Grid LOFAR Reduction Tools

Moving Radio Astronomy to a Grid location.. and beyond!

ISGC 2017

Taipei, Taiwan

Alexandar Mechev

March 10, 2017

Contents

LOFAR Introduction

LOFAR Low-Band Array Antenna

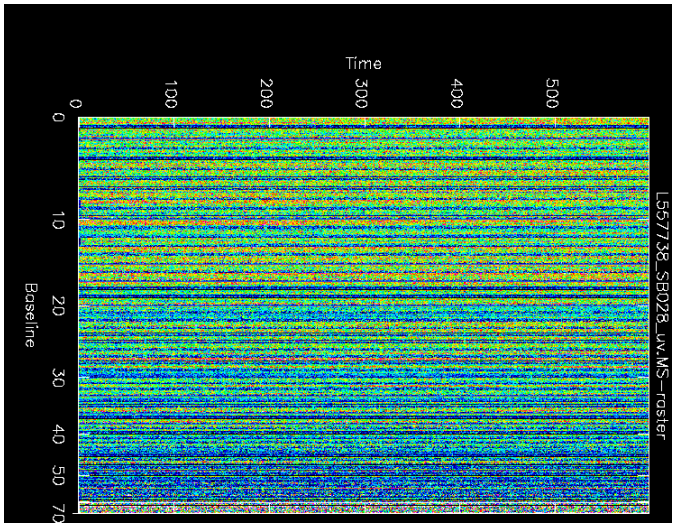
- 20-80MHz, 120-250MHz
- 7000 Antennae
- 24 core, 14 remote, 12 international stations
- 50m -> 1500 km baselines (van Haarlem et al. 2013)

LOFAR Introduction 2

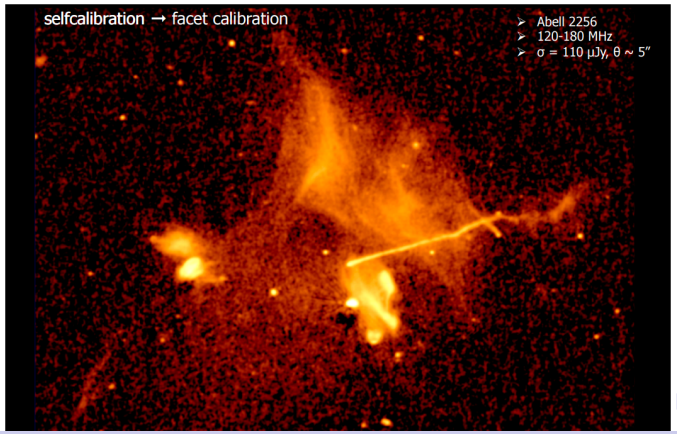
Raw data is 16 TB for 8h observation. Processing steps:

1. Flag RFI
2. *Average*
3. Demix Strong Sources
4. **Calibrate**
5. Image loop

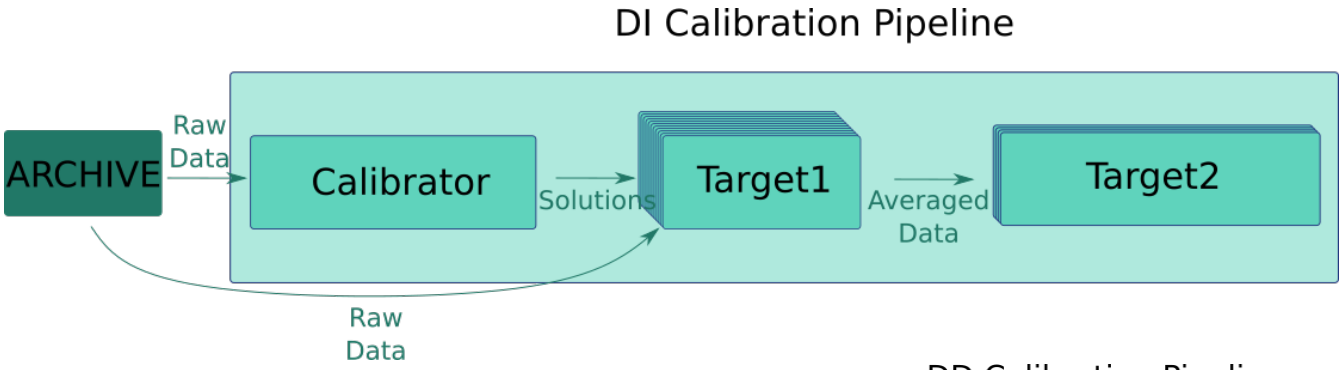
Goal: Go from initial data:



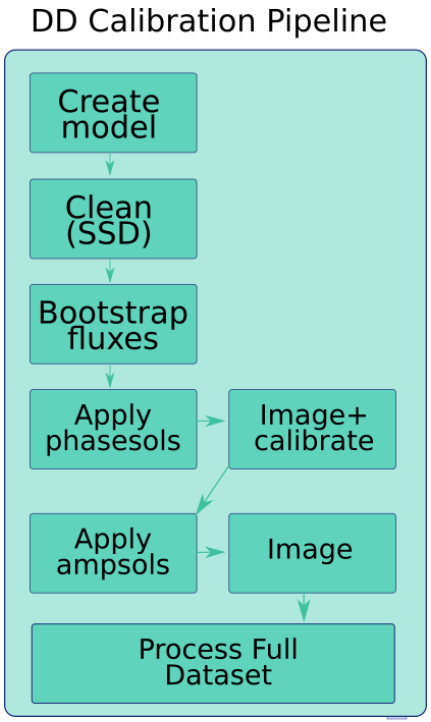
to Final image:



LOFAR Data Reduction

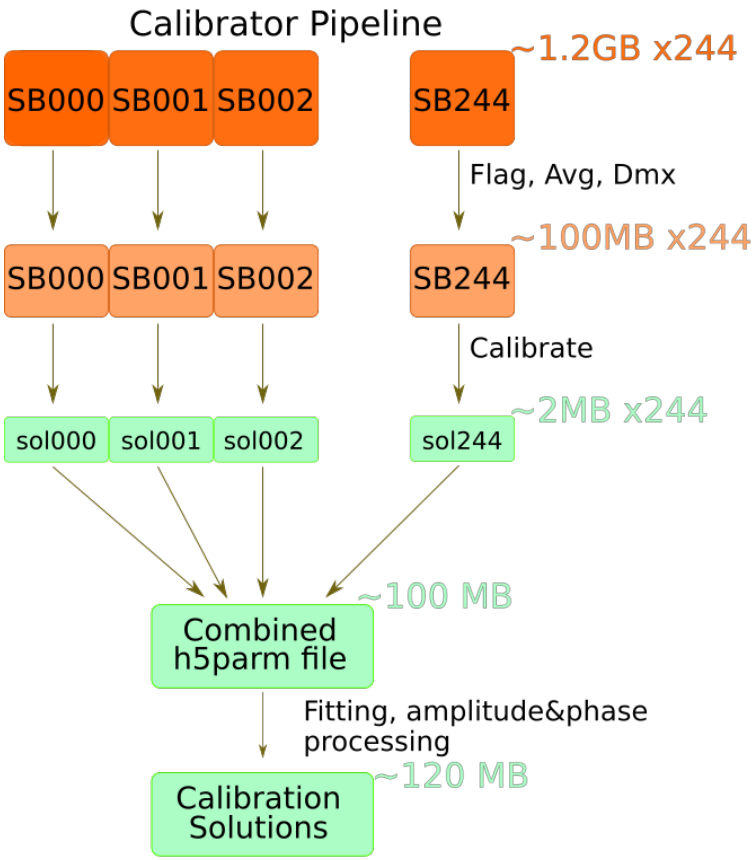


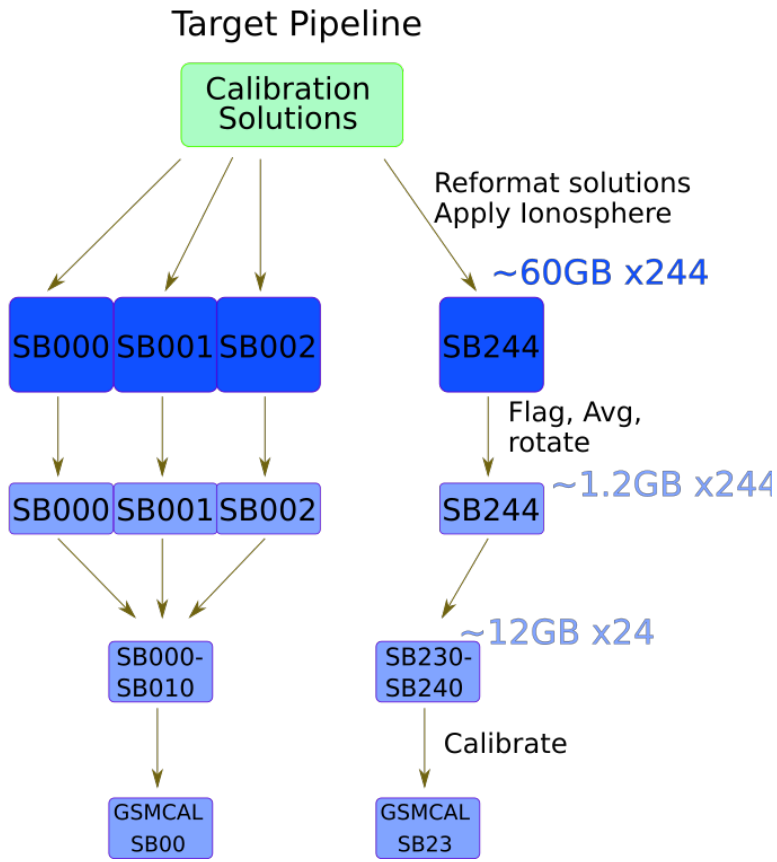
- Direction Independent/Dependent Effects
- Split data by frequency (DI) and by direction (DD)
- Majority of computation spent (LM) Minimization , FFT/iFFT and Gridding



LOFAR Data Reduction 2

Parallelization of pipeline steps.



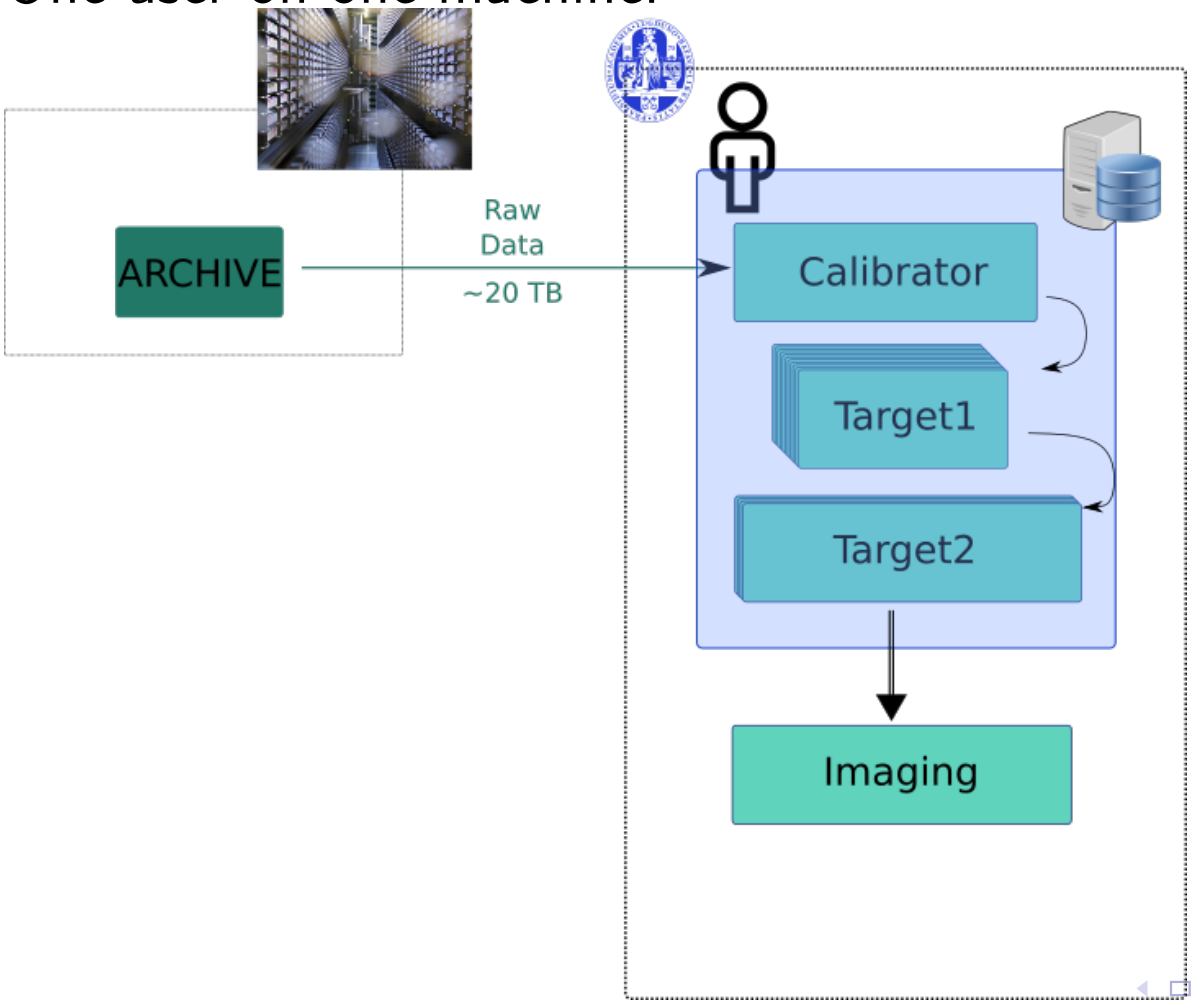


Data Rates and Processing Requirements

- Raw (antenna) data Rate: 80Gbps
- Correlated+Averaged Data: 50TB/6h
- Data produced in 2016: 5 PB
- LOFAR Survey:
  - 3100 observations of 16 TB each
  - Manual reduction takes $>$ 2wk/dataset
  - 5 Year deadline

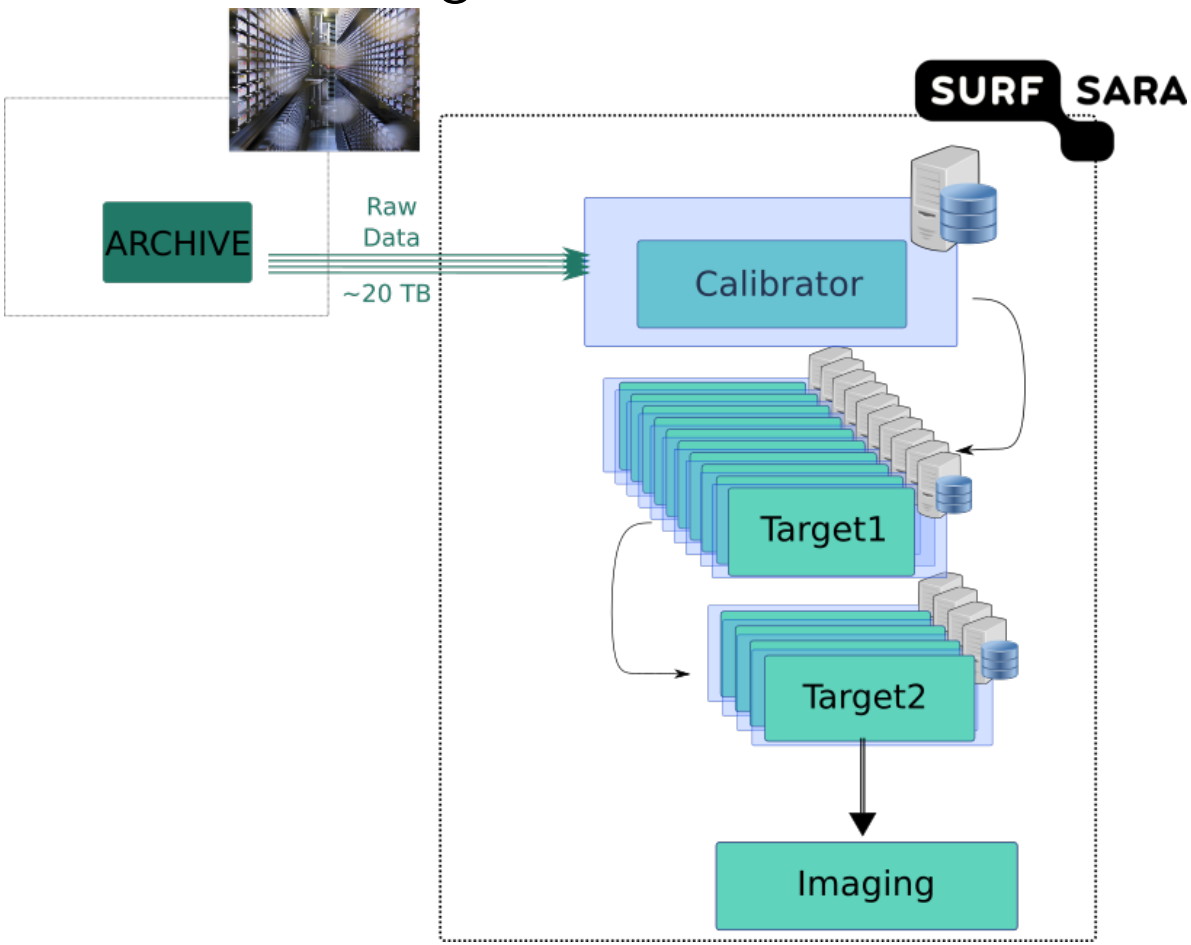
Reducing at institutes

One user on one machine:



Reducing at a GRID site

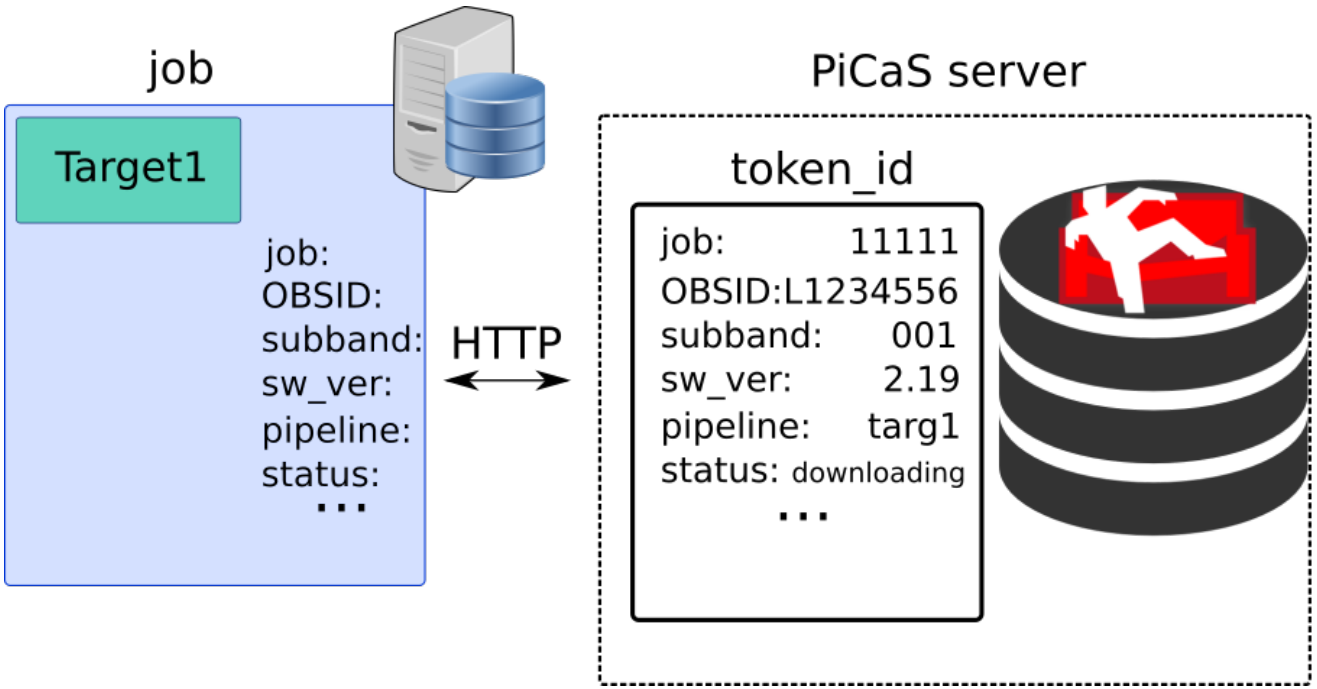
Automated management of data flow



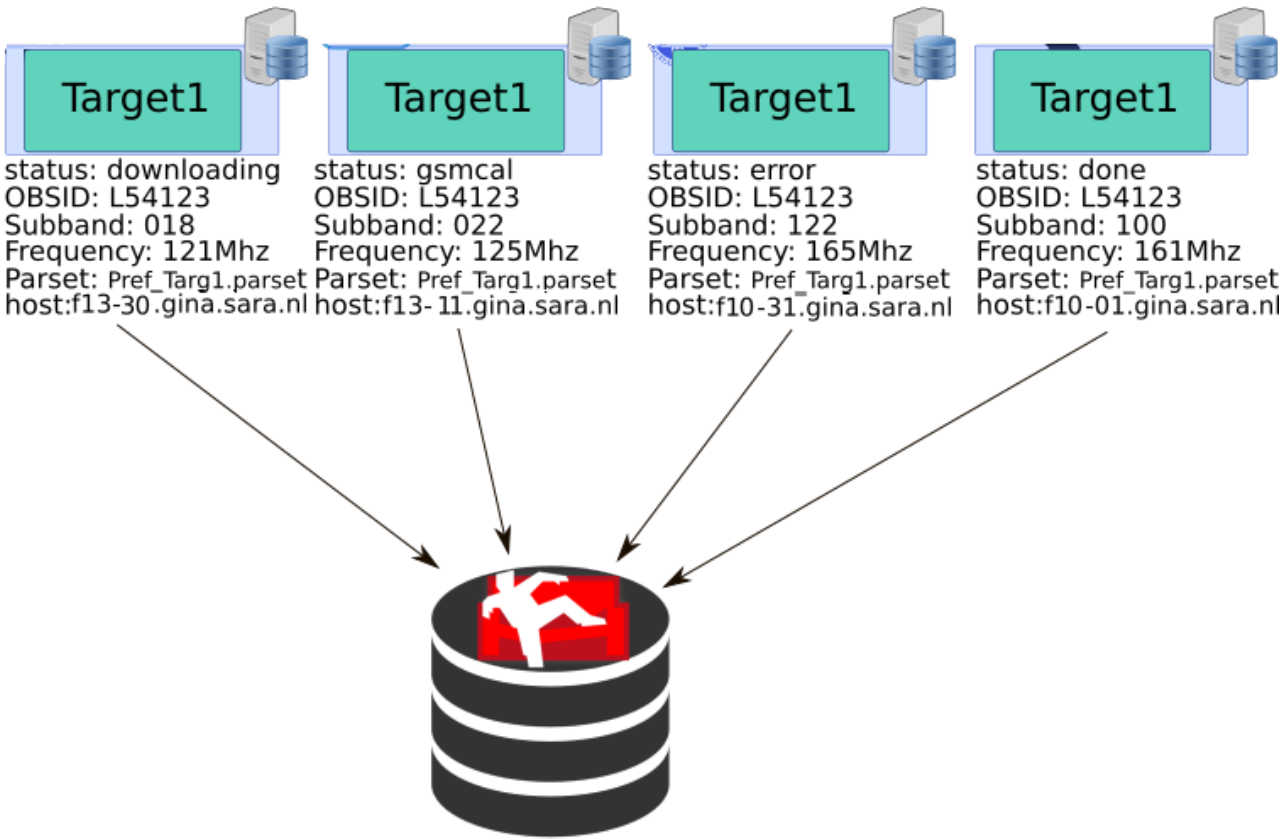
Challenges encountered

- No shared FS -> automate data retrieval/storage
- No login on worker machines -> automate logging
- Need to submit scripts as a pilot job.
  - i.e. submit a sandbox, sending files and commands to

PiCaS Tokens as 'job descriptors'



Storing status and metadata remotely



Views as 'linked jobs'

Can create views using any token field.

Ex: doc.status='error' && doc.output=11 (all error tokens with exit status 11)

Ex: doc.pipeline='targ1' && doc.status='error' (all error targ1 jobs)

Ex: Tracking concurrent reductions!

These are especially useful to list waiting jobs!

Connecting scripts with metadata

short scripts that send and pull metadata

Writing Data to CouchDB

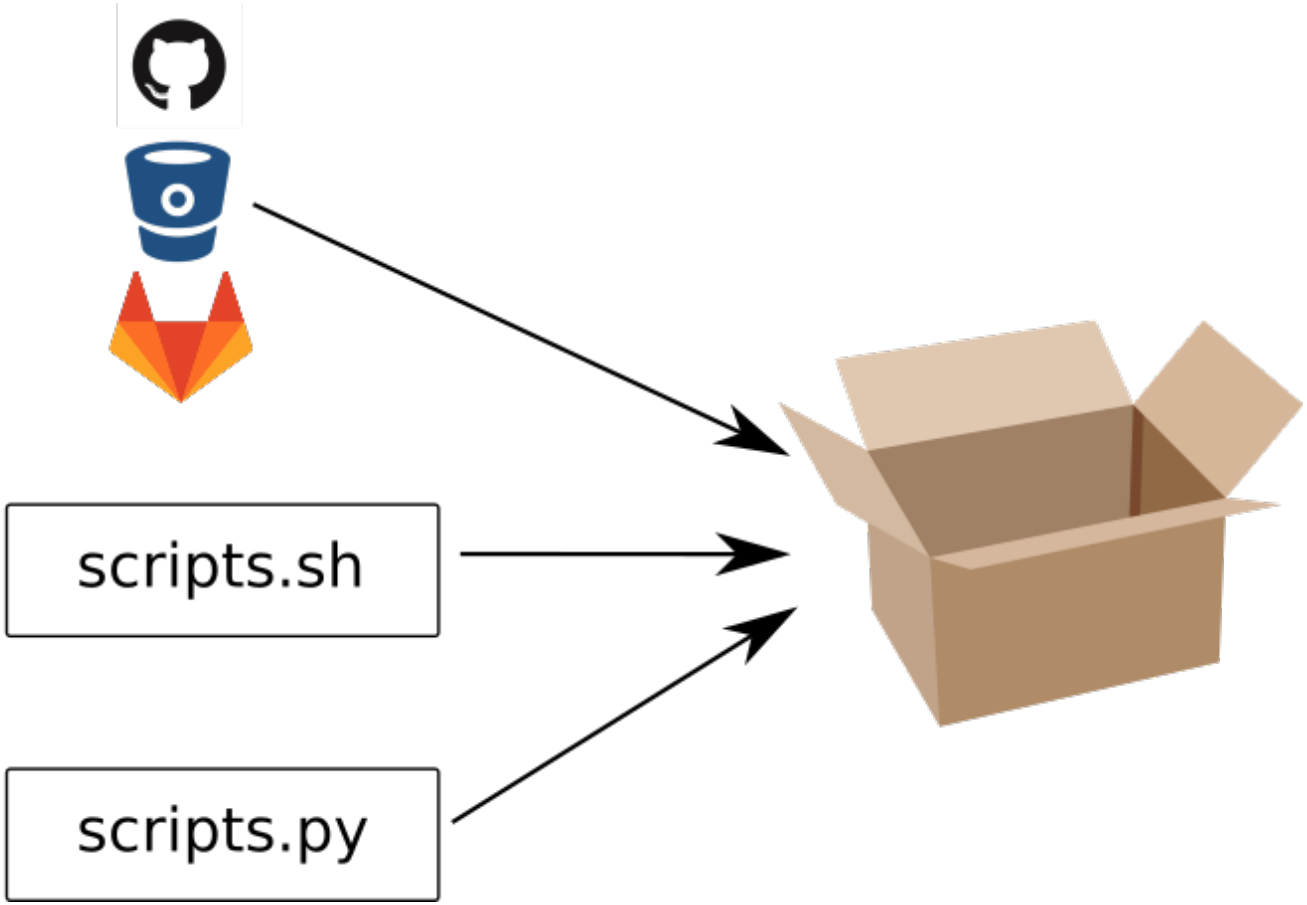
- PiCaS username
- PiCaS Password
- PiCaS database
- token_id
- field_name (key)
- value

Reading Data from CouchDB

- PiCaS username
- PiCaS Password
- PiCaS database
- token_id
- field_name (key)

Sandbox creation

Use a small yaml file to automate:



Sandbox upload and execution

- Sandboxes connected with observation (token 'type')
- upload location stored in token
  1. job starts on node
  2. job pulls helper scripts from git repo
  3. job locks a token (one token per job)
  4. job reads sandbox location and downloads sbx
  5. job untars sandbox and executes master.sh

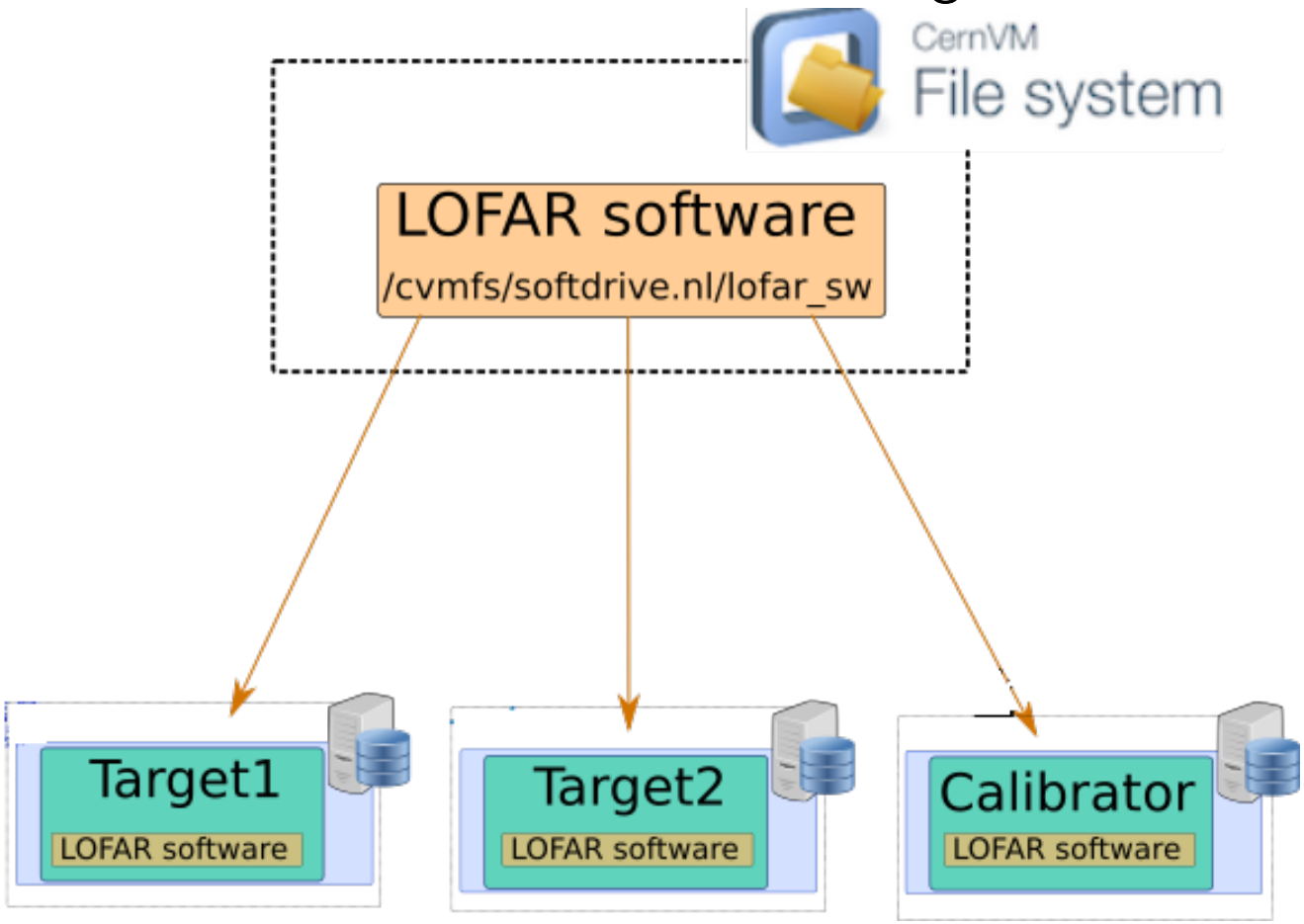
Sandbox execution on a node

master.sh procedure for LOFAR

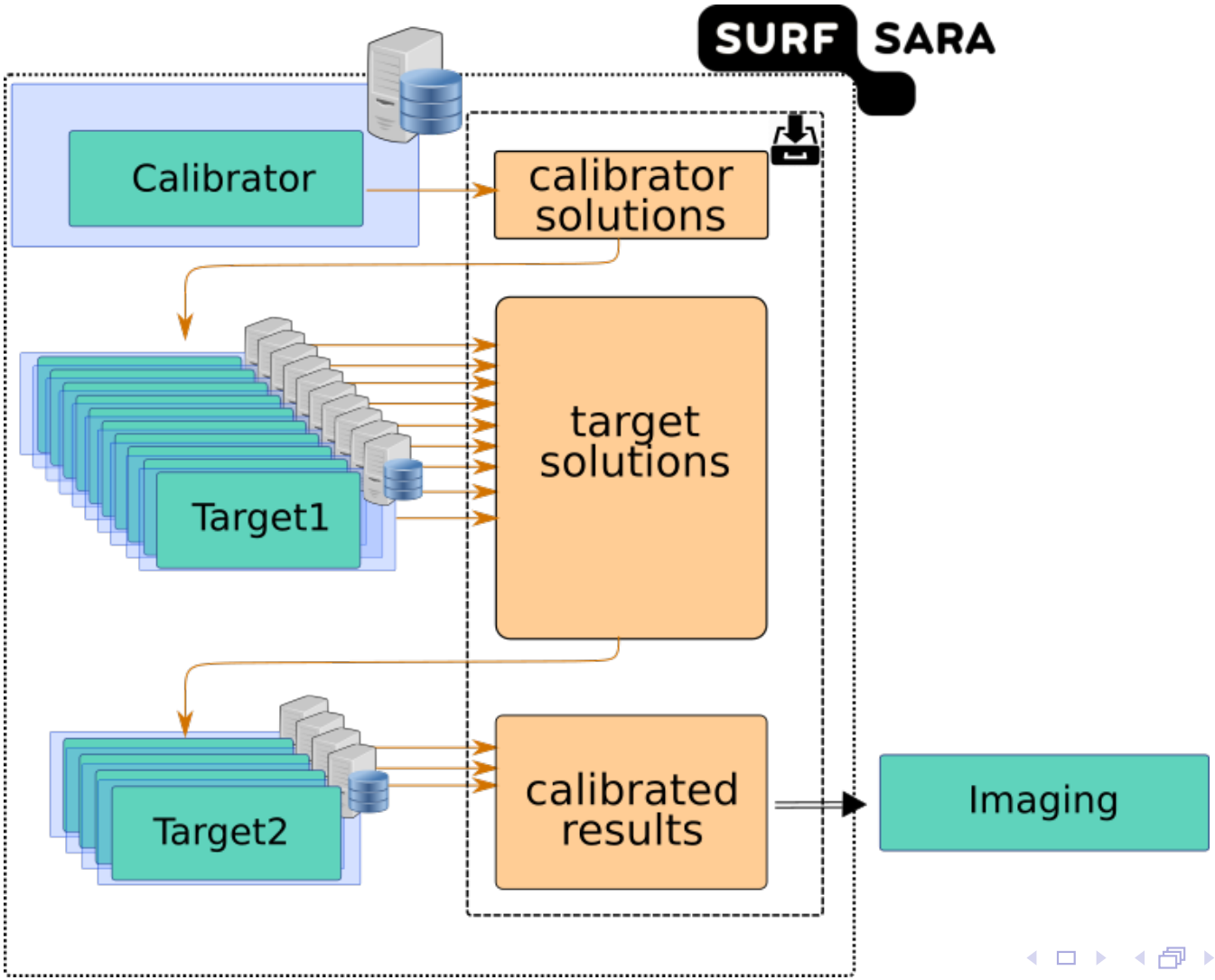
- Load functions into environment
- Read metadata and attachments from token
- Download data
- PROCESS DATA
- upload logs and plots to token

CernVM-FS software install

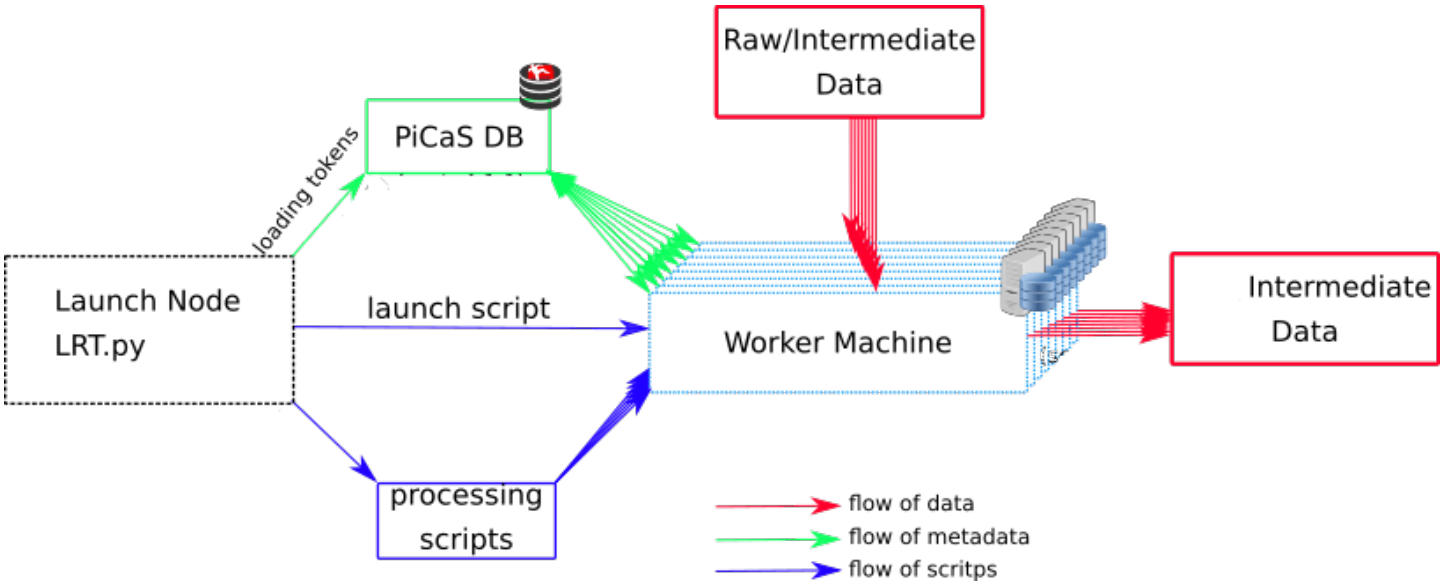
Global install of software mounted on grid nodes or cloud location



Using GRID Storage as an intermediate



Framework


Raw/Intermediate Data
PiCaS DB
loading tokens
Launch Node
LRT.py
launch script
Worker Machine
Intermediate Data
processing scripts
flow of data
flow of metadata
flow of scritps

Splitting Pipeline into Jobs

- Define each step:
  - Token structure
  - Sandbox:
    - scripts git
    - steps (set up env, download, process)
    - Token variables input
  - step_name
  - Parset, list of URLs
- Stage files
- Launch jobs

Submitting Jobs

Glite-WMS

Before

```
Executable = "/bin/sh";
Arguments = "startpilot.sh PICASDB USR PASSW ";
InputSandbox =
"sandbox/picas.tar","sandbox/couchdb.tar","sandbox/scripts.tar",
"sandbox/startpilot.sh","sandbox/pilot.py",
"sandbox/master_avg_dmx_v2.sh","sandbox/parsets.tar";
```

After

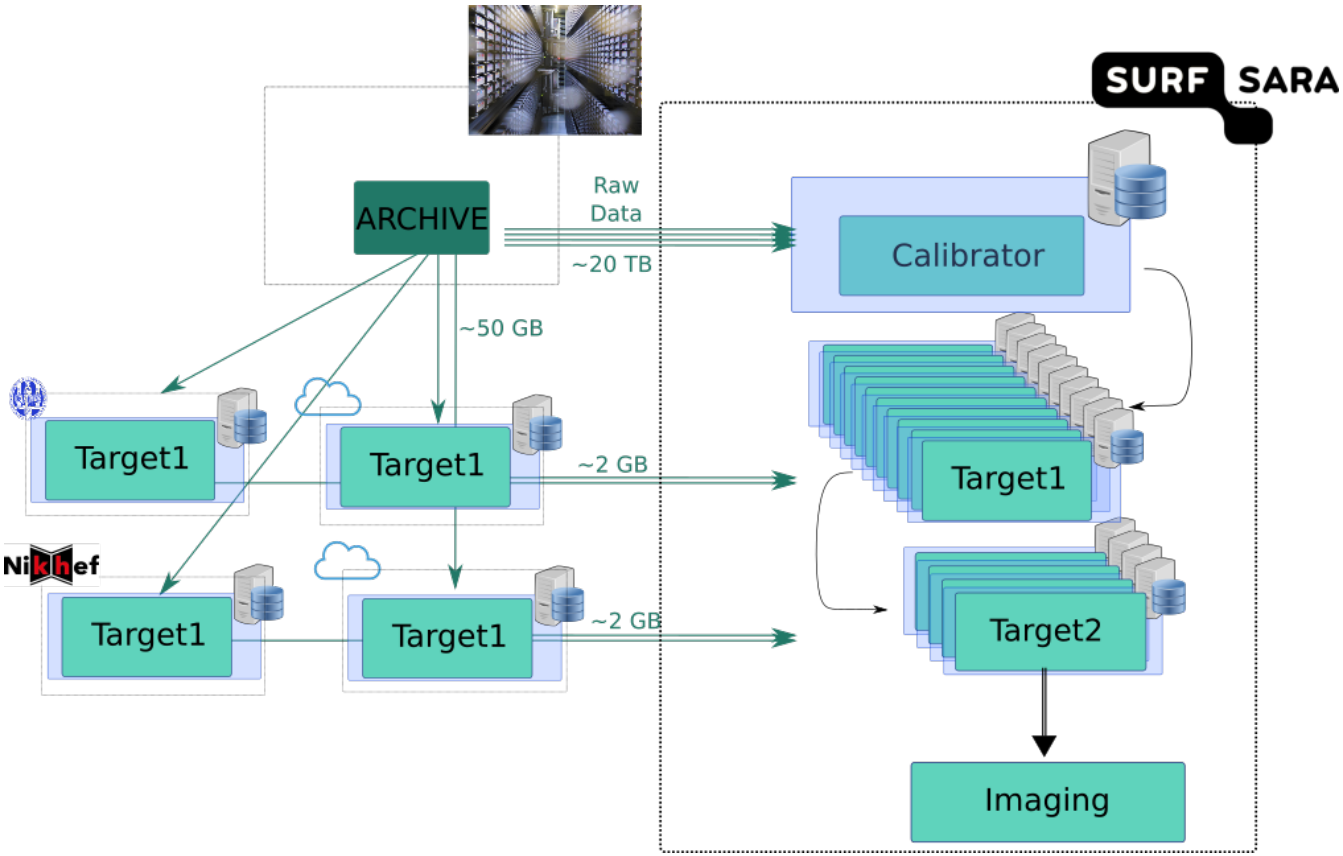
```
Executable = "/bin/sh";
Arguments = "run_remote_sandbox.sh PICASDB USR PASSW ";
InputSandbox = "run_remote_sandbox.sh";
```

Now it's easy to submit jobs! Just send a 20 line shell script
AND, can be submitted through other queues

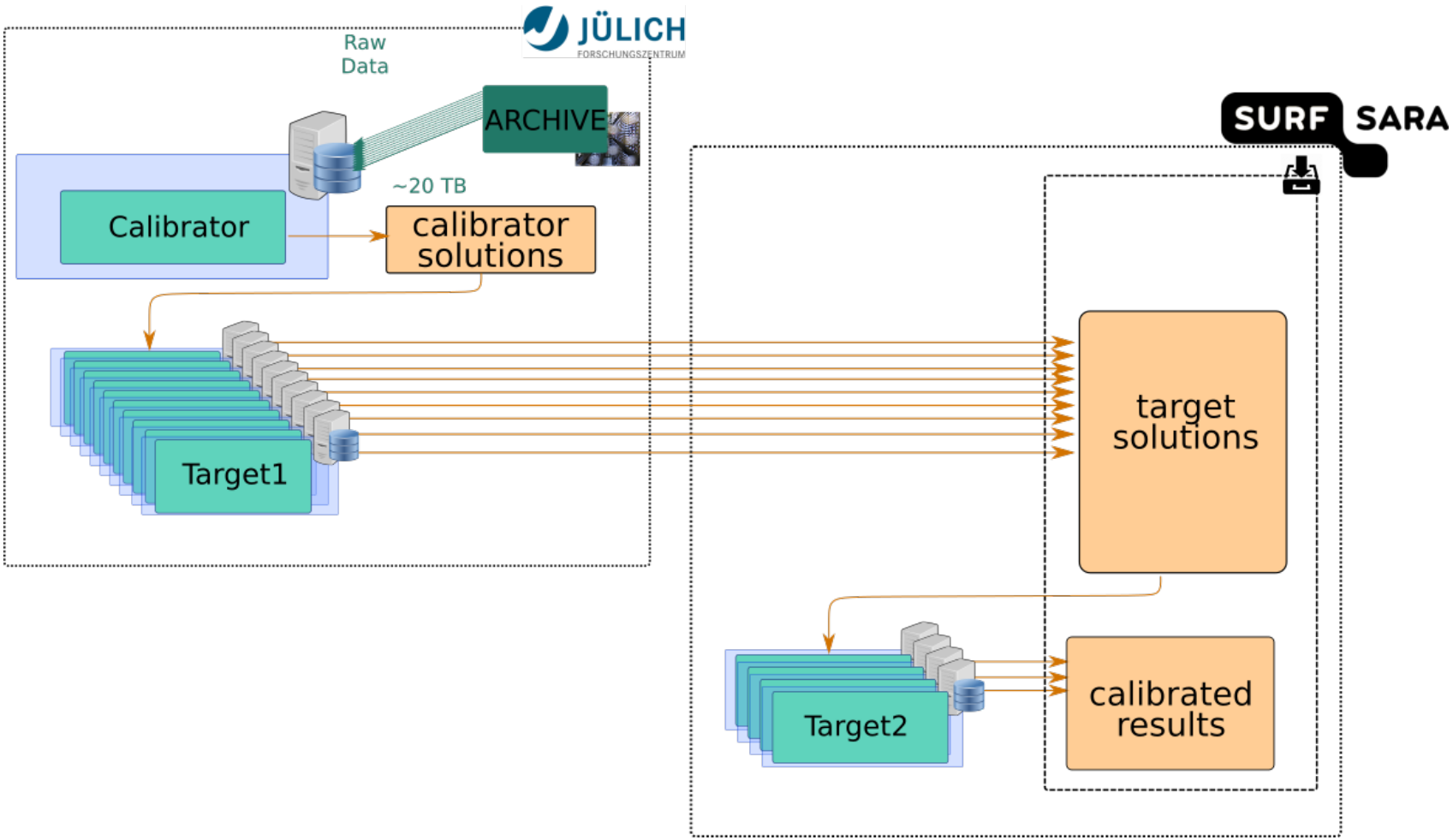
Results

15x speed-up out of the box
100+ Processed datasets in 4 Months
Can handle Multiple software packages (different pipelines)
Platform independence evolved naturally

Reducing on GRID + cloud (+institutes)

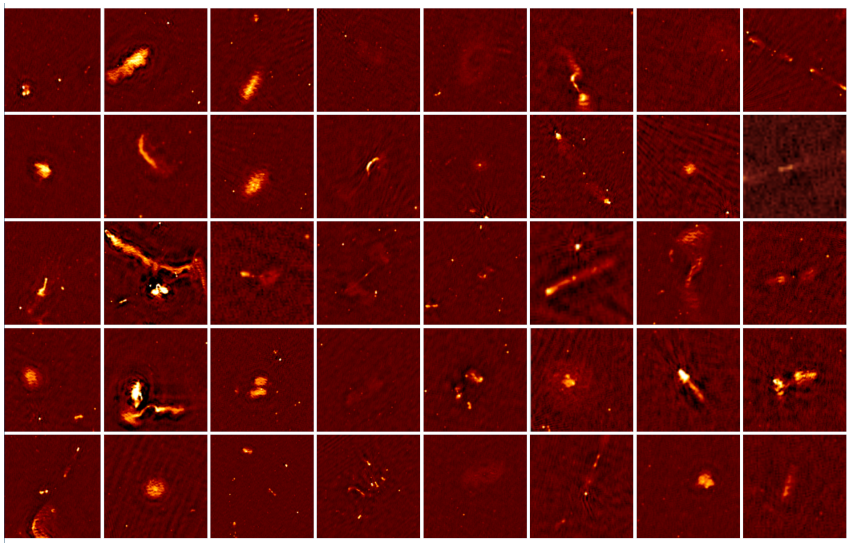


Processing Close to the Data



Overview

- Automating high-throughput processing
- Reduced unprecedented number of datasets
- Framework to send reductions at the archive sites
- Automate both pipelines
- Made images!



Acknowledgements and Source Code

Acknowledgements

Leiden University

Dr. H. Intema
Dr. A. Plaat
Dr. H. Rottgering
Dr. T.P. Stefanov
Dr. T. Shimwell
Dr. J.B.R Oonk

SURFsara

Dr. N. Danezi
Dr. C. Schrijvens

Sources

- Picas client: https://github.com/sara-nl/picasclient
- LOFAR Reduction Tools: https://github.com/apmechev/GRID_LRT
- Grid PiCaS tools: https://github.com/apmechev/GRID_picastools

Thanks for the attention!