

# Advancing Open and Reproducible Water Data Science by Integrating Data Analytics with an Online Data Repository

**Jeffery S. Horsburgh**

Utah State University

**Scott Black, Anthony Castronova**

Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI)



Utah Water Research Laboratory  
UtahStateUniversity





# Reproducibility is key

*“If I have seen further it is by standing on the shoulders of Giants.”*

Isaac Newton, 1625

Building trust in scientific research requires transparency  
and reproducibility



# Collaborative (Reproducible) Data Science Workflow

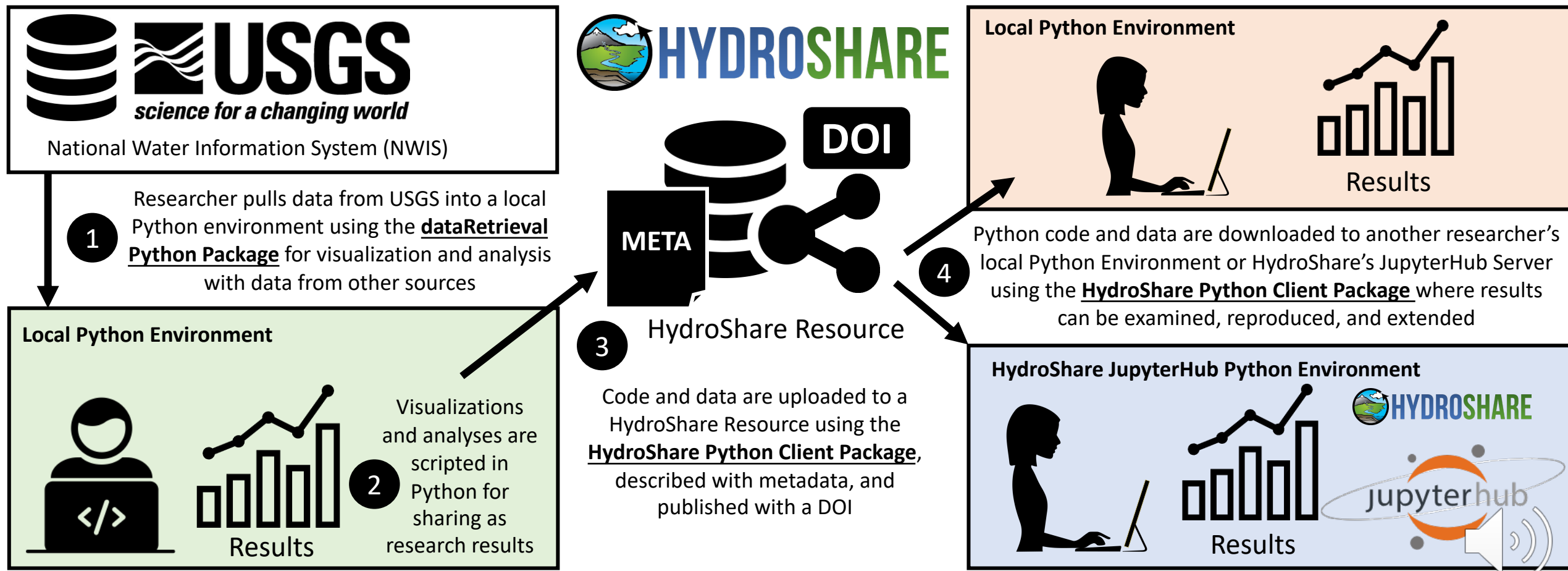
- Easily create a digital instance of a dataset or data science workflow
- Quickly share it with colleagues (perhaps privately at first)
- Add value through collaboration, annotation, and iteration
- Describe with metadata
- Eventually...share publicly or formally Publish so others can reuse



What is the role of data repositories in this scientific workflow?

# Connecting Visualization and Analysis with an Online Repository

- Better enabling collaborative data science workflows and reproducibility



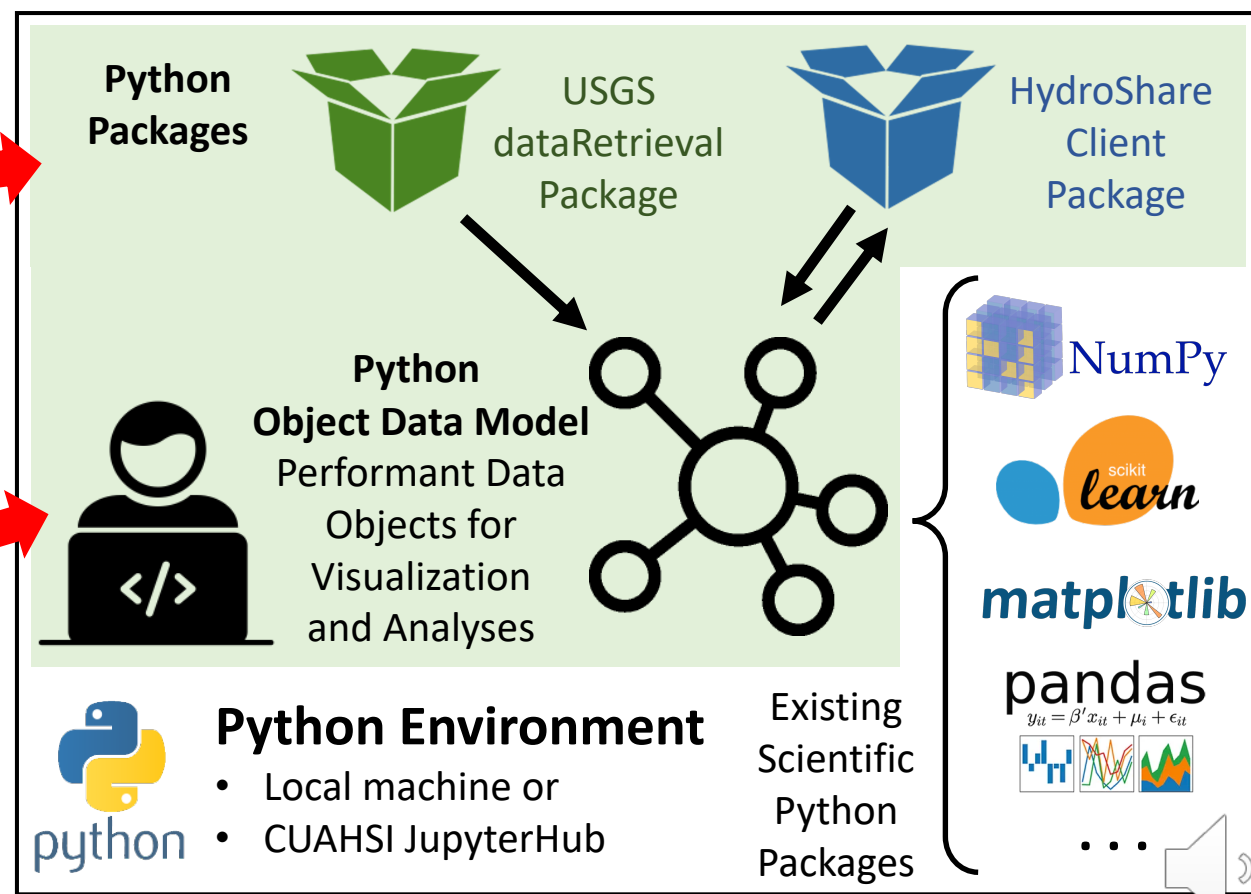
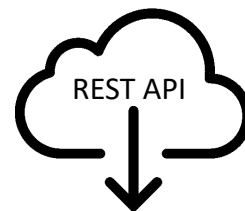


# The tools needed to make this work

Data repositories

Tools for accessing and interacting with those repositories

A Python representation of the data retrieved that can be operated on using existing data science tools





# HYDROSHARE

<http://www.hydroshare.org>

- A repository for sharing and publication that uses FAIR principles
- Operated by the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI)
- Creating and sharing data and models using a variety of file formats and flexible metadata
- Public-facing REST API and Python client enabling automated interactions

The screenshot displays the HydroShare website interface. The top navigation bar includes links for 'MY RESOURCES', 'DISCOVER', 'COLLABORATE', 'APPS', 'HELP', and 'ABOUT'. The main banner features a landscape image with a rainbow and the text 'Discover' and 'Discover content shared by your colleagues and other users. Access a broad range of resource types used in hydrology.'

Below the banner, the 'How it works' section is visible, with a numbered list starting with '1 Create data'. The text describes collecting data using the same methods as now, supported by a broad set of hydrologic data types. A blue icon with a plus sign is shown.

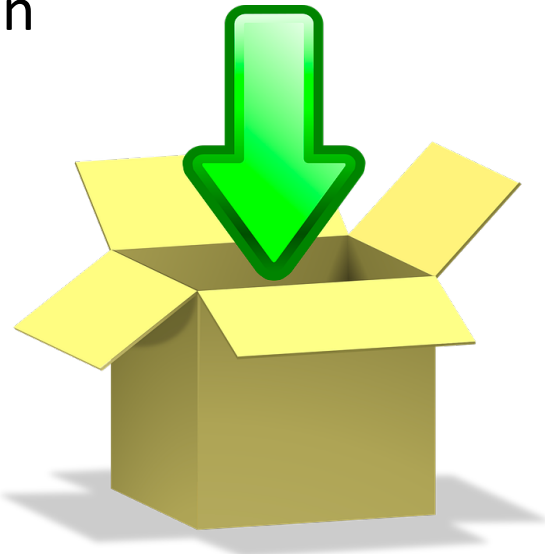
The 'What you can do' section lists several capabilities with green checkmarks: 'Share your data and models', 'Manage who has access to', 'Share, access, visualize and models', 'Use the web services API to', 'Publish data and models to plan', and 'Discover and access data and models'.

The right side of the screenshot shows a resource page titled 'Water Temperature in the Little Bear River at Mendon Road near Mendon, UT'. The page includes metadata such as 'Authors: Jeff Horsburgh, Amber Jones', 'Owners: Jeff Horsburgh', 'Resource type: Time Series', 'Created: June 6, 2015, 3:57 a.m.', and 'Last updated: June 6, 2015, 4:25 a.m. by Jeff Horsburgh'. The 'Abstract' section states: 'This dataset contains observations of water temperature in the Little Bear River at Mendon Road near Mendon, UT. Data were recorded every 30 minutes and represent the average values over the preceding time interval. The values were recorded using a HydroLab MS5 multi-parameter water quality sonde connected to a Campbell Scientific datalogger. Values represent quality controlled data that have undergone quality control to remove obviously bad data.' The 'Subject' section shows tags for 'Temperature', 'Water', 'Water quality', 'Little Bear River', and 'Utah'. The 'How to cite' section provides the citation: 'Horsburgh, J., A.Jones (2015). Water Temperature in the Little Bear River at Mendon Road near Mendon, UT, HydroShare, <http://www.hydroshare.org/resource/1a25b11fa1354773b6e9495e754f4e>'. The 'Sharing' section shows the status as 'Public' and the license as 'Creative Commons Attribution CC BY'. A 'Manage access' button is at the bottom.

# HydroShare “Resources”

- **Resource** = primary unit of digital content
  - Create, version, copy
  - Describe
  - Own, share, access
  - Discover
  - Formal Publication

A “Resource” is a container into which users can put digital content



Resources can be datasets, models, or other research products

My Resources

Type	Title	Owners	Last modified
Document	Survey of Stormwater Managers in Utah	iUTAH Data Manager	June 28, 2016, 6:25 p.m.
Document	Share and Publish your Data and Models with HydroShare	David Tarboton	June 28, 2016, 11:58 a.m.
Document	Cross section survey, Northwest Field Canal at 200 South	Jeffery Horsburgh	June 27, 2016, 7:40 p.m.
Figure	Water Temperature in the Little Bear River at Mendon Road near Mendon, UT	Jeffery Horsburgh	June 21, 2016, 7:14 p.m.
Model	HEC-HMS Version 4.1	David Tarboton	June 16, 2016, 8:20 p.m.
Dataset	Logan Digital Elevation Model	David Tarboton	June 14, 2016, 1:30 p.m.
Dataset	Logan Specific Catchment Area	David Tarboton	June 14, 2016, 1:30 p.m.
Figure	100yr flood every 3 years	Matthew Turner	May 3, 2016, 11:44 p.m.
Document	Hydrology Domain Cyberinfrastructures: Successes, Challenges, and Opportunities	Jeffery Horsburgh	Dec. 17, 2015, 9:07 a.m.
Document	CI-WATER Workshop - Tethys Provo Dam Break App Data	Nathan Swain	July 15, 2015, 8:56 p.m.

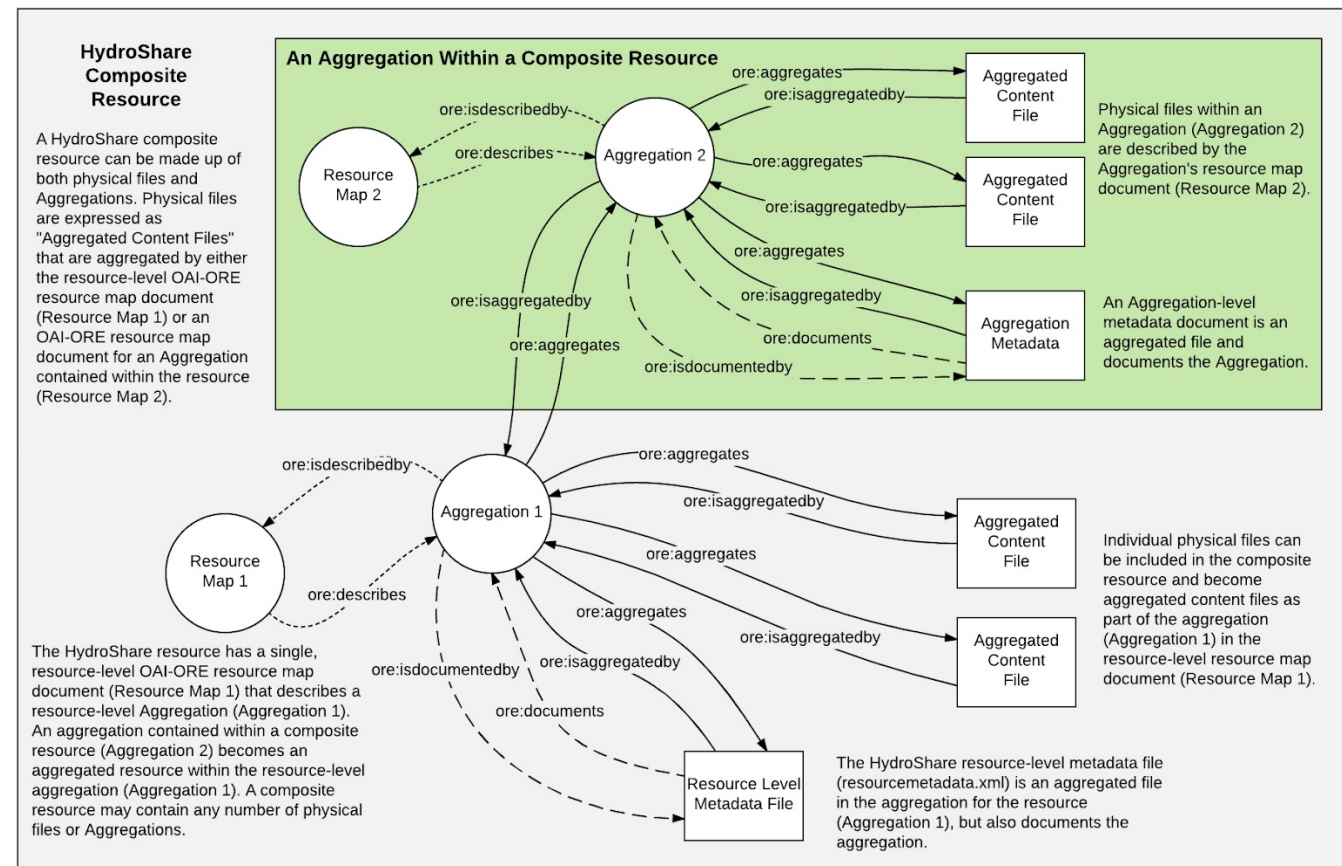
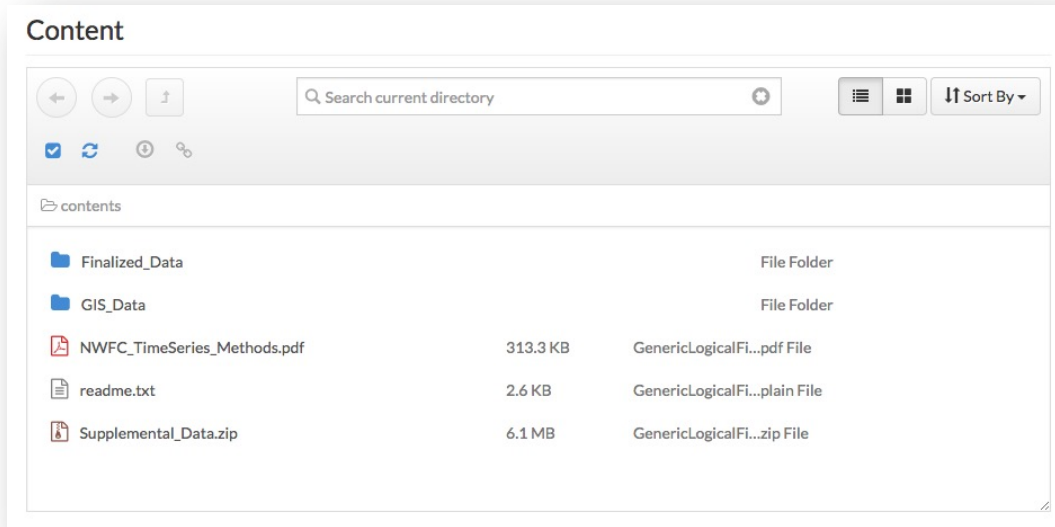
CONTACT US  
Email us at [hydroshare.org](mailto:hydroshare.org)

FOLLOW  
[Twitter](#) [Facebook](#) [YouTube](#) [LinkedIn](#)

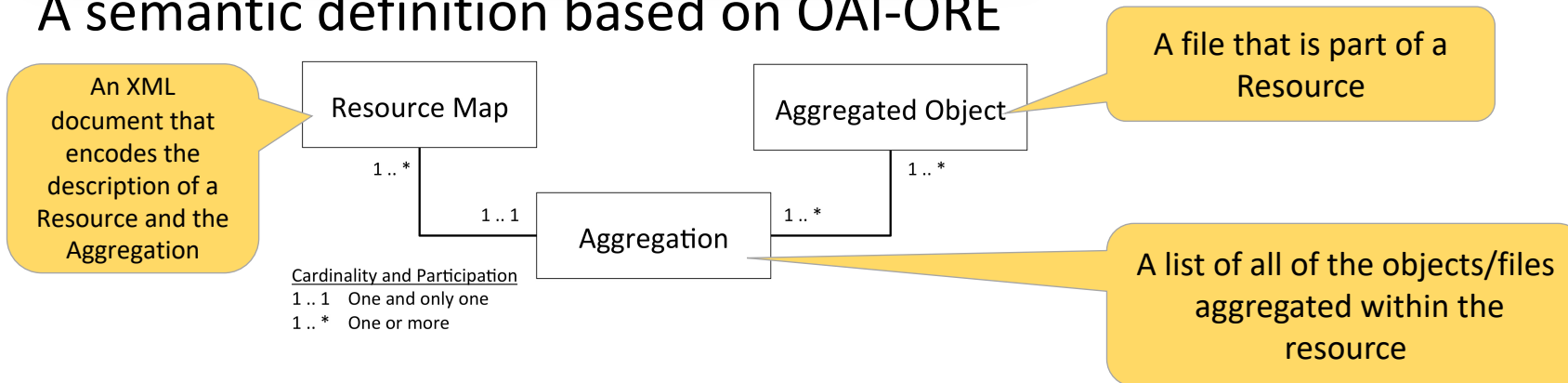
OPEN SOURCE  
HydroShare is Open Source. Find us on [GitHub](#).

# HydroShare “Resources”

## A file/content – based definition



## A semantic definition based on OAI-ORE



A profile of the Open Archives Initiative's Object Reuse and Exchange (OAI-ORE) standard



# HydroShare “Resource” as a Data Science Enabler

- When creating reproducible data science workflows – how to organize?
  - Eventual goal is to share the analysis
  - Need to be able to get data/code into a repository
  - Need straightforward ways to organize the content used
    - Potentially inputs, outputs, code, etc.
- The HydroShare Resource is a great organizing container
  - Think of it as a “Project Directory”
  - Existing Resource Data Model
  - Machine readable semantic representation of structure
  - Flexible
  - Existing “aggregation” types identify commonly used data
  - Can map the whole thing to Python for easy manipulation
  - Already handled by a repository!

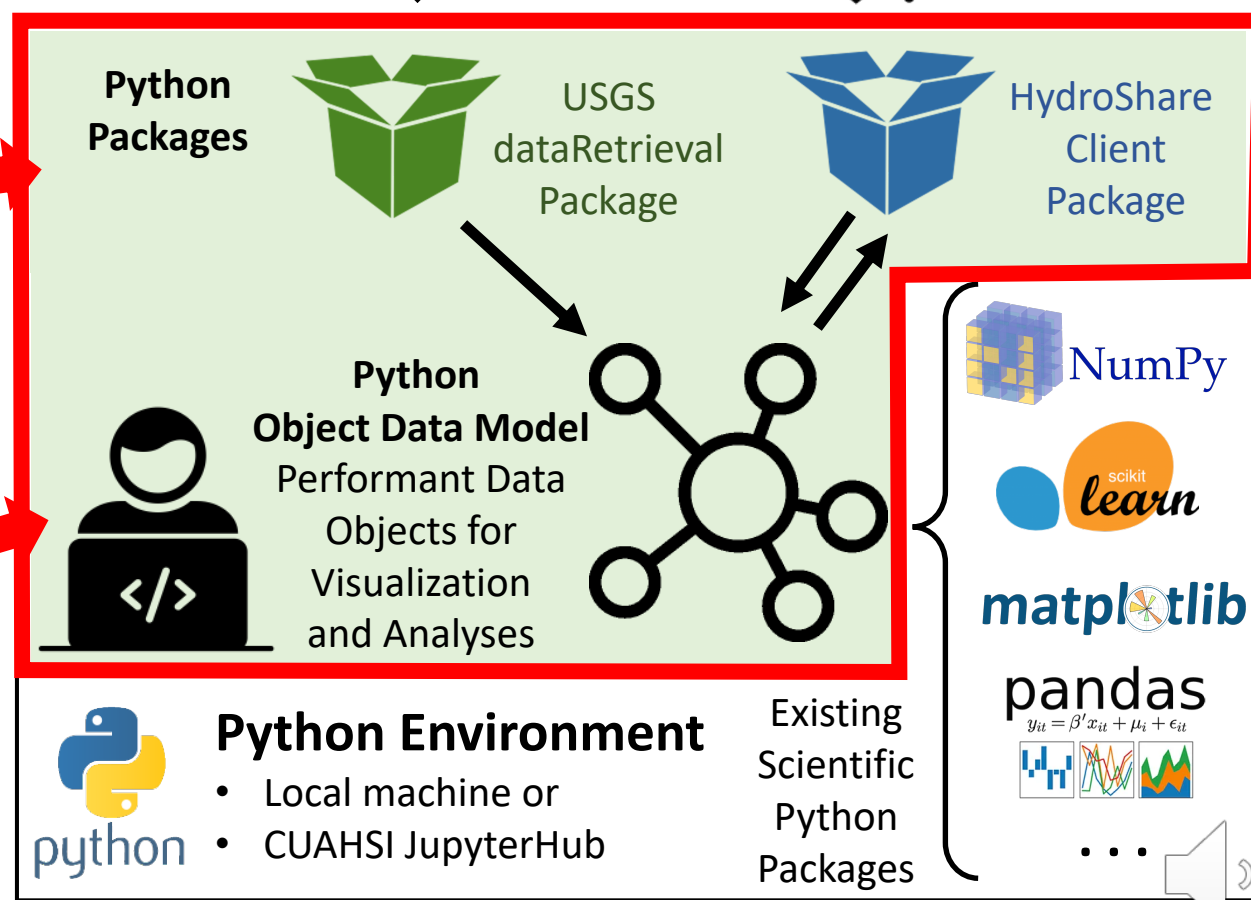


# The tools needed to make this work

Data repositories

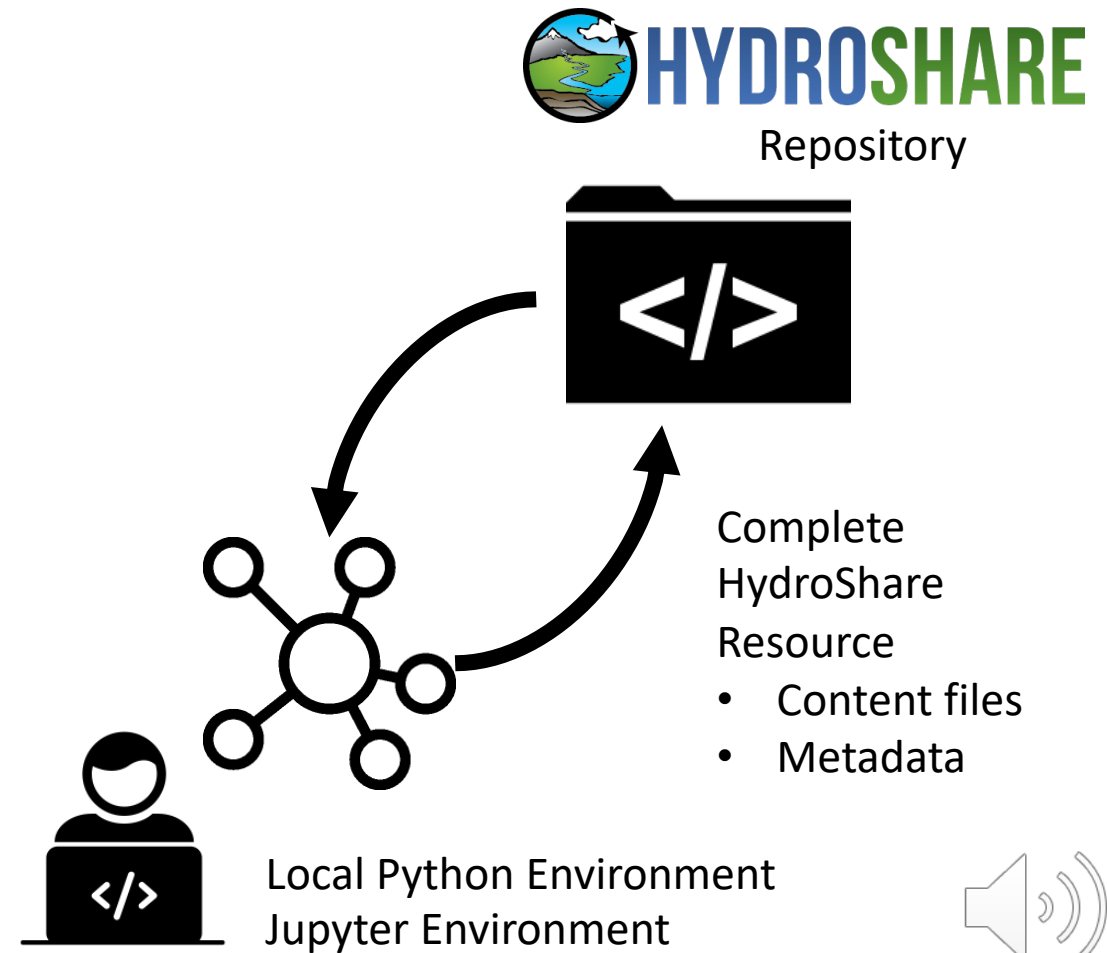
Tools for accessing and interacting with those repositories

A Python representation of the data retrieved that can be operated on using existing data science tools



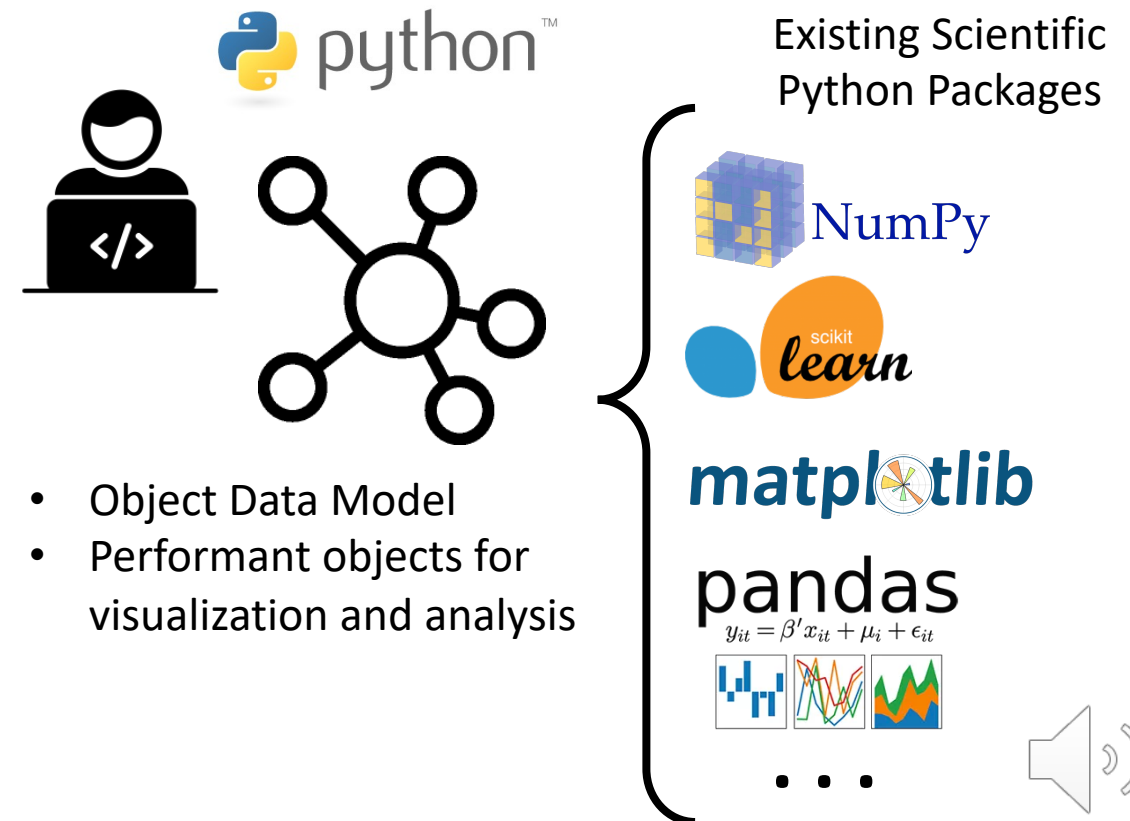
# HydroShare Python Client (hsclient)

- A set of Python functions for interacting with the HydroShare repository
- (Object) Structure of HydroShare resources is specified in the OAI-ORE RDF/XML resource map documents
- hsclient translates this structure to a Python object representation
  - Read the structure and metadata of a resource into Python objects
  - Manipulate it in your Python environment (local or Jupyter)
  - Save that structure back to HydroShare
  - Modify RDF/XML outside of HydroShare and send those files back to be ingested



# A flexible water-data science object data model (hsmodels)

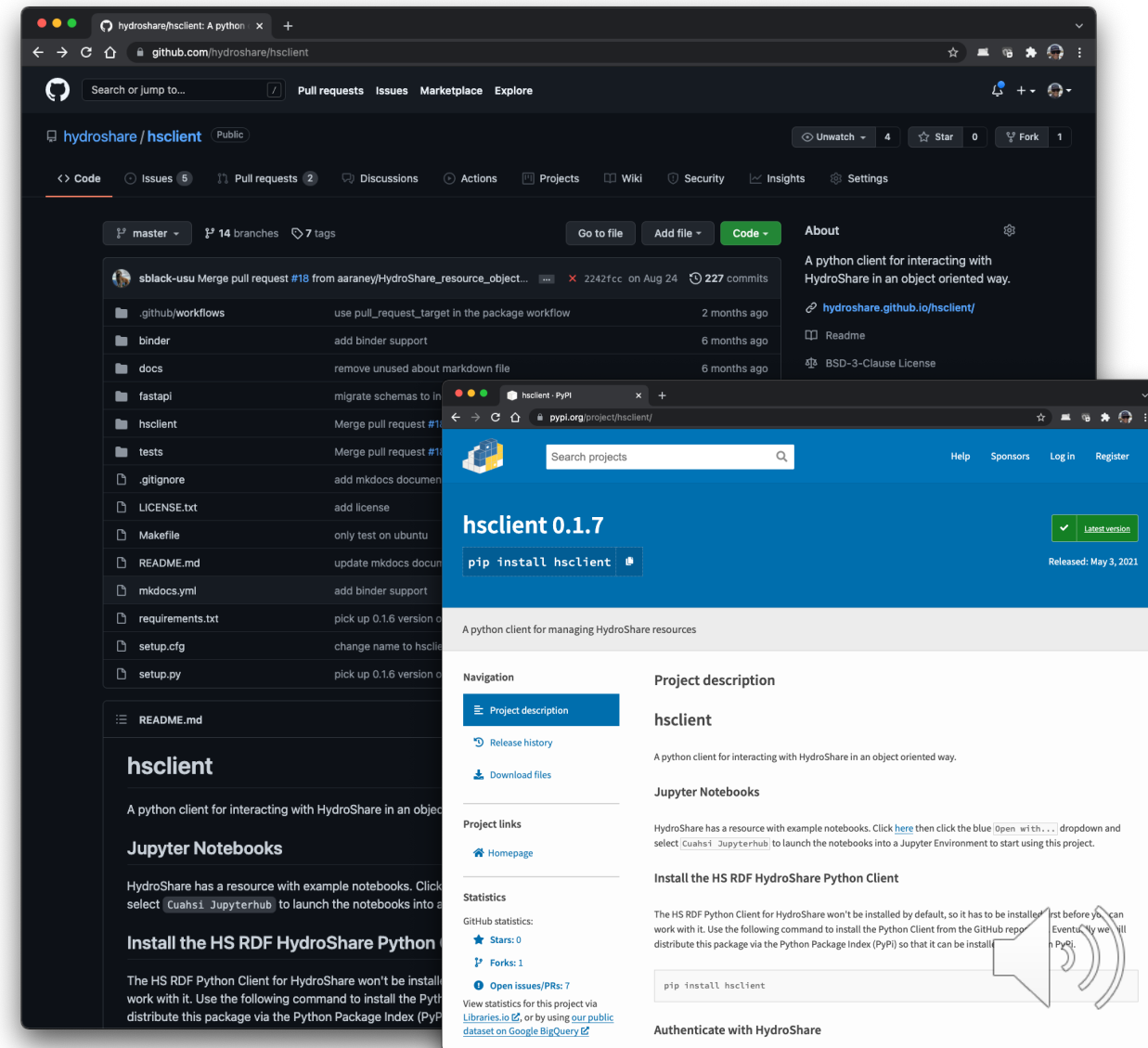
- Extending the HydroShare Resource Data Model to Python analysis environments
- Maps HydroShare's resource metadata to a set of Python objects (classes) defined using pydantic models
- Maps common water-related data types (HydroShare content types) to performant data structures within Python
- Load and stage data for visualization/analysis using common Python tools (pandas, matplotlib, etc.)





# HydroShare Python Client 'hsclient' package

- A set of Python functions for interacting with HydroShare
  - Resource creation/editing
  - Interact with resources in an interactive, object-oriented way
  - Integrate HydroShare resources into data science workflows
  - Reduce the time required to get data for analysis and then save results
- Example Jupyter Notebooks:  
<https://www.hydroshare.org/resource/7561aa12fd824ebb8edbee05af19b910/>
- GitHub Repository:  
<https://github.com/hydroshare/hsclient>



# USGS dataretrieval Python package

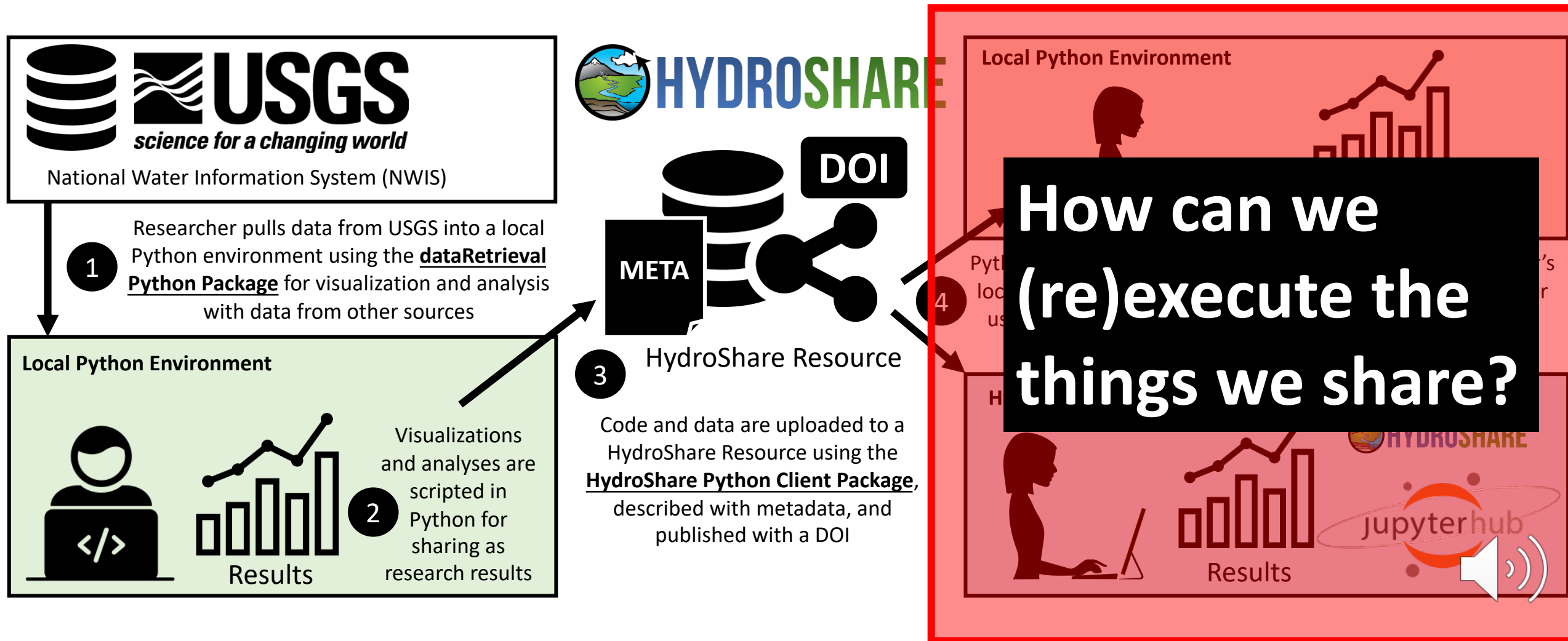
- Python mirror of the R dataRetrieval tool
- Currently has most of the same functions
- Very similar results
- Collaborating with Timothy Hodson at USGS
- Example Jupyter Notebooks: <https://www.hydroshare.org/resource/c97c32ecf59b4dff90ef013030c54264/>

<https://github.com/USGS-python/dataretrieval>

The image displays two overlapping web browser windows. The background window shows the GitHub repository for 'USGS-python / dataretrieval'. It features the repository's file structure, including folders like '.github/workflows', 'dataretrieval', 'demos', and 'tests', and files like '.gitignore', 'CONTRIBUTING.md', 'LICENSE.md', 'README.md', 'requirements.txt', and 'setup.py'. The README.md file is open, showing the title 'dataretrieval: Download hydrologic and climate data' and a section 'What is dataretrieval?' which describes it as a Python alternative to the USGS-R dataRetrieval package. The foreground window shows the PyPI package page for 'dataretrieval 0.5'. It includes a search bar, a 'pip install dataretrieval' button, and a 'Release history' section listing versions 0.1 through 0.5 with their respective release dates. The footer of the PyPI page contains links for 'Help', 'About PyPI', 'Contributing to PyPI', and 'Using'.

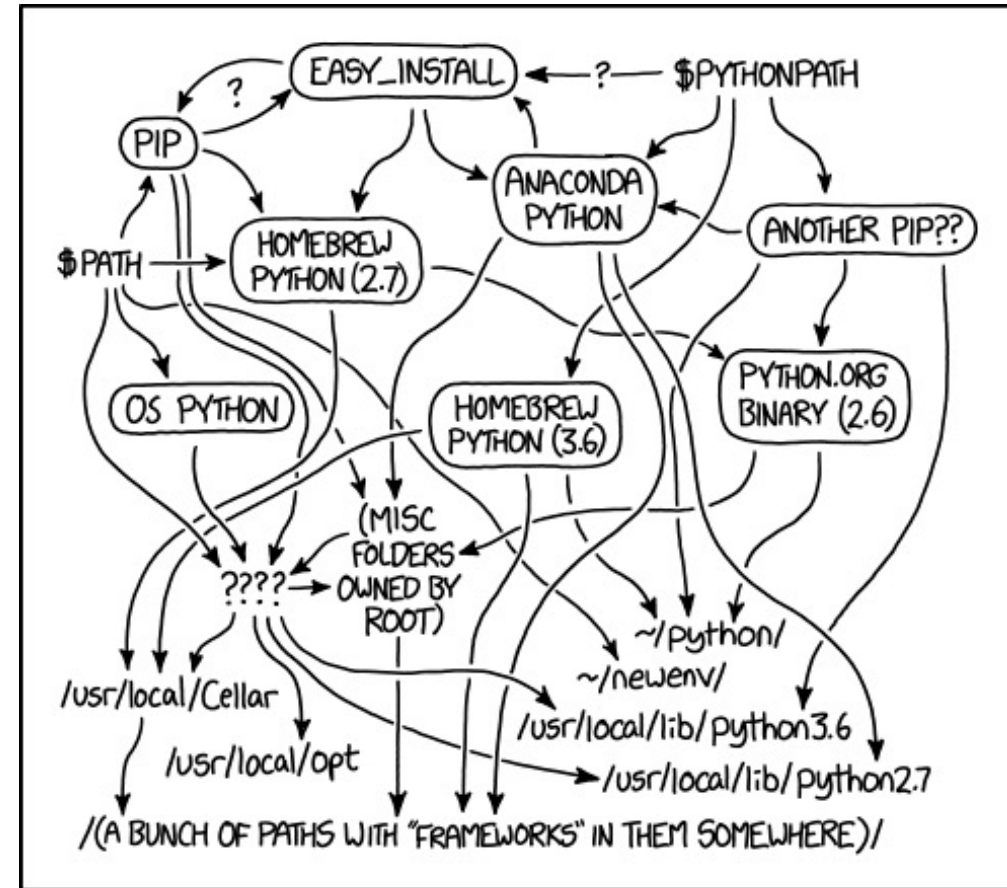
# Connecting Visualization and Analysis with an Online Repository

- Better enabling collaborative data science workflows and reproducibility



# One Option: Local Python Environment

- Set up a local environment
- Get the Python version right
- Install the right versions of all of the packages
- Cross your fingers and hope it will run . . .
- Virtual environments can help, but this can still be challenging



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED  
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

<https://xkcd.com/1987/>



# Collaborative and interactive computing for water-data scientists

## CUAHSI JupyterHub – Google Cloud

- Supports “unlimited” users (\$)
- Capable of creating classroom/workshop specific instances
- Completely customizable and uses the latest JH software

## CyberGIS-Jupyter for Water

- More available compute resources

## MATLAB Online

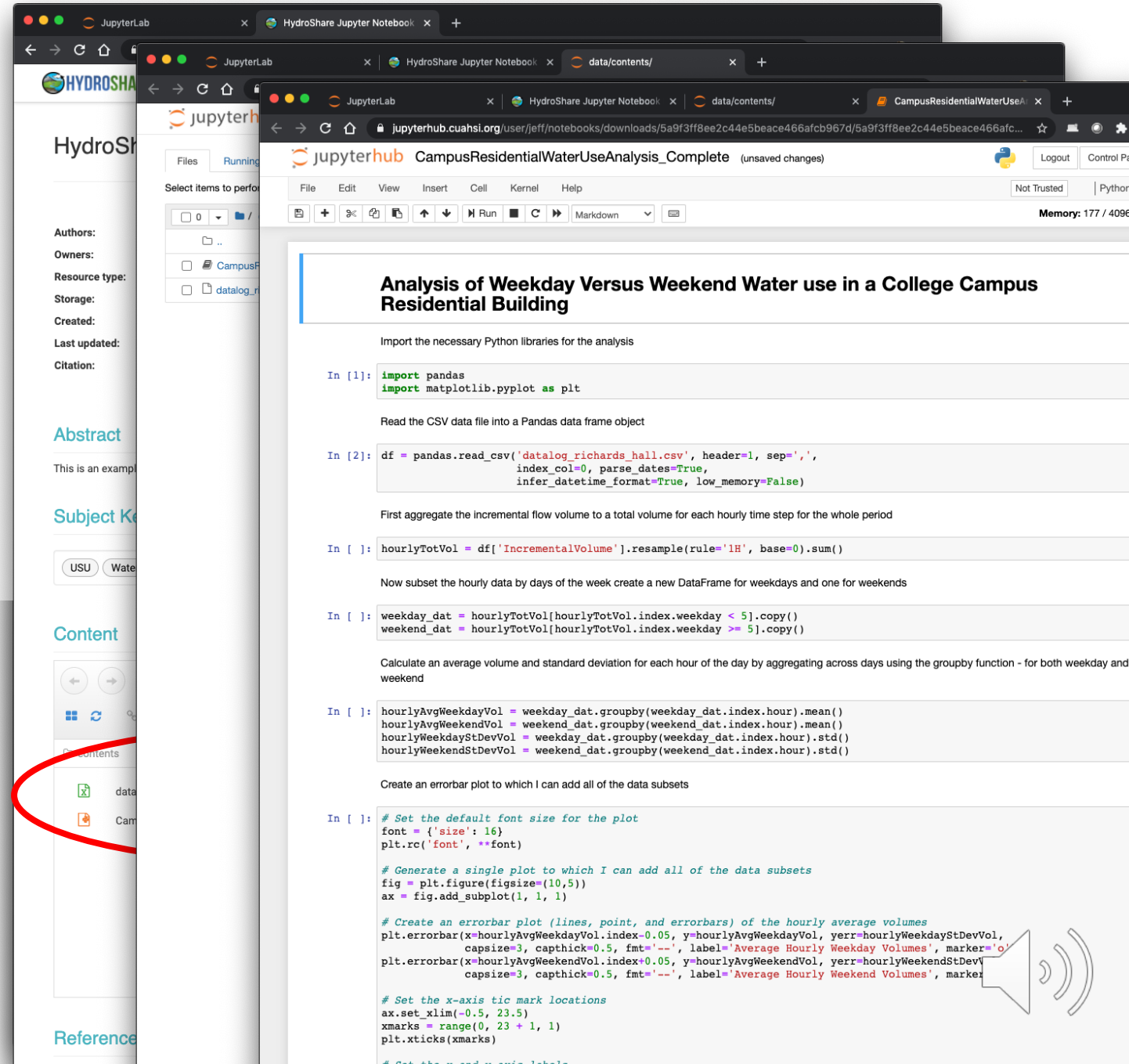
- 50 concurrent users
- Livescript support and m-file
- 20+ toolboxes





# Creating and Sharing Reproducible Analyses

- Reproducible analyses: Sharing data and code together in a repository
- Linking repositories with computational environments
- Repositories as a gateway to high performance computing and cloud services



The screenshot displays a JupyterLab environment with a notebook titled "CampusResidentialWaterUseAnalysis\_Complete". The notebook's content includes the following sections and code:

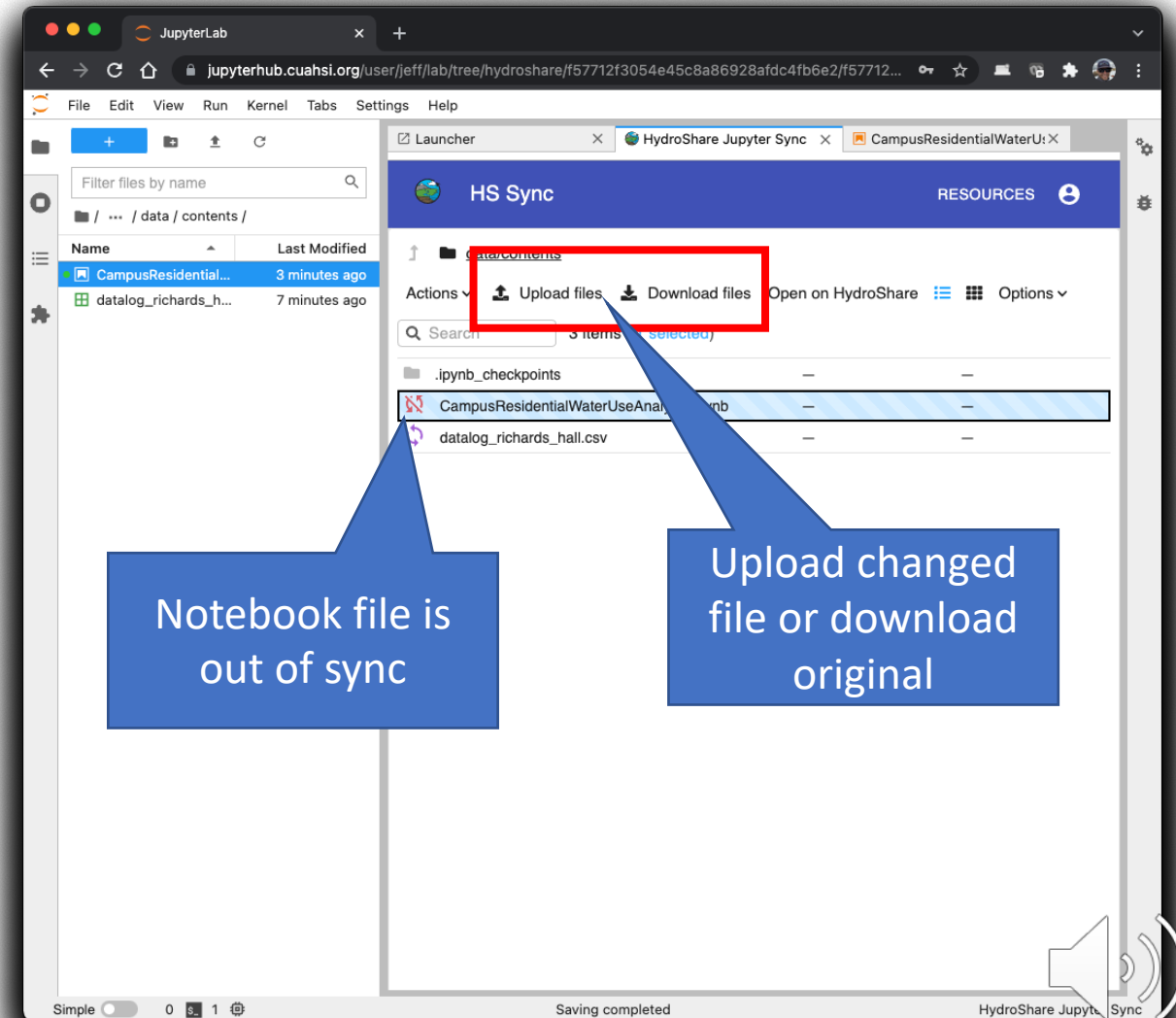
- Analysis of Weekday Versus Weekend Water use in a College Campus Residential Building**
- Import the necessary Python libraries for the analysis**  
`In [1]: import pandas  
import matplotlib.pyplot as plt`
- Read the CSV data file into a Pandas data frame object**  
`In [2]: df = pandas.read_csv('datalog_richards_hall.csv', header=1, sep=',',  
index_col=0, parse_dates=True,  
infer_datetime_format=True, low_memory=False)`
- First aggregate the incremental flow volume to a total volume for each hourly time step for the whole period**  
`In [ ]: hourlyTotVol = df['IncrementalVolume'].resample(rule='1H', base=0).sum()`
- Now subset the hourly data by days of the week create a new DataFrame for weekdays and one for weekends**  
`In [ ]: weekday_dat = hourlyTotVol[hourlyTotVol.index.weekday < 5].copy()  
weekend_dat = hourlyTotVol[hourlyTotVol.index.weekday >= 5].copy()`
- Calculate an average volume and standard deviation for each hour of the day by aggregating across days using the groupby function - for both weekday and weekend**  
`In [ ]: hourlyAvgWeekdayVol = weekday_dat.groupby(weekday_dat.index.hour).mean()  
hourlyAvgWeekendVol = weekend_dat.groupby(weekend_dat.index.hour).mean()  
hourlyWeekdayStDevVol = weekday_dat.groupby(weekday_dat.index.hour).std()  
hourlyWeekendStDevVol = weekend_dat.groupby(weekend_dat.index.hour).std()`
- Create an errorbar plot to which I can add all of the data subsets**  
`In [ ]: # Set the default font size for the plot  
font = {'size': 16}  
plt.rc('font', **font)  
  
# Generate a single plot to which I can add all of the data subsets  
fig = plt.figure(figsize=(10,5))  
ax = fig.add_subplot(1, 1, 1)  
  
# Create an errorbar plot (lines, point, and errorbars) of the hourly average volumes  
plt.errorbar(x=hourlyAvgWeekdayVol.index-0.05, y=hourlyAvgWeekdayVol, yerr=hourlyWeekdayStDevVol,  
capsize=3, capthick=0.5, fmt='--', label='Average Hourly Weekday Volumes', marker='o')  
plt.errorbar(x=hourlyAvgWeekendVol.index+0.05, y=hourlyAvgWeekendVol, yerr=hourlyWeekendStDevVol,  
capsize=3, capthick=0.5, fmt='--', label='Average Hourly Weekend Volumes', marker='o')  
  
# Set the x-axis tic mark locations  
ax.set_xlim(-0.5, 23.5)  
xmarks = range(0, 23 + 1, 1)  
plt.xticks(xmarks)  
  
# Set the y and y-axis labels`

A red circle in the left sidebar highlights the "Contents" tab, which lists files like "data" and "CampusResidentialWaterUseAnalysis\_Complete".

# CUAHSI JupyterSync App using hsclient

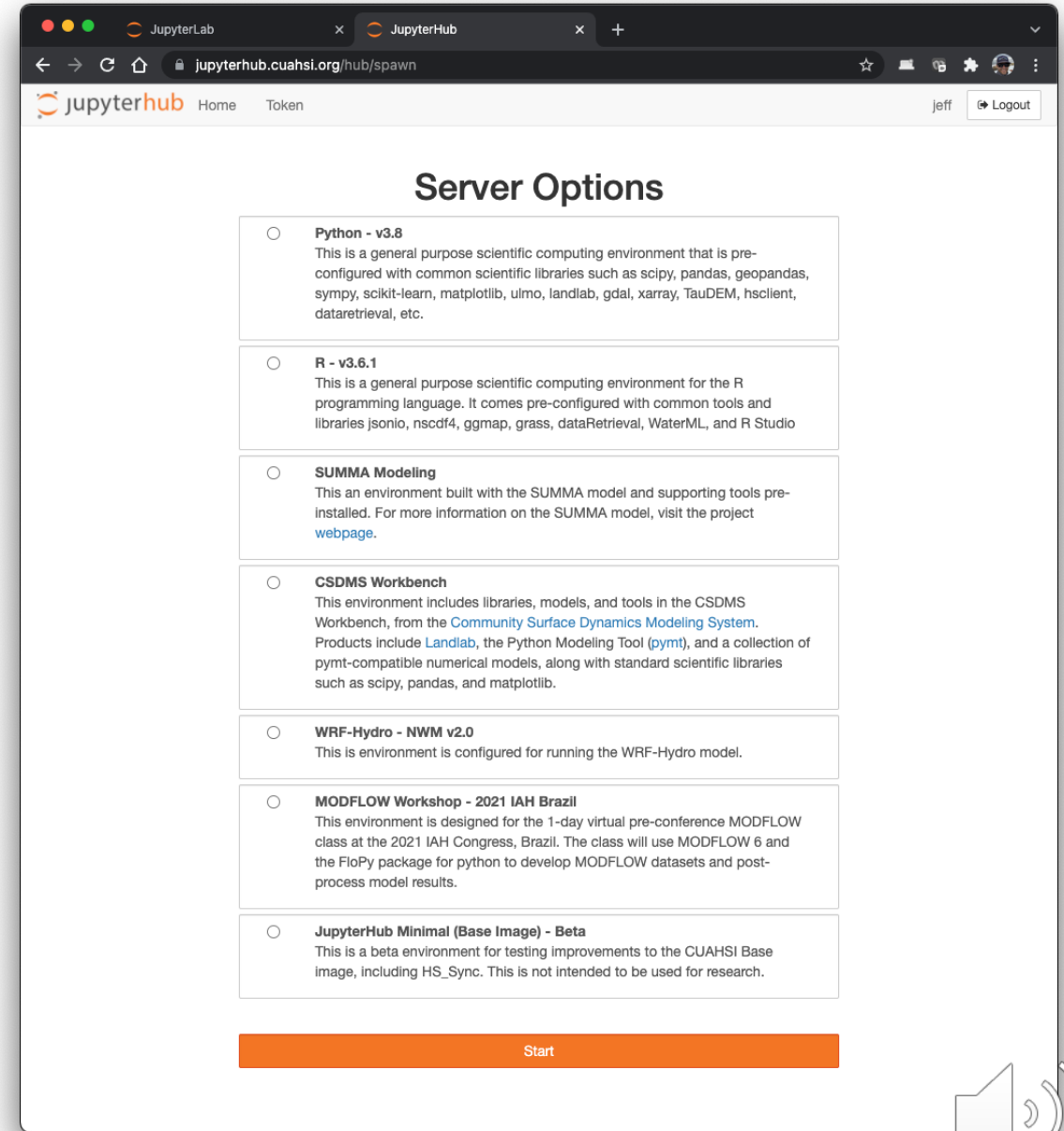
1. Launch CUAHSI Jupyterhub ([jupyterhub.cuahsi.org](http://jupyterhub.cuahsi.org))
2. Launch the Jupyter Sync App
3. Choose a HydroShare resource to work with
4. Select files to download to Jupyter environment
5. Open a file to edit or execute
6. Make changes to the file in the Jupyter Environment
7. Upload changed file to HydroShare or download original file to replace

Work by Tony Castronova and Austin Raney and students from Olin College of Engineering



# HydroShare's Linked JupyterHub Environments

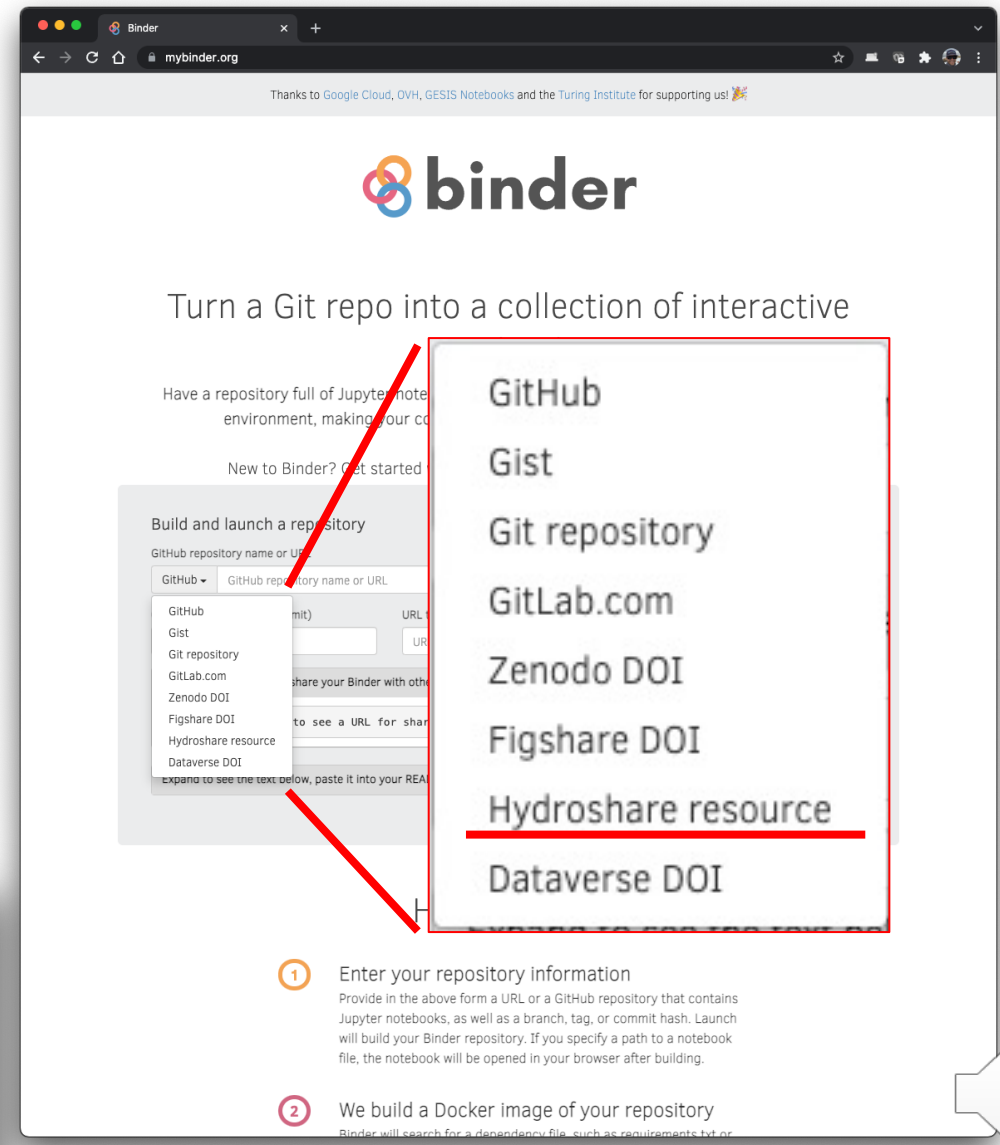
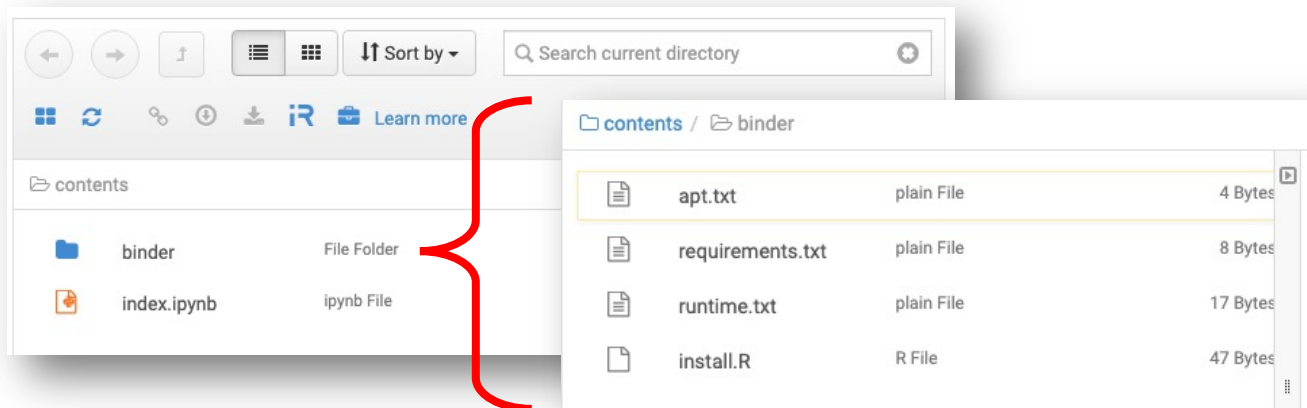
- Better because I don't have to set up the environment
- Some nice tools for interacting with HydroShare
- But, I still need an environment
- Some potential limitations:
  - Software dependencies
  - Legacy code
  - Long run times
  - Complicated and large input/output files





# Improving Reproducibility with Binder

- Custom computing environments
- Free, but limited resources
- Can lower the barrier of entry for water scientists
- Integrated with HydroShare
- Users can start with a HydroShare base image



Slide from Tony Castronova at CUAHSI



1664061  
1931297  
1931278

# Questions?

Jeffery S. Horsburgh

[jeff.horsburgh@usu.edu](mailto:jeff.horsburgh@usu.edu)



Utah Water Research Laboratory  
UtahStateUniversity

