration with Alessandro Pau @ EPFL

Integration with TokSearch for high level processing

Web user interface

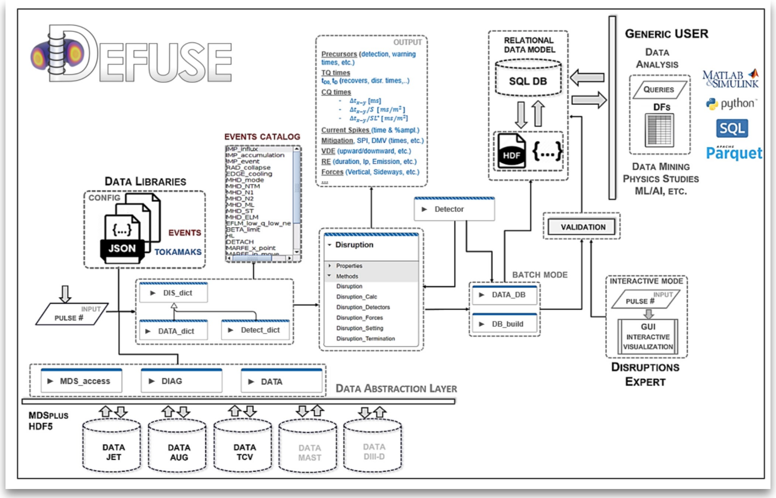
- Potentially looking at SciCAT

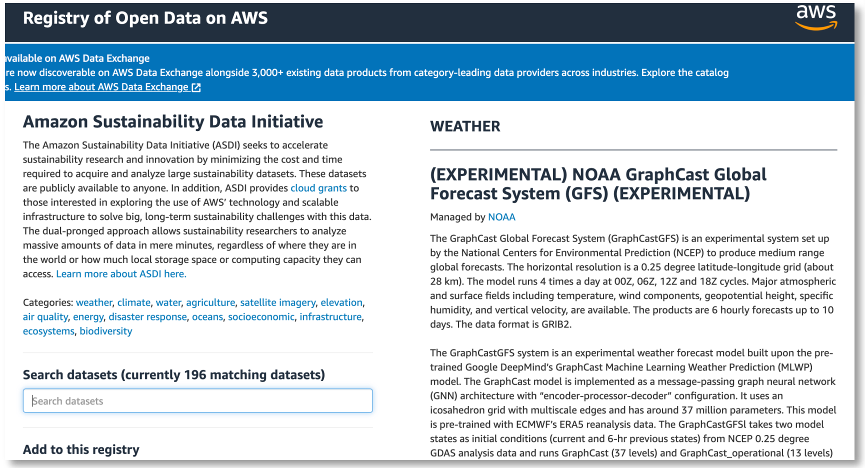
Data mirrors and hosting

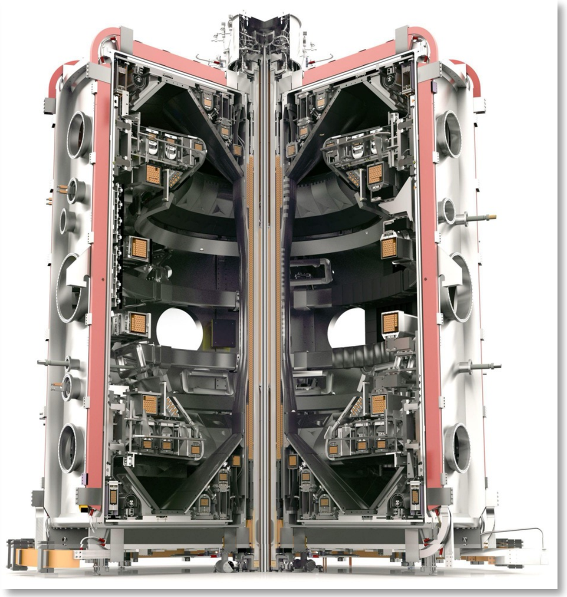
- AWS Sustainability Data Initiative
- A permanent home for metadata database

Rollout to MAST-U

- Authentication/hosting/data sharing needed for embargoed data
- Pipeline in development







Litaudon XL, et al. EUROfusion contributions to ITER Nuclear Operation. Nuclear Fusion. 2023.
SciCat
AWS Sustainability Initiative

Summary

Towards being FAIR

UK Atomic Energy Authority

"Perfect is the enemy of good" - Voltaire

FAIR Principle	Success	How?
Findable		
F1. (Meta)data are assigned a globally unique and persistent identifier		Yes. We assign UUID and S3 for each object. DOI etc. in future.
F2. Data are described with rich metadata (defined by R1 below)		Yes. All data have useful metadata accompanying them in file and in metadatabase
F3. Metadata clearly and explicitly include the identifier of the data they describe		Yes. Each item has a UUID as part of the metadata
F4. (Meta)data are registered or indexed in a searchable resource		Yes. Metadatabase APIs provide search and filtering
Accessible		
A1. (Meta)data are retrievable by their identifier using a standardised communications protocol		Yes. REST and GraphQL APIs support this
A1.1 The protocol is open, free, and universally implementable		Yes.
A1.2 The protocol allows for an authentication and authorisation procedure, where necessary		Yes, in future: ACL & Keycloak
A2. Metadata are accessible, even when the data are no longer available		Yes, metadatabase record
Interoperable		
I1. (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.		Yes. Metadata schema is on site. Ongoing UKAEA schema work.
I2. (Meta)data use vocabularies that follow FAIR principles		Ongoing UKAEA schema work
I3. (Meta)data include qualified references to other (meta)data		No. Ongoing UKAEA schema work.
Reusable		
R1. (Meta)data are richly described with a plurality of accurate and relevant attributes		Yes. But more work to do!
R1.1. (Meta)data are released with a clear and accessible data usage license		Yes.
R1.2. (Meta)data are associated with detailed provenance		No. But we have fusion-prov tool to extract this.
R1.3. (Meta)data meet domain-relevant community standards		No. Ongoing IMAS mapping work

GO FAIR: https://www.go-fair.org/fair-principles/

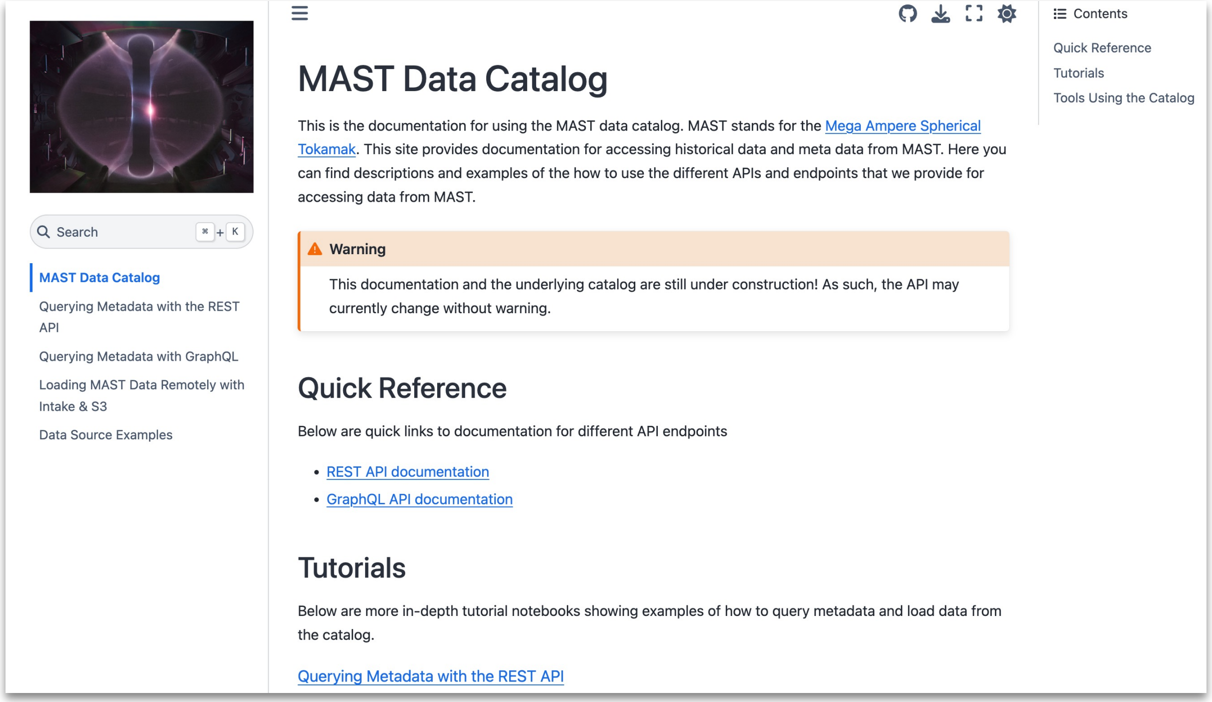
Summary

UK Atomic Energy Authority

We developed a data infrastructure solution for the history of the MAST experiment

We provide a public REST API for the metadata

We provide a public the history of the MAST data in cloud object storage



Test site:
https://mastapp.site/

With Thanks

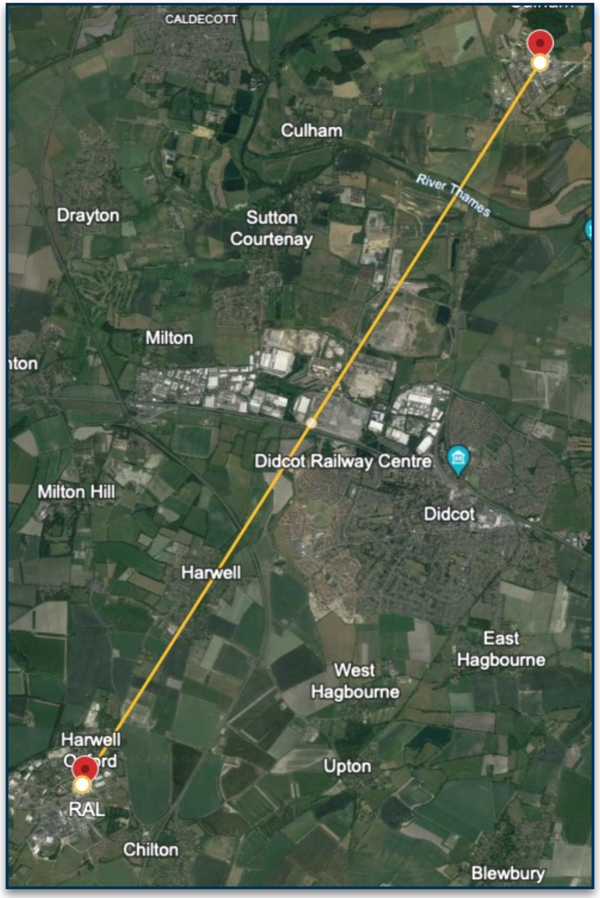
A cross-organisation collaboration between STFC and UKAEA and was funded as part of the Fusion Computing Lab programme.

STFC

Saiful Khan

Jeyan Thiyagalingam

Rutherford Appleton Laboratory



Culham Centre for Fusion Energy

UKAEA

Samuel Jackson

Nathan Cummings

James Hodson

Shaun De Witt

Stanislas Pamela

Rob Akers

With special thanks to Jonathan Hollocombe, Stephen Dixon, Jimmy Measures, Lucy Kogan, Adam Parker, Deniza Chekrygina, Alejandra Gonzalez-Beltran and the STFC Cloud and STFC Data Services Groups.

(C) Google (2024) Didcot Area, Accessed June 2024