

*show your work!*

For a reproducible, extensible, and  
transparent research



- Scientific results should be reproducible and transparent
- **Challenge:** Integrating code, data, and manuscript in a clean, automated way.

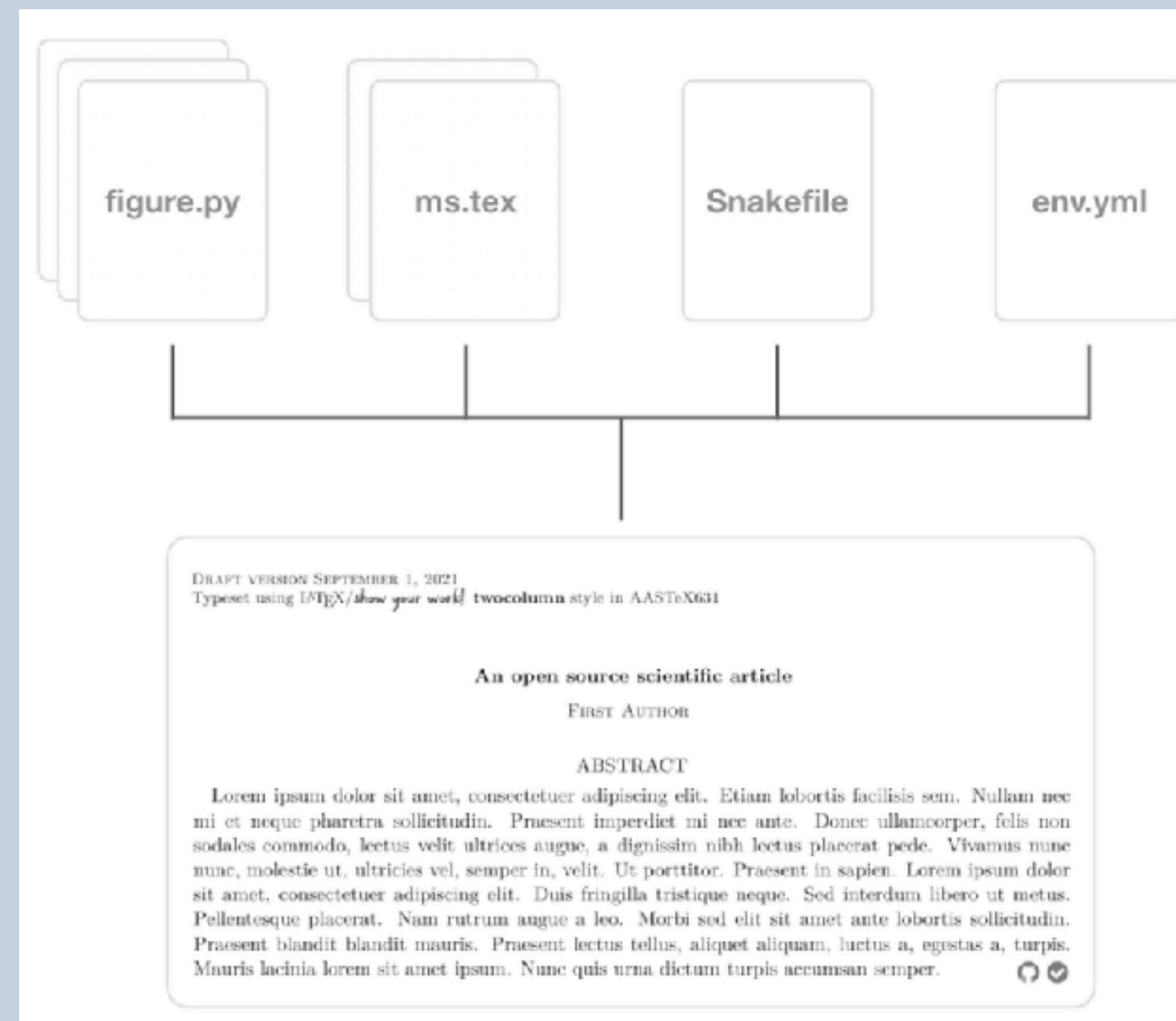
**i The showyourwork philosophy**

Scientific papers should exist as GitHub repositories comprised of LaTeX files, figure scripts, rules to access datasets, a platform/environment specification, **and nothing else**. Anyone should be able to re-generate the article PDF from scratch at the click of a button.

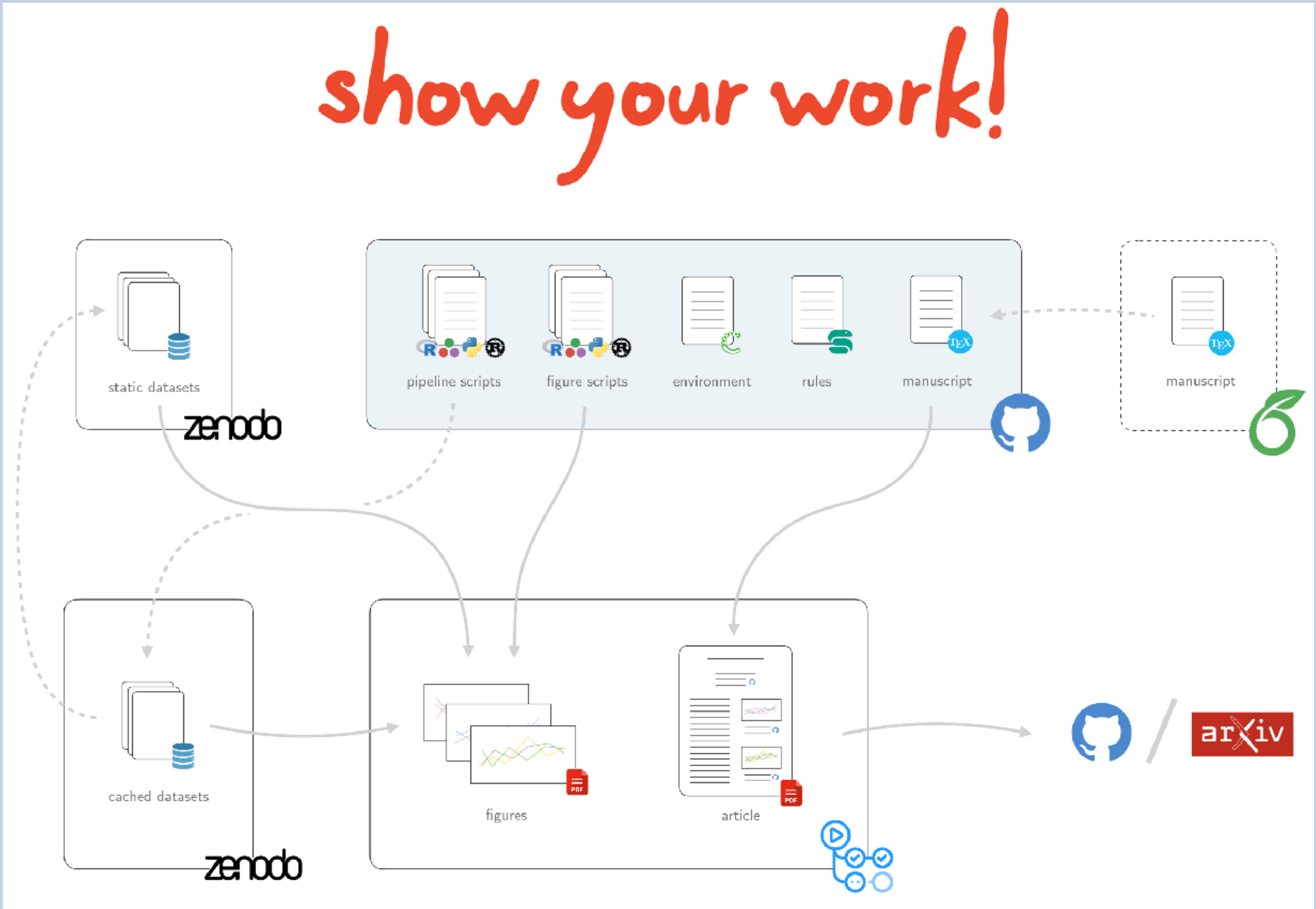
- The showyourwork! workflow is intended to help authors publish **open source**, **replicable**, and **reproducible** scientific articles.
- It ensures that the **article PDF** is always in sync with all of the scripts, code, and data used to generate it.
- It represents all steps required to go from raw data to publication figures in a workflow file that can be used to reproduce all or part of the elements in a paper (potentially including numerical simulations, data reduction, fetching data from third-party releases, and plotting)
- It has been developed by astronomers at Flatiron Institute (mainly Rodrigo Luger)

- It does this automatically with the help of:
  - the **Snakemake workflow management system** (a tool to create reproducible and scalable data analyses. Workflows are described via a human readable, Python based language. They can be seamlessly scaled to server, cluster, grid and cloud environments. They can entail a description of required software, which will be automatically deployed to any execution environment. Workflow runs can be automatically turned into interactive portable browser based reports) -> any volunteer to present this at a code coffee?
  - the **tectonic typesetting engine** (modernized, complete, self-contained TeX/LaTeX engine, powered by XeTeX and TeXLive.)
  - **Github Actions CI**

- articles exist as GitHub repositories with a specific layout
- Whenever new commits are pushed to the remote repository, a GitHub action is triggered that automatically builds the article from the input figure scripts, manuscript files, and conda environment file, following the instructions specified in the Snakefile.









- Git repo: <https://github.com/showyourwork/showyourwork>
- Documentation: <https://show-your.work/en/latest/intro/>
- Example of paper developed with showyourwork!
  - <https://github.com/matscke/hz-inner-edge-discontinuity>
  - <https://raw.githubusercontent.com/matscke/hz-inner-edge-discontinuity/main-pdf/ms.pdf>



## Zenodo integration

- showyourwork! integrates with the Zenodo and Zenodo Sandbox services to host datasets and simulation results used in the workflow.
- There are two main ways in which this integration occurs:
  - static datasets: your workflow depends on data that cannot be programmatically generated (e.g. data collected from a telescope) that should be made available to anyone trying to reproduce your results. You can archive it on an online open-access file-hosting service, like Zenodo (showyourwork! deals with all the communicating back-and-forth for you once you specify the archive ID)
  - dynamic datasets: showyourwork! can cache the results of intermediate steps in your pipeline alongside a record of all the inputs (e.g. useful when workflows entail running lengthy computations, simulations, that third-party users may not want to run on their own and helps the limited compute resources for builds on GitHub Actions). If, on subsequent runs of the workflow, the inputs remain unchanged, showyourwork! will simply download the cached results, maintaining the guarantee that the output you get follows deterministically from the given inputs.