Workflow Organization with REANA (Hands-On)

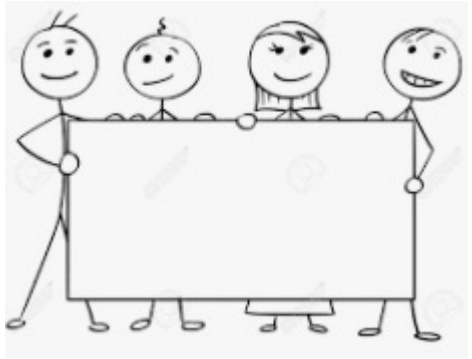
Arman Khalatyan, Harry Enke

Leibniz Institute for Astrophysics Potsdam

AG-2023, TUBerlin

Scientific life

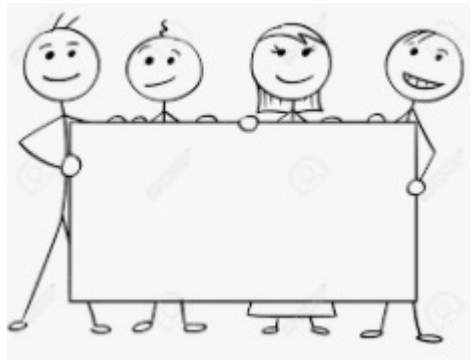
Idea → Collaborate → Publish

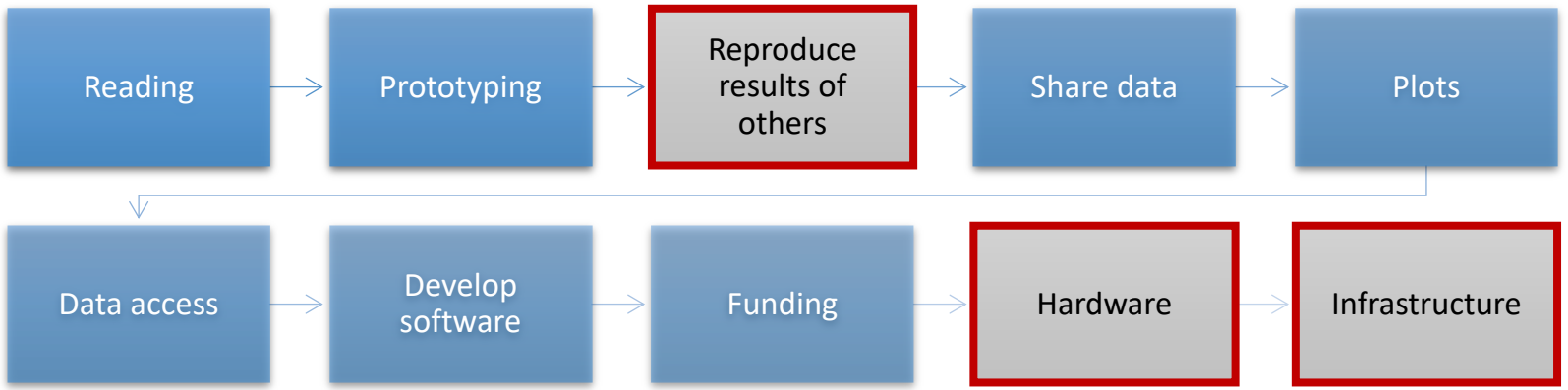


Reading → Prototyping → Reproduce results of others → Share data → Plots

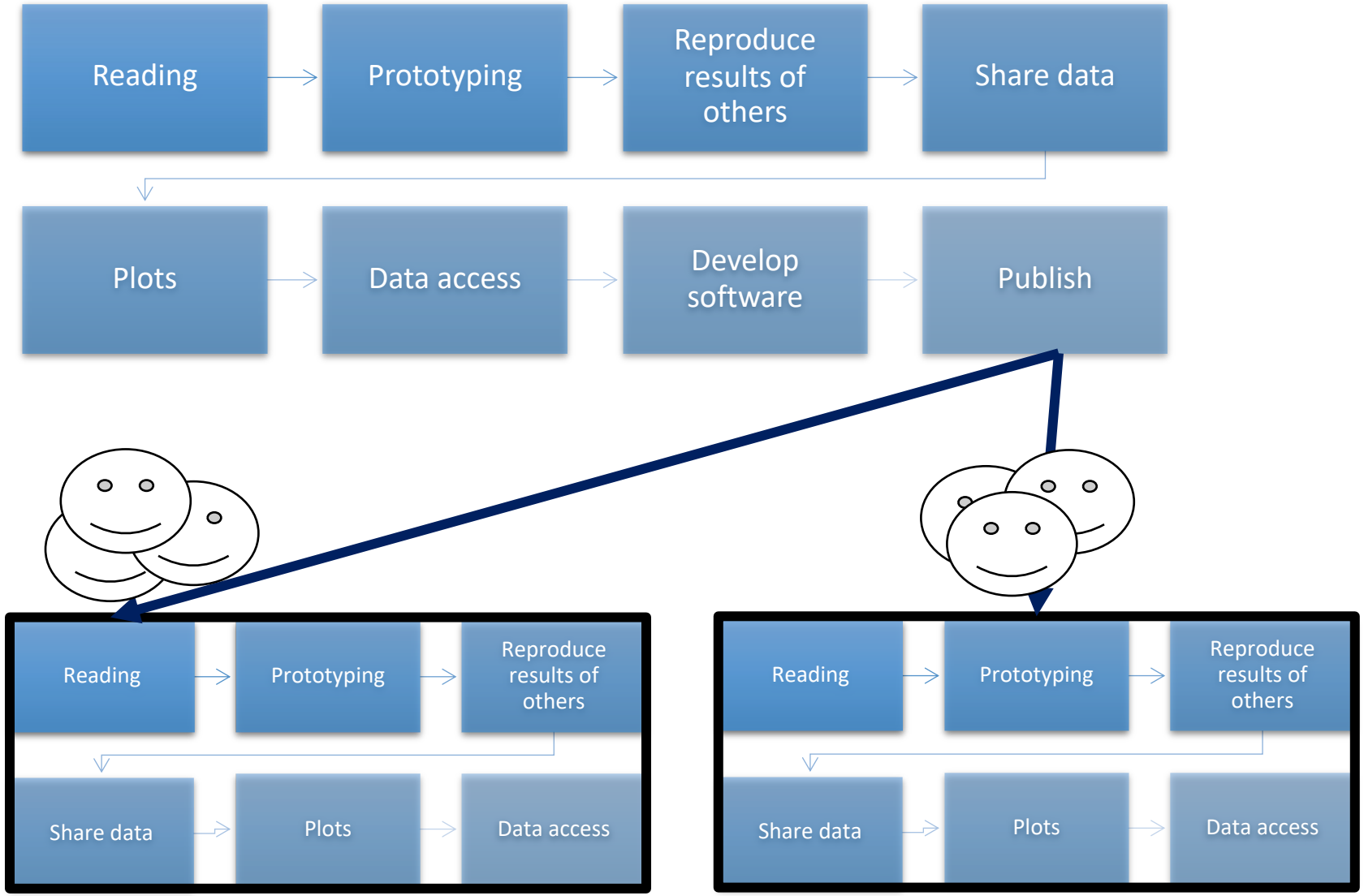
Data access → Develop software → Funding → Hardware → Infrastructure

Scientific life (top to down)

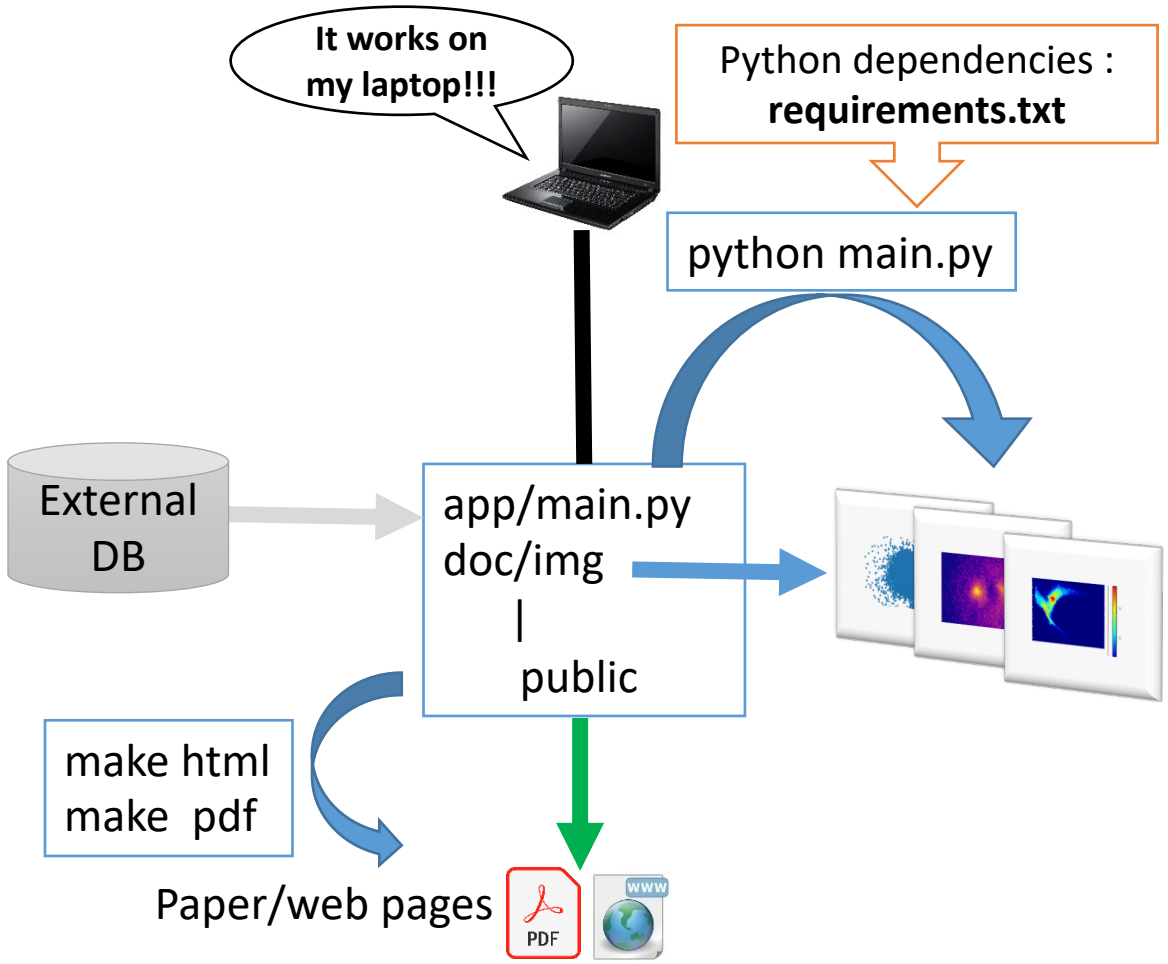




Scientific products



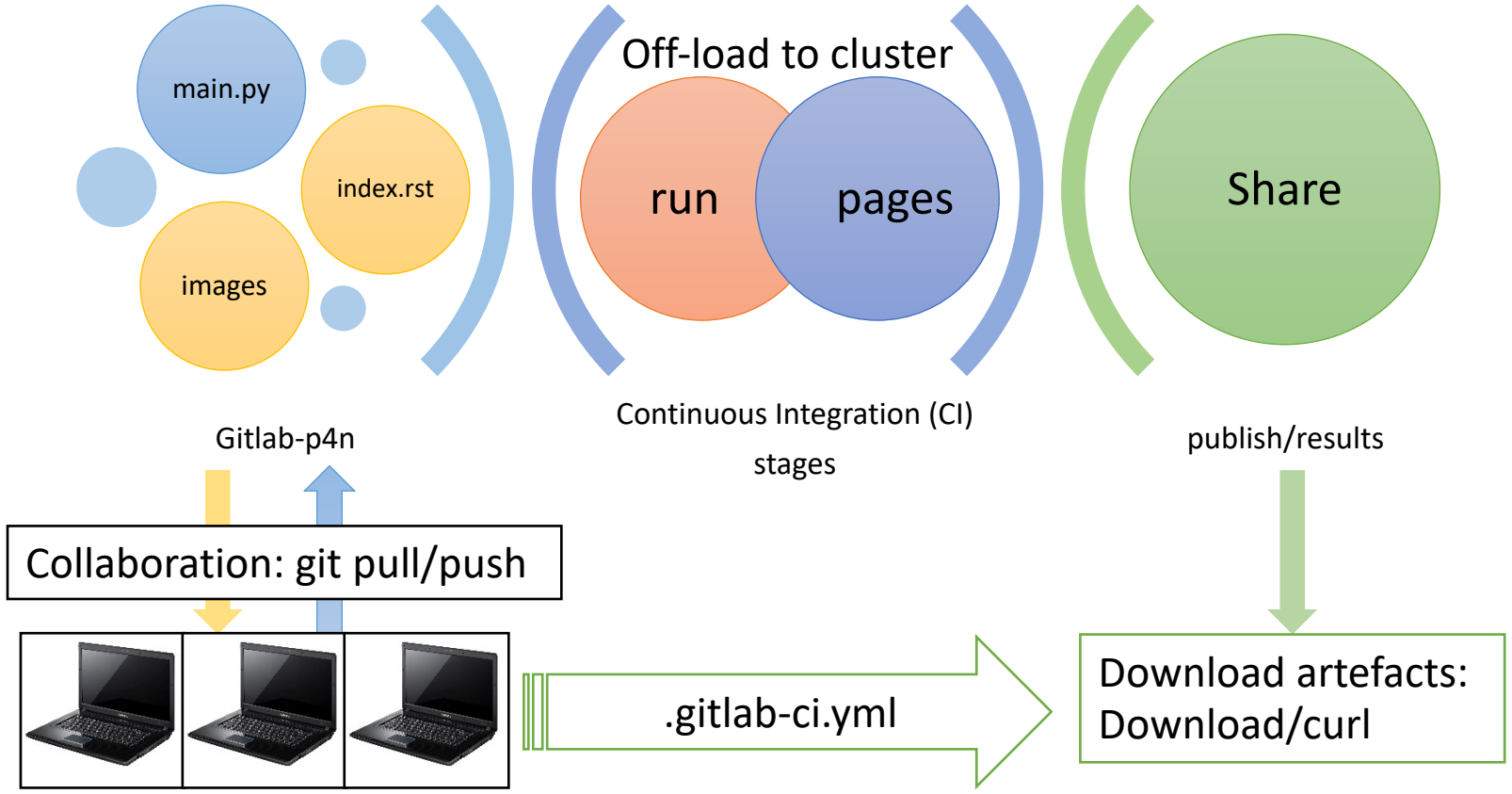
Local workflow:
”..but it works on my laptop”



Stages

- RUN
  - edit **main.py**
  - run: **main.py**
  - inspect images
- PAGES
  - edit **doc/index.rst** … etc.
  - compile
  - Inspect pdf/html
  - Publish

Global reproducible workflow: "…works everywhere"



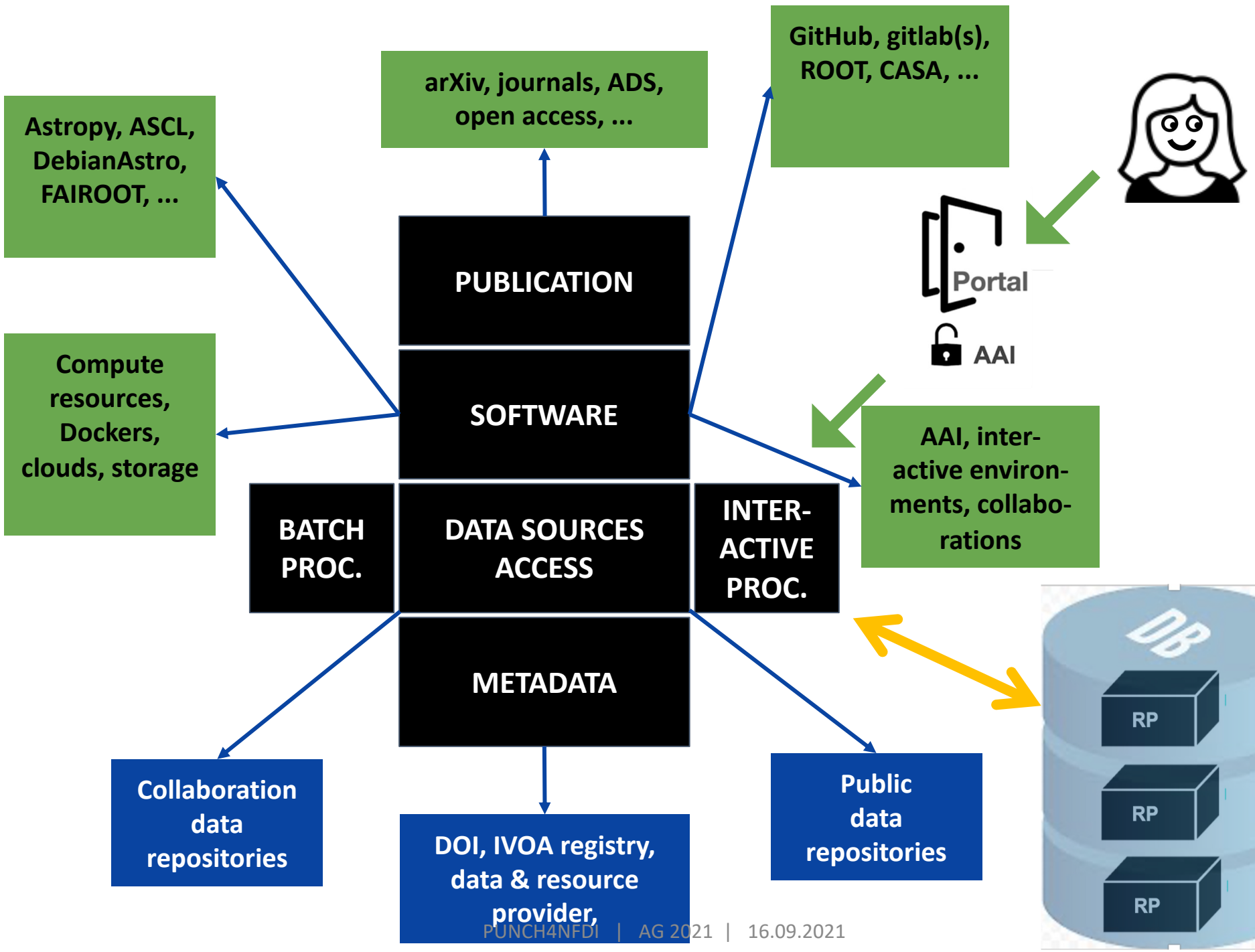
Stages

- RUN
  - e… ain.py
  - …
  - ins…
- PAGES
  - … … etc.
  - c… …e
  - Inspect pdf/html
  - Publish

Stages

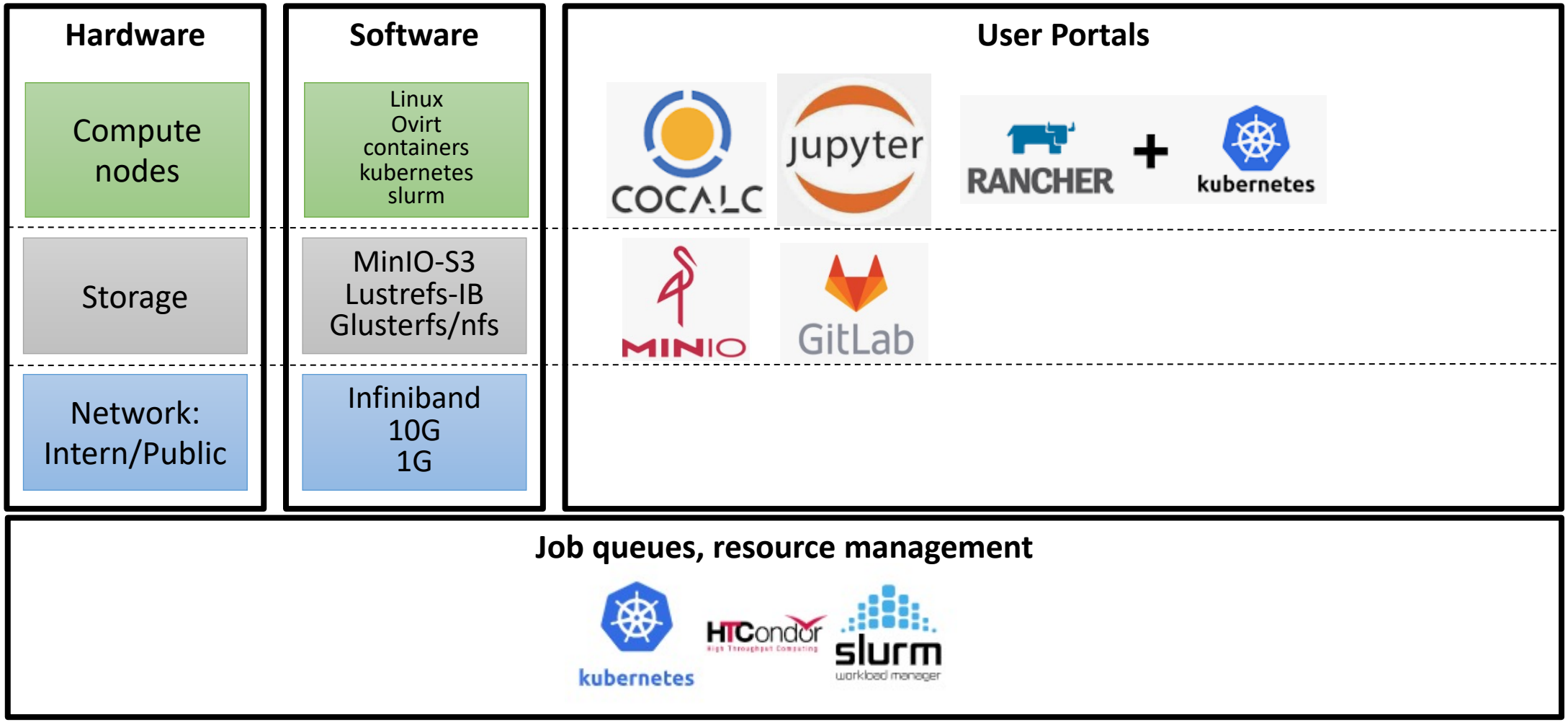
- EDIT
  - **main.py**
  - **index.rst**
- PUBLISH
  - **artefacts**

Digital Research Product



- Improves reproducibility and re-usability
- guided by FAIR
- Access via portal
- Interaction with other RPs
- Interfaces to tools and infrastructures
- Built on available developments of the communities our

IAAS and PAAS



https://reanahub.io/

reana

Reproducible research data analysis platform

Flexible

Run many computational workflow engines.

COMMON WORKFLOW LANGUAGE

Scalable

Support for remote compute clouds.

kubernetes HTCondor slurm workload manager

Reusable

Containerise once, reuse elsewhere. Cloud-native.

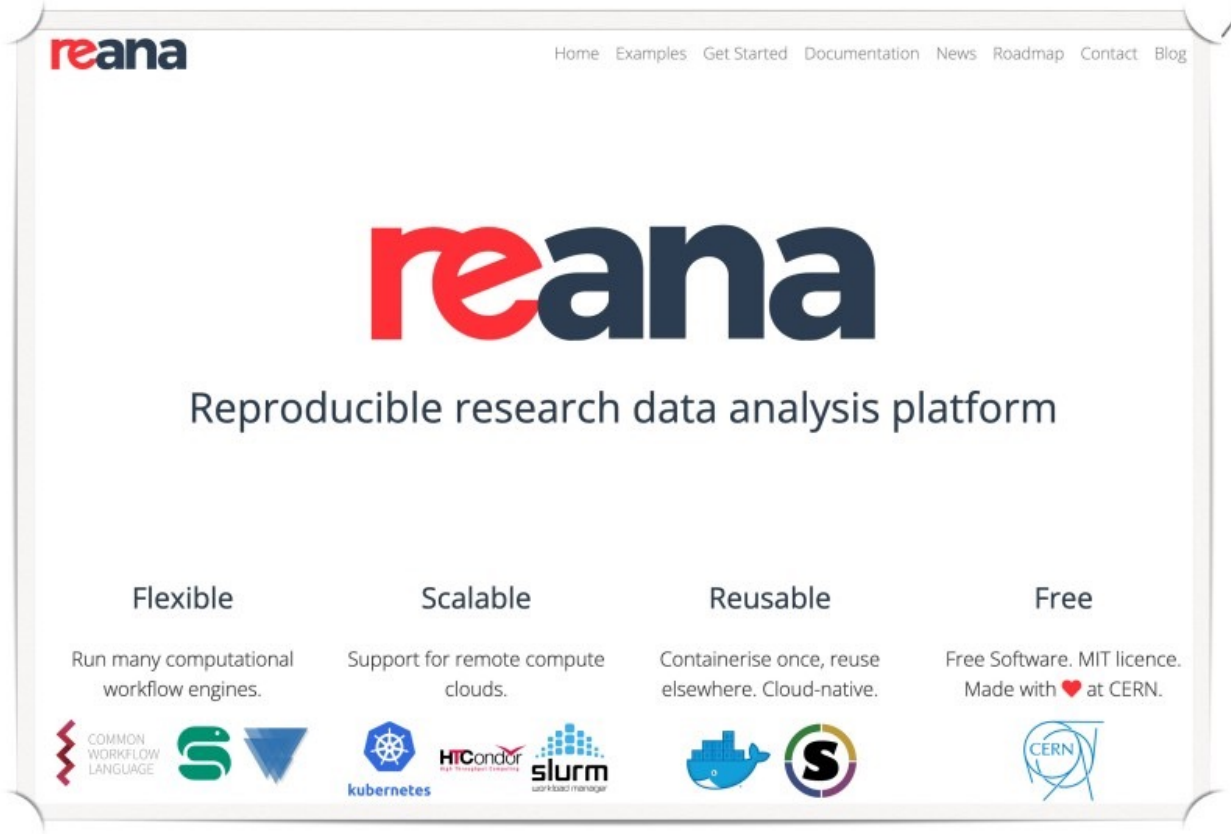
Free

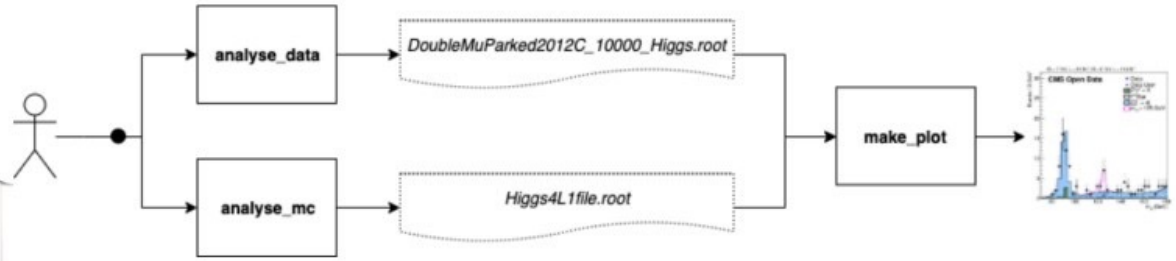
Free Software. MIT licence.
Made with ❤ at CERN.

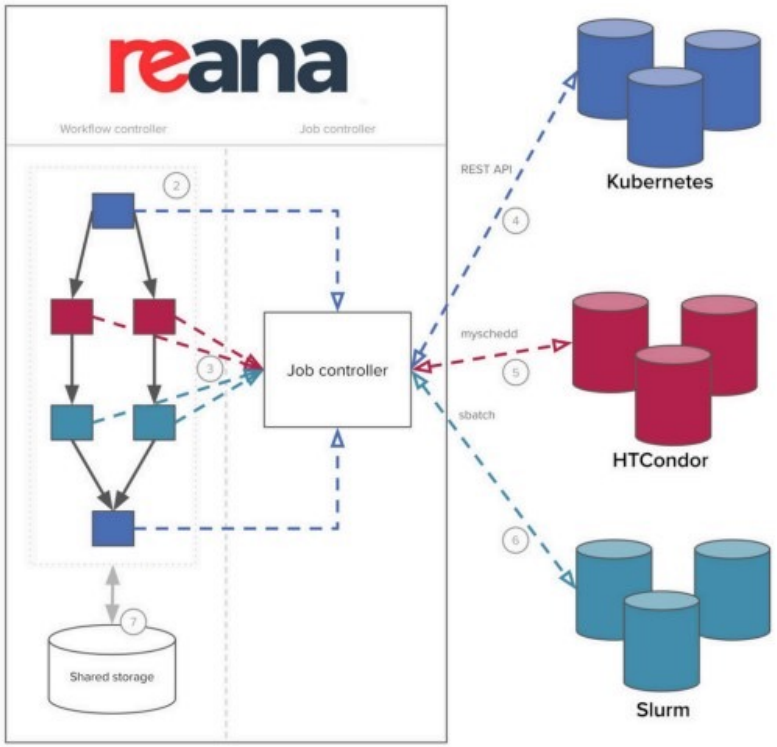
CERN

by Tibor Simko

What is REANA?







by Tibor Simko

"Preproducible" analyses

Nature **557** (2018) 613

WORLD VIEW · 24 MAY 2018

Before reproducibility must come preproducibility

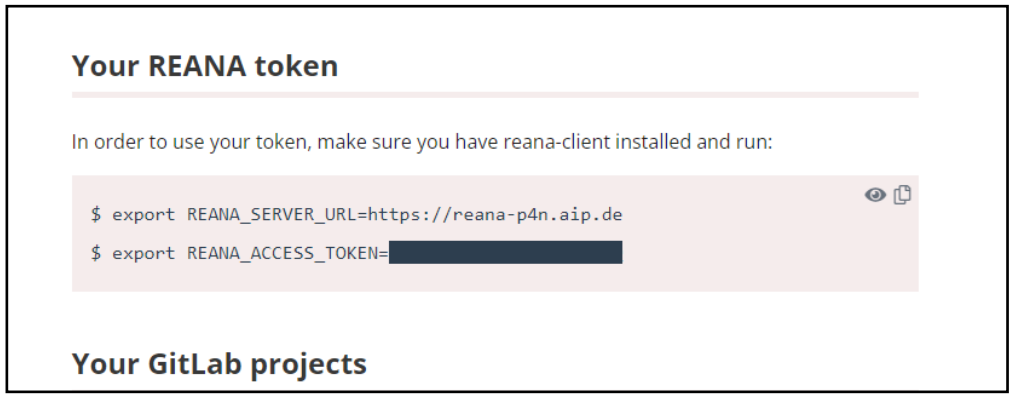
Instead of arguing about whether results hold up, let's push to provide enough information for others to repeat the experiments, says Philip Stark.

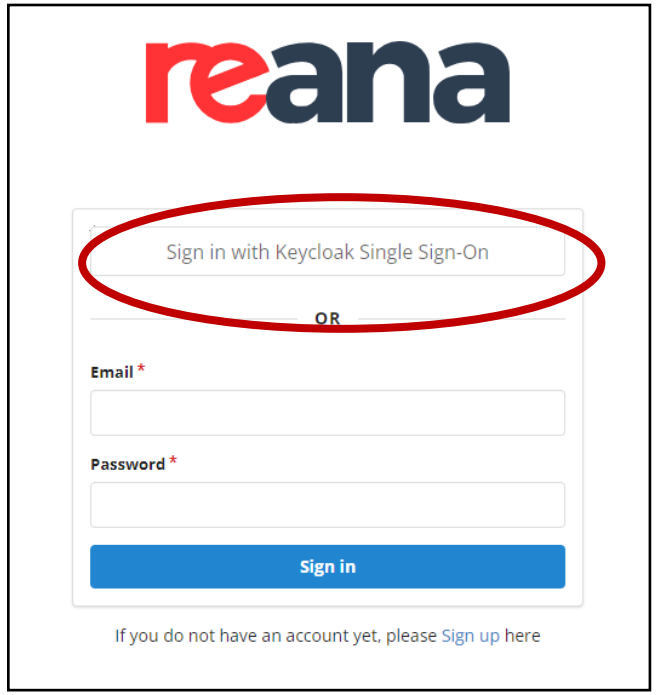
https://doi.org/10.1038/d41586-018-05256-0

Driving preproducibility via Continuous Integration with source code management systems

Getting started with REANA at AIP

- https://reana-p4n.aip.de/
- Login using your DFN-AAI





Setup environment on client

- conda create -n reana python=3.10
- conda activate reana
- pip install reana-client

```
(reana) basti@dragon ~ % pip install numpy matplotlib pandas
Collecting numpy
  Downloading numpy-1.25.0-cp310-cp310-macosx_10_9_x86_64.whl (20.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20.1/20.1 MB 39.1 MB/s eta 0:00:00
Collecting matplotlib
  Downloading matplotlib-3.7.2-cp310-cp310-macosx_10_12_x86_64.whl (7.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.4/7.4 MB 38.6 MB/s eta 0:00:00
Collecting pandas
  Downloading pandas-2.0.3-cp310-cp310-macosx_10_9_x86_64.whl (11.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.8/11.8 MB 43.4 MB/s eta 0:00:00
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.1.0-cp310-cp310-macosx_10_9_x86_64.whl (243 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 243.6/243.6 kB 8.9 MB/s eta 0:00:00
Collecting cycler>=0.10 (from matplotlib)
  Using cached cycler-0.11.0-py3-none-any.whl (6.4 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.40.0-cp310-cp310-macosx_10_9_x86_64.whl (2.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.1/2.1 MB 30.7 MB/s eta 0:00:00
Collecting kiwisolver>=1.0.1 (from matplotlib)
  Downloading kiwisolver-1.4.4-cp310-cp310-macosx_10_9_x86_64.whl (65 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 65.5/65.5 kB 2.2 MB/s eta 0:00:00
Collecting packaging>=20.0 (from matplotlib)
  Using cached packaging-23.1-py3-none-any.whl (48 kB)
Collecting pillow>=6.2.0 (from matplotlib)
  Downloading Pillow-10.0.0-cp310-cp310-macosx_10_10_x86_64.whl (3.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.4/3.4 MB 34.9 MB/s eta 0:00:00
Collecting pyparsing<3.1,>=2.3.1 (from matplotlib)
  Downloading pyparsing-3.0.9-py3-none-any.whl (98 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 98.3/98.3 kB 3.5 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.7 in ./miniconda3/envs/reana/lib/python3.10/site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in ./miniconda3/envs/reana/lib/python3.10/site-packages (from pandas) (2023.3)
Collecting tzdata>=2022.1 (from pandas)
  Downloading tzdata-2023.3-py2.py3-none-any.whl (341 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 341.8/341.8 kB 12.2 MB/s eta 0:00:00
Requirement already satisfied: six>=1.5 in ./miniconda3/envs/reana/lib/python3.10/site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Installing collected packages: tzdata, pyparsing, pillow, packaging, numpy, kiwisolver, fonttools, cycler, pandas, contourpy, matplotlib
  Attempting uninstall: pyparsing
    Found existing installation: pyparsing 3.1.0
    Uninstalling pyparsing-3.1.0:
      Successfully uninstalled pyparsing-3.1.0
Successfully installed contourpy-1.1.0 cycler-0.11.0 fonttools-4.40.0 kiwisolver-1.4.4 matplotlib-3.7.2 numpy-1.25.0 packaging-23.1 pandas-2.0.3 pillow-10.0.0 pyparsing-3.0.9 tzdata-2023.3
(reana) basti@dragon ~ % export REANA_SERVER_URL=https://reana-p4n.aip.de
(reana) basti@dragon ~ % export REANA_ACCESS_TOKEN=86hWnxfZqR9sRBf6_94J0Q
(reana) basti@dragon ~ % git clone https://github.com/reanahub/reana-demo-root6-roofit
Cloning into 'reana-demo-root6-roofit'...
remote: Enumerating objects: 314, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 314 (delta 10), reused 18 (delta 8), pack-reused 286
Receiving objects: 100% (314/314), 88.00 KiB | 4.40 MiB/s, done.
Resolving deltas: 100% (159/159), done.
(reana) basti@dragon ~ % cd reana-demo-root6-roofit
(reana) basti@dragon reana-demo-root6-roofit % reana-client run -w root6-roofit
```

First workflow

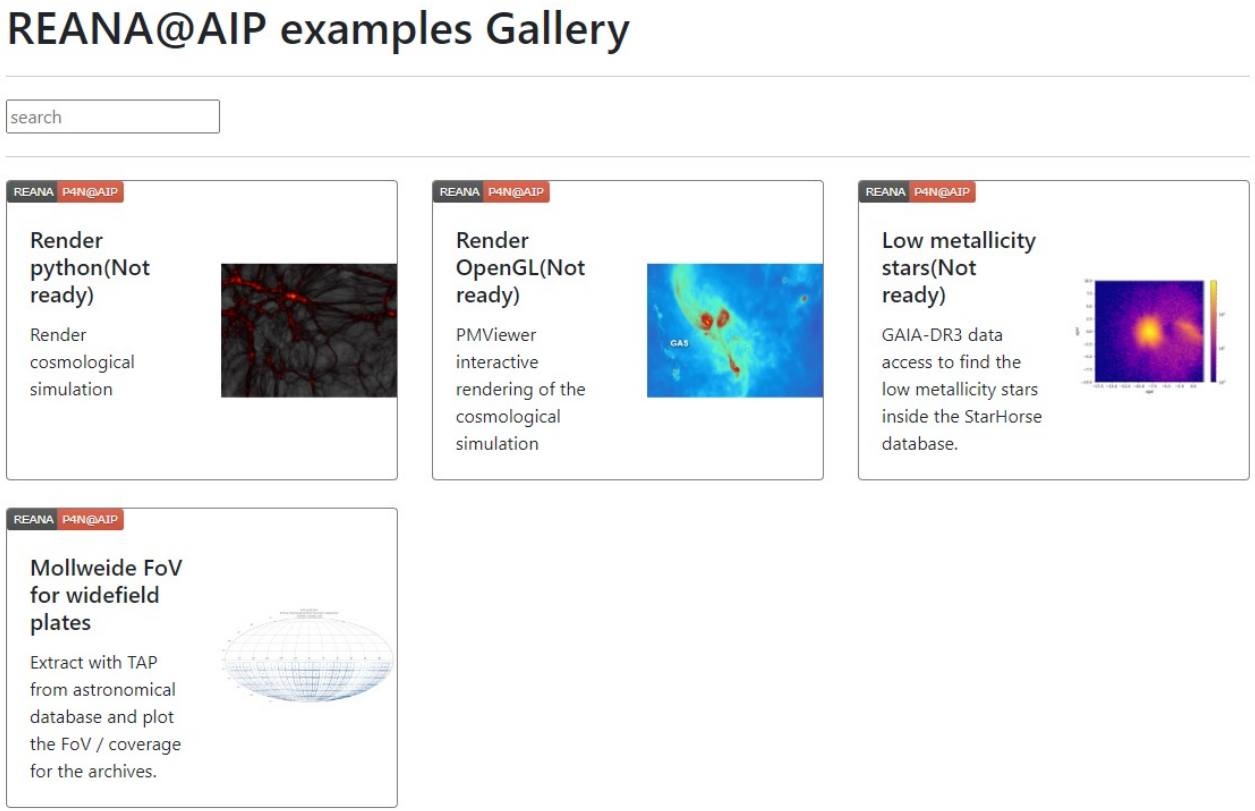
- Create: reana.yaml

```
version: 0.9.0
inputs:
  files:
    - plot_poly.py
workflow:
  type: serial
  specification:
    steps:
      - environment: gitlab.aip.de:5005/akhalatyan/gpu-on-newton:reana-python.117
        commands:
          - python plot_poly.py
outputs:
   files:
     - plot.png
```

- reana-client ping
- reana-client run -w plot1
- reana-client download -w plot1

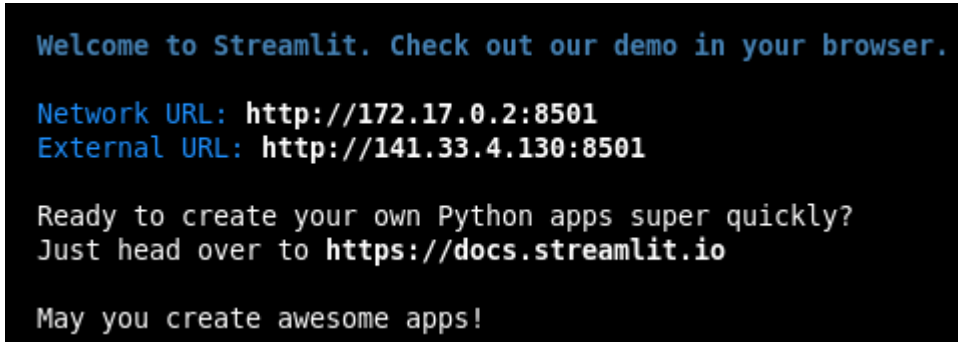
More examples:

- https://reana-p4n.aip.de/44cc543d-b1ac-4c88-9be8-2ad56c70f57d/?token=oIQmPUuTh4LOCEPVyixAHQ



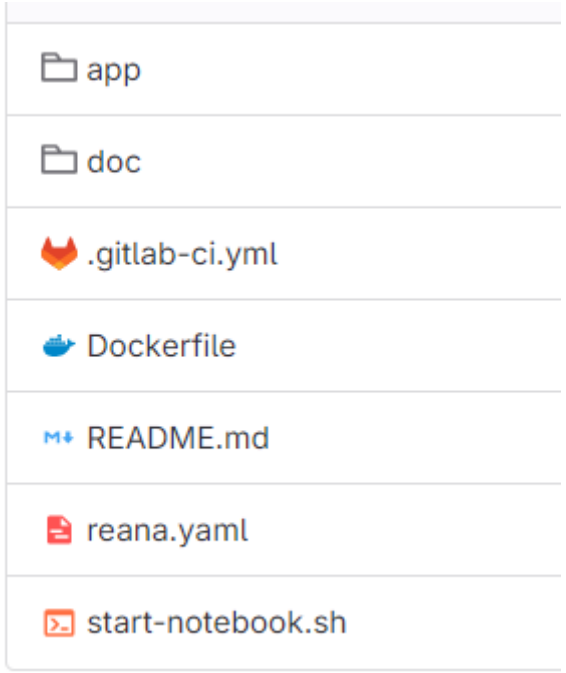
AIP container registry usage example:

- git clone git@gitlab.aip.de:akhalatyan/okd-aiko.git
- cd okd-aiko/
- docker login gitlab.aip.de:5005
- docker build -t gitlab.aip.de:5005/akhalatyan/okd-aiko .
- docker push gitlab.aip.de:5005/akhalatyan/okd-aiko
- docker run -it gitlab.aip.de:5005/akhalatyan/okd-aiko
- ENJOY:



own custom containers:

- https://gitlab-p4n.aip.de/arm2arm/reanatest



reana.yaml 278 B

```yaml
version: 0.3.0
inputs:
  parameters:
    shrepo: https://gitlab-p4n.aip.de/arm2arm/reanatest.git
workflow:
  type: serial
  specification:
    steps:
      - environment: jupyter/scipy-notebook
        commands:
          - git clone "${shrepo}"
          - echo "Hello1"
```