

ReproServer: Making Reproducibility Easier and Less Intrusive

Remi Rampin^{1,2}, Vicky Steeves^{2,3}, Fernando Chirigati^{1,2}, and Juliana Freire^{1,2}

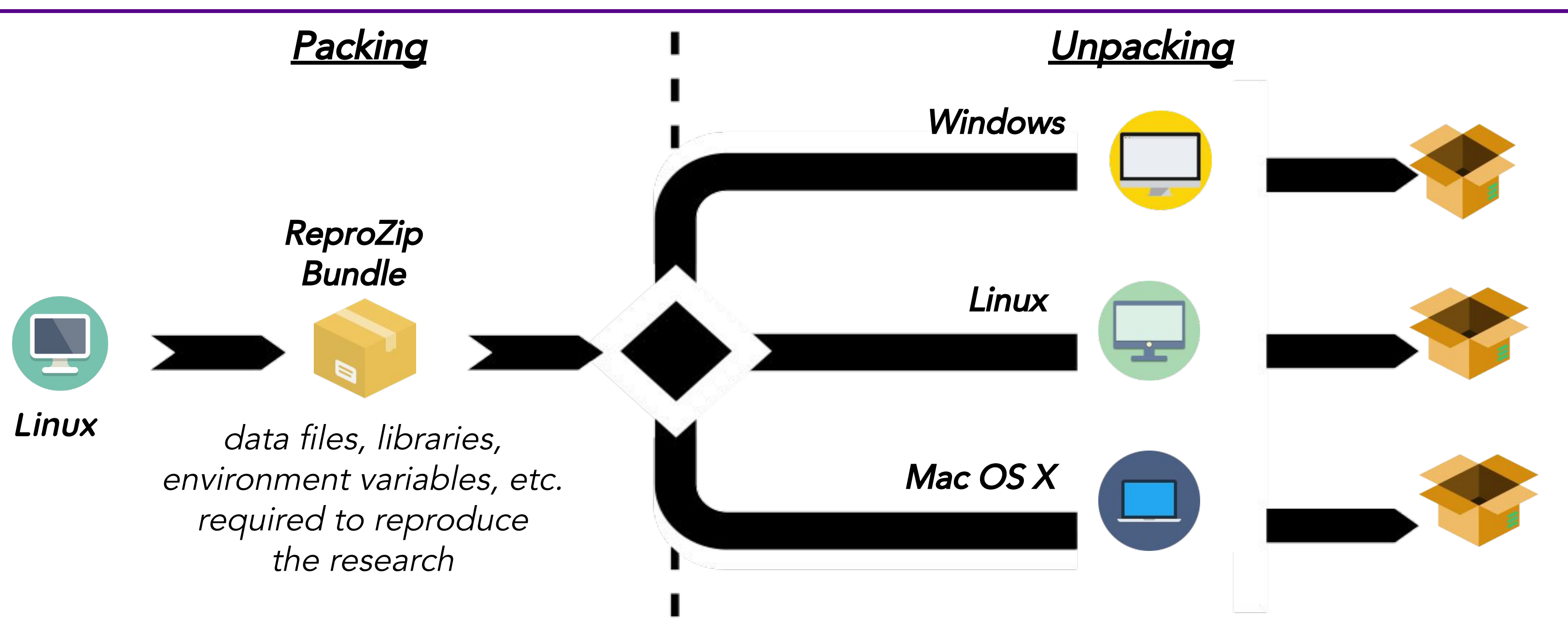


¹NYU Tandon School of Engineering, ²NYU Center for Data Science, ³NYU Division of Libraries



ReproServer is a browser-based application for interacting with and reproducing the contents of **ReproZip** bundles! **ReproZip** (reprozip.org) is an open source tool used to reproduce data analysis scripts, software, graphical tools, interactive tools, client-server applications, Jupyter Notebooks, and most processes that can run on a computer!

Before: ReproZip



Users first have to trace and pack their work using **ReproZip** on their working computer, which must be running Linux.

Users run **ReproZip** at the same time as running their analysis, and **ReproZip** will capture all the provenance, data, environmental variables, source code, etc. required to re-run the research.

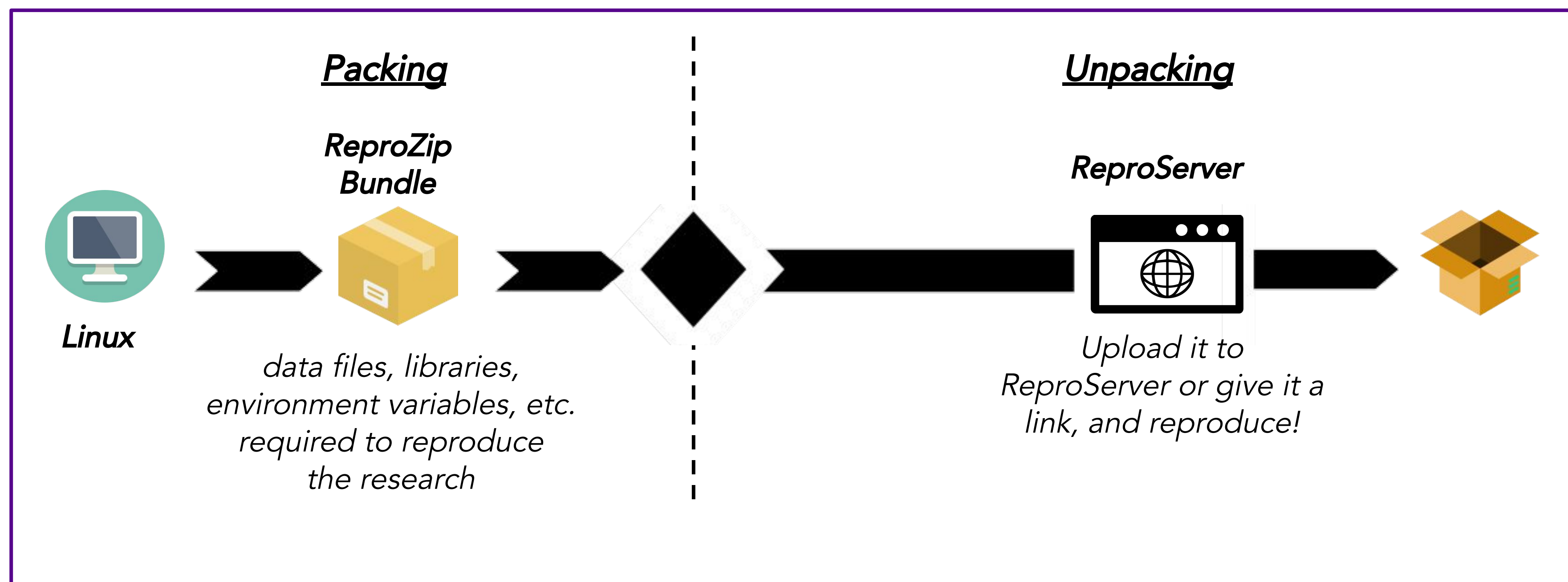
Then, they can give the `.rpz` file to others to reproduce across OS's, using an unpacker (e.g. docker, singularity, vagrant) installed on the secondary user's computer.

Now: ReproZip + ReproServer

Using **ReproServer**, the `.rpz` file or a link to one is enough to reproduce the contents of the **ReproZip** bundle!

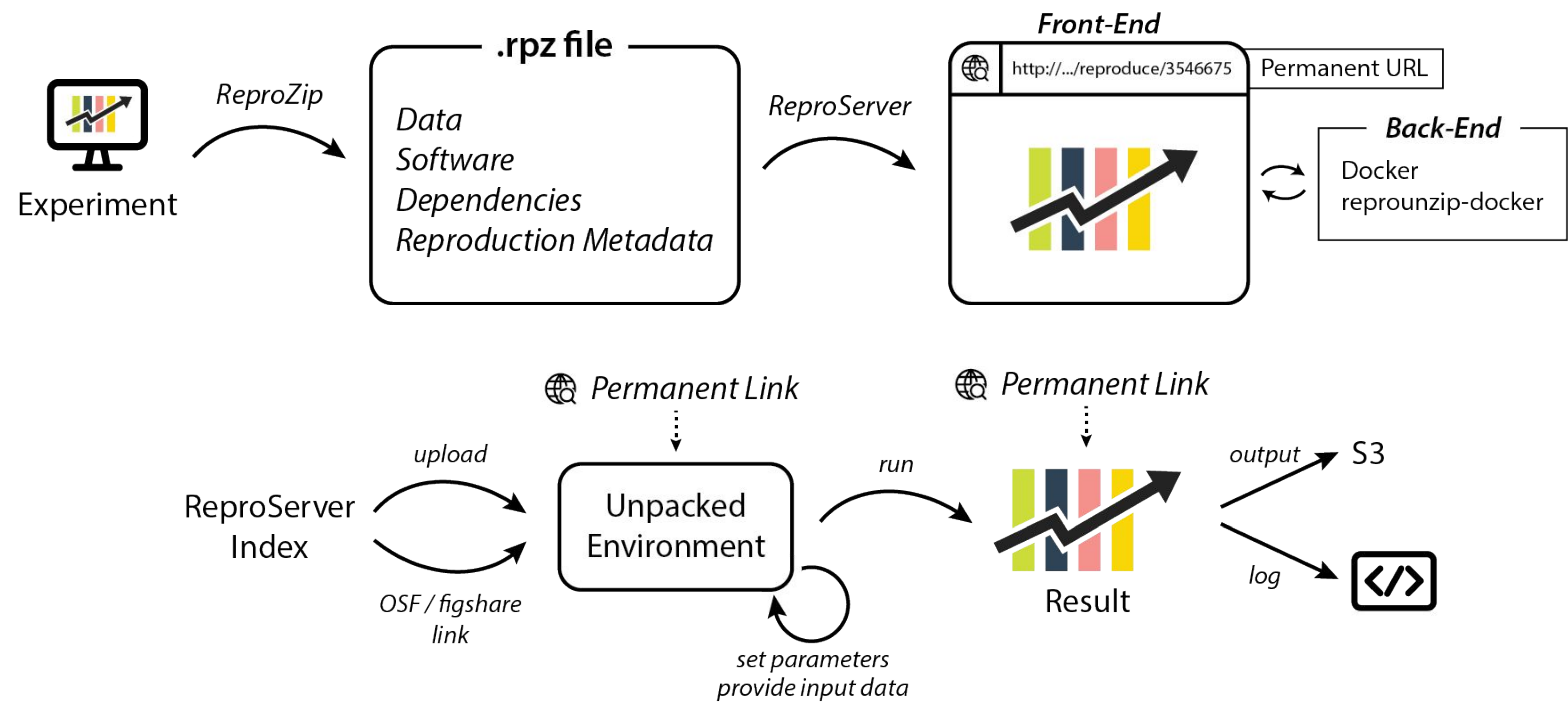
Users still pack their research using **ReproZip** on their local computer. They can then deposit the `.rpz` file in a repository (e.g. OSF, Figshare) or send the file to someone else.

Using a link or a file, the secondary user can unpack and interact with the contents of a **ReproZip** bundle in their browser using **ReproServer**, no local installs required!

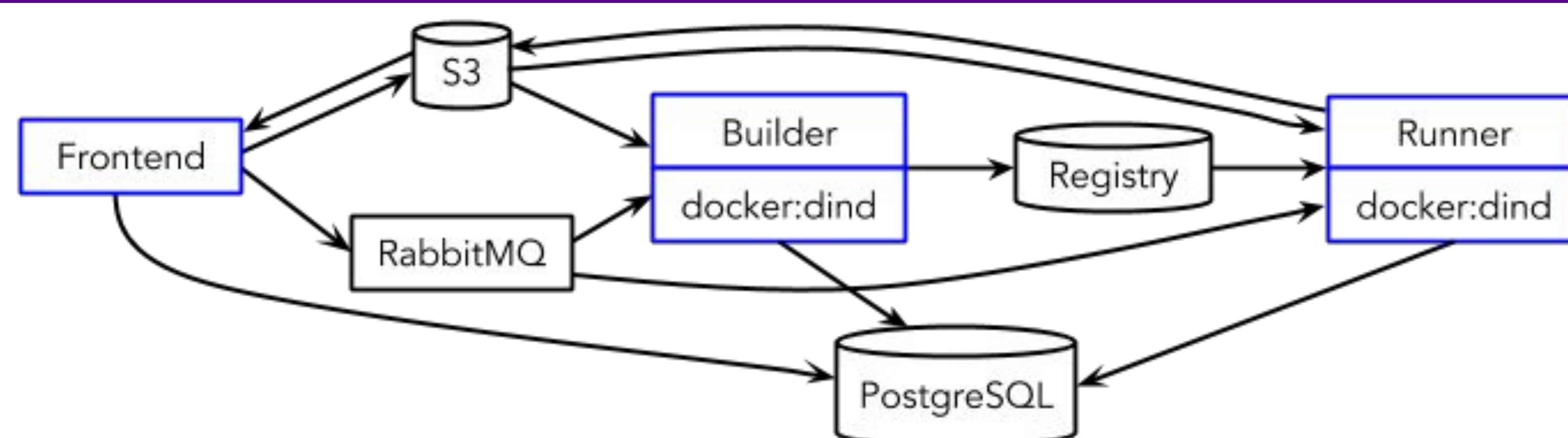


ReproServer: Workflow & Perks

- Allows **changing** input data, configuration, command-lines
- Gives you a **URL to include in papers** to reproduce your experiment
- Offloads archiving responsibility to people who are good at it (repositories)
- **No lock-in**: build on your laptop, pack automatically, reproduce anywhere



ReproServer: Architecture Overview



Frontend: Web application allowing users to select or upload an experiment, edit parameters, upload input files. After running, shows the log and lets the user download output files.

Builder: From the RPZ file, builds a Docker image and caches it on a private Docker registry for the runners.

Runner: From the cached Docker image, input files, and parameters, runs the research and stores the results.

Acknowledgements: This work was supported by the Moore-Sloan Data Science Environment at NYU.