Reduction of MeerKAT interferometric data in PUNCH4NFDI

A new cross-TA use case

In collaboration with Kristof Rozgonyi & Joe Mohr (LMU Munich)

Nicola Malavasi (LMU Munich) - Cross-TA meeting - 05/07/2023

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

PUNCH 4NFDI

Goal of the use case: radio data reduction

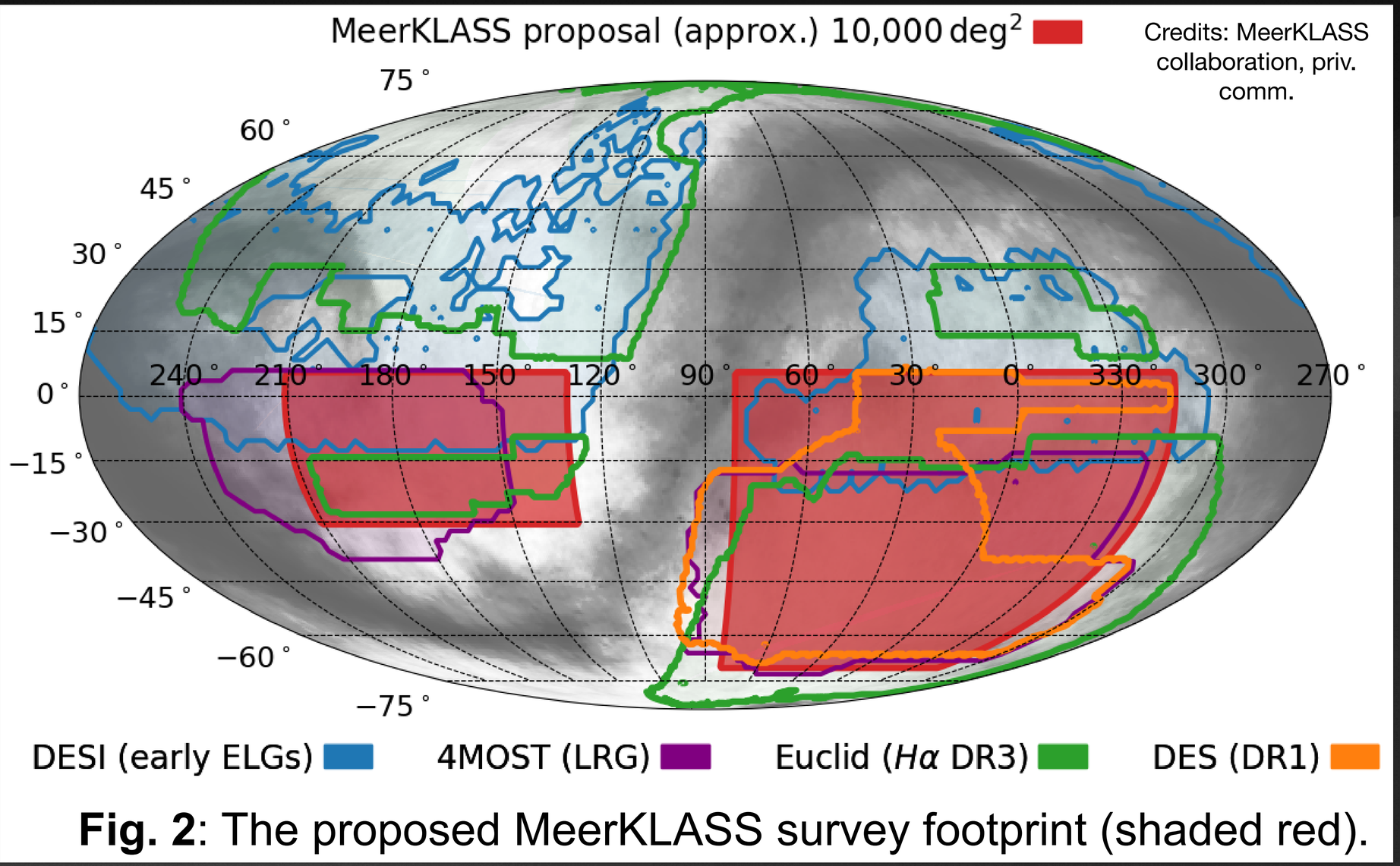
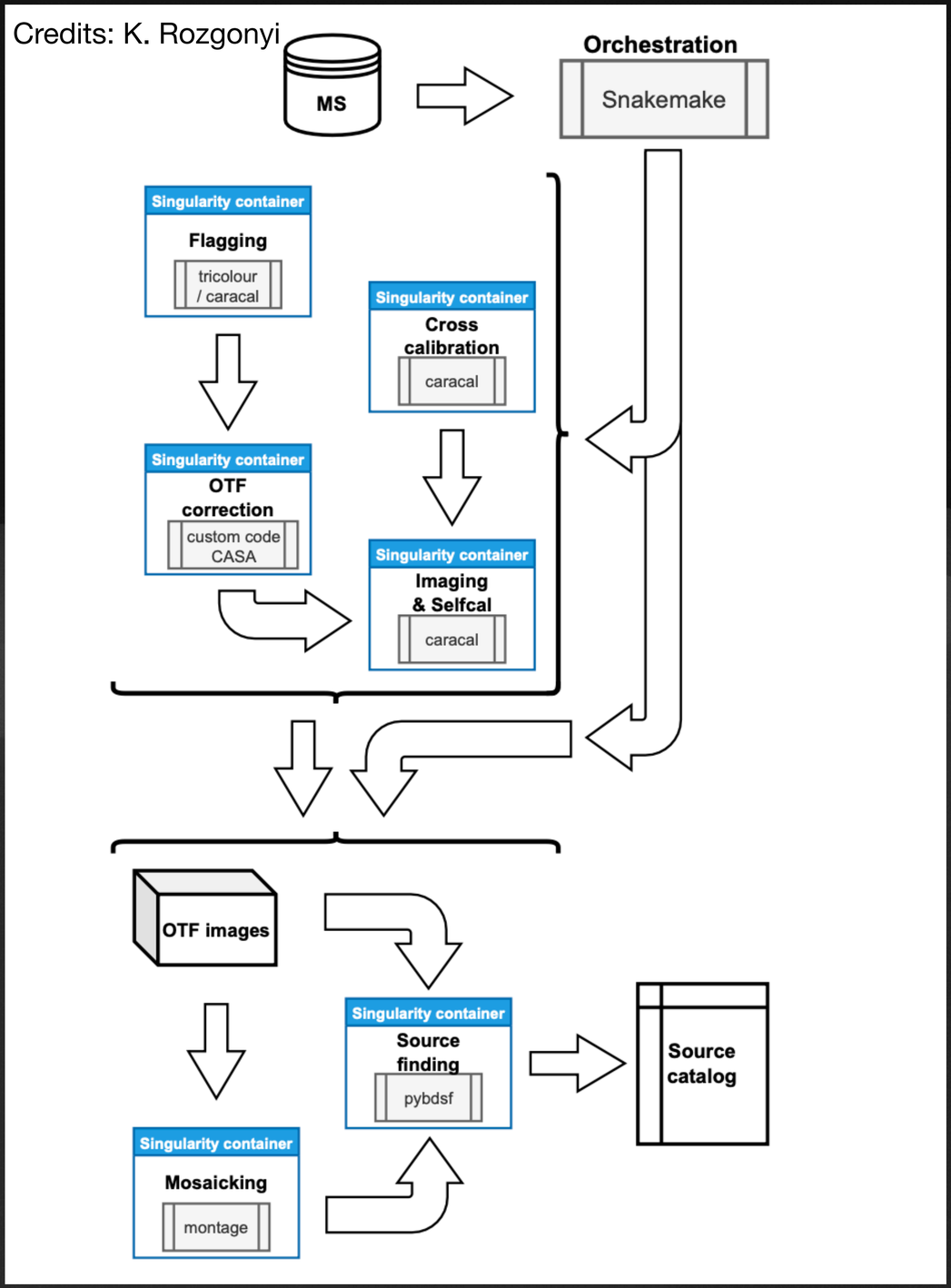


Fig. 2: The proposed MeerKLASS survey footprint (shaded red).

The MeerKLASS survey

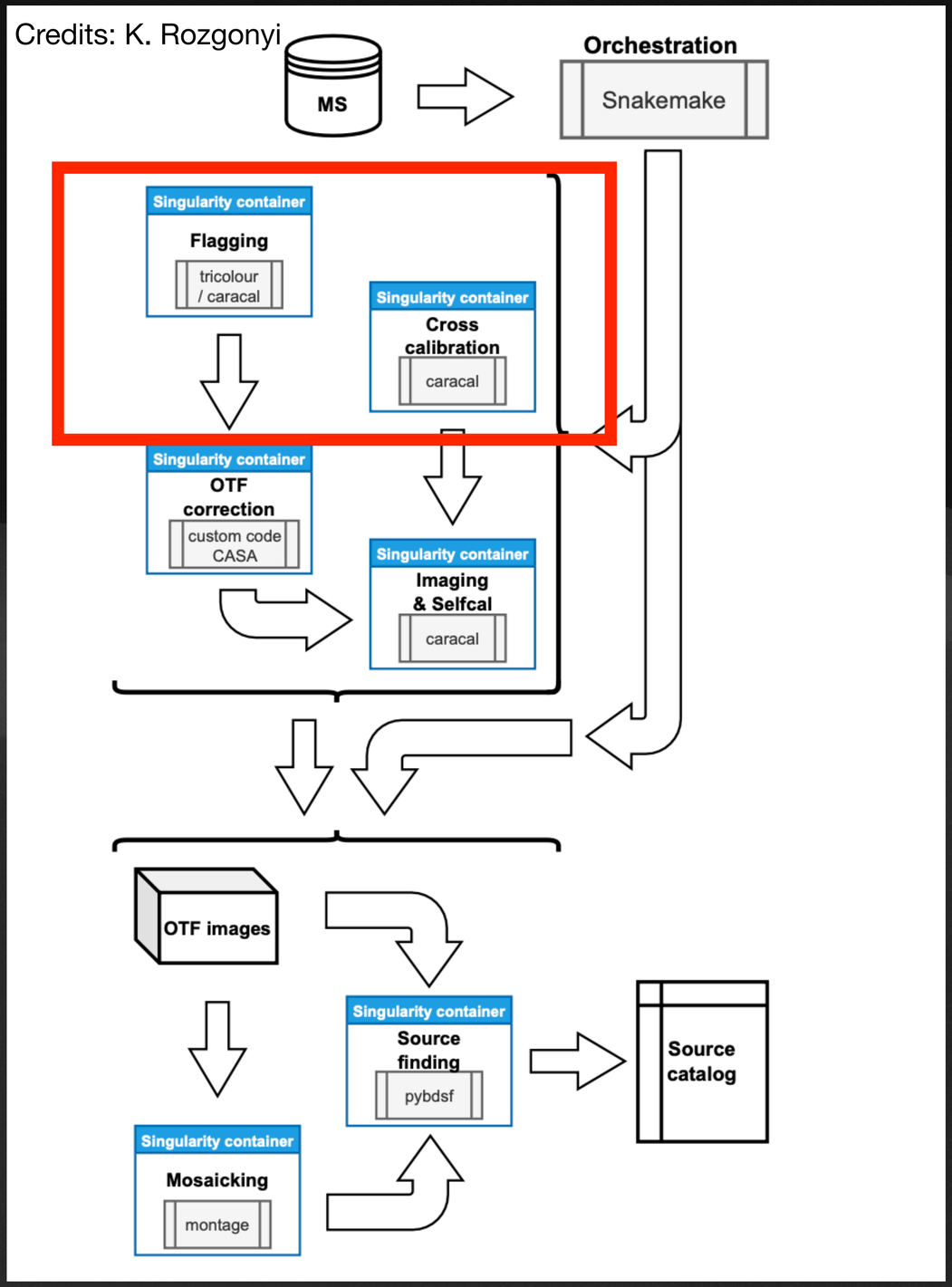
- 10˙000 sqdeg.
- ~2500 h of observations.
- 300 sqdeg already available (MeerKLASS pilot observations).
- New observing mode for MeerKAT: **On The Fly interferometric mode**.

Ad-hoc reduction pipeline



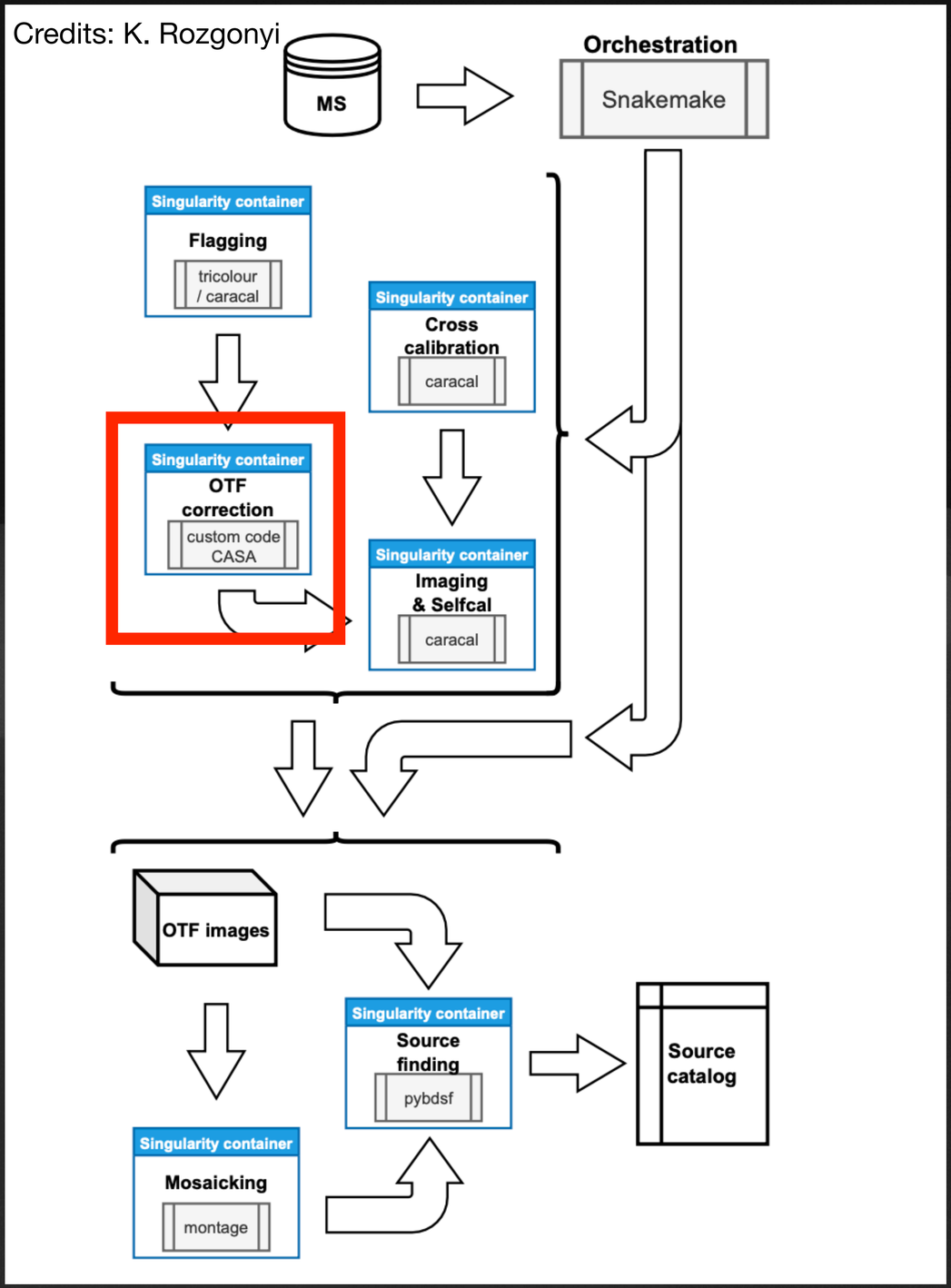
Ad-hoc reduction pipeline

- Flagging, cross- and self-calibration prepare the data (rather standard).



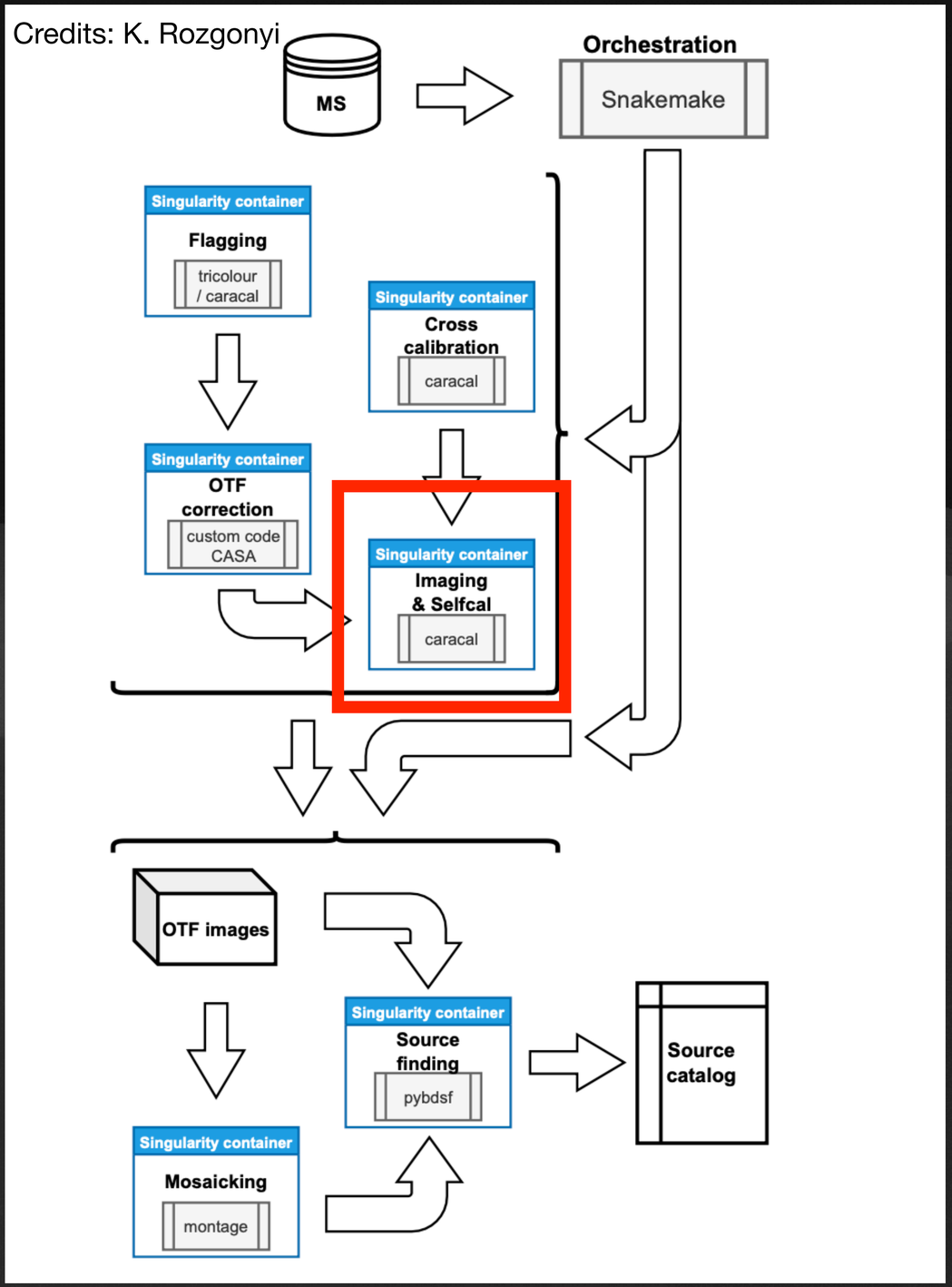
Ad-hoc reduction pipeline

- Flagging, cross- and self-calibration prepare the data (rather standard).
- OTF correction (**NEW**).



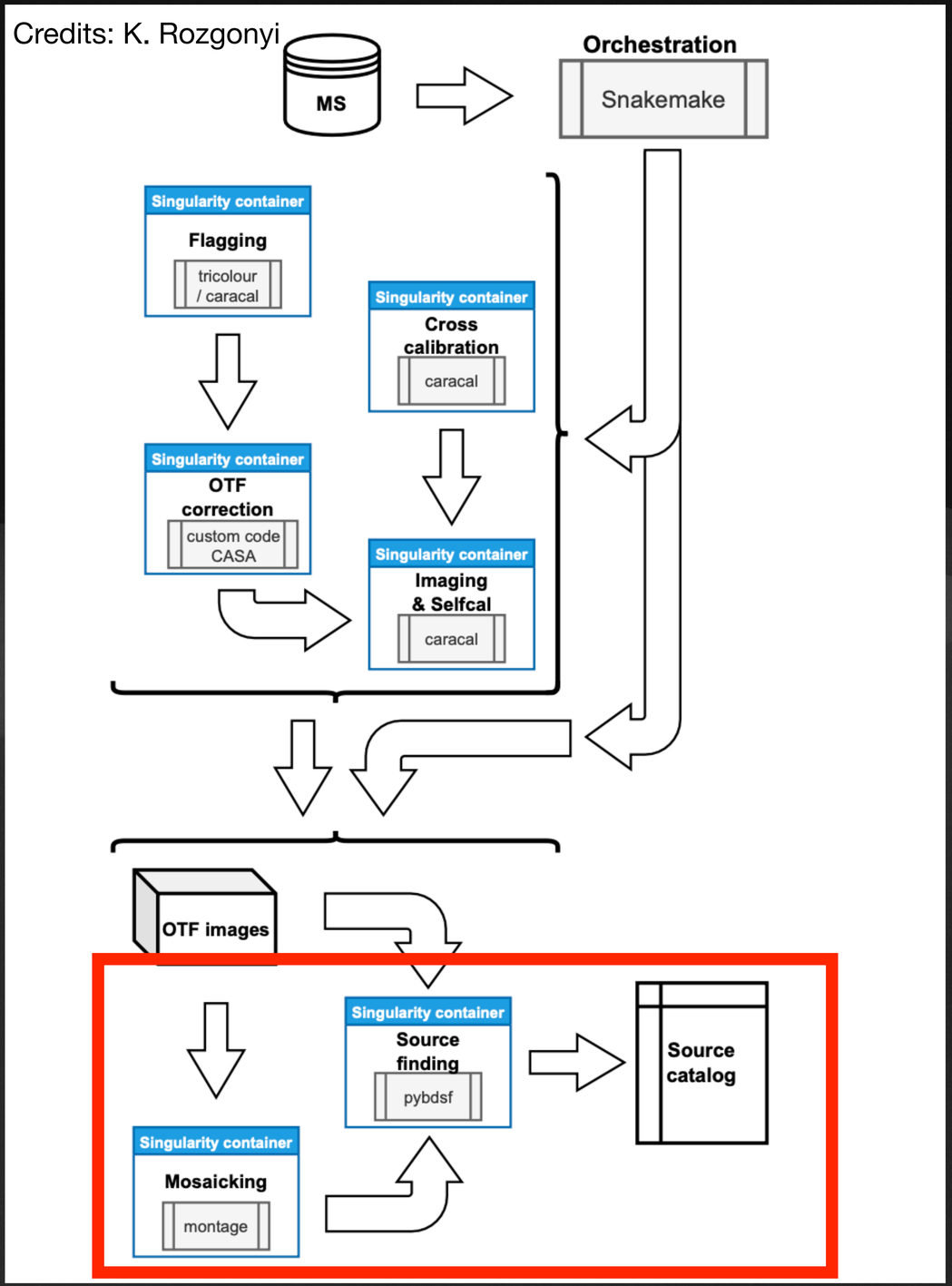
Ad-hoc reduction pipeline

- Flagging, cross- and self-calibration prepare the data (rather standard).
- OTF correction (**NEW**).
- Imaging (1GB input data —> ~5GB output data; 3h using ~30 cores).



Ad-hoc reduction pipeline

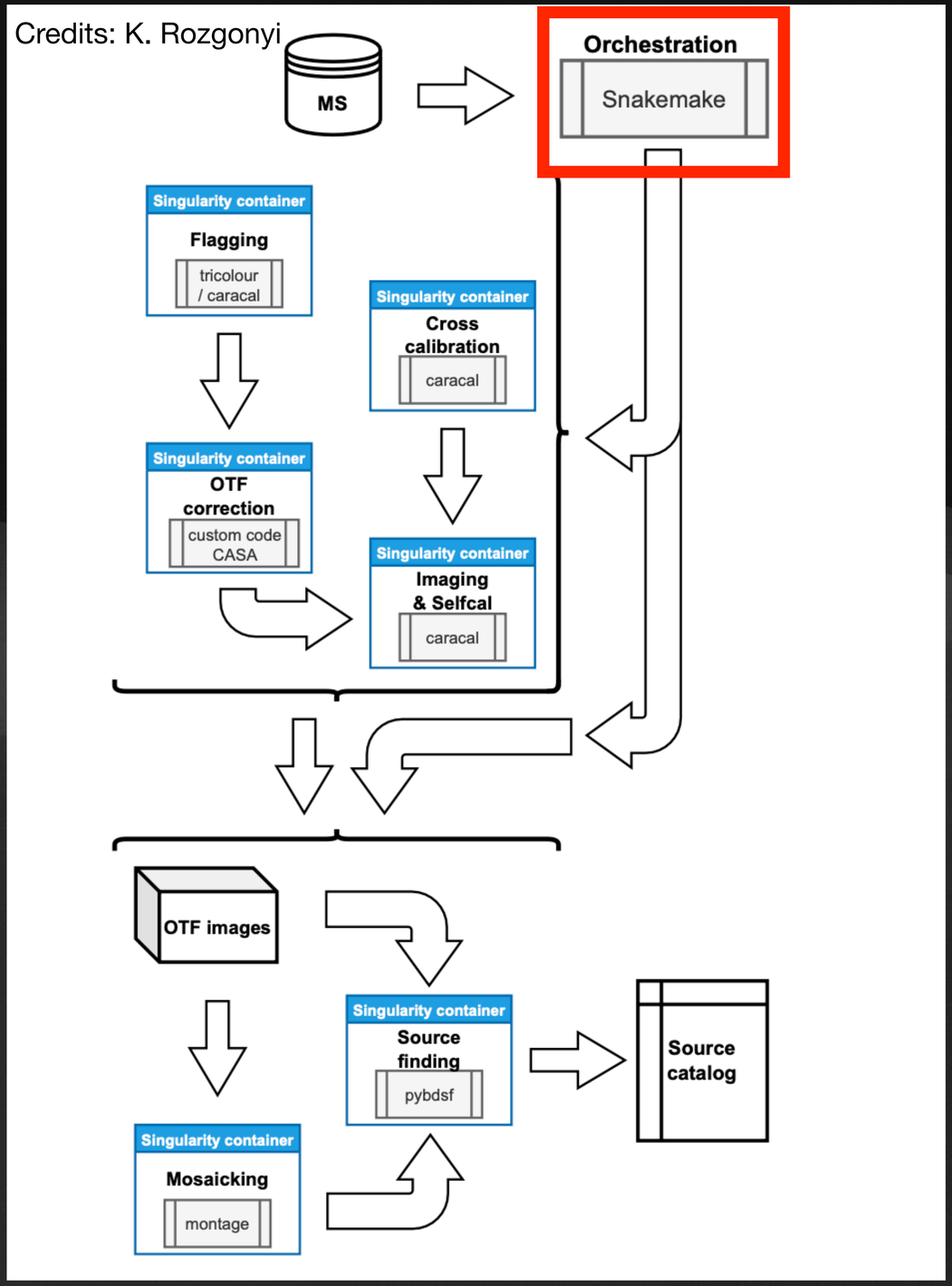
- Flagging, cross- and self-calibration prepare the data (rather standard).
- OTF correction (**NEW**).
- Imaging (1GB input data —> ~5GB output data; 3h using ~30 cores).
- Mosaicking: finding all the tiles (1 deg region of the sky) that cover a given area of scientific interest on the sky.
- Source extraction, catalogue creation (metadata).



Ad-hoc reduction pipeline

- Flagging, cross- and self-calibration prepare the data (rather standard).
- OTF correction (**NEW**).
- Imaging (1GB input data —> ~5GB output data; 3h using ~30 cores).
- Mosaicking: finding all the tiles (1 deg region of the sky) that cover a given area of scientific interest on the sky.
- Source extraction, catalogue creation (metadata).
- Everything managed by orchestration/workflow manager.

Credits: K. Rozgonyi

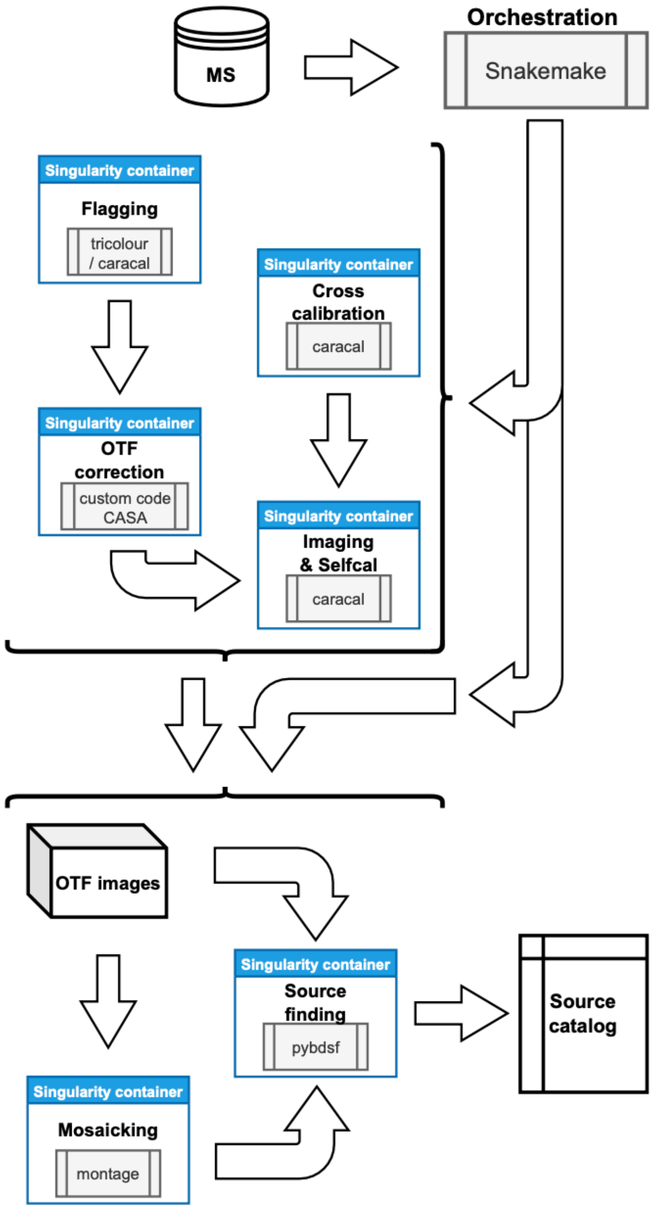


Definition of the use case

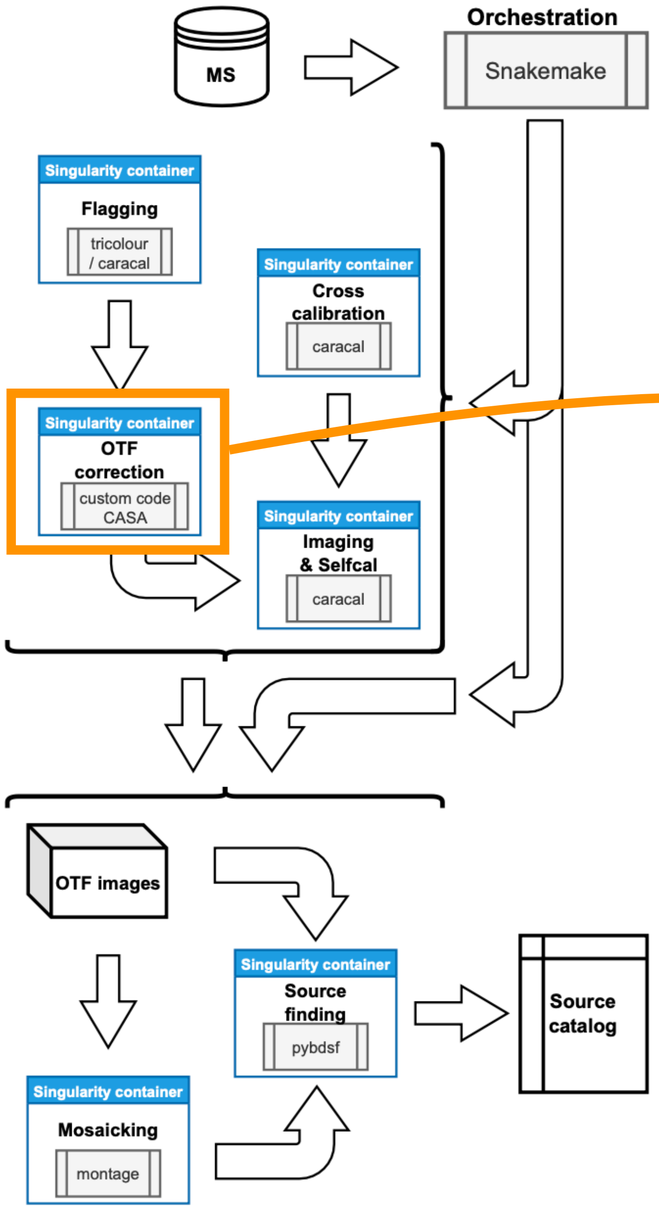
Use case: 2.36

- Implement this data reduction pipeline in PUNCH4NFDI.
- Use of Compute4PUNCH and Storage4PUNCH resources.
- Use of workflow managers.
- Storing metadata for sources.
- Make reduced data available for scientific analyses.

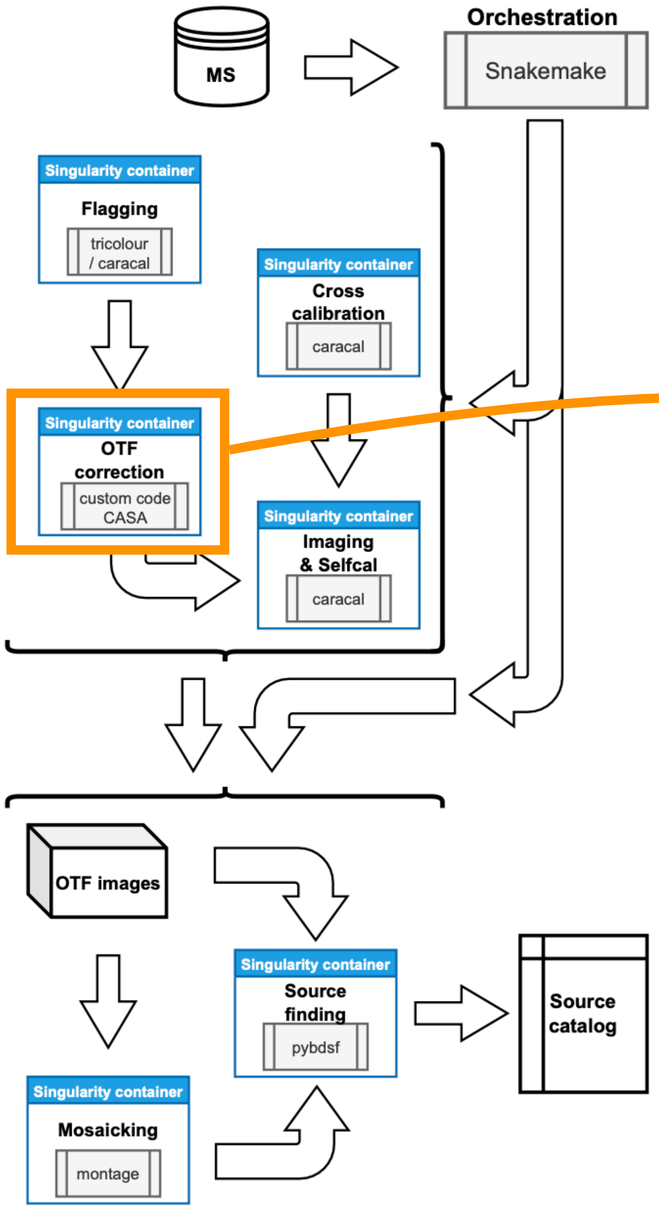
Achievements - OTF correction

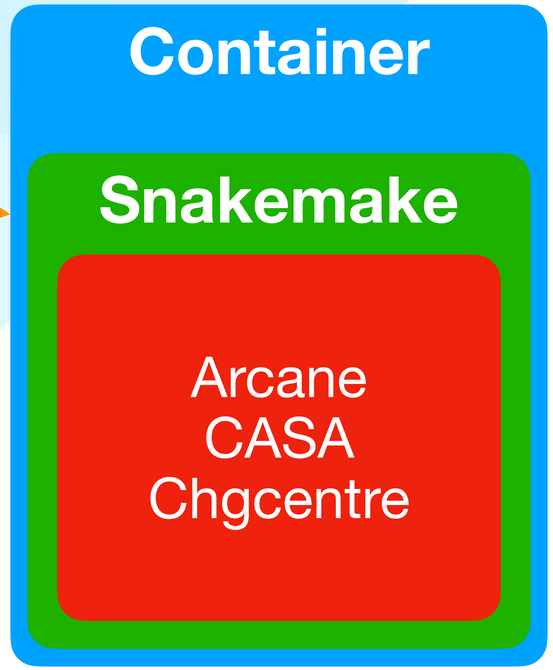


Achievements - OTF correction

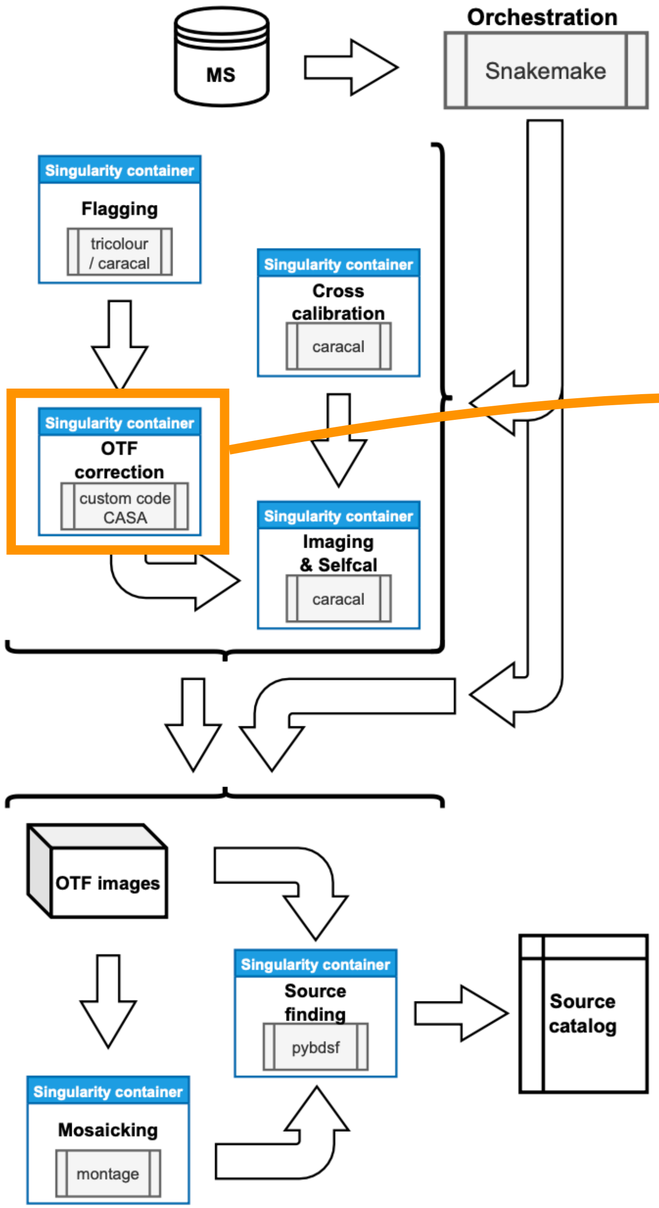


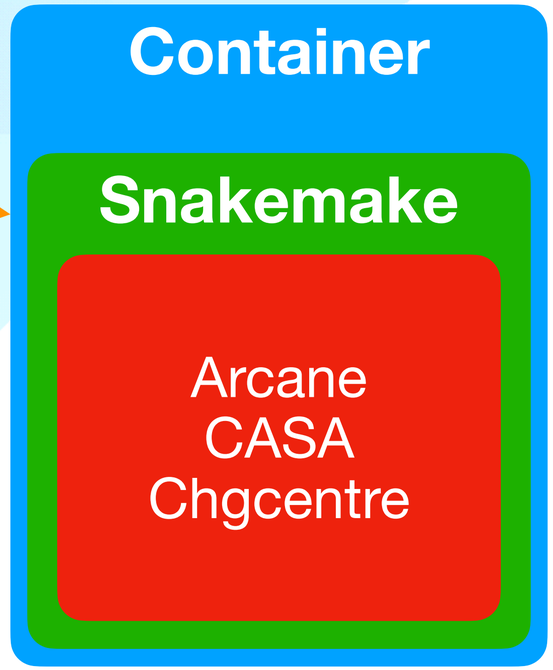
Achievements - OTF correction





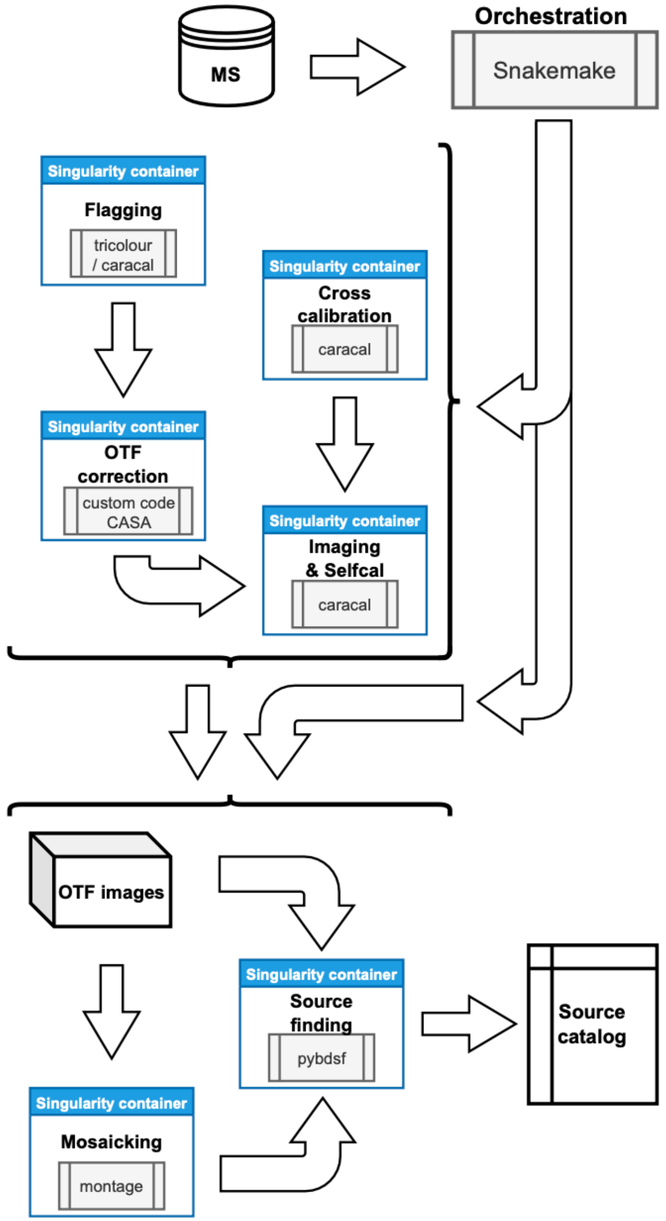
Achievements - OTF correction



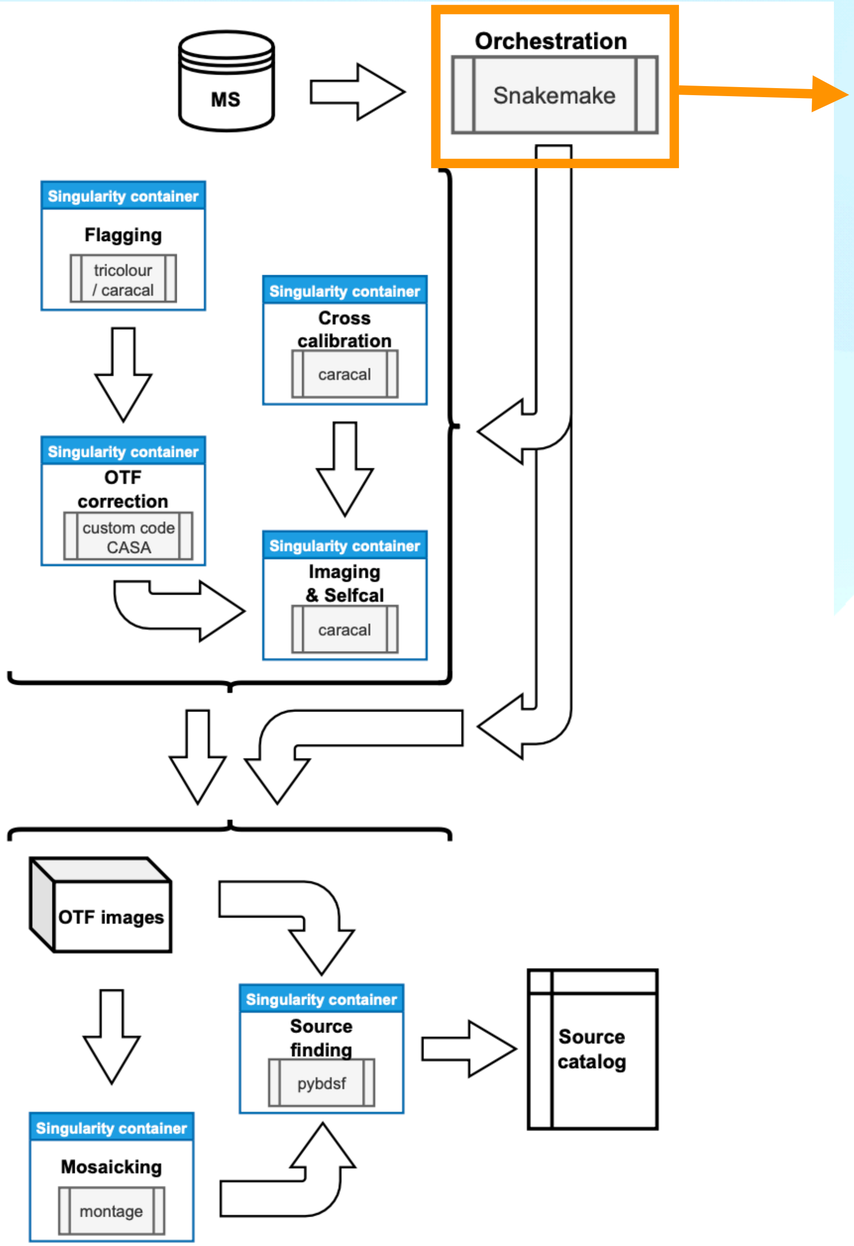


- Container submitted to Compute4PUNCH container stack.
- Test job ran requesting container from list.
- Container available to use on other clusters, **solution open to be scaled up**.

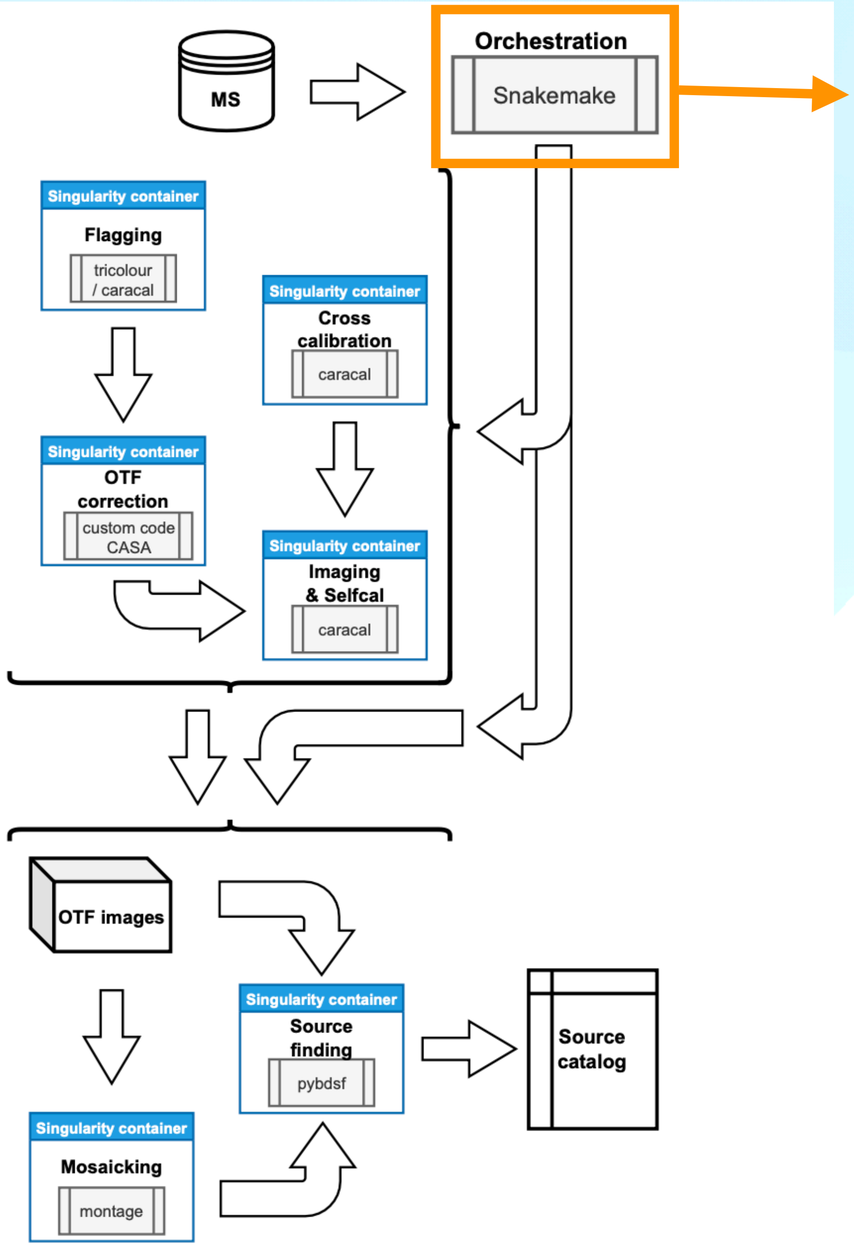
Achievements - orchestration

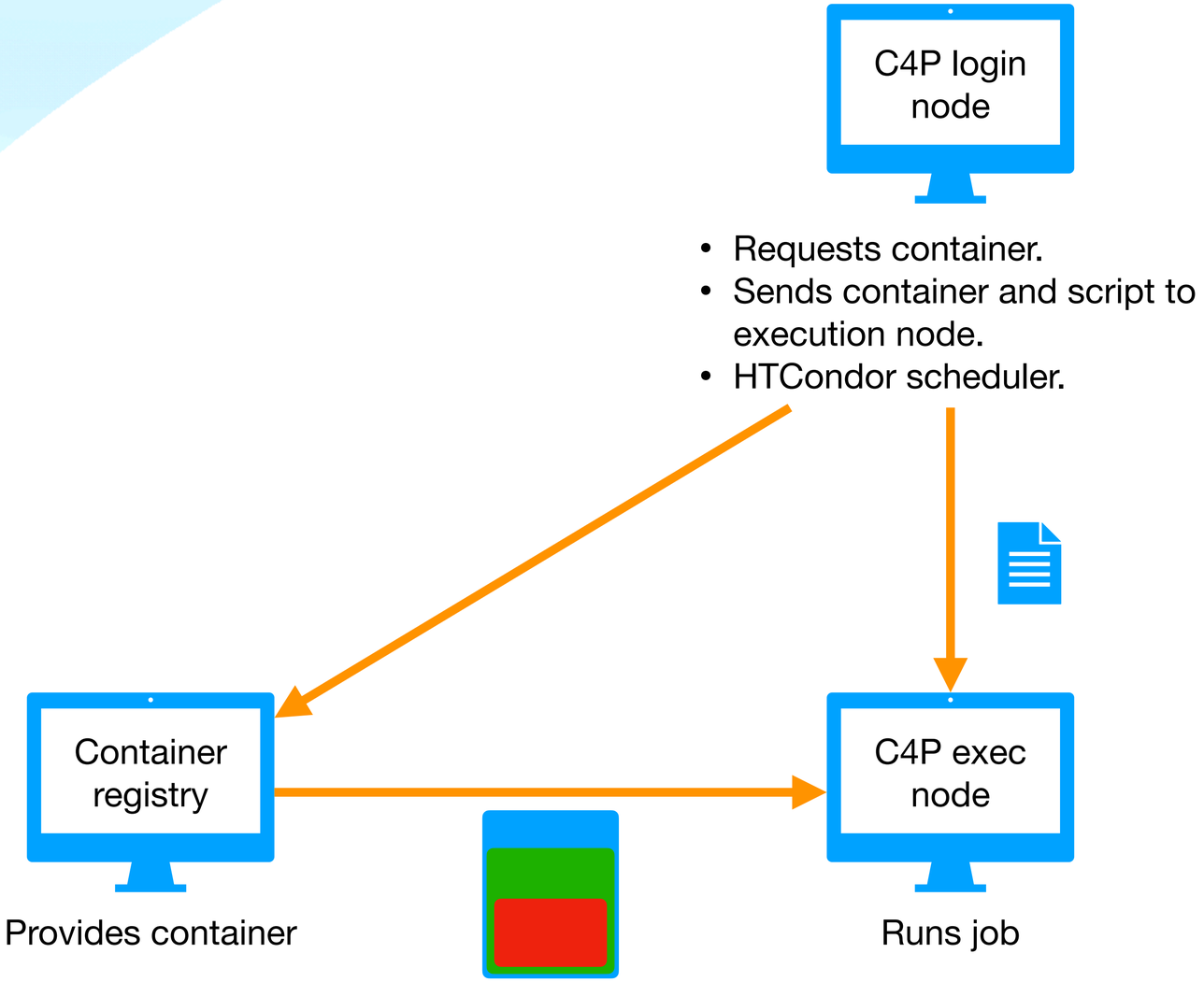


Achievements - orchestration

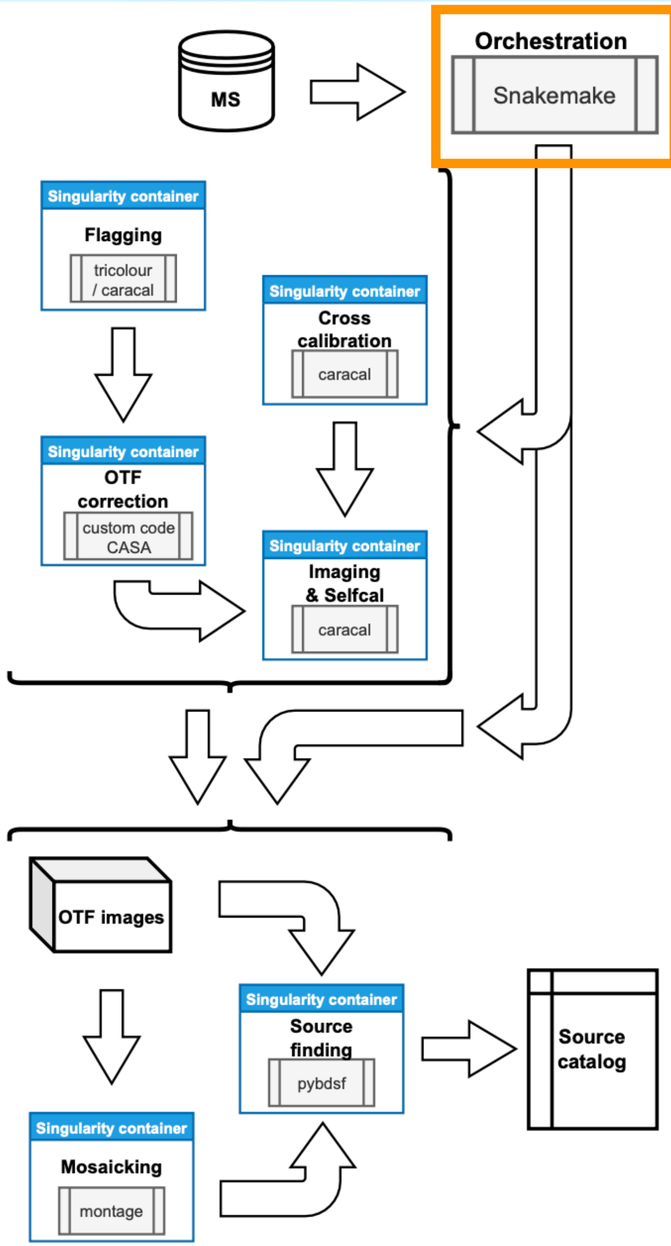


Achievements - orchestration

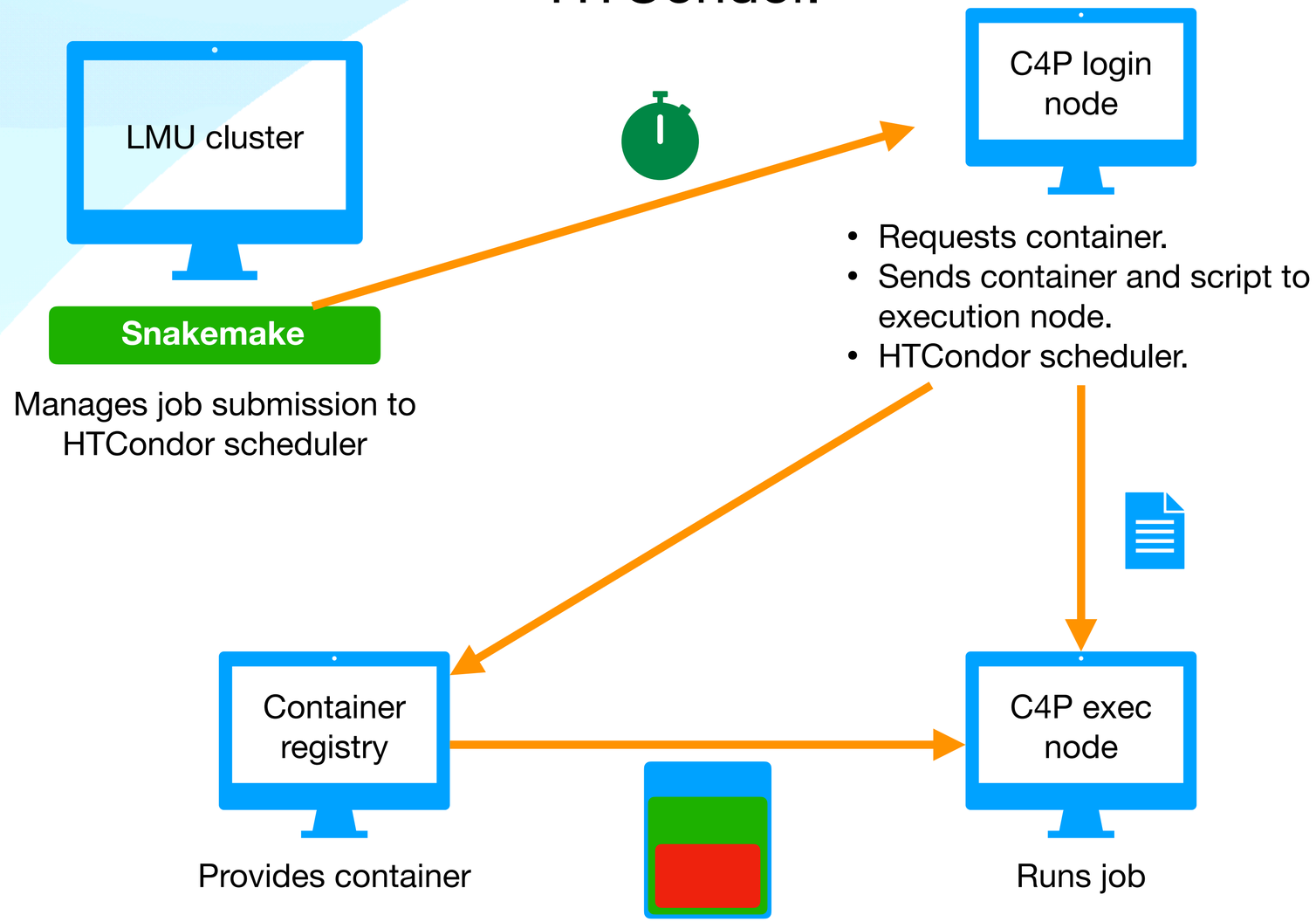




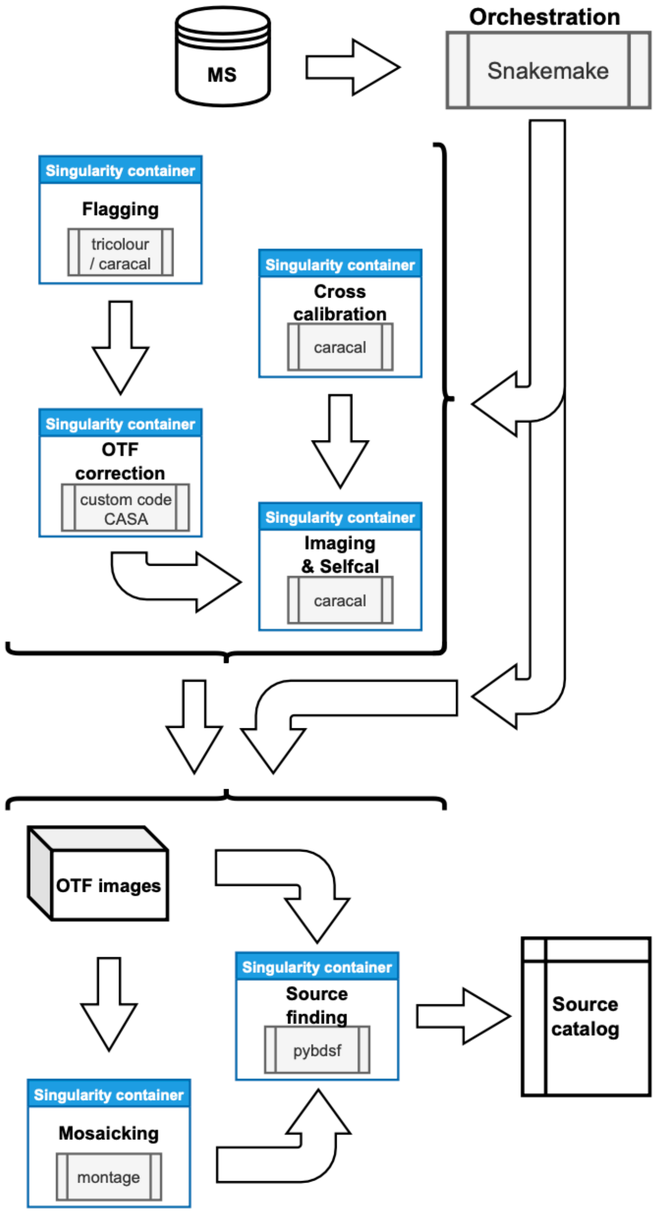
Achievements - orchestration



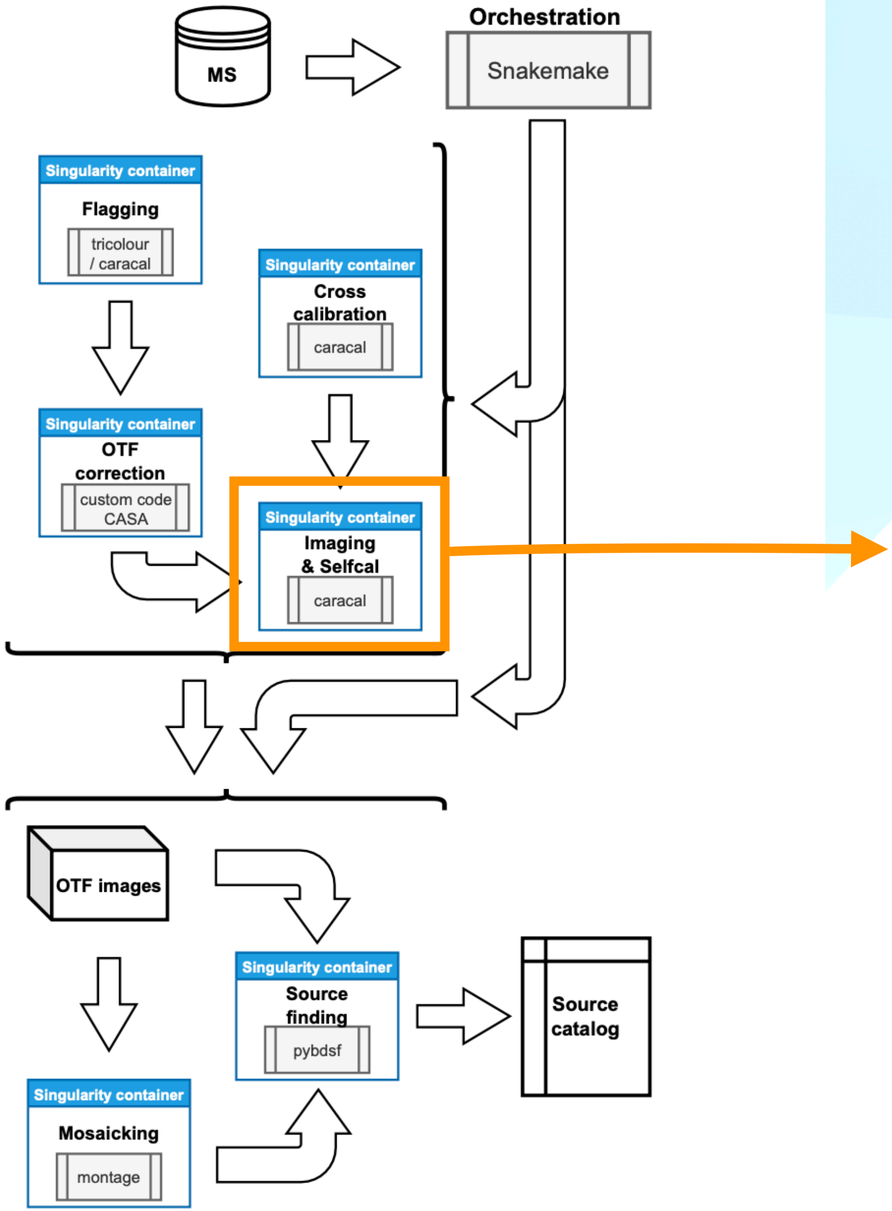
Job submission to Compute4PUNCH done from external cluster via snakemake+ssh command to HTCondor.



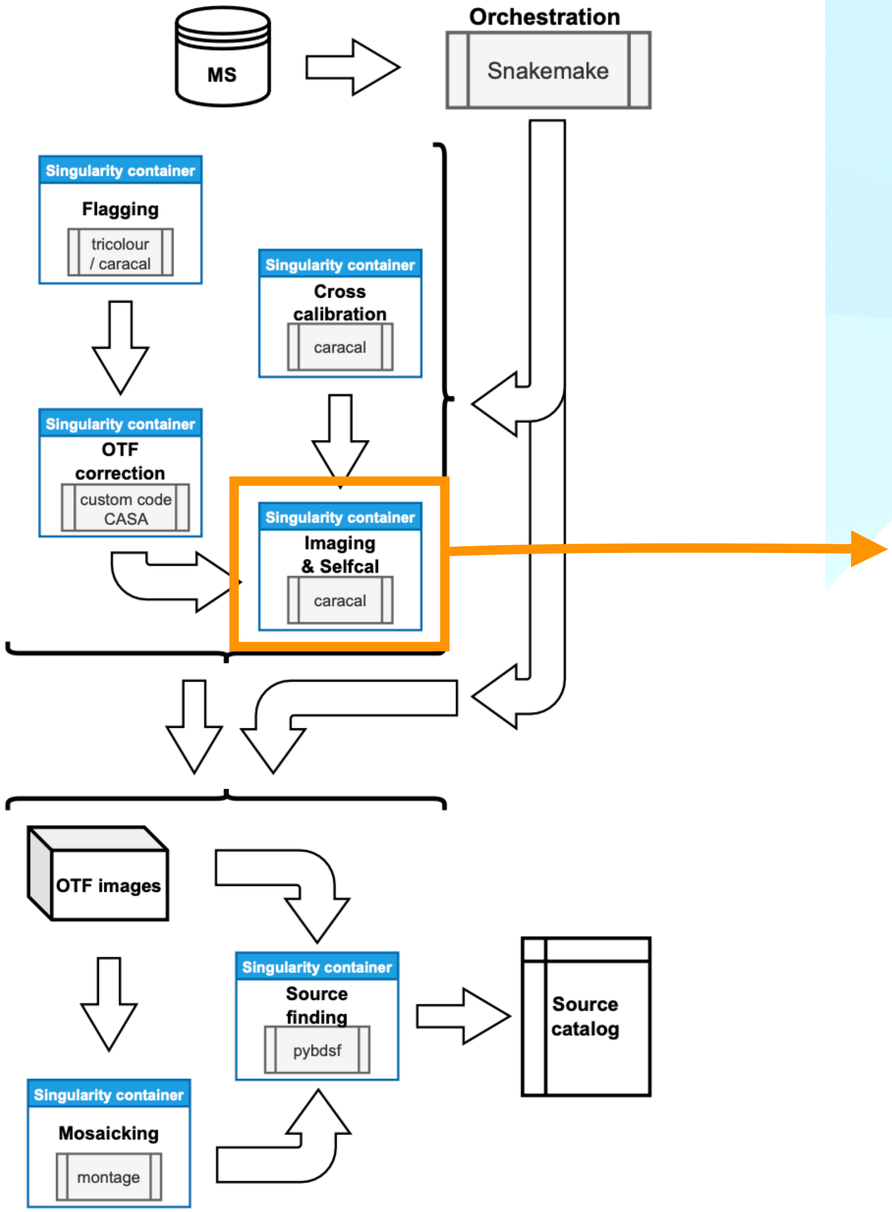
Achievements - Imaging



Achievements - Imaging

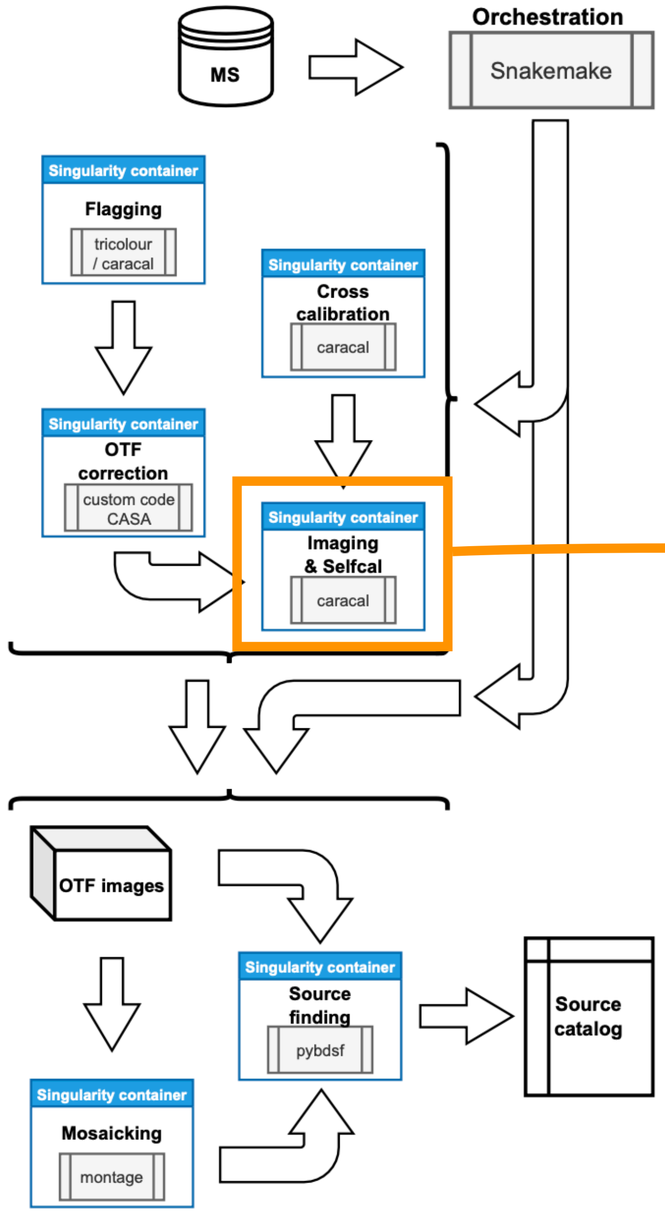


Achievements - Imaging



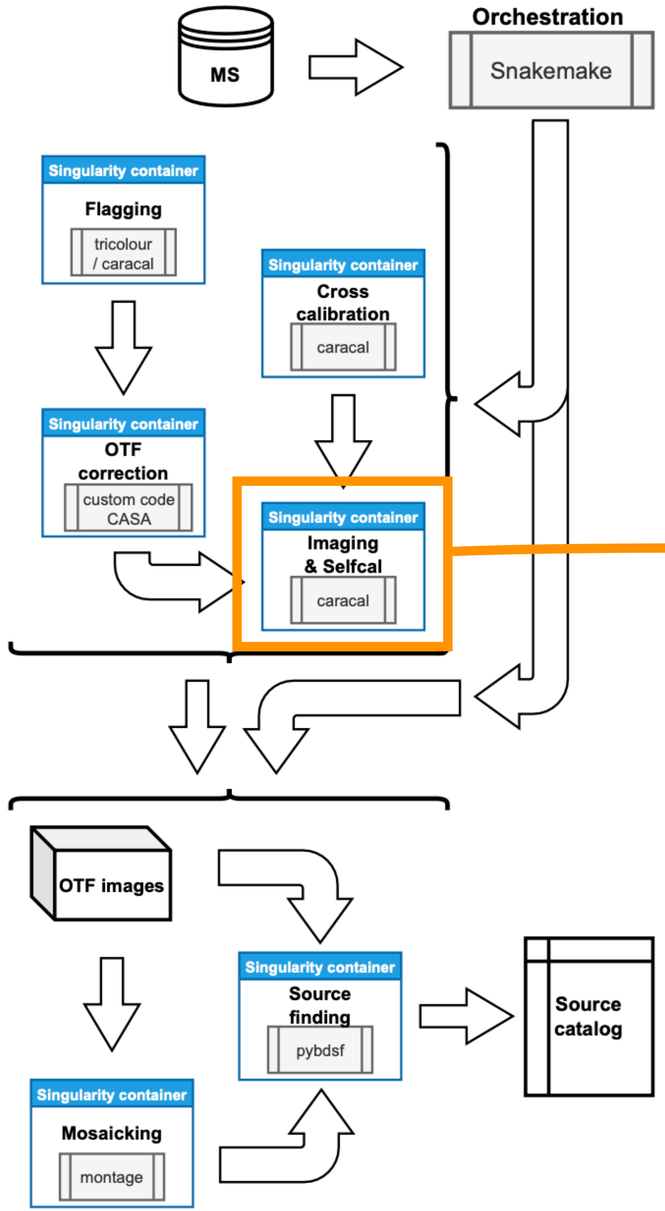
- To be run on Compute4PUNCH caracal needs to be containerized.

Achievements - Imaging



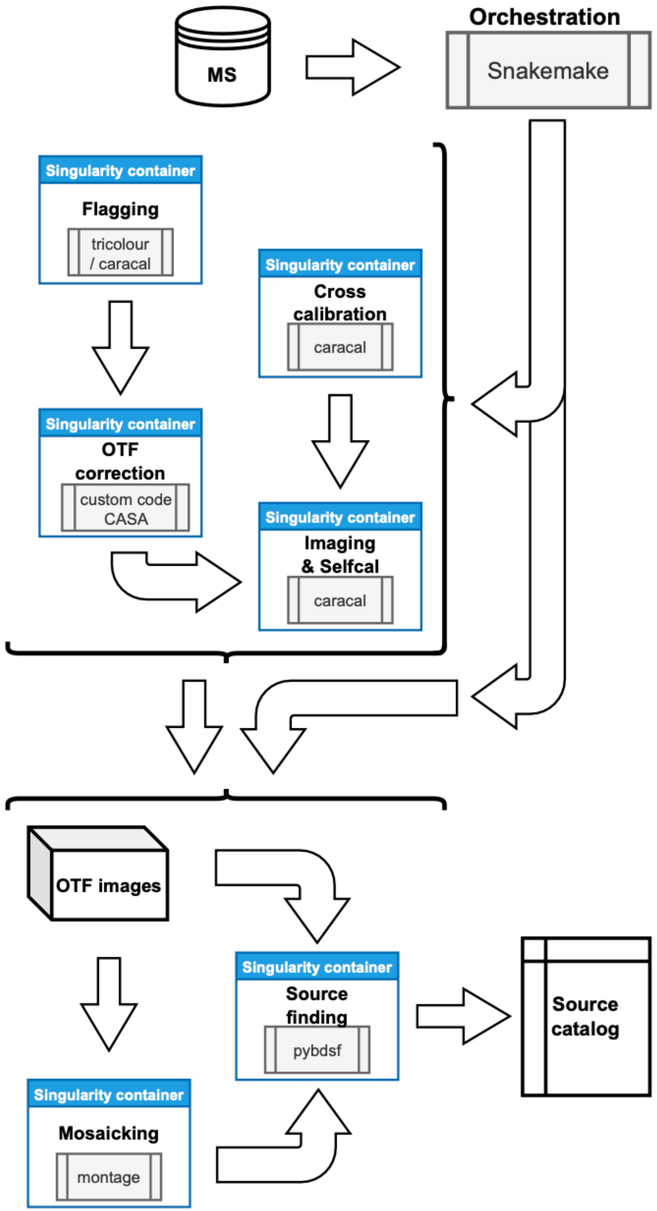
- To be run on Compute4PUNCH caracal needs to be containerized.
- PROBLEM: caracal uses stimela which provides containerized software for specific tasks.

Achievements - Imaging

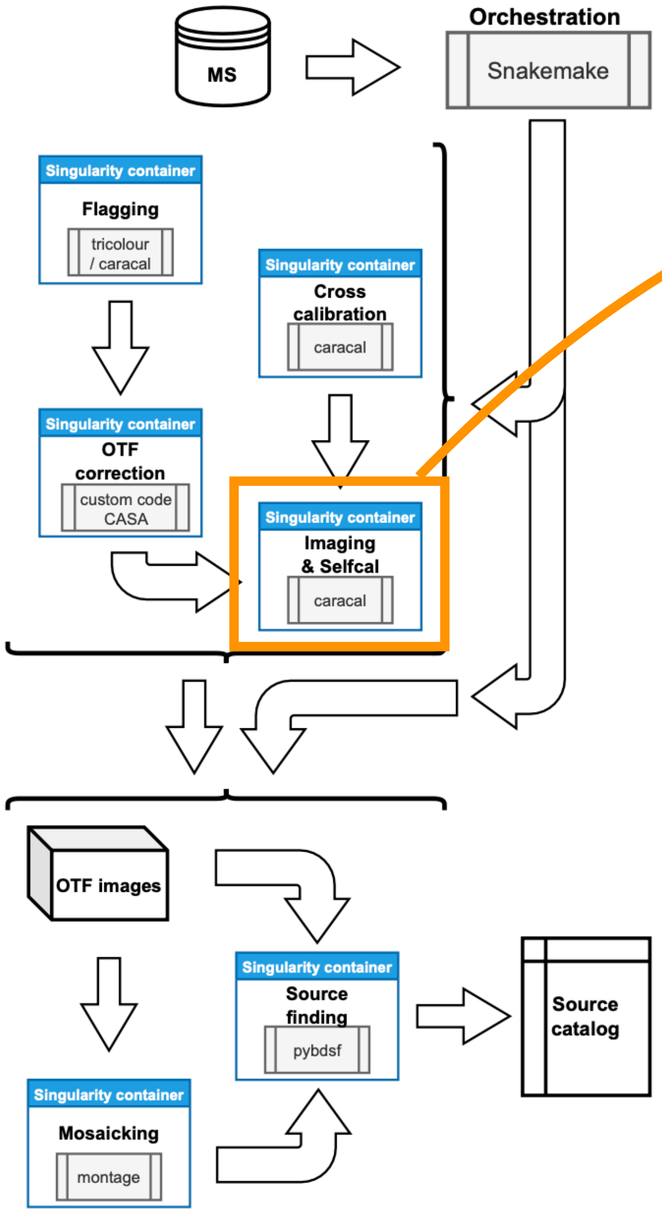


- To be run on Compute4PUNCH caracal needs to be containerized.
- PROBLEM: caracal uses stimela which provides containerized software for specific tasks.
- Need of a “container in container” solution.

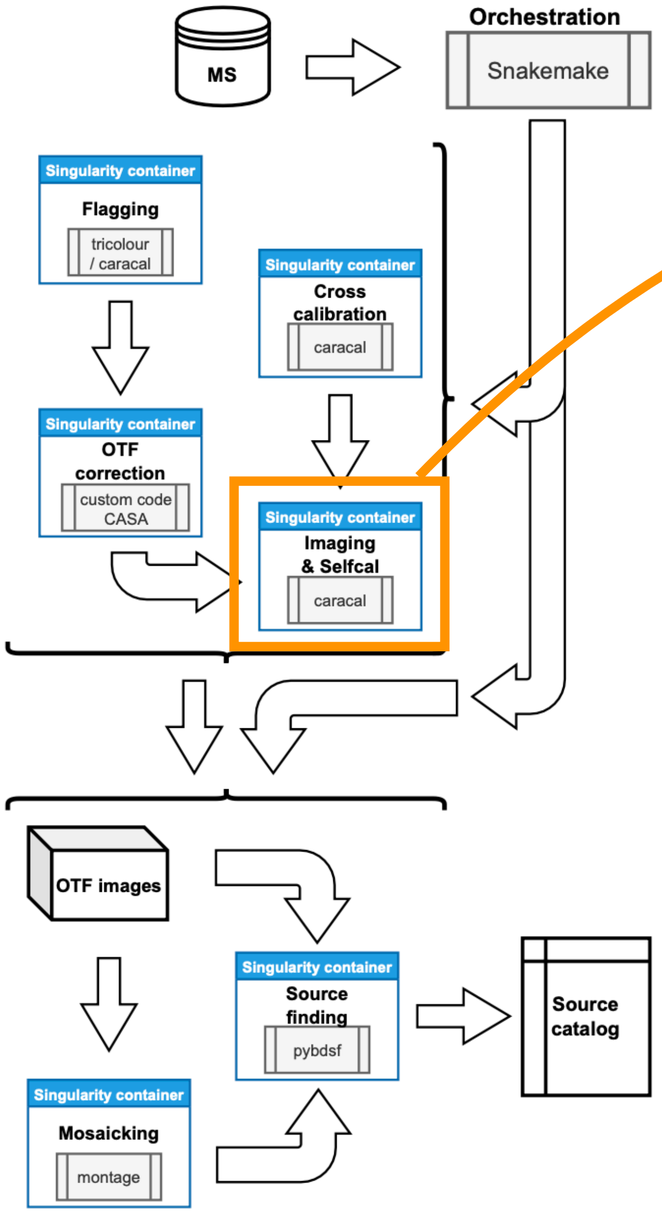
Achievements - Imaging



Achievements - Imaging

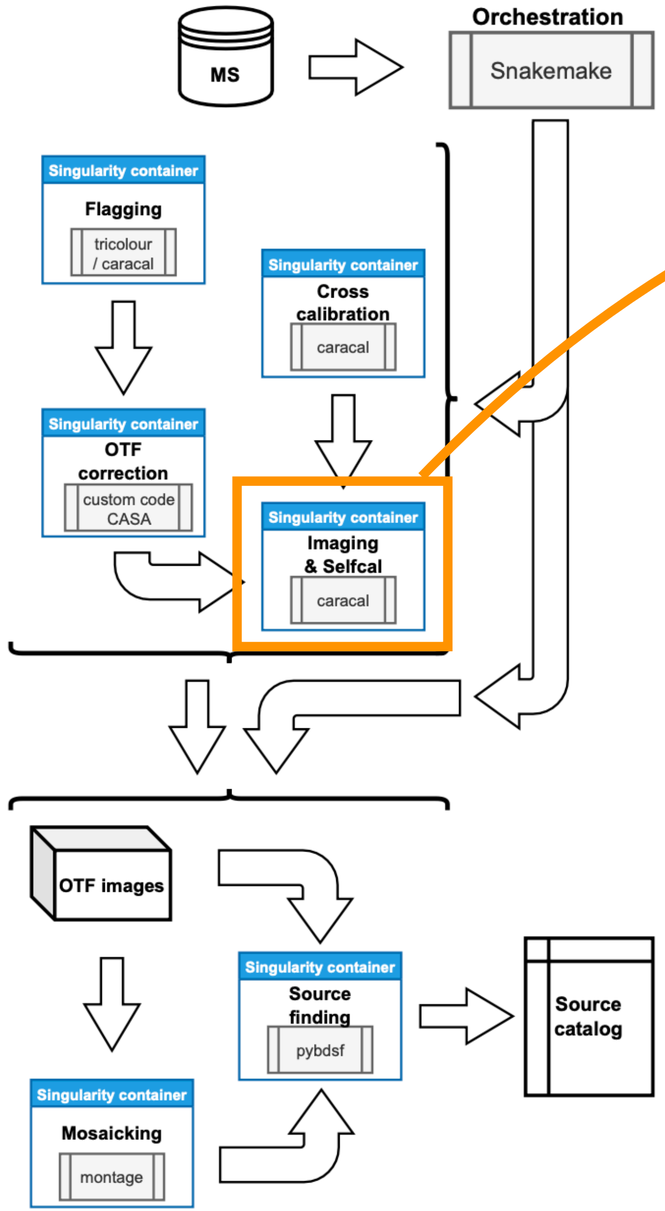


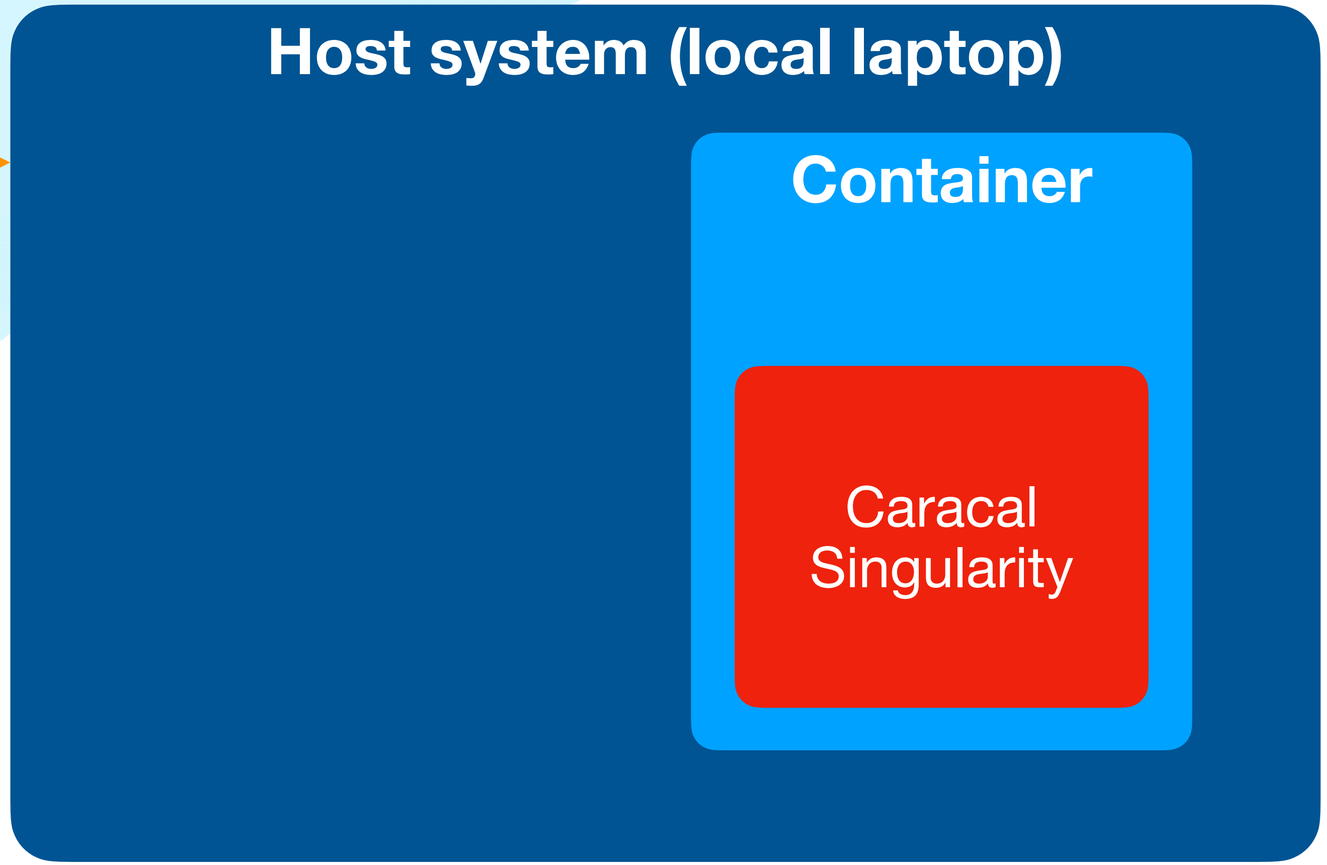
Achievements - Imaging



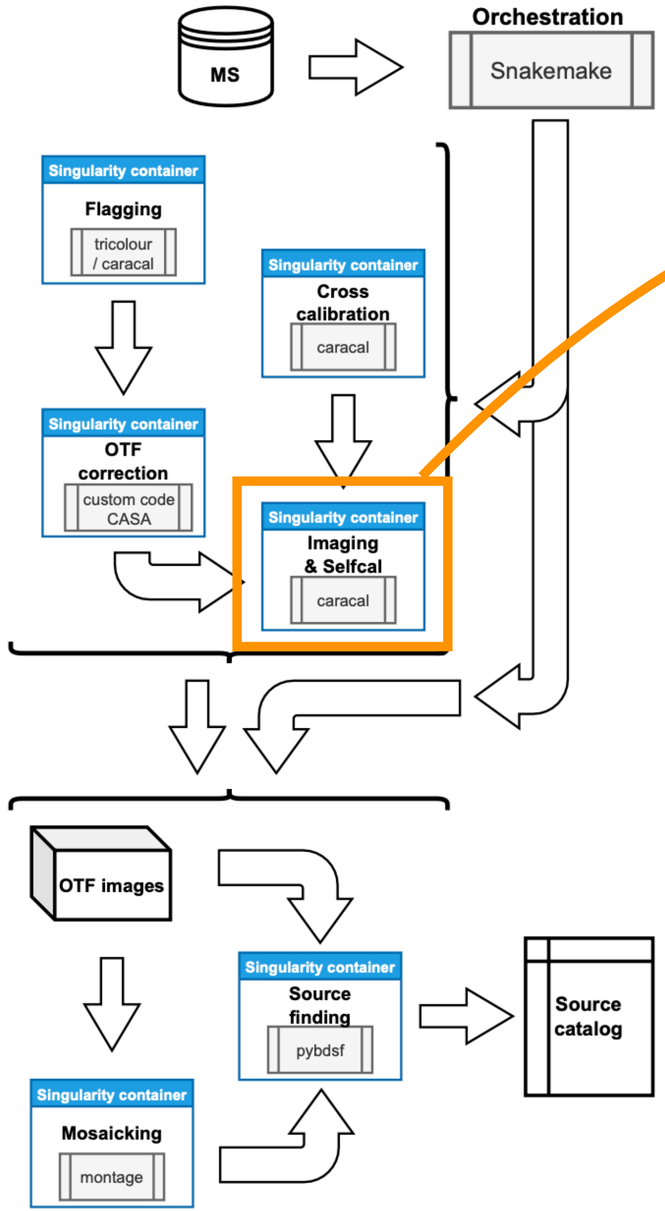
Host system (local laptop)

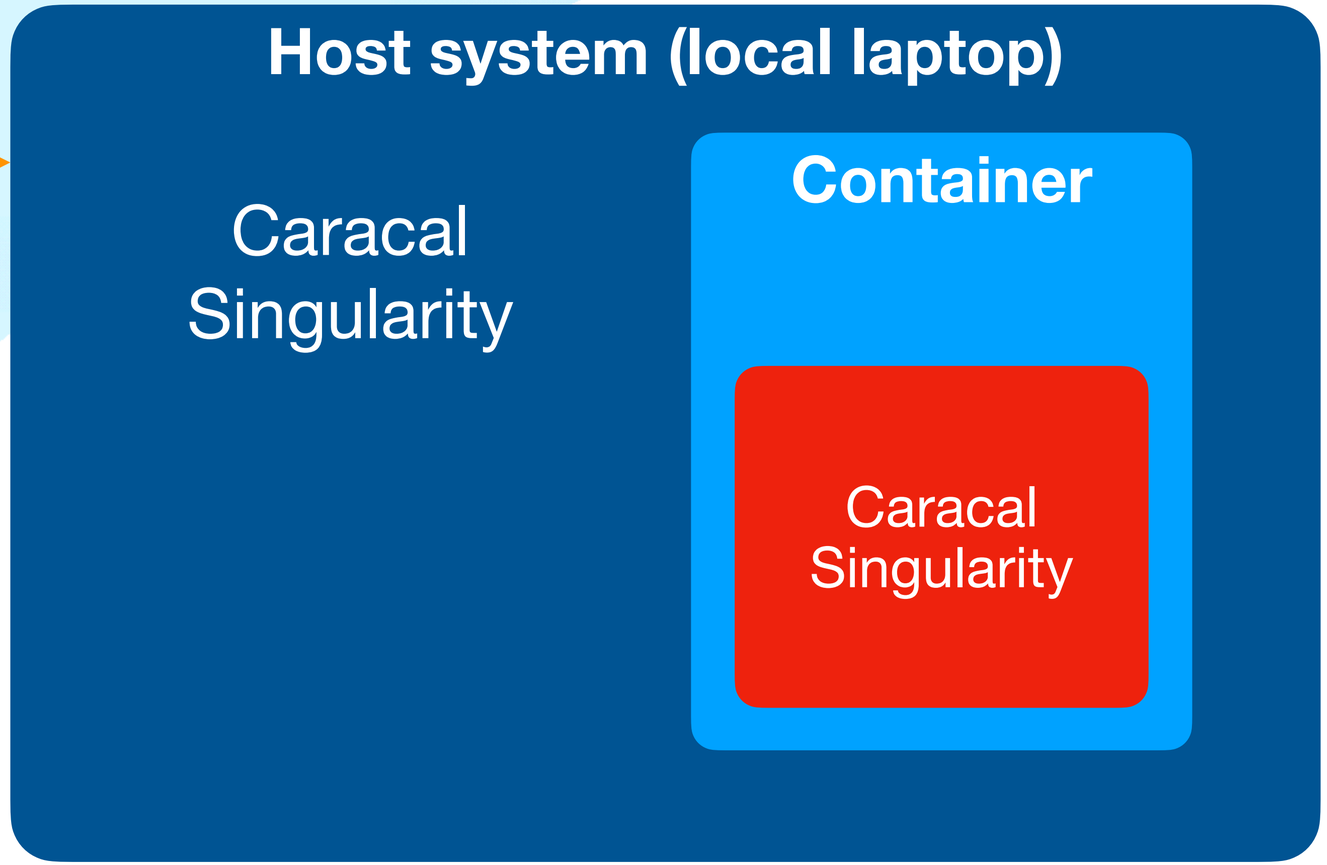
Achievements - Imaging



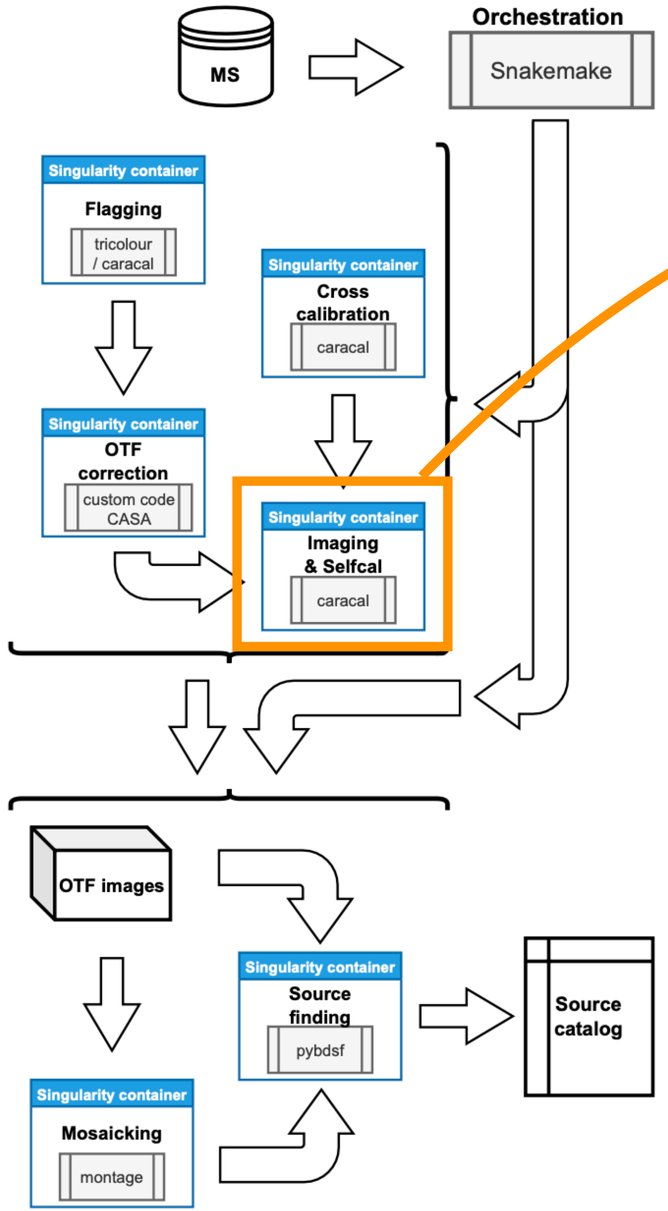


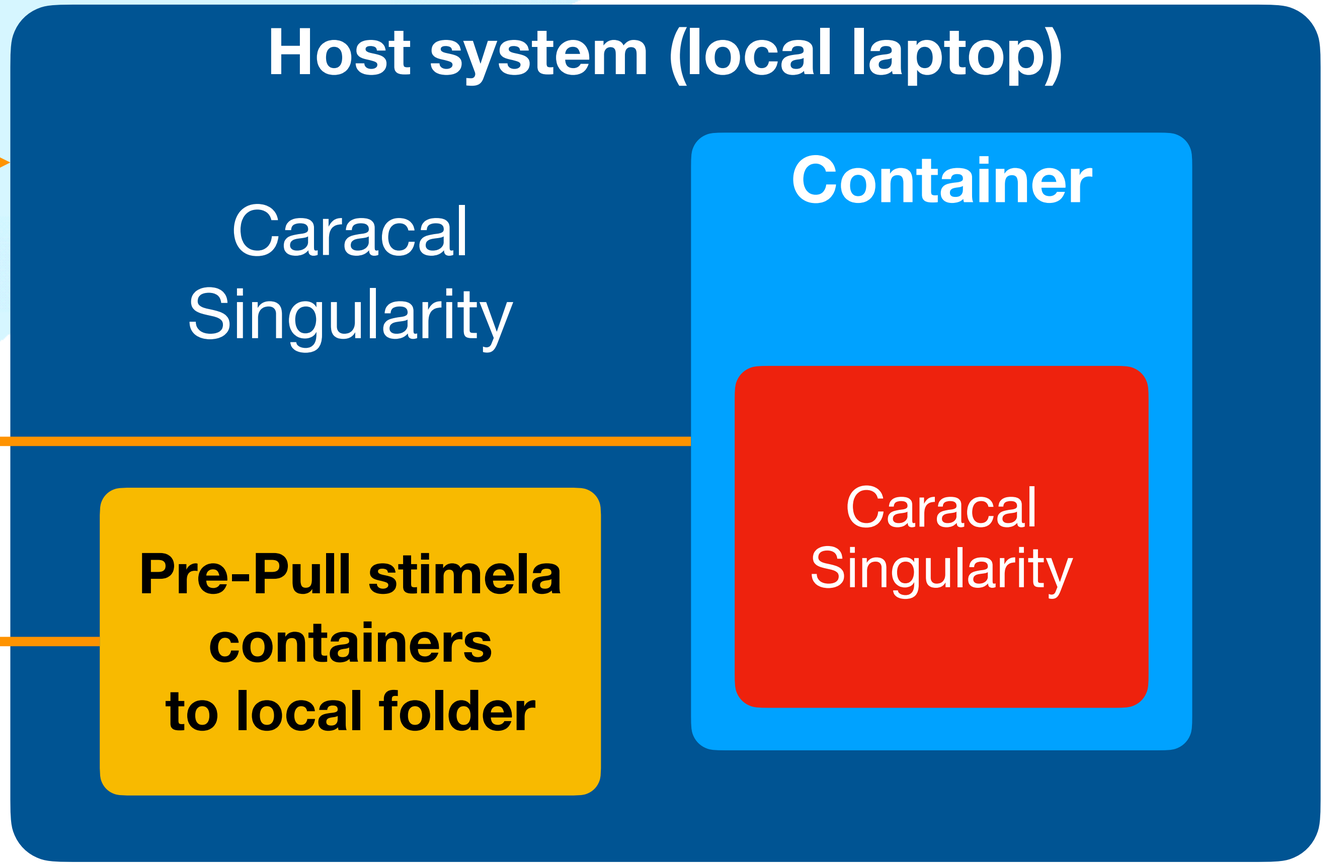
Achievements - Imaging





Achievements - Imaging





- Provide to Compute4PUNCH.
- Caracal in container uses containerized singularity to access pre-pulled cabs.

Development plans

- Scale up OTF step on Compute4PUNCH.
- Implement imaging step on Compute4PUNCH.
- Use Storage4PUNCH resources to store data in connection to Compute4PUNCH.
- Set up orchestration: entire workflow of the pipeline managed by workflow manager.

We are facing challenges which require synergy between task areas to be solved.

Development plans

- OTF step: from test jobs to large data volumes.
- Imaging: solve “container in container” issue for caracal.

Likely to be solved with support from TA2 and inclusion of Munich cluster in C4P.

Need access to a **shared file system** or **efficient transfer of data from Storage4PUNCH** to handle ~350GB sized files.

Need to solve a problem with missing user information on C4P: user on C4P worker nodes is “nobody”, but caracal requires user data.

Development plans

- Use Storage4PUNCH resources to store data to be **input or output from Compute4PUNCH**.

Streaming/mounting from Storage4PUNCH directly into Compute4PUNCH worker nodes?

Likely to be solved with support from TA2 and inclusion of Munich cluster in C4P.

Development plans

- Set up orchestration: entire workflow of the pipeline managed by workflow manager.

Use REANA instance
to manage a
snakemake workflow.

Support from TA4.

Conclusions

Interesting use case that will lead to scientific analyses, test of a lot of PUNCH infrastructure, synergy of PUNCH task areas.

- TA2: storage and compute.
- TA3: support to pipeline development & management.
- TA4: workflow implementation & management.

Requirements from other task areas

- Support resolving nested container problems (associated with user “nobody”) on Compute4PUNCH.
- Support using Storage4PUNCH to store significant volumes of data to increase efficiency of Compute4PUNCH imaging and OTF jobs.
- Support in getting remote submission to Compute4PUNCH working.
- Support in implementing REANA management of the job workflow.