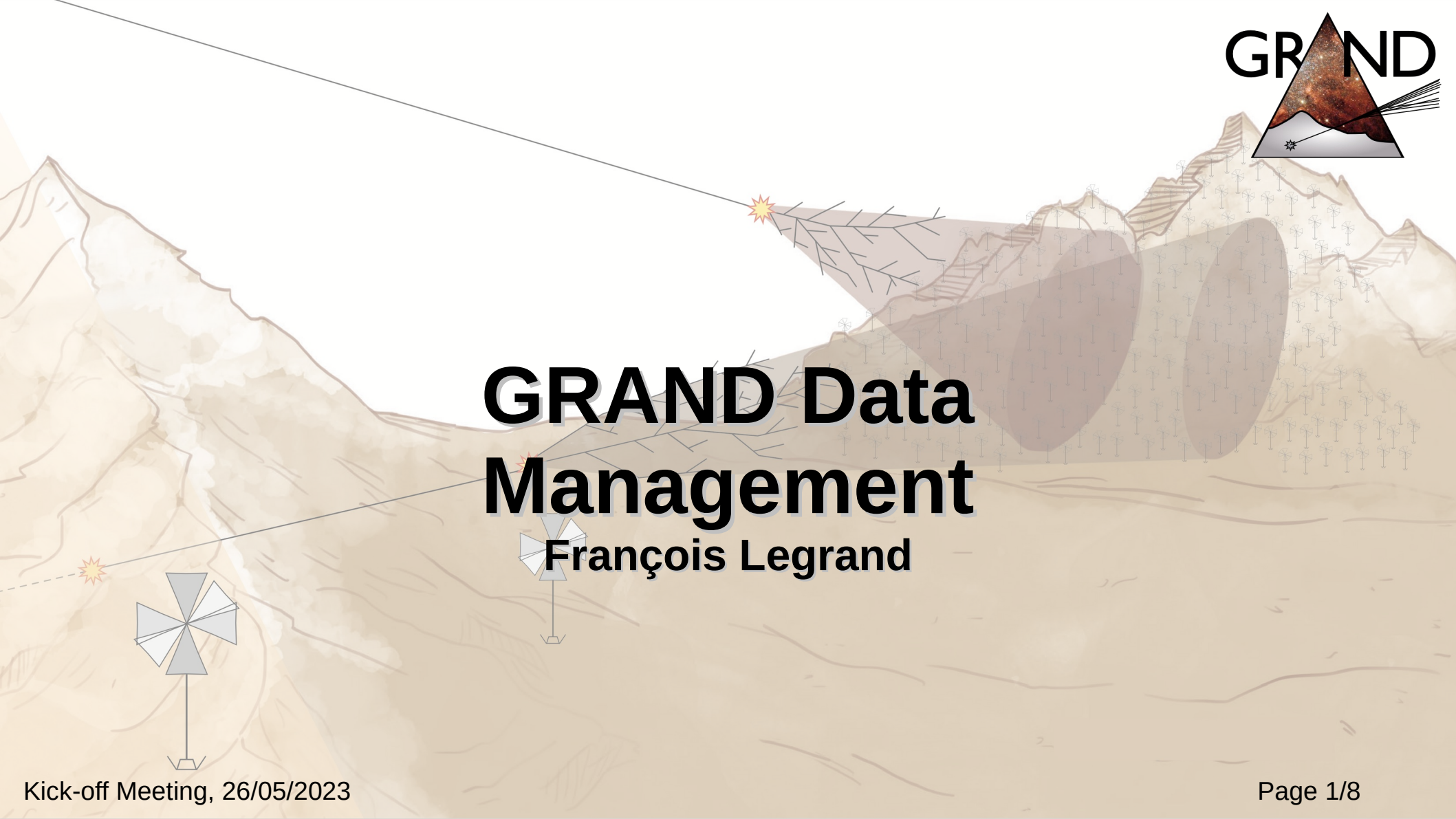


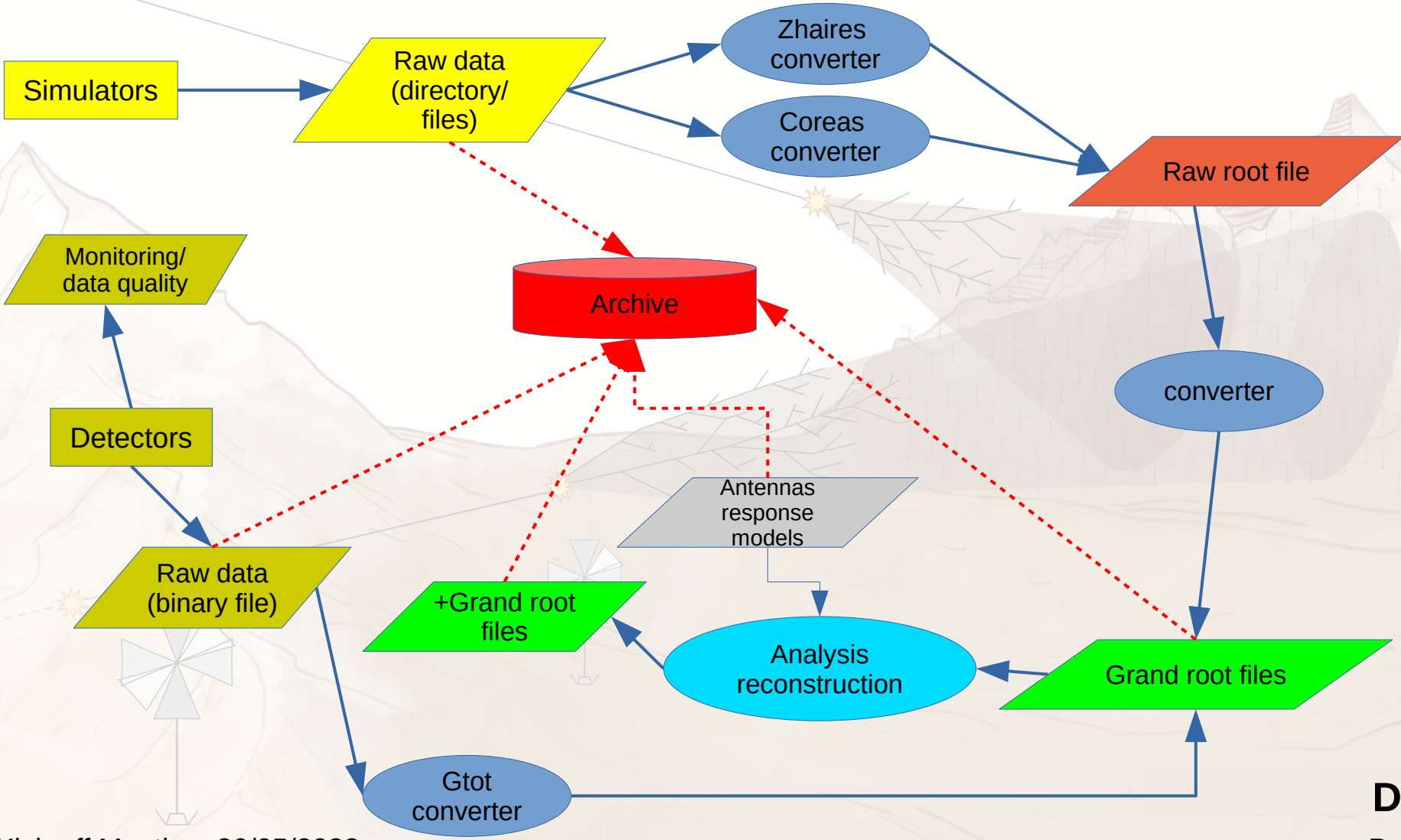
# GRAND Data Management

François Legrand

## Objectives

- Identify products
  - Type of data (sims, obs, calib,…) and type of files (root, npy, npz, txt, bin,...)
  - What should be kept and referenced
  - What is linked/needed by what ?

- Define storage strategy
  - Grand Root files
    - Structure (splitting of files, directory storage,…)
    - Updates strategy (add new files vs modify existing files,… )
    - Naming convention
  - Other files


Simulators
Raw data (directory/ files)
Zhaires converter
Coreas converter
Raw root file
Monitoring/ data quality
Archive
converter
Detectors
Antennas response models
Raw data (binary file)
+Grand root files
Analysis reconstruction
Grand root files
Gtot converter

# Dataflow

| Source | Type | Format | To be saved | Comments |
|---|---|---|---|---|
| Simulators | Simulations | Simulator custom (dir+files) | **Yes** | Tar to save. |
| Simulators converters | Simulations | Raw root files | no | We keep these files during first stage for testing but they should be removed at the end |
| Raw to Grand converter | Simulations | Grand root files | **Yes** | |
| Simulators | Models | | no | Part of simulators |
| Detectors | Row data observations | bin | **Yes** | |
| Detectors | Antennas response model | npy | **Yes** | |
| Detectors | Monitoring, data quality, ... | ? | ? | Not existing now. Should be saved (maybe in another DB) when available |
| Gtot converter | Observations | Grand root files | **Yes** | |
| Codes | Reconstructions etc... | git | **Yes** | |

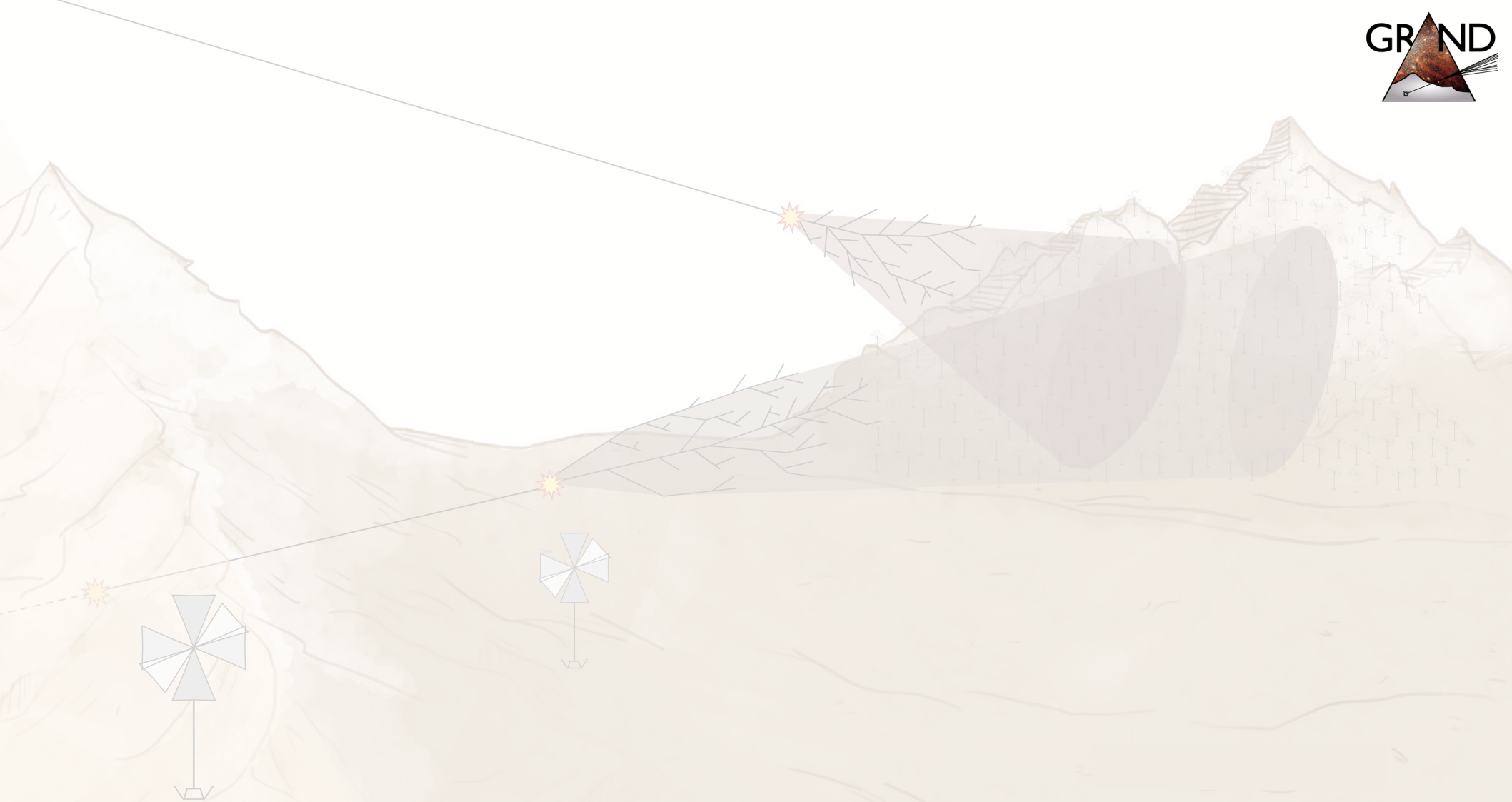
## Datas

# GRAND root files storage

- An observation or a simulation (run ?) will be stored into a directory
- Directory will contains all root files related to the run/events (at least one with the trun and tadc for observations or trun and tshowersim for simulations (TBC) ?)
- Initial files produced should go into a single file (or not !)
  - ⇒ identify what is produced initially
  - ⇒ naming convention for initial files needed (directory_name.root ? Pb if renaming, inirun.root ?)
- Files will not be modified once they are created/registered
- Any step in the analysis or reconstruction producing new data or modifying existing ones will generate a new file in the directory
  - ⇒ mechanism to trace processing needed (should be in the DB but it would be nice if it can be also in the file)
  - Do we produce a new file by ttree or group ttrees by type (run/event) or by analysis level, … ?
  - ⇒ naming convention for new files needed (depends on previous question, can be initial_name+treetype+version.root, initial_name+analysis_level.root, ...)

# GRAND root directory naming

- Directory named from the source. **This have to be unique (at least for official repo) !!!**
  - Name can be created in gtot for observations and in raw root to grand root converter for simulations ⇒ users cannot choose the name for their files !
  - or when registering (renaming) into the official repository (+DB) ⇒ users can name the files as they want but they will be renamed (keeping trace of the original name in the DB) when integrated into the official repositories
  - Observation could be :
    - concatenation of site + date + ??
    - Hash (sha256) of root file (first trun produced ?)
  - Simulations :
    - Hash (sha256) of root file (first trun produced ?)



# Data management next topics

- Data inventory and flow : identify all types of data (and format) we have to manage and determine what should be kept and referenced and what is available elsewhere and then useless to store.
- Data organization (format, which level of splitting, directory grouping, naming conventions, updates -or not- rules,...)
- Data structure (additional fields at file level e.g. unique identifier for all files related to a same observation/simulation,...)
- Versioning (versioning of root file structure in files and code release to read/write data files, tag in code/libs and files, ...)
- Database functionalities/Datamanager (DB content, use cases e.g. useful search, interface -web/python/?-, methods to retrieve data, …)
- Data hosting (Where ? Access rules/security/confidentiality ? Access protocols -scp,ftp,irods,http,..- ? Archiving and replication, …)
- Registration rules (data integrity check, validation process, …)