Leibniz-Institut für
Astrophysik Potsdam

KuFAIR

a kubernetes/docker based workflow with FAIR principles

One use-case from astronomy

The components of the use case:

(F) Found a paper with a published KuFAIR package (DOI / URL)

(A) Access the package

(I) Open the package in your browser (locally or in the portal)

(R) Interact with the data and metadata (provenance)

Kubernetes 4 FAIRness: KuFAIR

What can you do with KuFAIR v0.0.0 ?

DEMO

1. Create a „Package“ from a repository

2. Run a docker container setup from a „Package“ and
   Generate its FAIR related metadata (Provenance).

3. Explore a docker container interactively:
   explore the env., soft., deps. and products in your browser.

What do you have in a KuFAIR package?

1) Setup instruction for the container

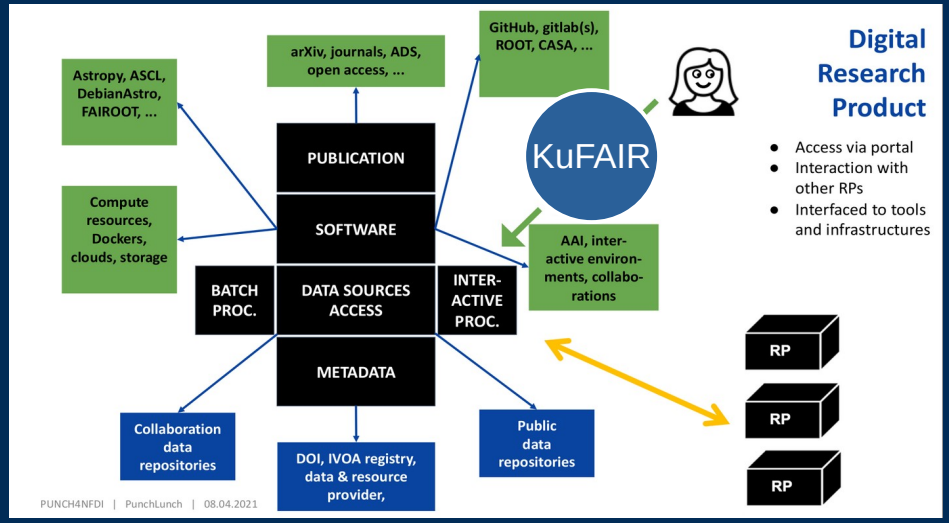
2) Trace of all related volumes: parameters, dependencies, and products

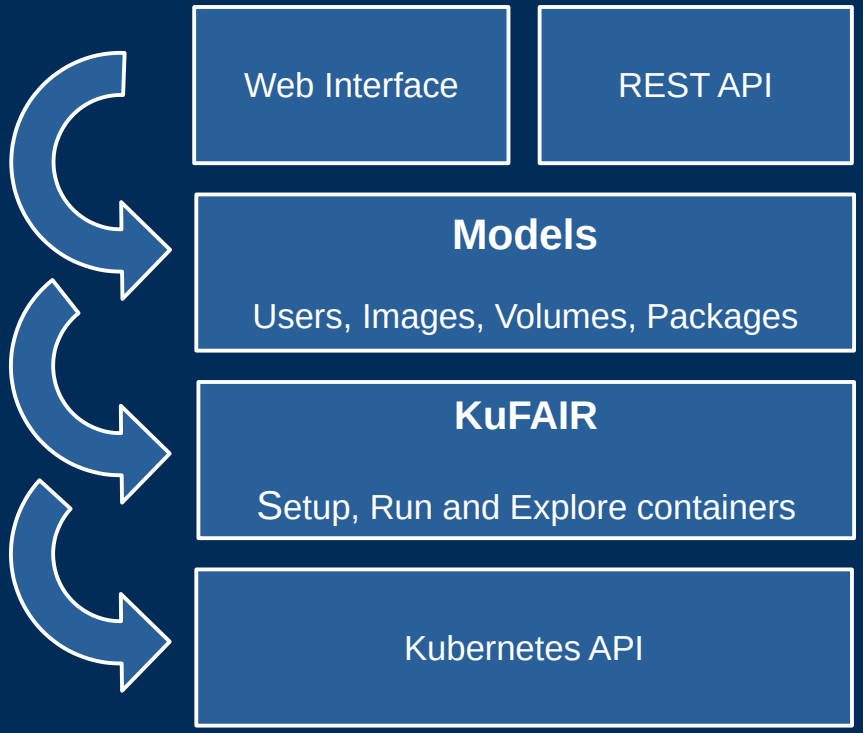
3) Documentation, authorship

4) Docker Image (optional)

5) Snapshot of the git-repository (optional)

KuFAIR in NFDI context



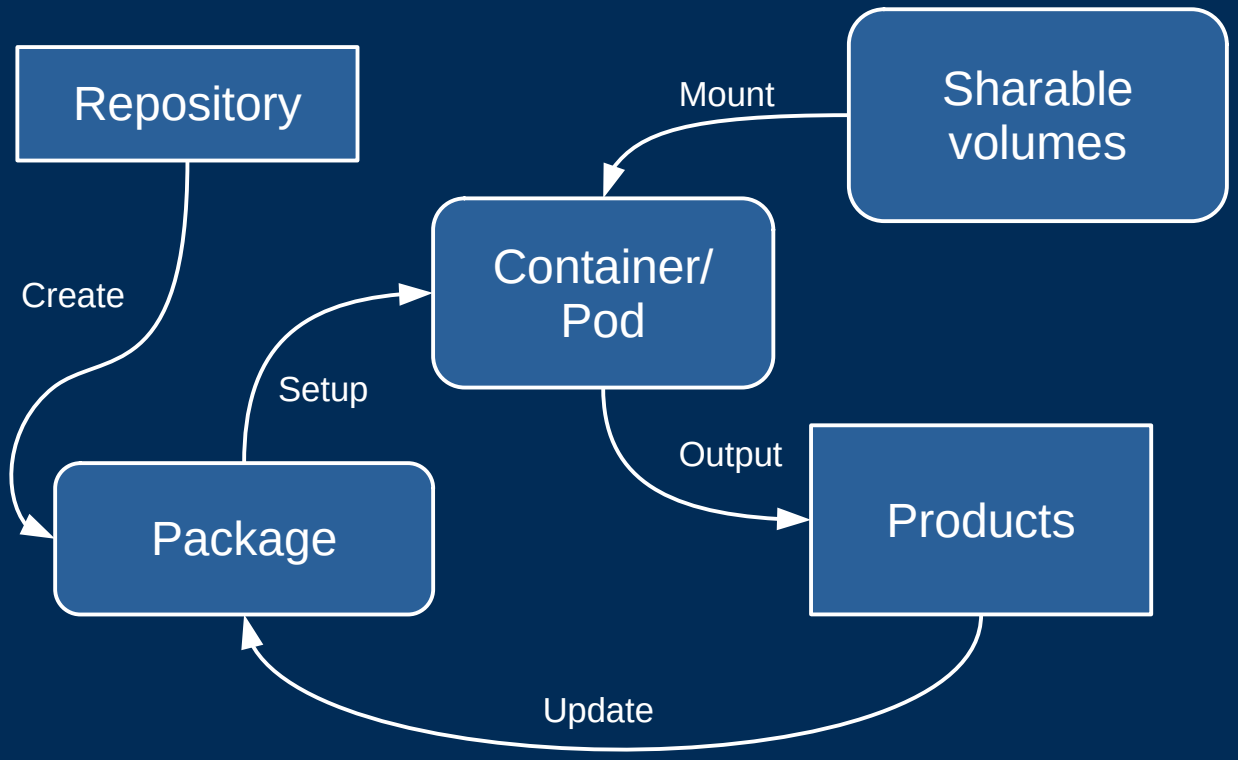


What is KuFAIR ?

KuFAIR is a layer between the **users** and **containers** deployed on kubernetes

It offers:
- A simple API: *create, run, explore*
- Guarantee the FAIRness of the data (end-to-end FAIRness)

What is the difference between KuFAIR packages and Jupyter Notebook?
- KuFAIR contains the full executable environment (no install, no broken-kernel)
- KuFAIR links the (full identified) data. Does not store them.
- KuFAIR can be reproducible (full provenance)
- KuFAIR can be reusable: plug-and-play your data workflow

How KuFAIR makes products FAIR?
- **Findable**: VO, OAI, S3 address seriously this issue (KuFAIR builds on them)
- **Accessible**: Package are small data < 100 MB. Not a significant problem (URL, S3 for data)
- **Interoperable**: interact with any docker(local)/kubernetes(cluster) instance
- **Reusable**: Provide the full provenance, and executable environment.

How KuFAIR works ?