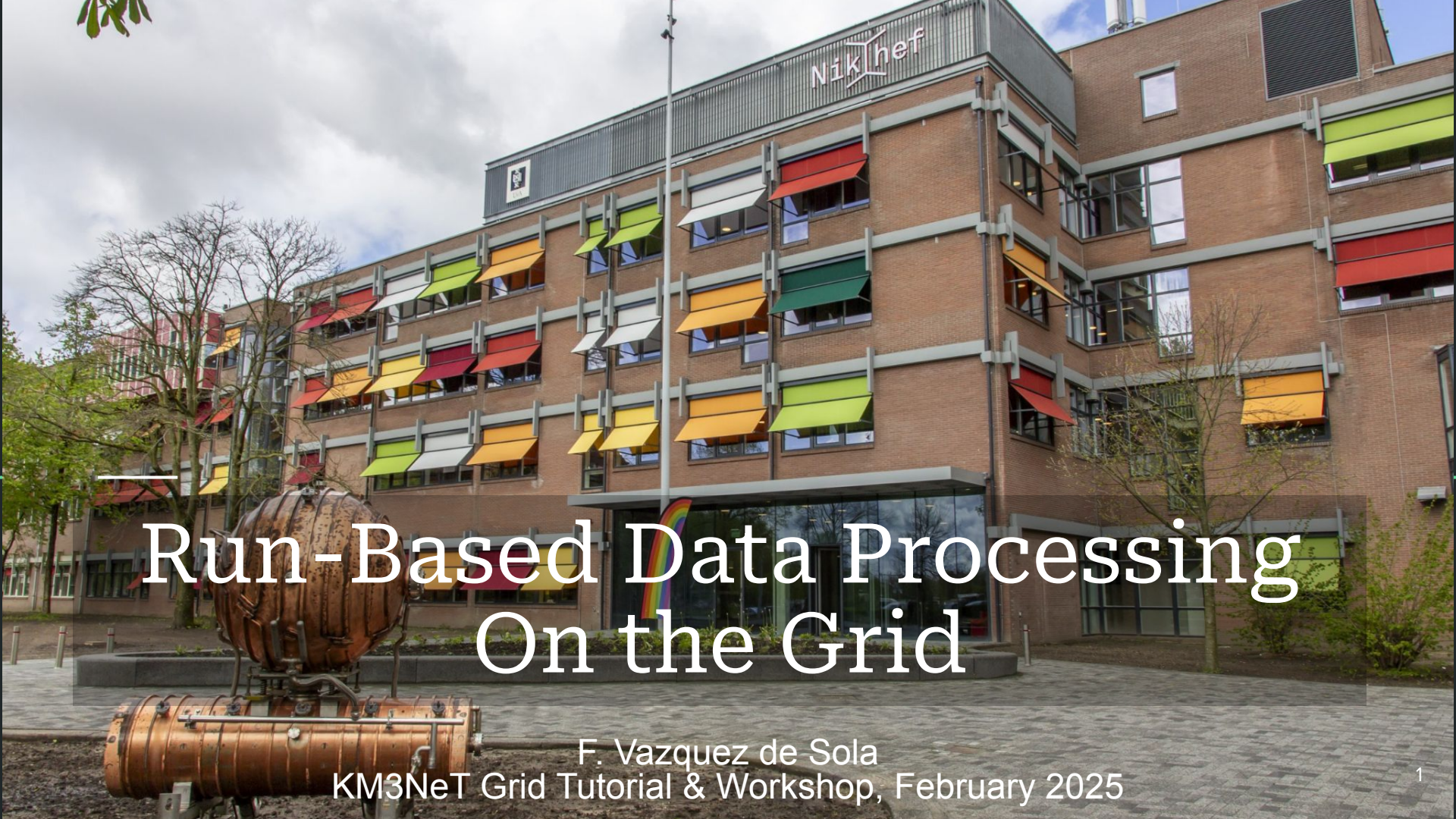
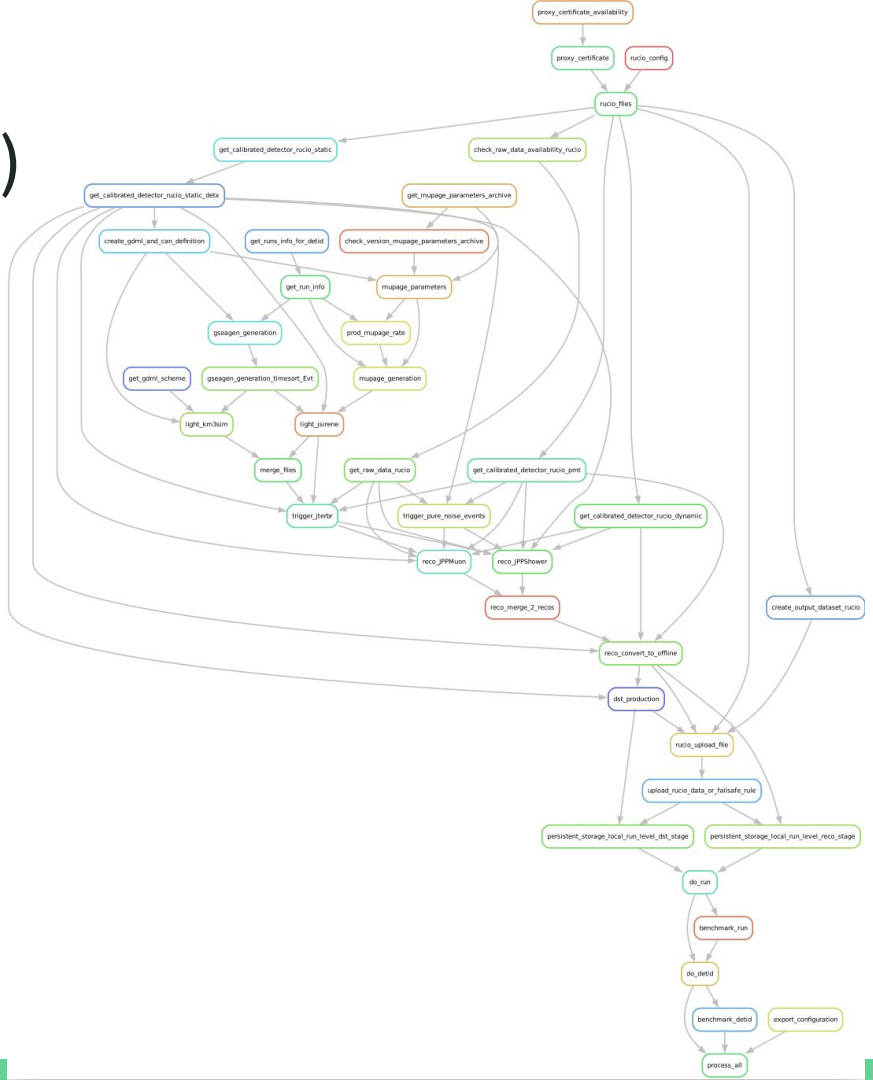


Run-Based Data Processing On the Grid

F. Vazquez de Sola

KM3NeT Grid Tutorial & Workshop, February 2025

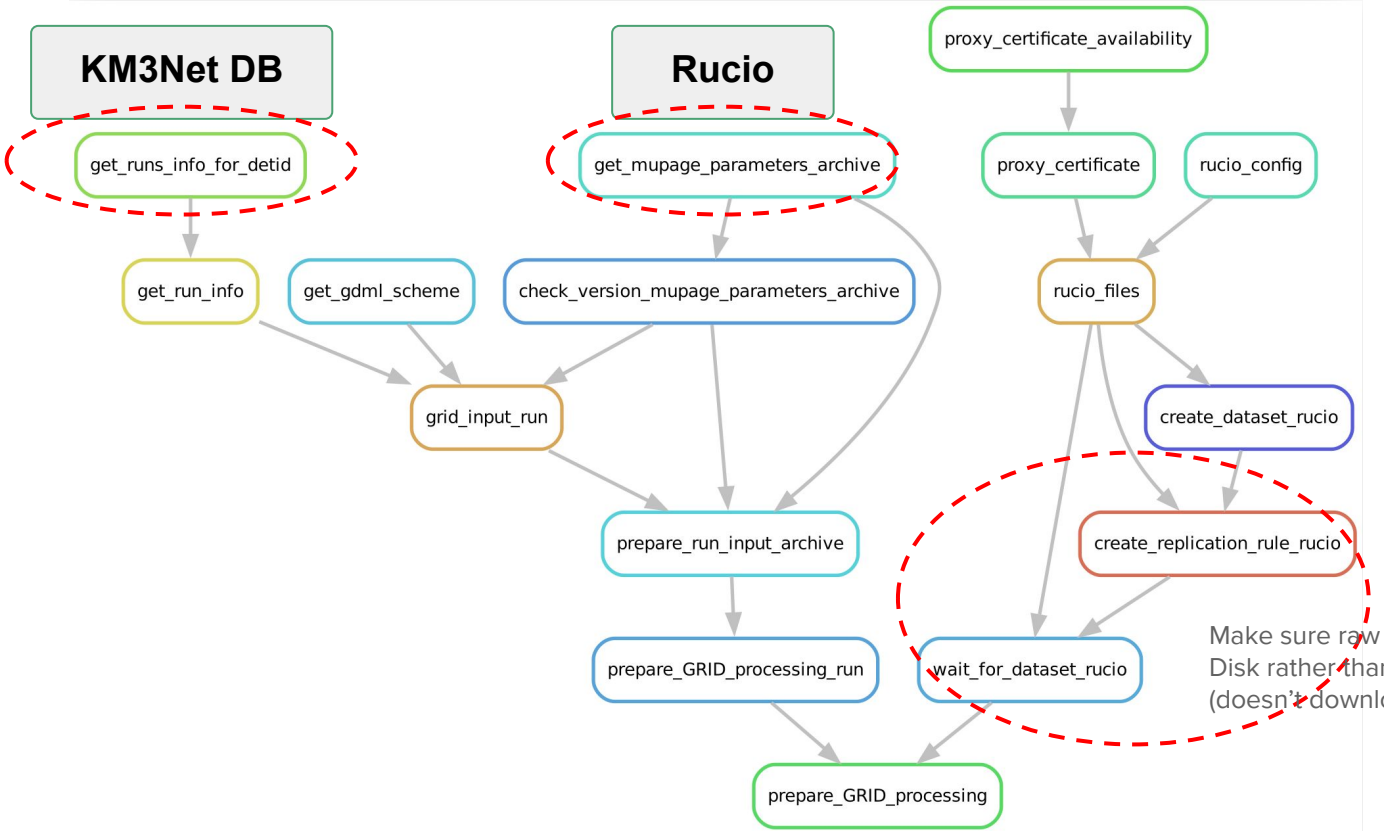
Processing (rulegraph)



Gridification of run-based data processing

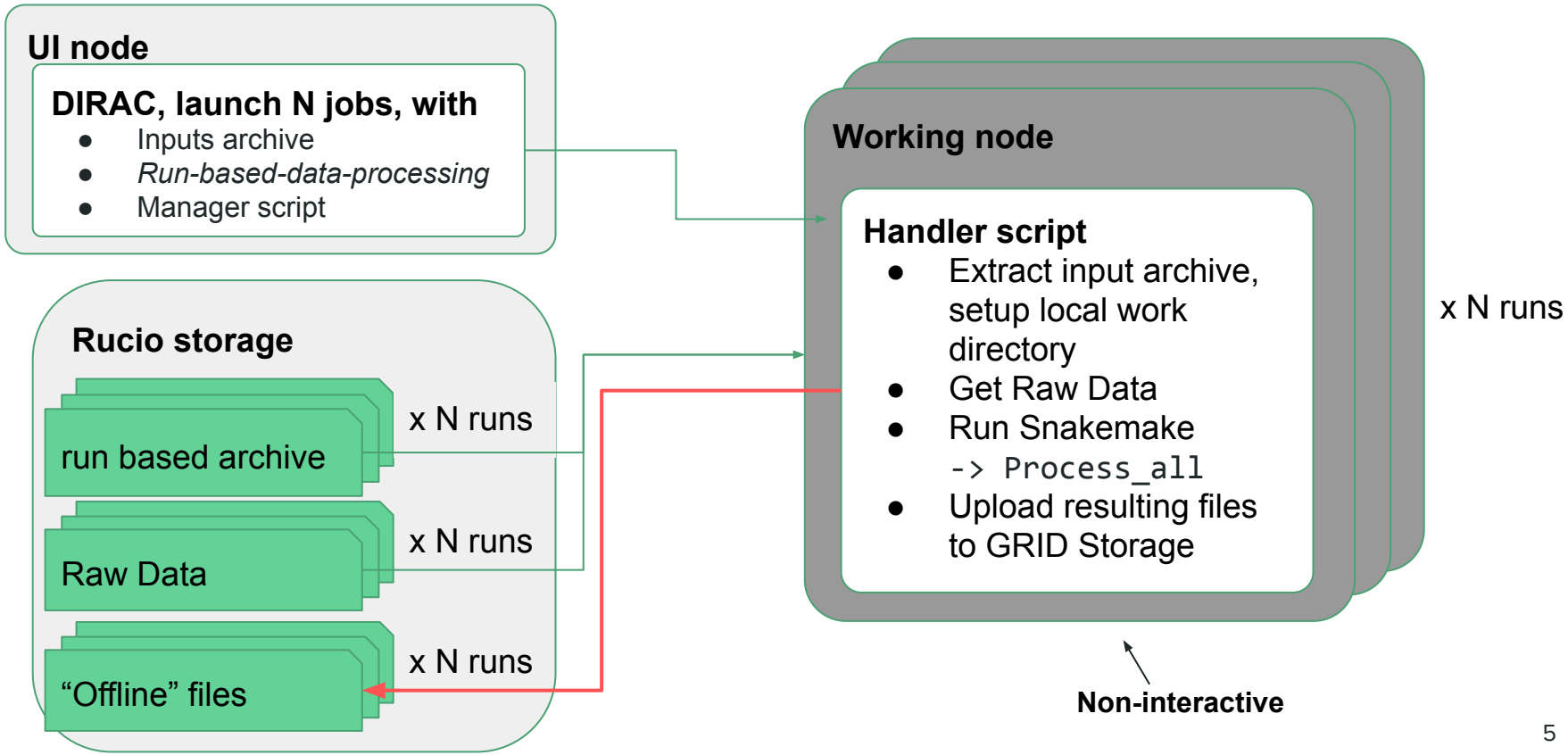
- Executables on CVMFS, in a container
  - /cvmfs/km3net.egi.eu/sw-v0.3/anaconda3/etc/profile.d/conda_init_km3net
  - except run-based data processing code itself, which gets sent in input sandbox for now
- Cannot run full snakemake process on the grid
  - Requires authentication with km3net database, and to wait for files from tape
- Conversely, *splitting* processing options are limited
  - By default, time of job submission by DIRAC is random (i.e. no “scheduling” with respect to each other)
  - Snakemake does not scale to track thousands of jobs
- Idea: Prepare files locally, then fully process every run separately in WNs
- Save output files at the end of job, with option to store progress logs in grid storage regularly during processing for debugging

Pre-submission preparation (rulegraph)

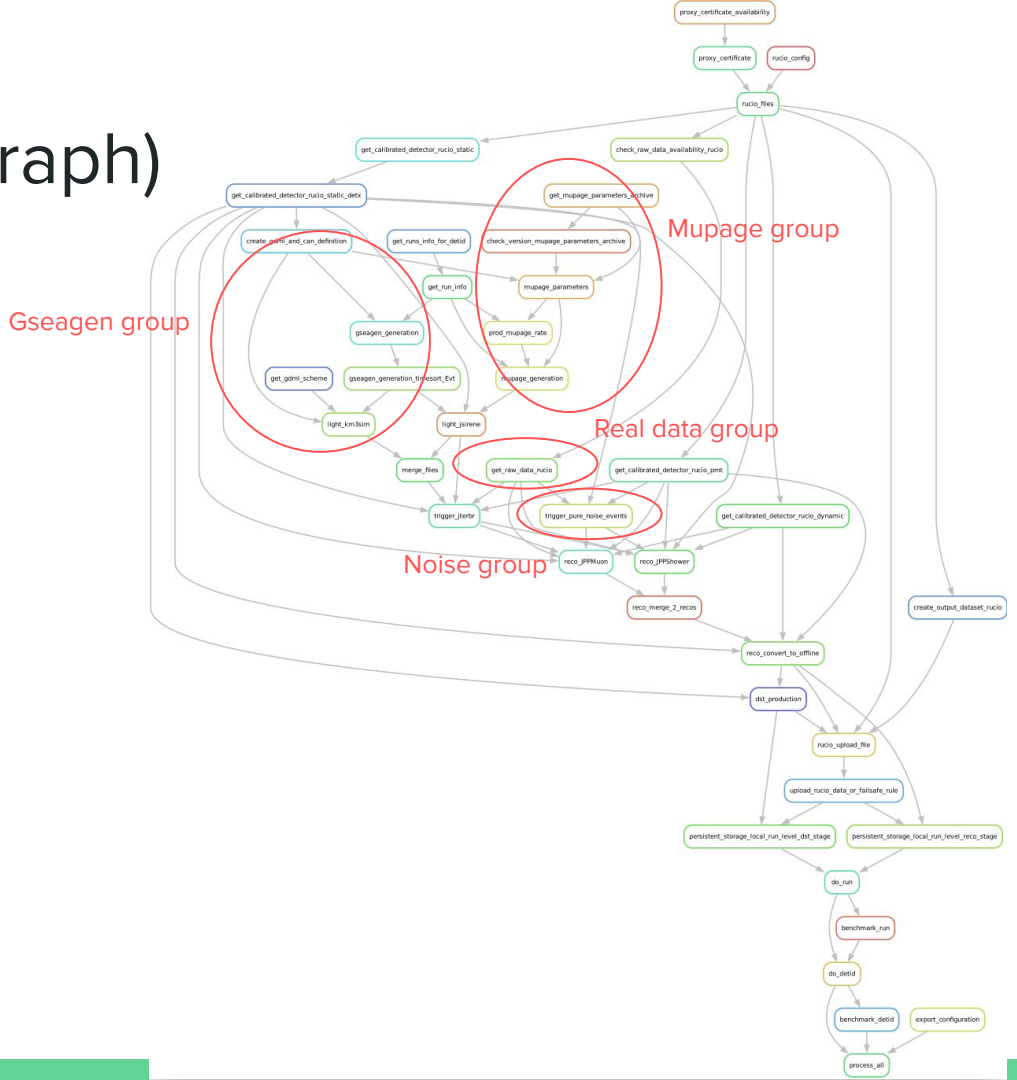


Make sure raw files are in Disk rather than Tape (doesn't download anything)

“Snakemake” on the Grid



Processing (rulegraph)



Outputs

All outputs from snakemake are saved with Rucio, including the archived logs and benchmarks. Currently saved for an indefinite lifetime to Disk. We recommend to define a manual review before moving to Tape.

During development, temporary naming convention for datasets is:
“ {rucio_scope}:RBDP_{config_version}_{ruciodatasetsuffix}_{detid}_{run} ”,
but can easily be changed for mass productions, or add containers that group multiple datasets (e.g. per detid or per config_version)

```
(Linux-x86_64)(improve_workflow) ui: grid-snakemake$ rucio list-files Snakemake_testing:RBDP_v9.0_complete_00000132_00014590
```

SCOPE:NAME	GUID	ADLER32	FILESIZE	EVENTS
Snakemake_testing:KM3NeT_00000132_00014590.data.jppmuon_jppshower-upgoing_dynamic.offline.dst.v9.0_complete.root	A7C90E27-84AB-478E-870E-86D463C8F4C4	ad:c4db4fc2	77.633 MB	
Snakemake_testing:KM3NeT_00000132_00014590.data.jppmuon_jppshower-upgoing_dynamic.offline.v9.0_complete.root	FF61D6E9-890F-42A6-84C9-AD0E3AAD31B3	ad:c3e6188c	2.349 GB	
Snakemake_testing:KM3NeT_00000132_00014590.jppmuon_jppshower-upgoing.v9.0_complete.archived_logs.tar.gz	B4B5DCF7-E3D2-49B0-B709-20CE7461F105	ad:69597686	24.494 MB	
Snakemake_testing:KM3NeT_00000132_00014590.mc.gsg_neutrinos.km3sim_sirene_merged.jterbr.jppmuon_jppshower-upgoing_static.offline.dst.v9.0_complete.root	6D36FE98-190E-4471-889A-82D8306B75A1	ad:8f23144b	2.485 MB	
Snakemake_testing:KM3NeT_00000132_00014590.mc.gsg_neutrinos.km3sim_sirene_merged.jterbr.jppmuon_jppshower-upgoing_static.offline.v9.0_complete.root	F9B21856-BFA8-4AA2-8899-024B2793D721	ad:fa17a562	55.725 MB	
Snakemake_testing:KM3NeT_00000132_00014590.mc.mupage_default.sirene.jterbr.jppmuon_jppshower-upgoing_static.offline.dst.v9.0_complete.root	FD0B92A0-FD59-4E43-B9B2-7D1F78D7360D	ad:1298b23f	98.040 MB	
Snakemake_testing:KM3NeT_00000132_00014590.mc.mupage_default.sirene.jterbr.jppmuon_jppshower-upgoing_static.offline.v9.0_complete.root	46BDE537-6866-460B-A140-772E45C50D8E	ad:b8a0326d	2.664 GB	
Snakemake_testing:KM3NeT_00000132_00014590.mc.pure_noise.jterbr.jppmuon_jppshower-upgoing_static.offline.dst.v9.0_complete.root	C2958719-E7B2-4D1D-88E3-C19354E24380	ad:66cfb245	8.530 MB	
Snakemake_testing:KM3NeT_00000132_00014590.mc.pure_noise.jterbr.jppmuon_jppshower-upgoing_static.offline.v9.0_complete.root	FBDC3D1E-BE08-442E-8885-373EFF368EAE	ad:3db17105	189.602 MB	

```
Total files : 9
Total size : 5.470 GB
```

Production Status

- Full DetID processing attempted with DetID 132 (ORCA6)
  - See some failed jobs (issues at JEvtMerge or conversion to dst, usually because an earlier step failed “silently”)
- “Standalone approach” requires individual snakemake productions to be short enough to fit in available queues :
  - Most queues are 1-core 24h, with some up to 8-cores 96h.
  - Around 70-80h (single-core) for one full *ORCA6* run. And this will grow…
- Splitting of to_produce files might not be enough. Further splitting
  - “Horizontal”: multi-threading, file splitting then re-merging, and/or
  - “Vertical” : grid orchestrator to run sequential steps separately (discouraged?)

For testing, added options to RBDP to finish test run in 5 minutes instead

How to use (your turn!)

- Some setup (beyond standard certificate and DIRAC access):

  $ git clone git@git.km3net.de:rucio/grid-snakemake.git
  $ cd grid-snakemake
  $ git clone -b grid git@git.km3net.de:rucio/run-based-data-processing.git

  This part to be voided once run-based data processing made part of KM3NeT CVMFS environment

- Run:

  $ ./start-snakemake-singleWN.py -R <runlistfile> -I <configfile> -T </path/local/scratch> -S -k <km3env_ver> \

  -o <ruciodatasetsuffix> [-l if you want to make it local] [-m if you want to split into four jobs per run]

  (ignore Dirac error about executables not being “findable locally”, and note down the JobID in the “Submission Result” line)

- You can then check the status, and eventually get the logs:

  $ ./Status-API.py <JobID>
  $ ./Output-API.py <JobID>
  $ cd JobOutputs/<JobID> # to check Run-Snakemake-GridWN.log and other log files
  $ ../../Output_RBDP-logs.sh Run-Snakemake-GridWN.log # to download all snakemake log files (if debugging)

- To get the output files, use Rucio:

  $ rucio did list <rucio_scope>:RBDP_<config_version>_<ruciodatasetsuffix>_<detid>_<run> #etc

Modifications to workflow: https://git.km3net.de/rucio/run-based-data-processing

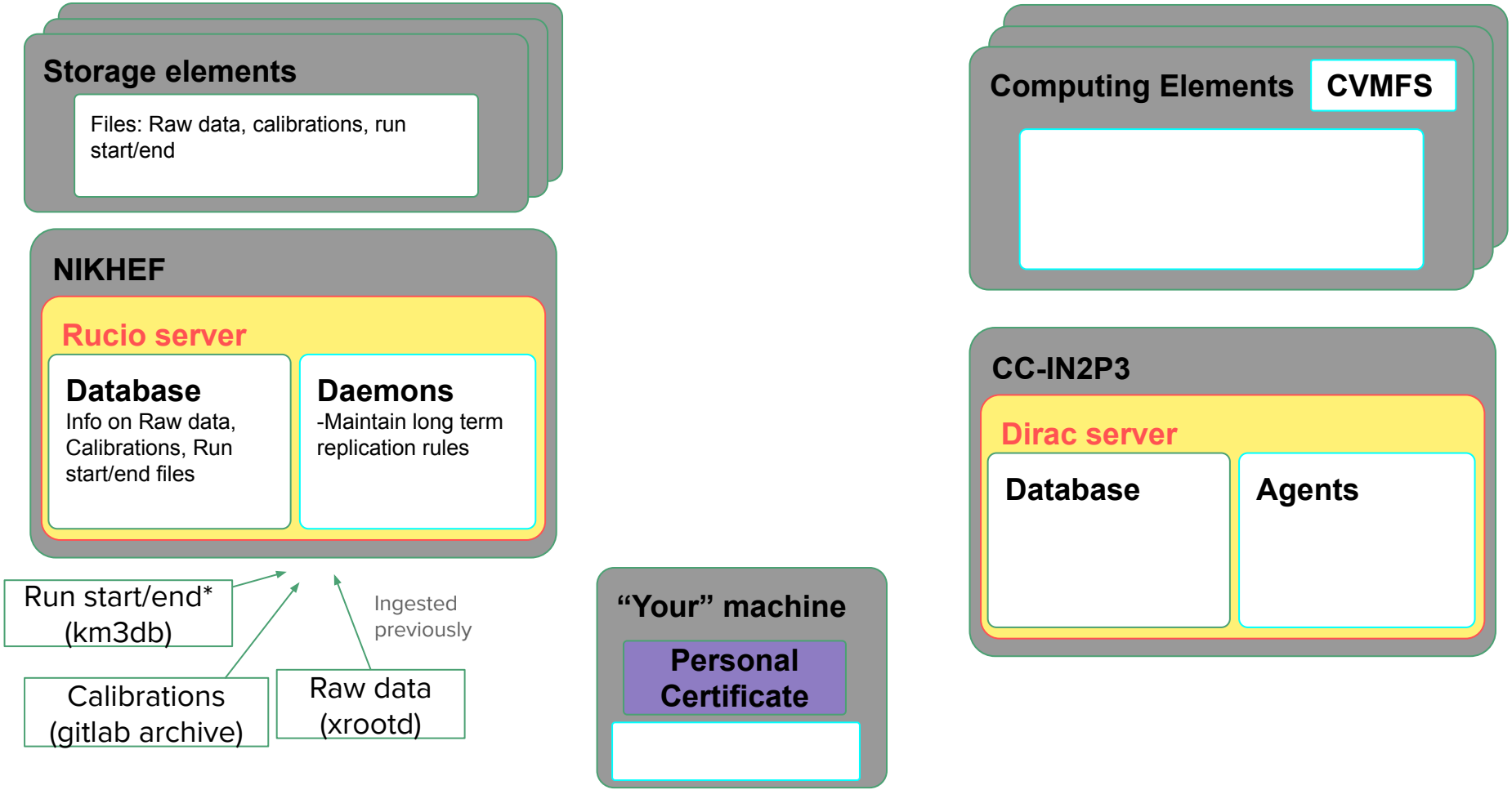
Job submission: https://git.km3net.de/rucio/grid-snakemake

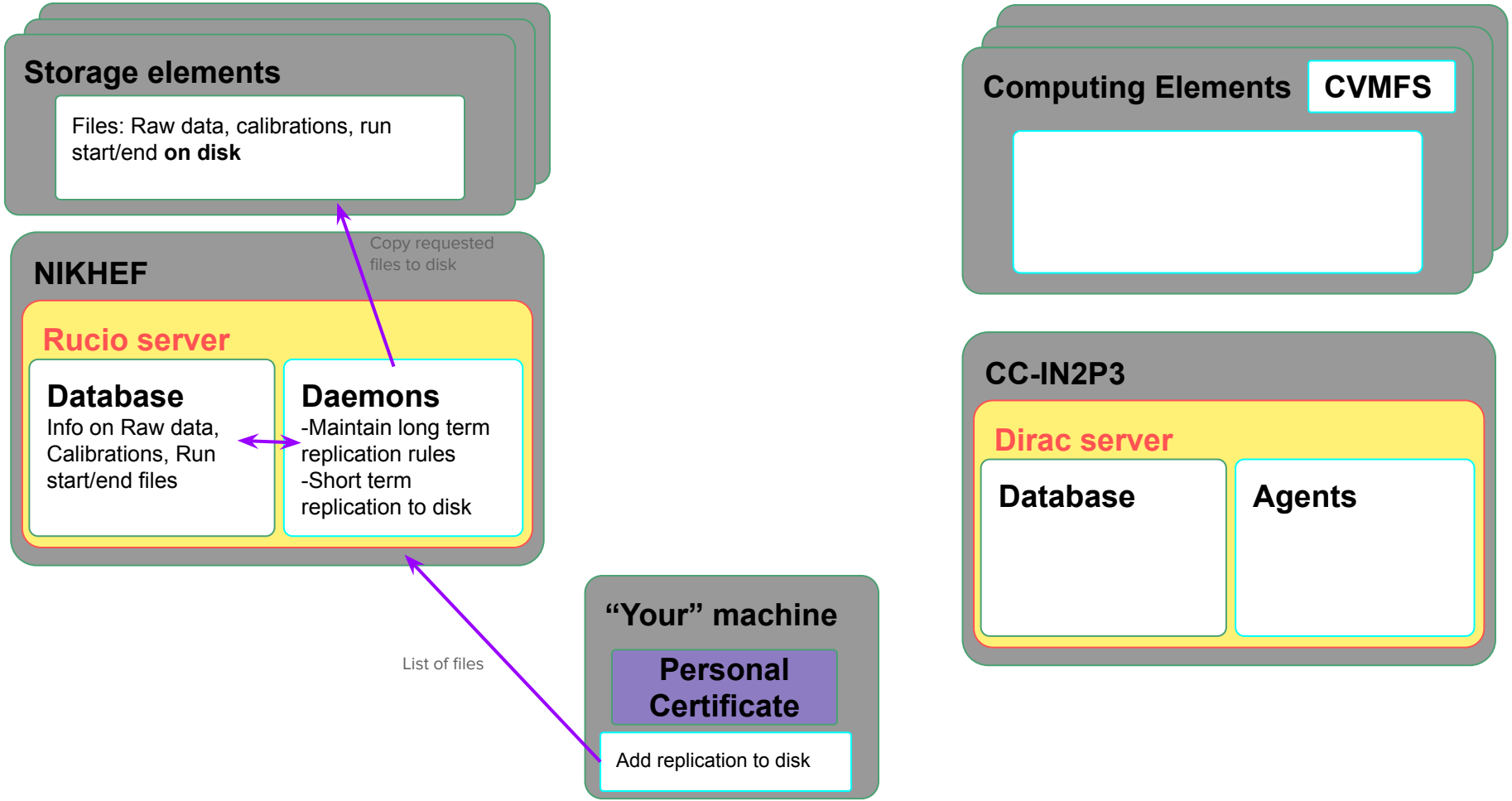
Summary of changes

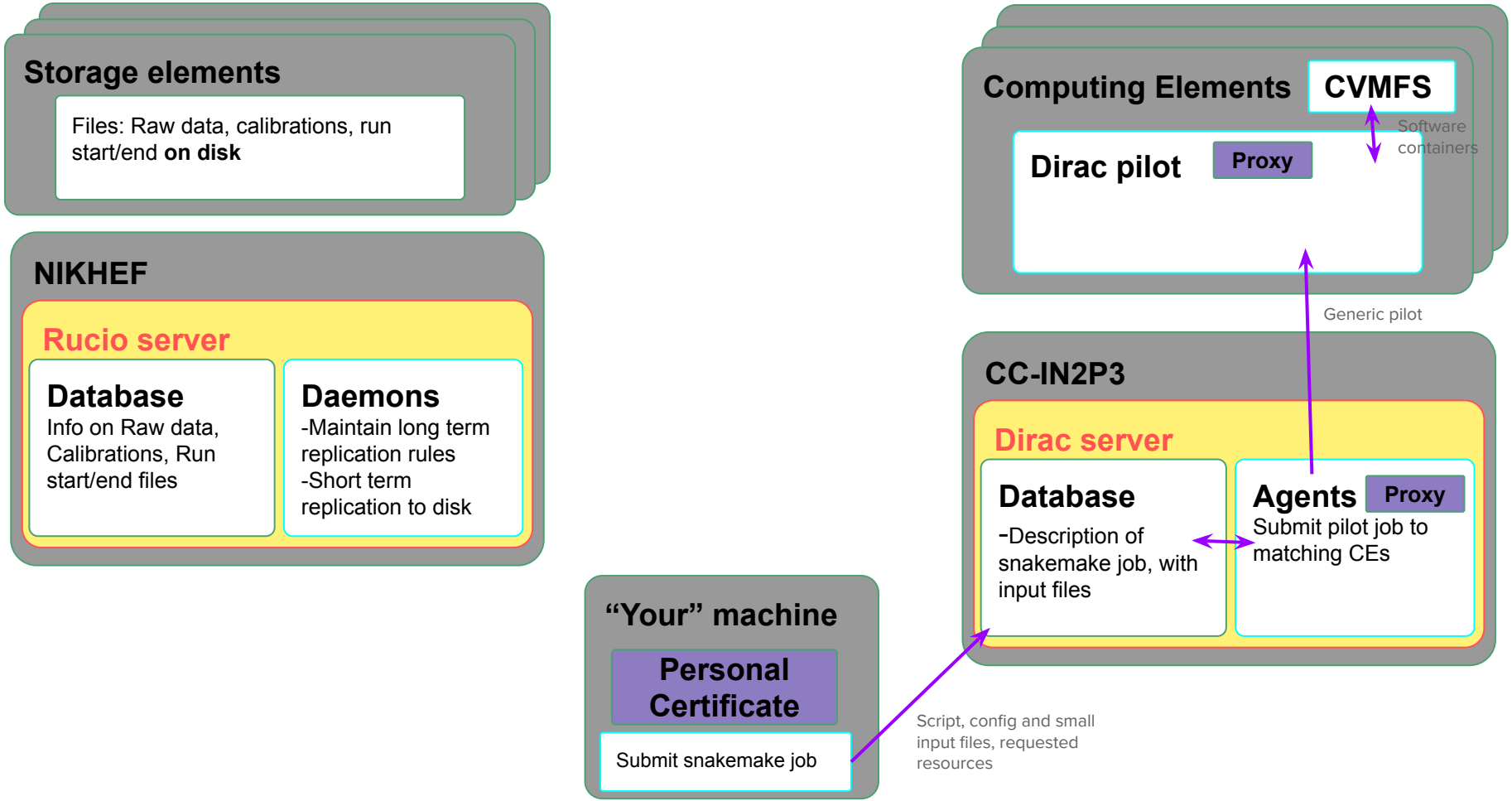
- Ingestion of all* our inputs into Rucio, and everything accessible with X509 certificate through snakemake rules!
  - This means inputs can be acquired from within grid job, no extra authentication necessary

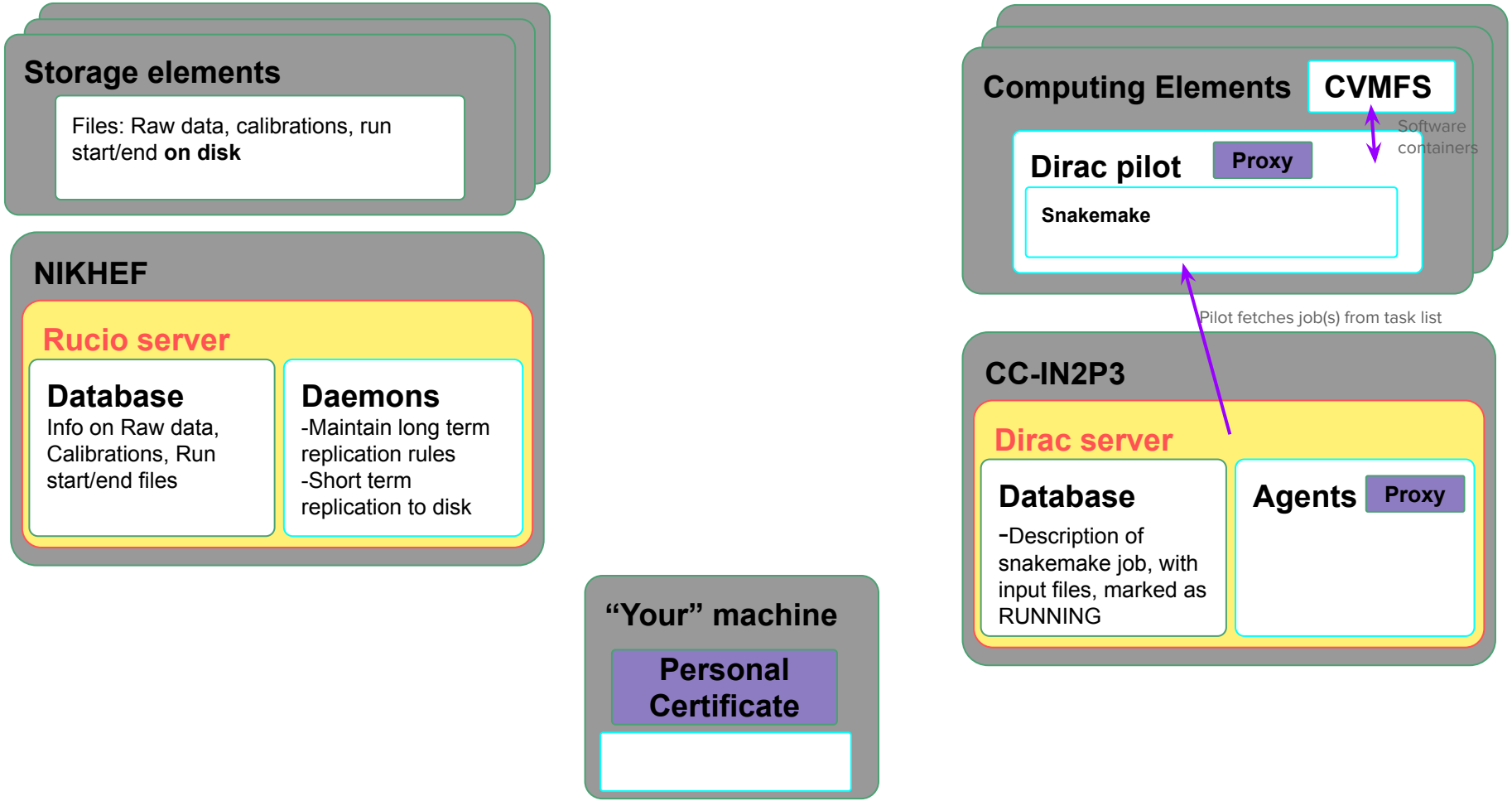
    * exception is km3db, where run start/end hasn't been ingested (yet?)
  - Only preparation step is ensuring data is replicated to disk before job submission
- Saving of outputs with Rucio integrated directly in Run-Based Data Processing rules; logs and benchmarks also saved automatically at the end
- Very basic splitting done automatically at submission level: identify all files in "to_produce" list with matching data type (data|pure_noise|mupage|gsg) and send them together as one job

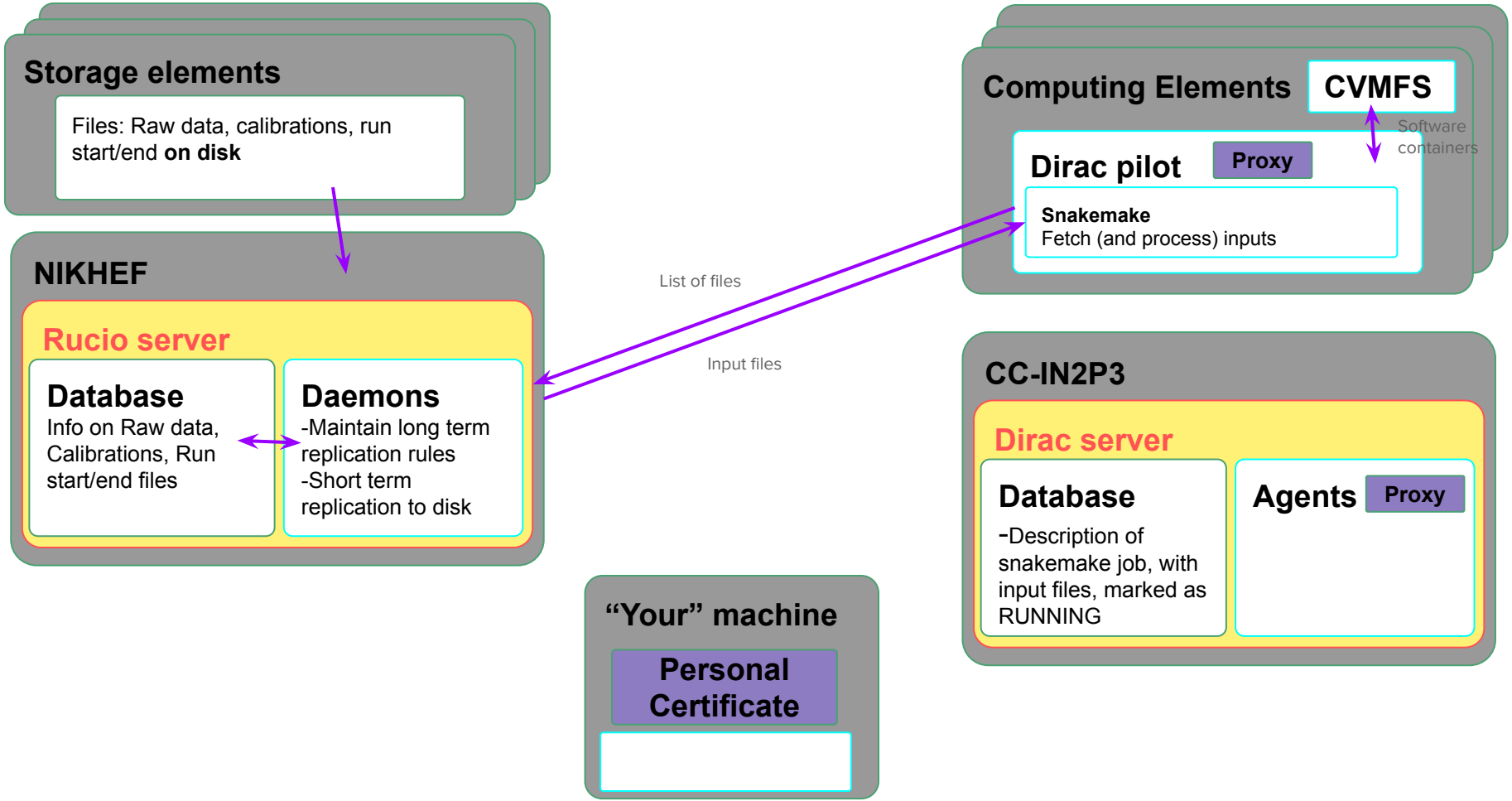
How does it work?

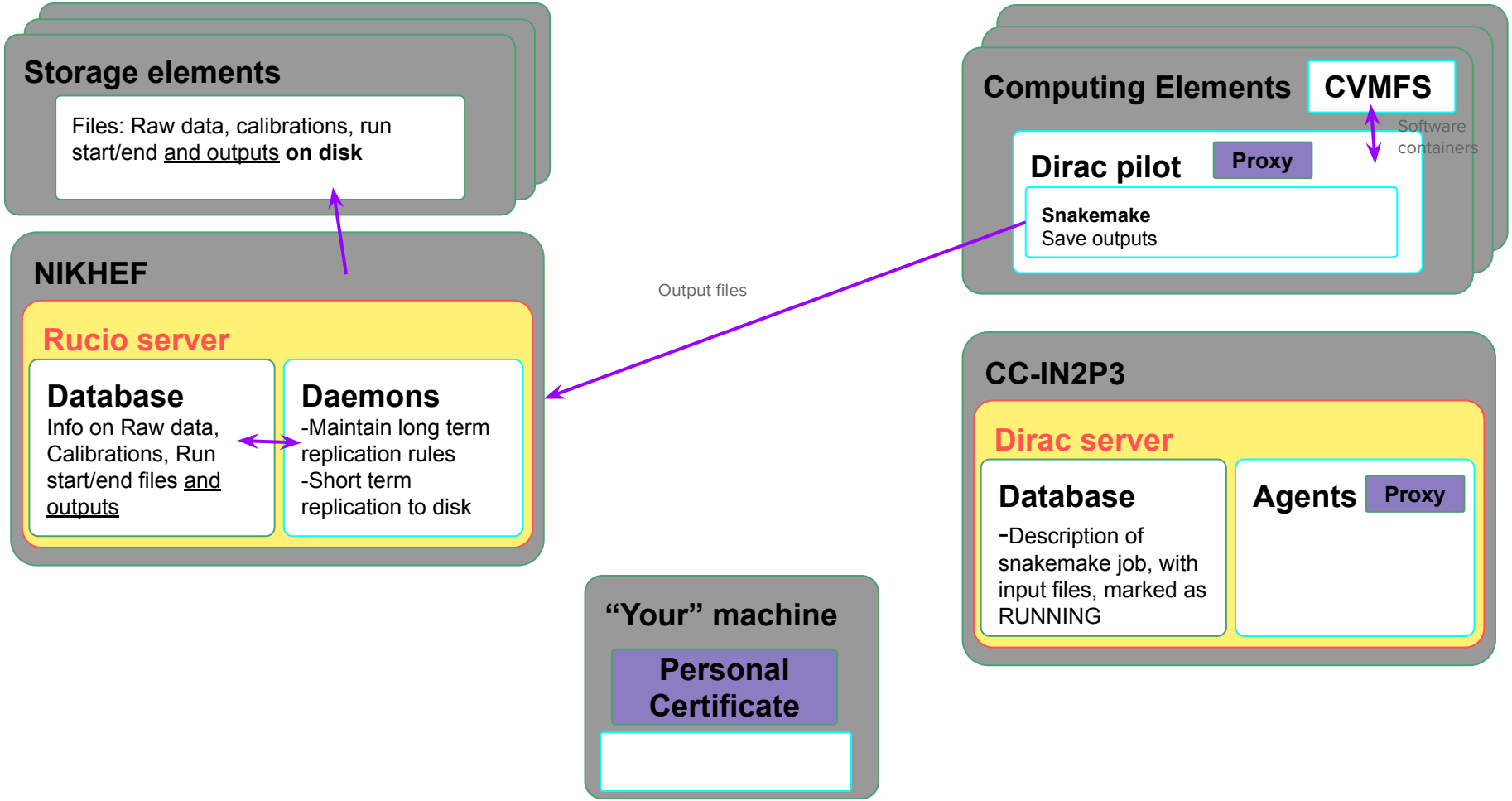

Storage elements
Files: Raw data, calibrations, run start/end
NIKHEF
Rucio server
Database
Info on Raw data, Calibrations, Run start/end files
Daemons
-Maintain long term replication rules
Run start/end* (km3db)
Calibrations (gitlab archive)
Raw data (xrootd)
Ingested previously
"Your" machine
Personal Certificate
Computing Elements
CVMFS
CC-IN2P3
Dirac server
Database
Agents

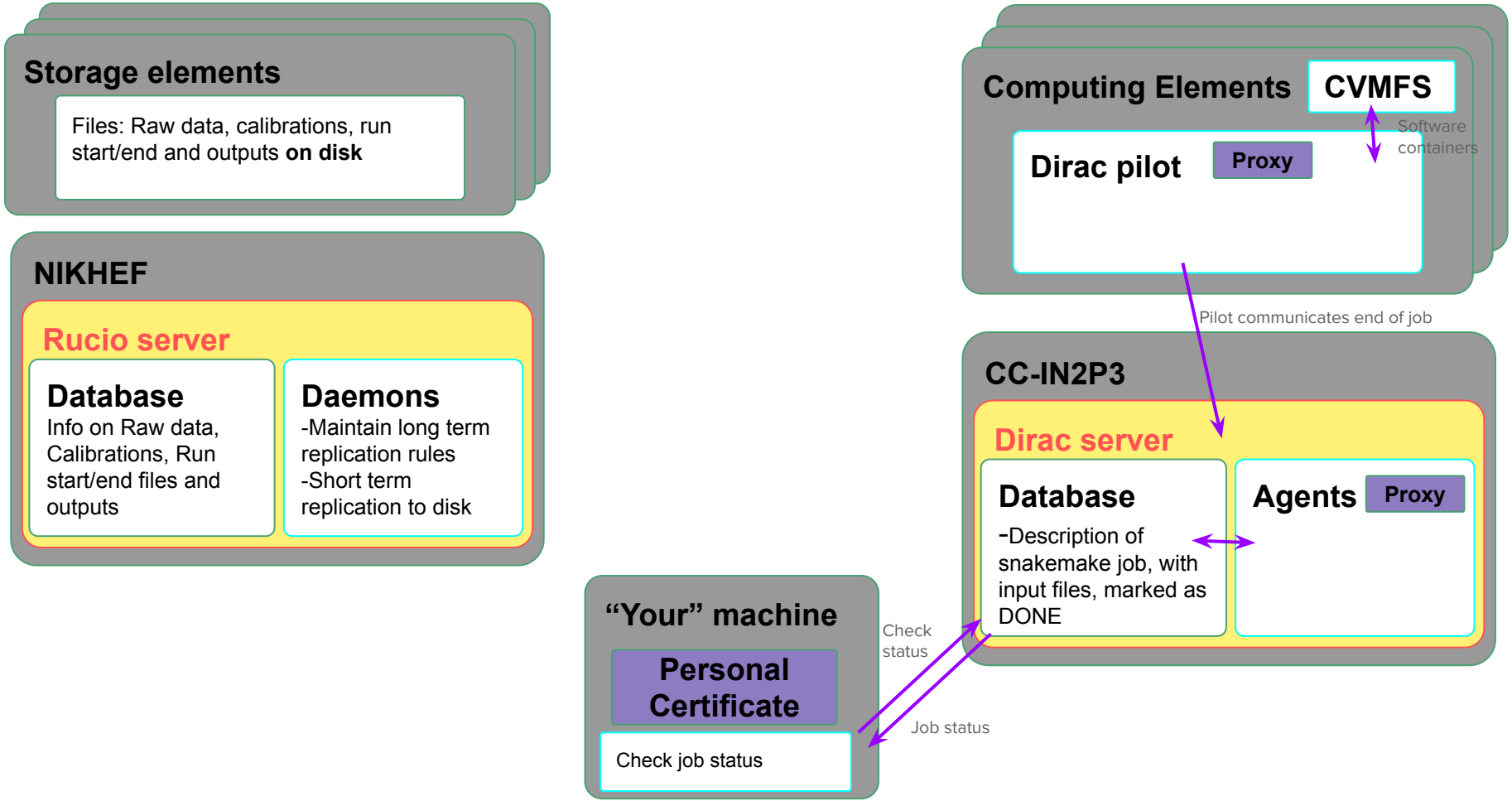

Storage elements
Files: Raw data, calibrations, run start/end on disk
Computing Elements
CVMFS
Copy requested files to disk
NIKHEF
Rucio server
Database
Info on Raw data, Calibrations, Run start/end files
Daemons
-Maintain long term replication rules
-Short term replication to disk
CC-IN2P3
Dirac server
Database
Agents
List of files
“Your” machine
Personal Certificate
Add replication to disk

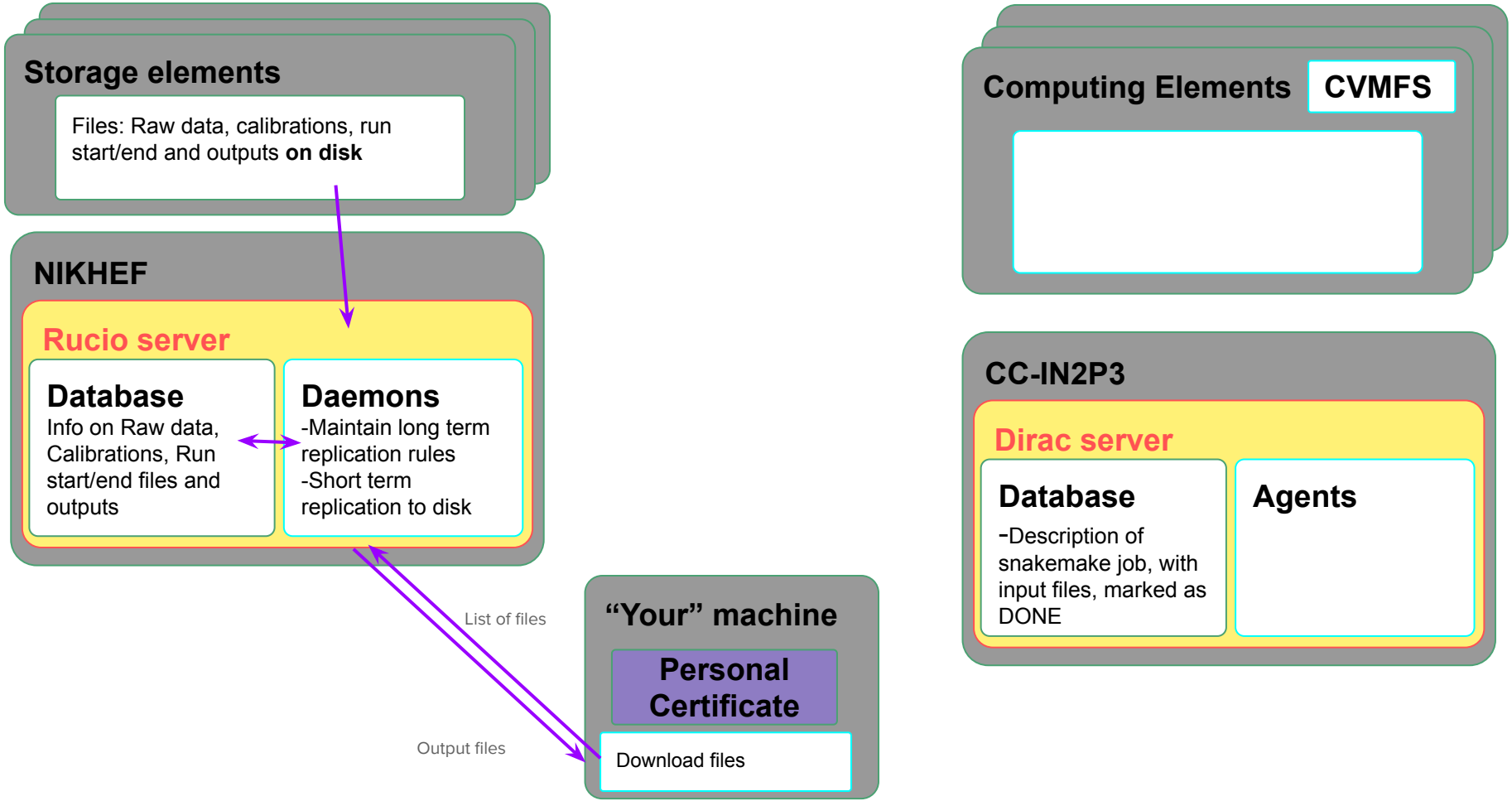

Storage elements
Files: Raw data, calibrations, run start/end on disk
NIKHEF
Rucio server
Database
Info on Raw data, Calibrations, Run start/end files
Daemons
-Maintain long term replication rules
-Short term replication to disk
Computing Elements
CVMFS
Software containers
Dirac pilot
Proxy
Generic pilot
CC-IN2P3
Dirac server
Database
-Description of snakemake job, with input files
Agents
Proxy
Submit pilot job to matching CEs
"Your" machine
Personal Certificate
Submit snakemake job
Script, config and small input files, requested resources


Storage elements
Files: Raw data, calibrations, run start/end on disk
NIKHEF
Rucio server
Database
Info on Raw data, Calibrations, Run start/end files
Daemons
-Maintain long term replication rules
-Short term replication to disk
"Your" machine
Personal Certificate
Computing Elements
CVMFS
Software containers
Dirac pilot
Proxy
Snakemake
Pilot fetches job(s) from task list
CC-IN2P3
Dirac server
Database
-Description of snakemake job, with input files, marked as RUNNING
Agents
Proxy


Storage elements
Files: Raw data, calibrations, run start/end on disk
NIKHEF
Rucio server
Database
Info on Raw data, Calibrations, Run start/end files
Daemons
-Maintain long term replication rules
-Short term replication to disk
List of files
Input files
Computing Elements
CVMFS
Software containers
Dirac pilot
Proxy
Snakemake
Fetch (and process) inputs
CC-IN2P3
Dirac server
Database
-Description of snakemake job, with input files, marked as RUNNING
Agents
Proxy
"Your" machine
Personal Certificate


Storage elements
Files: Raw data, calibrations, run start/end and outputs on disk
NIKHEF
Rucio server
Database
Info on Raw data, Calibrations, Run start/end files and outputs
Daemons
-Maintain long term replication rules
-Short term replication to disk
Output files
Computing Elements
CVMFS
Software containers
Dirac pilot
Proxy
Snakemake
Save outputs
CC-IN2P3
Dirac server
Database
-Description of snakemake job, with input files, marked as RUNNING
Agents
Proxy
"Your" machine
Personal Certificate


Storage elements
Files: Raw data, calibrations, run start/end and outputs on disk
NIKHEF
Rucio server
Database
Info on Raw data, Calibrations, Run start/end files and outputs
Daemons
-Maintain long term replication rules
-Short term replication to disk
Computing Elements
CVMFS
Software containers
Dirac pilot
Proxy
Pilot communicates end of job
CC-IN2P3
Dirac server
Database
-Description of snakemake job, with input files, marked as DONE
Agents
Proxy
"Your" machine
Personal Certificate
Check job status
Check status
Job status


Storage elements
Files: Raw data, calibrations, run start/end and outputs **on disk**
NIKHEF
Rucio server
Database
Info on Raw data, Calibrations, Run start/end files and outputs
Daemons
-Maintain long term replication rules
-Short term replication to disk
List of files
Output files
"Your" machine
Personal Certificate
Download files
Computing Elements
CVMFS
CC-IN2P3
Dirac server
Database
-Description of snakemake job, with input files, marked as DONE
Agents