



# StraboExperimental / LAPS



**Matej Pec**  
**Ulrich Mok**  
**Ekaterina Bolotskaya**  
**Ben Holtzman**

**Andreas Kronenberg**  
**Noah Phillips**  
**Julie Newman**  
**Hannah Cunningham**

**Basil Tikoff**  
**Jason Ash**





## What is StraboExperimental?



A **digital** repository for experimental rock deformation data. As part of the Strabo Ecosystem, it utilizes features from Strabo **in addition** to capabilities specifically for experimental data.

## What is LAPS?

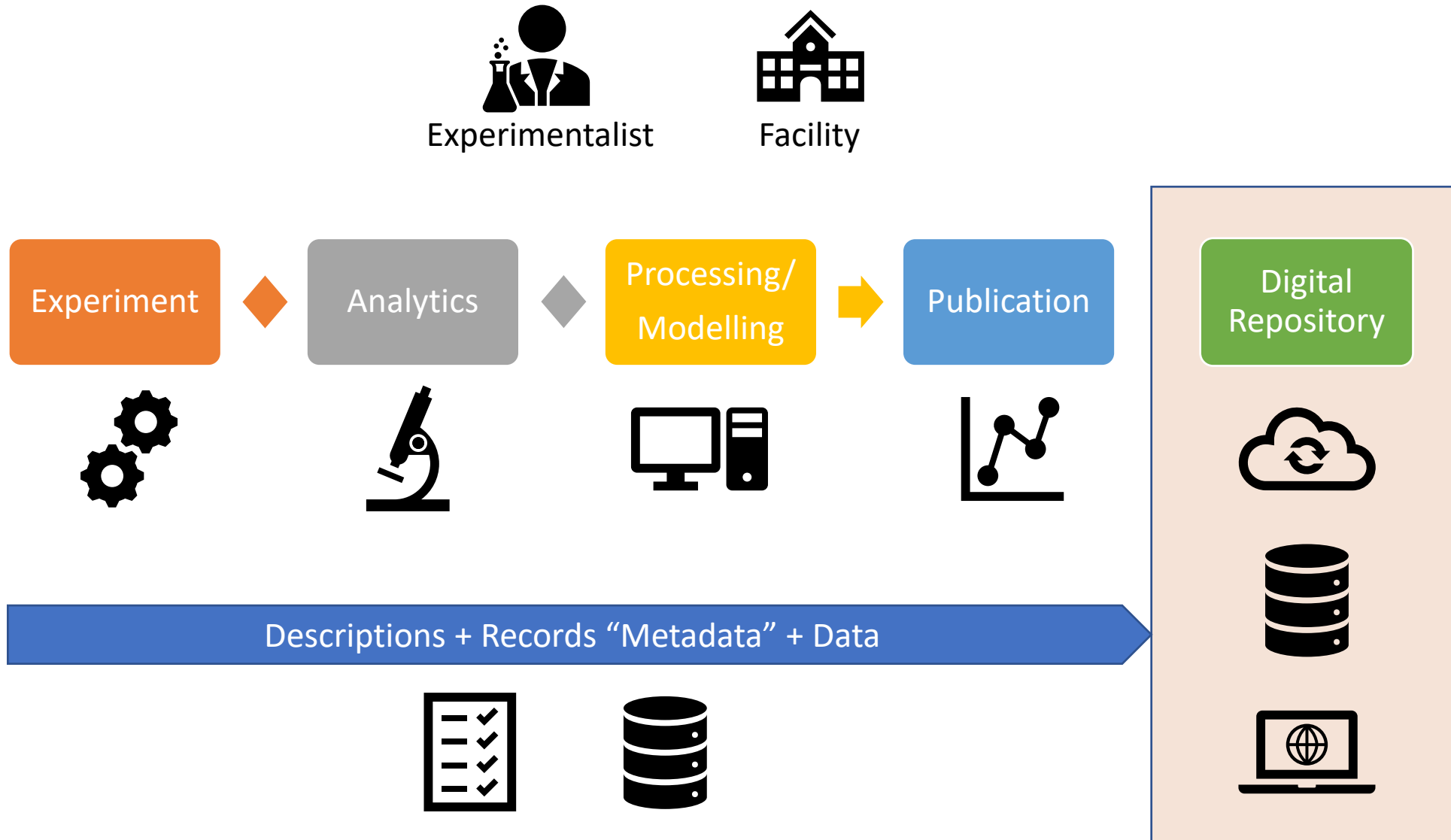


\*Laboratory Acquisition & Protocol Standards

LAPS is a framework for preparing and storing experimental data **locally**. It comprises a set of instructions and workflow tools for gathering and organizing experimental results. Functionality includes import/export features and compatibility with **StraboExperimental**.



# LAPS - Experimental Workflow





## A. Technical Challenges

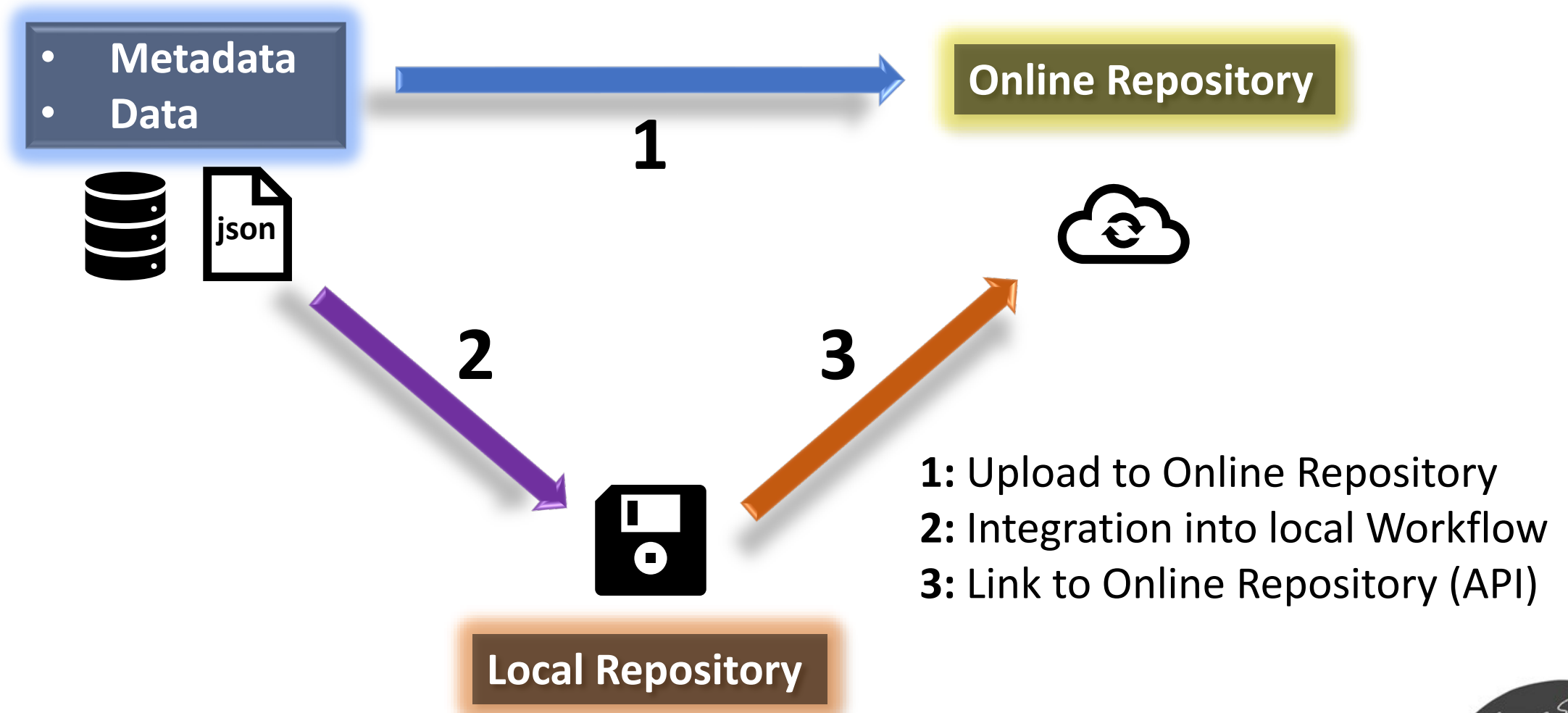
- Fragmented Data and Formats (Digital & Analog)
- Repository: Database Maintenance and Longevity
- Data Access (Software, API)
- Processing Tools

## B. Societal Challenges - Community Standards

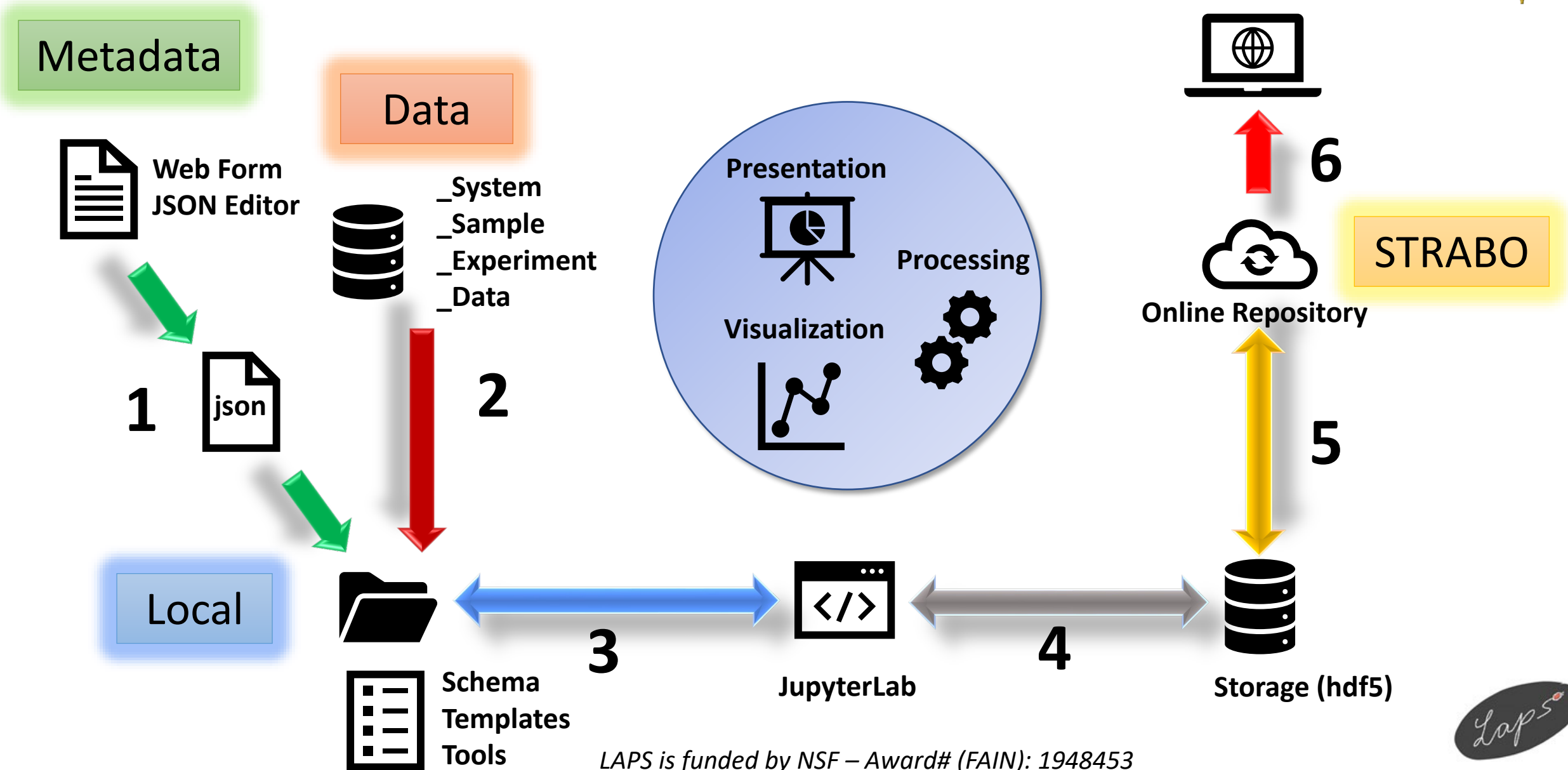
- No Metadata Standard
- Which data to include (processed-unprocessed)?
- Legacy Data, Pilot Data (published + unpublished)
- Data Quality Criteria
- Copyright, Citations, Privacy Issues
- Time Management



# Data Management - Options



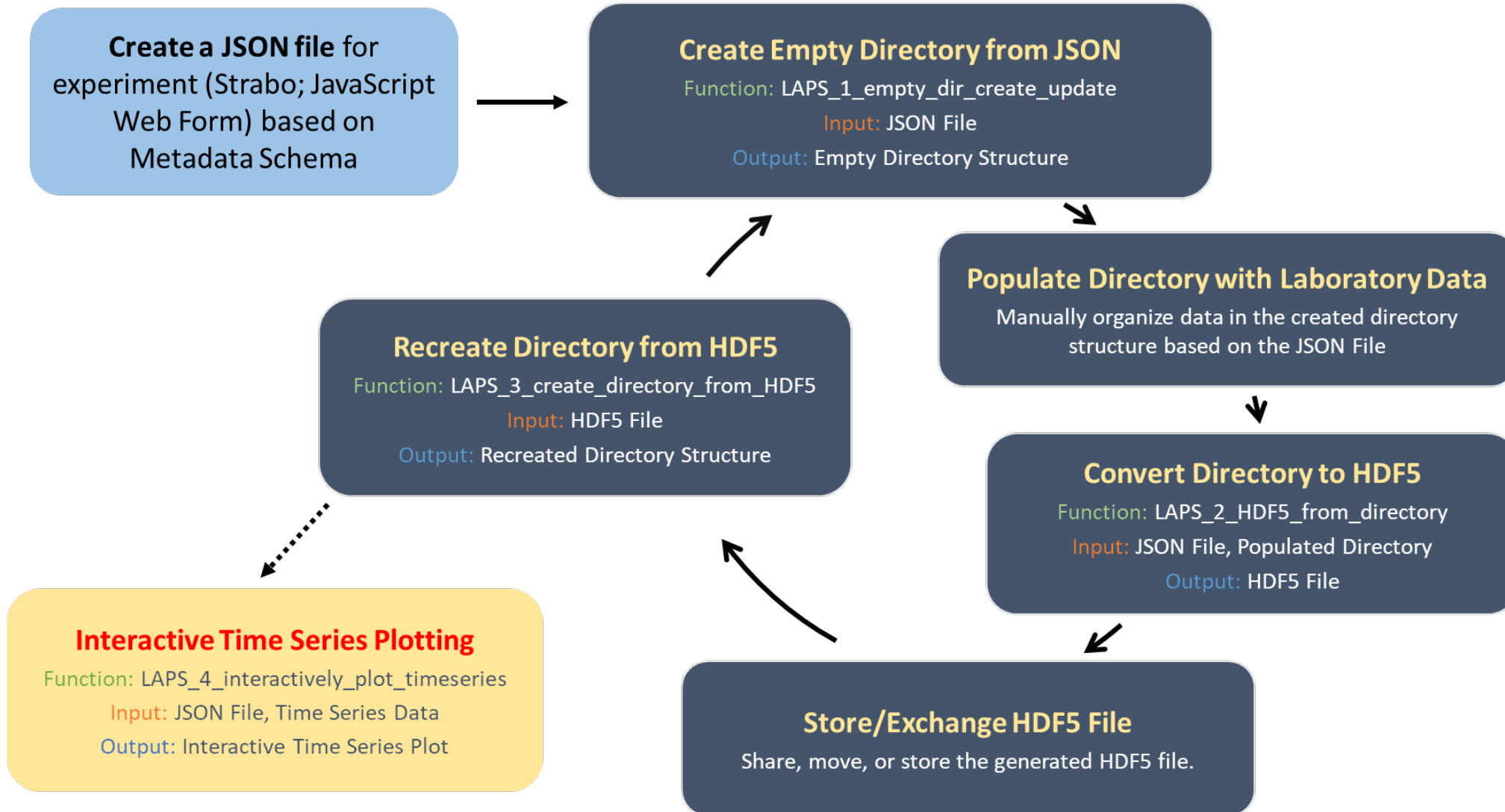
# LAPS - Workflow Details



LAPS is funded by NSF – Award# (FAIN): 1948453

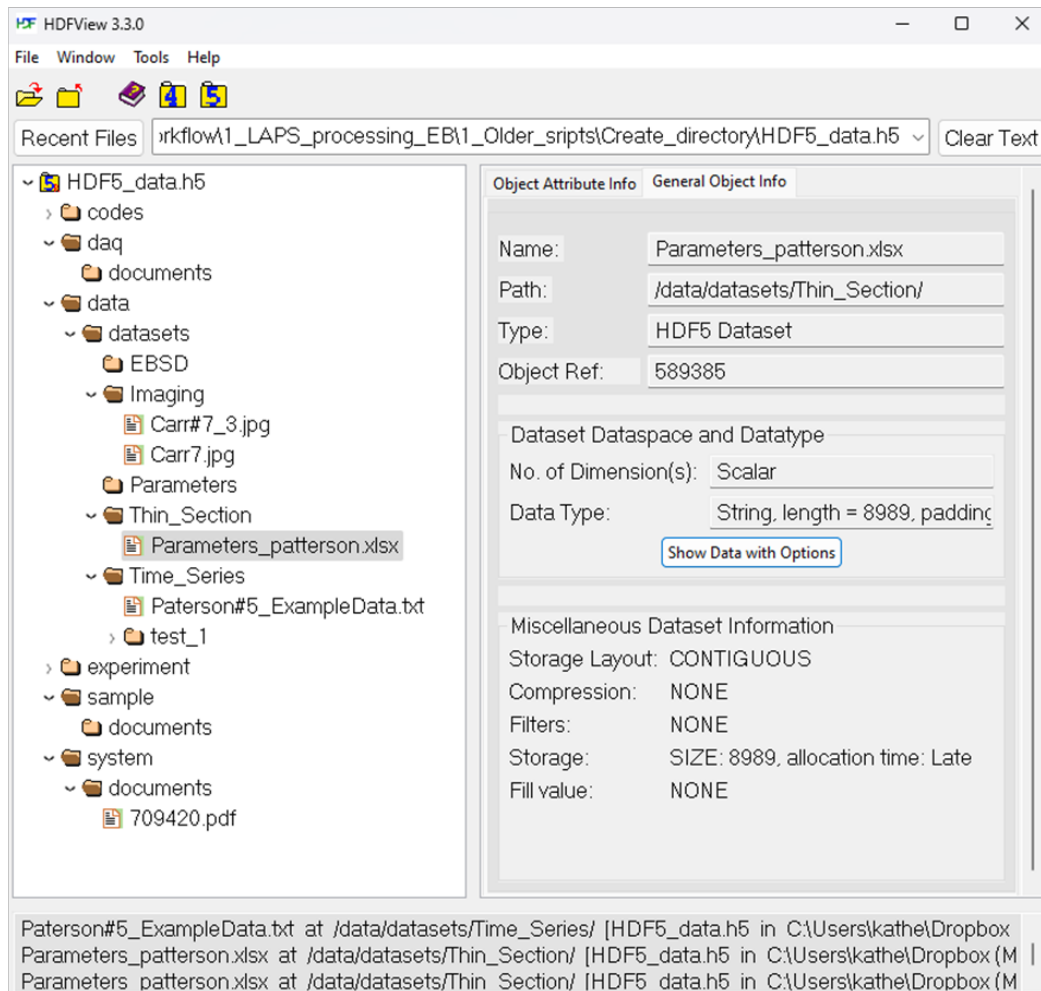
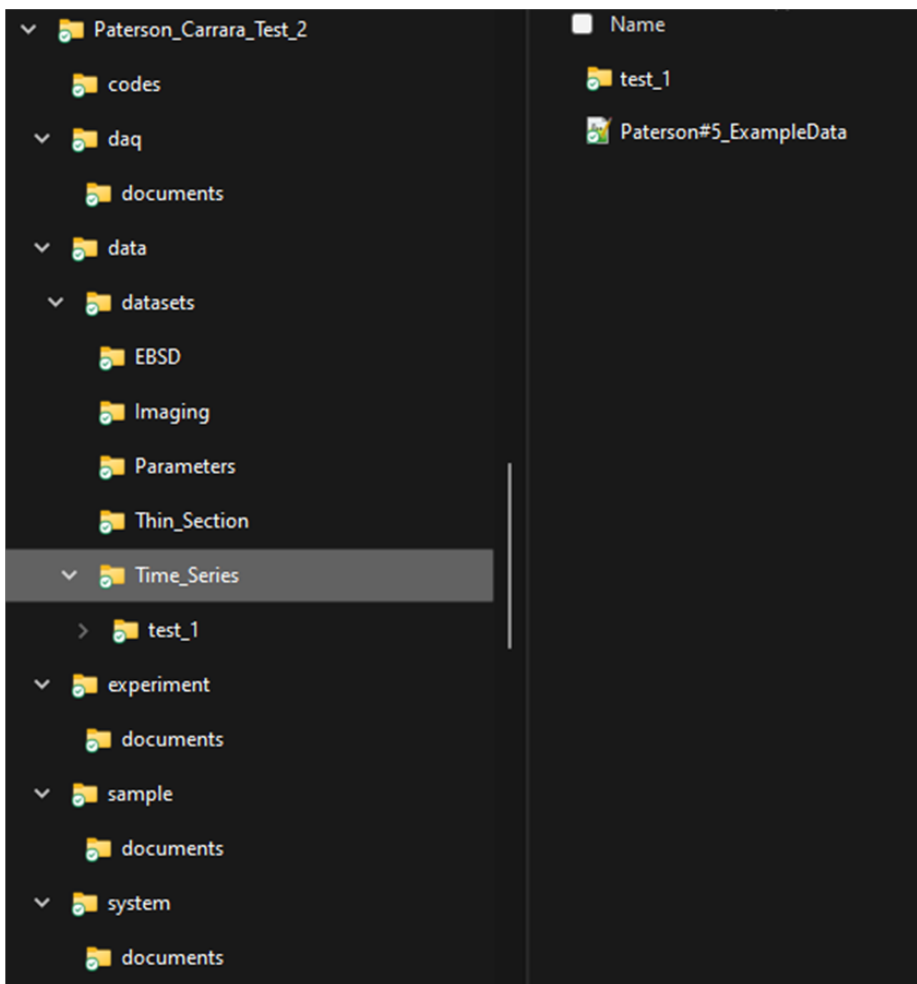


# LAPS – Anaconda/JupyterLab/Python Tools





## Folder structure to HDF5 file structure





# LAPS – Python Tools/Interactive Plots (Example)



## Interactive time series data plotting

Project: Yield Point Study Carrara  
Apparatus: Paterson Rig #5  
No sample name

Datasets:

Time\_Series

- Time Relative Differential , hour
- Time Relative Differential , sec
- Load Axial Internal , KN
- Displacement Axial External High Gain (Fine) , mm
- Displacement Axial External Low Gain (Coarse) , mm
- Displacement Volumetric Pore Volume , mm
- Pressure Pore Upstream , MPa
- Pressure Pore Downstream , bar
- Load Axial External , KN
- Pressure Confining Vessel , MPa
- Temperature Sample Average , deg C

Parameters

No headers

Imaging

No headers

Thin\_Section

No headers

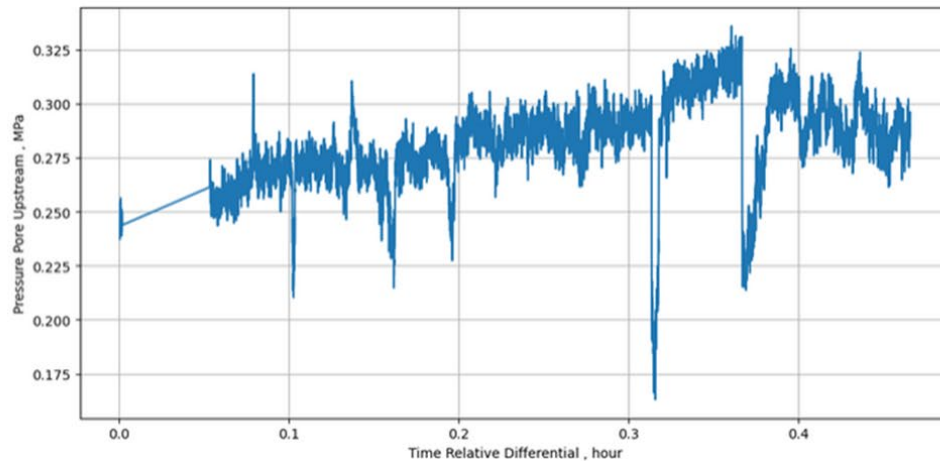
EBSD

No headers

X-axis: Time Relative Differential , hour

Y-axis: Pressure Pore Upstream , MPa

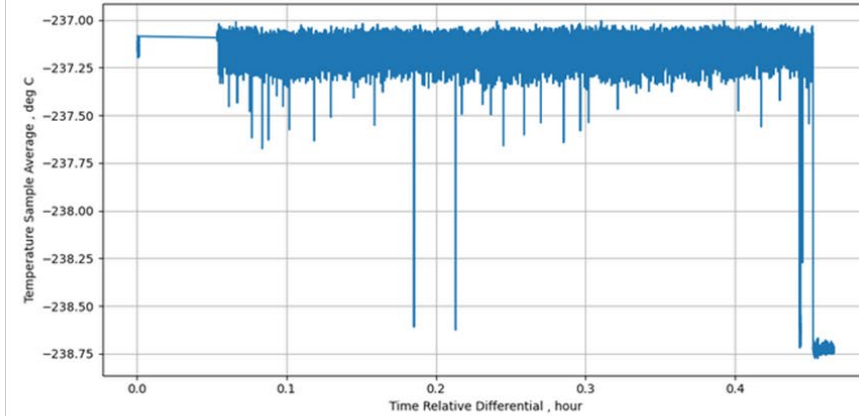
Plot



X-axis: Time Relative Differential , hour

Y-axis: Temperature Sample Average , deg C

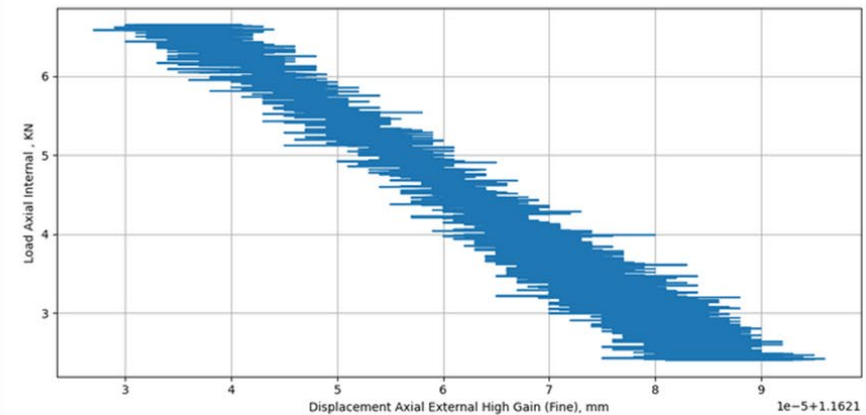
Plot



X-axis: Displacement Axial External High Gain (Fine) , mm

Y-axis: Load Axial Internal , KN

Plot





## In Development

- Strabo Metadata/Data upload and download via API
- Parsing tools for other Open Data Repositories
- Data Processing Tools (e.g., Calibrations)
- Acoustic Emissions Machine Learning Code
- Time Synchronization (between DAQ Systems)



# Comparison - Table



	STRABO	LAPS
Digital Repository	✓	✓ (single Test only)
Public Access	✓ (online)	✗ (offline)
Access Control	✓	✗
Metadata Standards	✓	✓
Structured Data	✓	✓ (json+hdf5)
Open Source	○	✓
Data Entry Forms	✓	✓
Database	✓	✗
External Data Reference	✓	✓
Workflow Integration	✗ (standalone)	✓
Strabo Integration	✓	○
API/Import Export	✓ / ✓	✗ / ✓
Readiness	2024	2024



# LAPS – Python/Html/Javascript Form Tool



```
[4]: ## LAPS project data conversion, storage, and plotting based on the
# Ekaterina Bolotskaya
# 07/17/2023

# -----
## Import modules
# -----

import os
import json
import h5py
import numpy as np

[5]: # -----
## Import functions
# -----

from LAPS_1_empty_dir_create_update import *
from LAPS_2_HDFS_from_directory import *
from LAPS_3_create_directory_from_HDFS import *
from LAPS_4_interactively_plot_timeseries import *

[6]: # -----
## Create the initial directory structure based on the JSON schema
# -----
## Define paths (platform independent)
cwd = os.getcwd() # get the current directory
schema_path = os.path.join(cwd, 'Test_input_output_data', 'Paterson5_CarraraTest.json')
empty_dir = os.path.join(cwd, 'Test_input_output_data', 'Paterson_Carrara_Test')

## Call the function to create the initial directory structure based on the JSON schema
LAPS_1_empty_dir_create_update(schema_path, empty_dir, json, os)

Empty directory created successfully. Please populate it with data for further processing.
C:\Users\ul1\Documents\AEdata_to_h5\LabData\workflow\1_LAPS_processing_EB\2_dir_to_HDFS_to_dir_v2\Test_input_output_data\Paterson_Carrara_Test
```

### Apparatus Form

Apparatus (DUT) Information (Description, Features, Parameter, Manuals and Documents)

Apparatus Name	Apparatus Type	Location	Apparatus ID
Paterson Rig #5	Triaxial (conventional)	E54-715	Pat_54_715

**Description**

This apparatus is capable of reaching confining pressures up to 500 MPa and temperatures up to 1300°C with Argon gas serving as the confining medium. It was designed by Prof. Mervyn Paterson and is one of 13 existing apparatus of this type world-wide. This apparatus is uniquely suited for high-resolution deformation experiments of relatively large samples up to 15 mm in diameter and 30 mm in length thanks to its internal load cell and compensated piston design which eliminates any seal friction from load measurements. The apparatus has a three-zone furnace with temperature gradients of <math>1\text{ }^\circ\text{C}/\text{mm}</math> and a hot zone <math>40\text{ mm}</math> in length.

**Apparatus Features:**

- Loading  Unloading  Heating  Cooling  High Temperature  Ultra-High Temperature  Low Temperature
- Sub-Zero Temperature  High Pressure  Ultra-High Pressure  Hydrostatic Tests  HIP  Synthesis  Deposition/Evaporation
- Mineral Reactions  Hydrothermal Reactions  Elasticity  Local Axial Strain  Local Radial Strain  Elastic Moduli
- Yield Strength  Failure Strength  Strength  Extension  Creep  Friction  Frictional Sliding  Slide Hold Slide
- Stepping  Pure Shear  Simple Shear  Rotary Shear  Torsion  Viscosity  Indentation  Hardness  Dynamic Tests
- Hydraulic Fracturing  Hydrothermal Fracturing  Shockwave  Reactive Flow  Pore Fluid Control  Pore Fluid Chemistry
- Pore Volume Compaction  Storage Capacity  Permeability  Steady-State Permeability  Transient Permeability
- Hydraulic Conductivity  Drained/Undrained Pore Fluid  Uniaxial Stress/Strain  Biaxial Stress/Strain  Triaxial Stress/Strain
- Differential Stress  True Triaxial  Resistivity  Electrical Resistivity  Electrical Capacitance  Streaming Potential
- Acoustic Velocity  Acoustic Events  P-Wave Velocity  S-Wave Velocity  Source Location  Tomography  In-Situ X-Ray
- Infrared  Raman  Visual  Other

Select Apparatus Test Capabilities

**Parameters**

System Parameters and Limits

**Documents**





The screenshot shows a JupyterLab interface. On the left is a file browser showing a directory structure: / ... / Test\_input\_output\_data / Paterson\_Carrara\_Test / sample (4 minutes ago), facility (4 minutes ago), experiment (4 minutes ago), data (4 minutes ago), daq (4 minutes ago), and apparatus (4 minutes ago). The main area displays a README.txt file with the following content:

```
65 You can either manually fill the form or use templates to generate the required .json file.
66 Save the created .json schema file to a location on your computer.
67
68 2. Place Main.ipynb and the provided function files (`LAPS_1_empty_dir_create_delete.py`,
69 `LAPS_1_empty_dir_create_update.py`, `LAPS_2_HDF5_from_directory.py`,
70 `LAPS_3_create_directory_from_HDF5.py`, `LAPS_4_interactively_plot_timeseries.py`)
71 in the same directory (not necessarily where you want to manage your laboratory data).
72
73 3. Specify the path to the JSON schema file: schema_path.
74 This file defines the structure of the data to be stored. Specify the name for the empty
75 directory to be created: empty_dir.
76
77 4. Run either version of **LAPS_1_empty_dir_create** function to create the initial directory
78 structure based on the JSON schema.
79 This function will create an empty directory structure in a folder named empty_dir.
80 Please see comments above on the difference between the two versions if rerunning the function.
81
82 5. Populate the generated directory structure with your laboratory data.
83 You can organize the data according to the subfields defined in the JSON schema.
84 If you want these scripts to be part of the HDF5, add them into the directory.
85
86 6. Specify the path to the populated directory: popul_dir (if you populated the original
87 directory, the path should be the same as empty_dir).
88 Specify the name for the HDF5 file to be created: hdf5_file.
89
90 7. Run the **LAPS_2_HDF5_from_directory** function to convert the populated directory structure
91 into an HDF5 file.
92 This function will read data from the populated directory, generate an HDF5 file that
93 represents the same structure, and dump the data there as binary files.
94
95 8. The HDF5 file can now be stored, exchanged, or used for further analysis.
96 You may choose to move it to a different location or share it with others.
97
98 9. To reproduce the original directory structure from the HDF5 file, specify the directory name:
99 output_dir,
100 the file name for the recreated schema: recr_scdir, and the HDF5 file name: hdf5_file.
101 Run the **LAPS_3_create_directory_from_HDF5** function. It will recreate the directory
102 structure.
```



# LAPS – Anaconda/JupyterLab/Python Tools



The screenshot shows the JupyterLab interface. On the left is a file browser with a search bar and a list of files and folders. The selected file is 'Main.ipynb'. The right pane shows the code editor for 'Main.ipynb' with Python code. The code is organized into cells, with comments and code blocks. The code includes imports for 'os', 'json', 'h5py', and 'numpy', and defines paths for the current working directory, schema path, and empty directory. It also calls a function to create the initial directory structure.

```
[4]: ## LAPS project data conversion, storage, and plotting based on the .json schema file
# Ekaterina Bolotskaya
# 07/17/2023

# -----
## Import modules
# -----
import os
import json
import h5py
import numpy as np

[5]: # -----
## Import functions
# -----
from LAPS_1_empty_dir_create_update import *
from LAPS_2_HDF5_from_directory import *
from LAPS_3_create_directory_from_HDF5 import *
from LAPS_4_interactively_plot_timeseries import *

[6]: # -----
## Create the initial directory structure based on the JSON schema
# -----
## Define paths (platform independent)
cwd = os.getcwd() # get the current working directory
schema_path = os.path.join(cwd, 'Test_input_output_data', 'Paterson5_CarraraTest.json')
empty_dir = os.path.join(cwd, 'Test_input_output_data', 'Paterson_Carrara_Test')

## Call the function to create the initial directory structure based on the JSON schema
LAPS_1_empty_dir_create_update(schema_path, empty_dir, json, os)

Empty directory created successfully. Please populate it with data for further processing.
C:\Users\uli\Documents\AEdata_to_h5\LabData\Workflow\1_LAPS_processing_EB\2_dir_to_HDF5_to_dir_v2\Test_input_output_data\Paterson_Carrara_Test
```

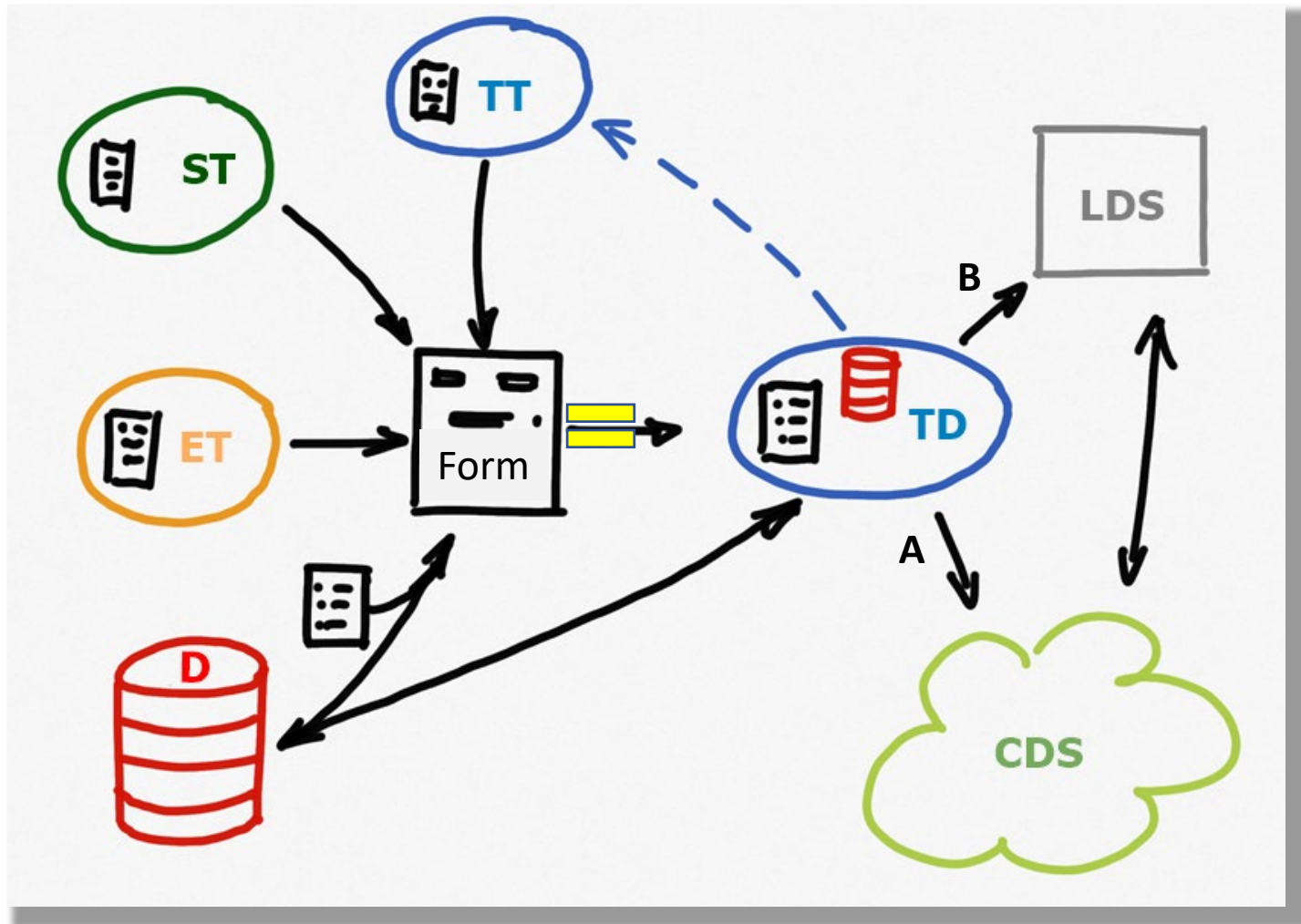
LAPS is funded by NSF – Award# (FAIN): 1948453



# Laboratory Workflow



## Combine Experimental Information + Data



ST: System Template  
ET: Experimental Template  
Sample Template  
D: Data

TD: Test Dataset  
TT: Test Template

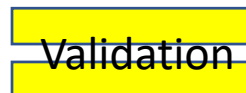
LDS: Local Data System  
CDS: Cloud Data System



Meta Data



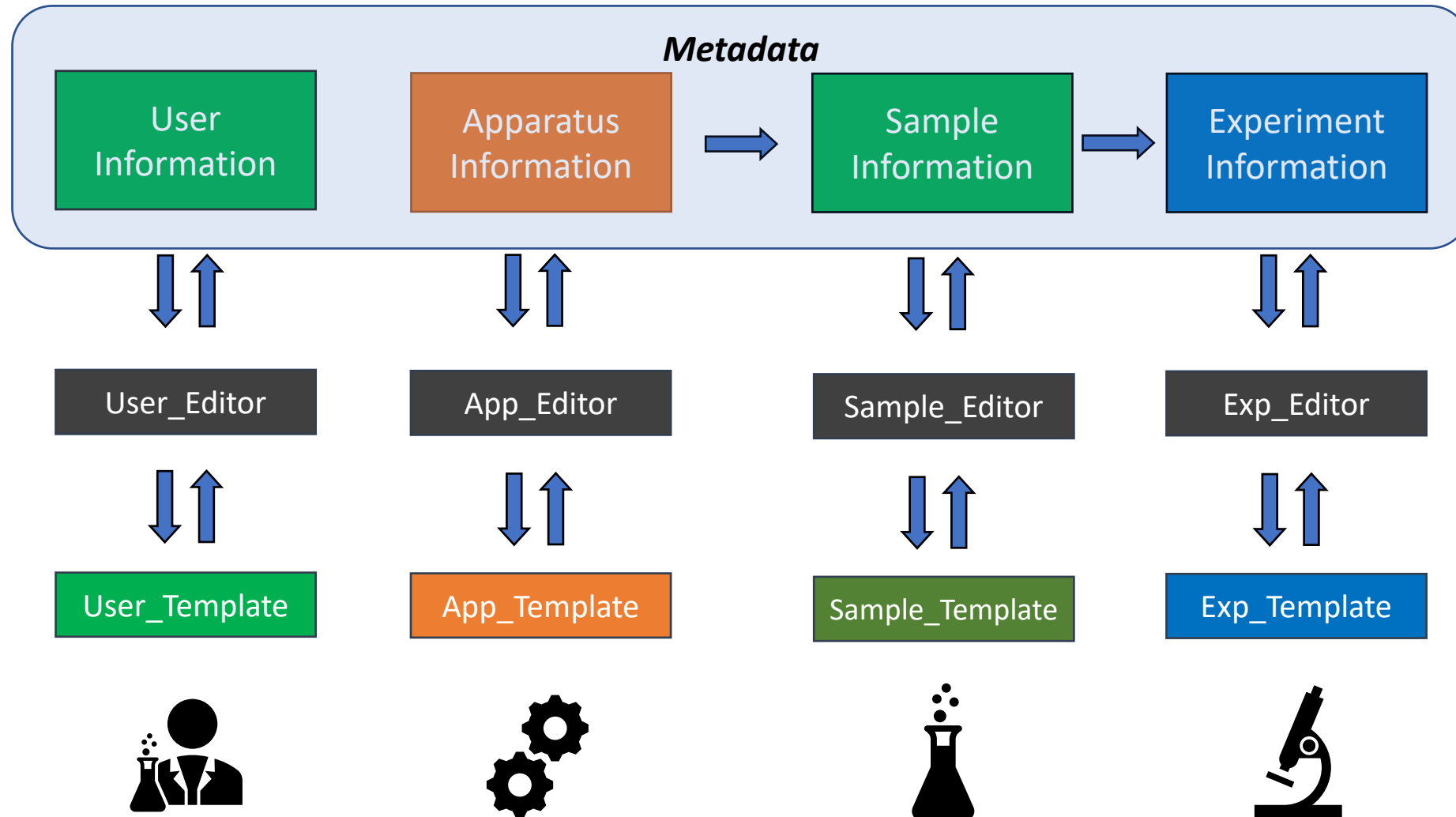
Data Set (e.g., Time Series)



Validation



## Templates simplify user input



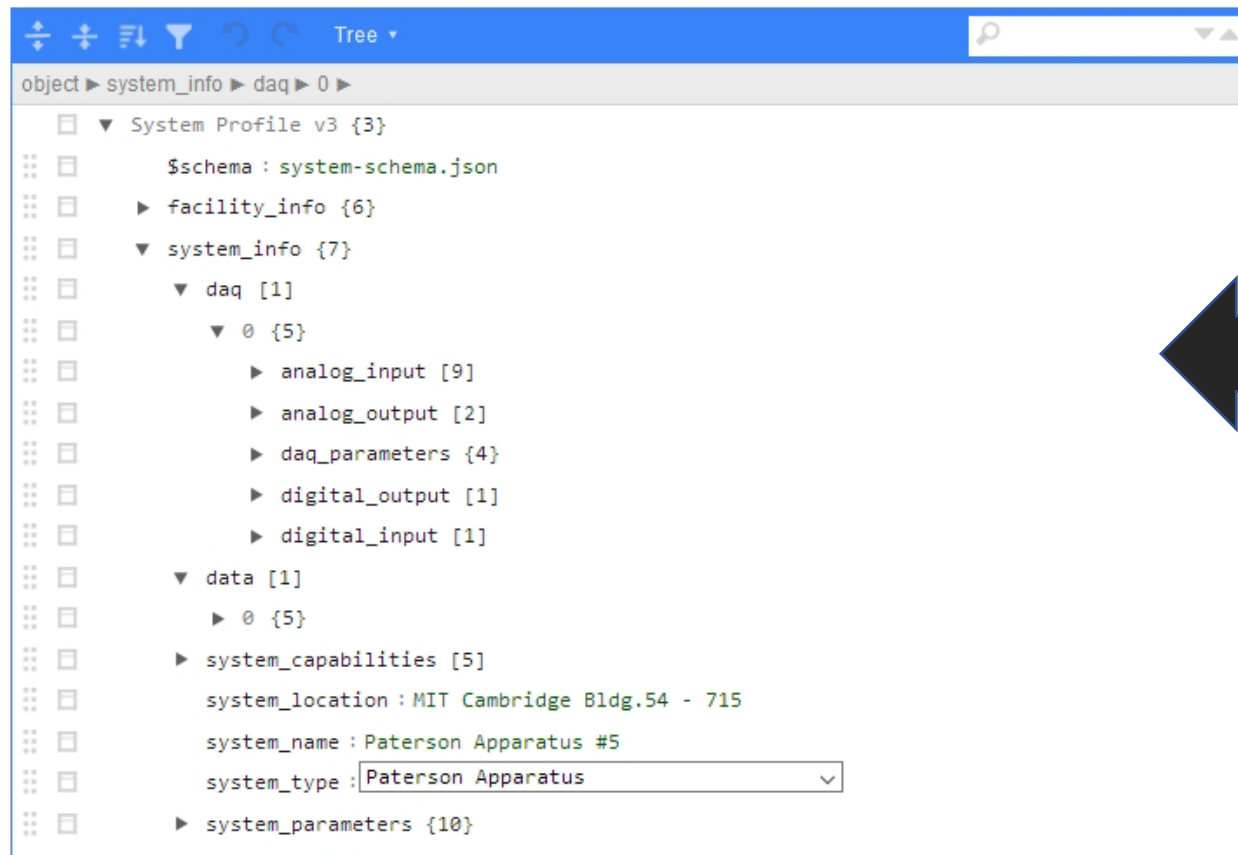


## Many ways to modify and edit Experimental Data and Metadata

Load JSON document:  No file selected.

Save JSON document:

Download MIT Machine Templates here: [Paterson#5](#) - [Permeameter](#)



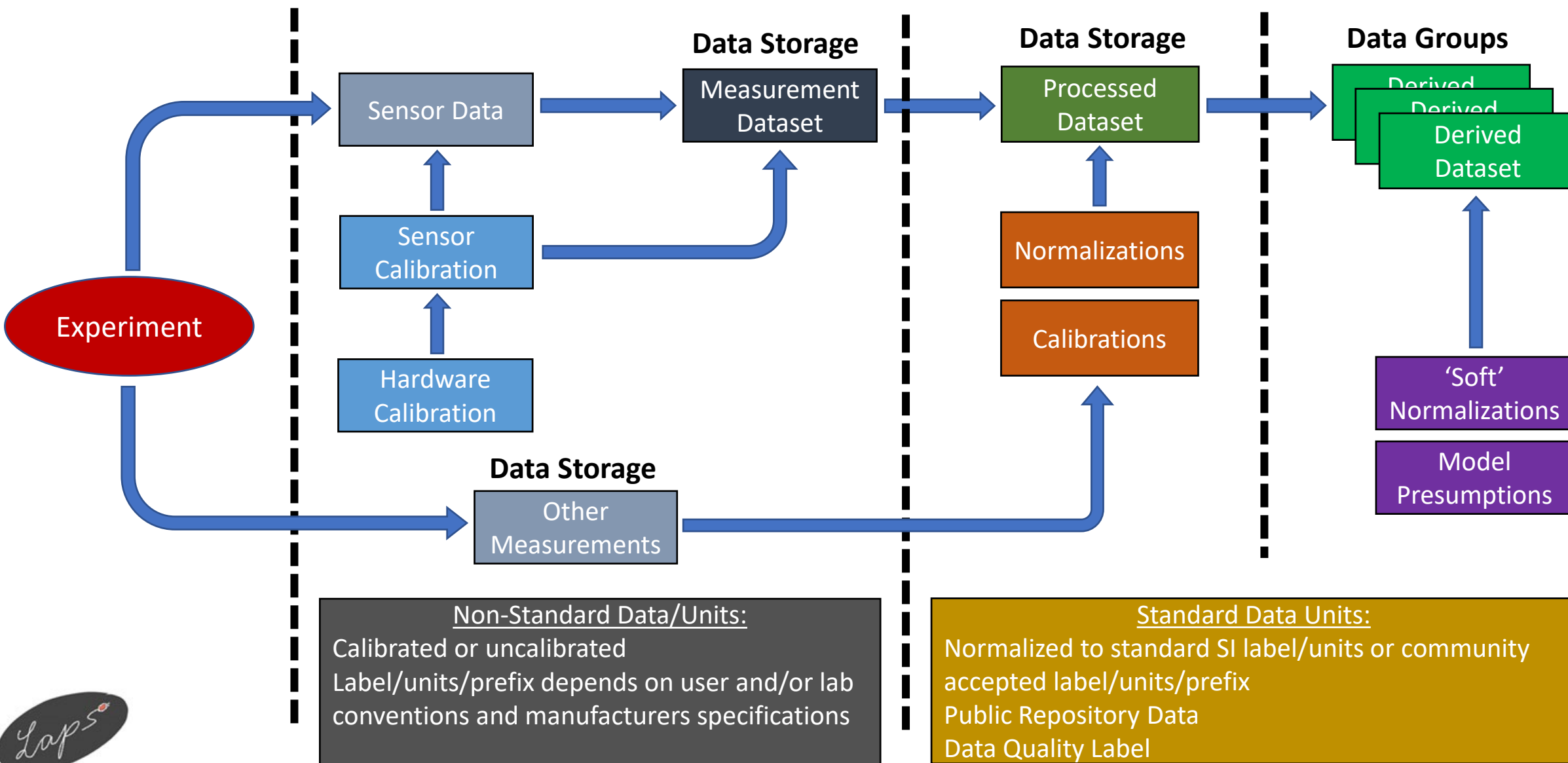
object ▶ system\_info ▶ daq ▶ 0 ▶

- System Profile v3 {3}
  - \$schema : system-schema.json
  - facility\_info {6}
  - system\_info {7}
    - daq [1]
      - 0 {5}
        - analog\_input [9]
        - analog\_output [2]
        - daq\_parameters {4}
        - digital\_output [1]
        - digital\_input [1]
    - data [1]
      - 0 {5}
    - system\_capabilities [5]
    - system\_location : MIT Cambridge Bldg.54 - 715
    - system\_name : Paterson Apparatus #5
    - system\_type : Paterson Apparatus
    - system\_parameters {10}



```
Paterson.json x experimental-schema.json Solution Explorer - Folder View
Schema: system-schema.json
1
2 "$schema": "system-schema.json",
3
4 "facility_info": {
5   "facility_address": "...",
14  "institute_name": "MIT",
15  "facility_type": "University Lab",
16  "facility_name": "Rock Physics Laboratory",
17  "facility_contact": "...",
25  "facility_id": "45re-tyu"
26
27 },
28 "system_info": {
29   "daq": [
30     {
31       "analog_input": "...",
177      "analog_output": "...",
207      "daq_parameters": {
208        "daq_name": "NI F",
209        "daq_manufacturer": "...",
210        "daq_resolution": "...",
211        "daq_samplingrate": "...",
212      },
213      "digital_output": "...",
214      "digital_input": "...",
483    },
484    "system_capabilities": [
485      "Acoustic Emission",
486      "Permeability",
487      "Stress Deformation",
488      "Strain Deformation",
489      "Hydrostatic Stress"
490    ],
491    "system_location": "MIT",
492    "system_name": "Paterson Apparatus #5",
493    "system_type": "Paterson Apparatus",
494    "system_parameters": {
495      "app_distortion": 75,
496      "f_max": 100,
497      "disp_min": 20,
498      "disp_max": 49,
499      "p_min": 0,
500      "p_max": 500,
501      "pp_min": 0,
502      "pp_max": 100,
503      "t_min": 20,
504      "t_max": 1400
505    }
506  ]
507 }
```

# Laboratory Data Workflow - Scenario





# Table Headers - Schema

