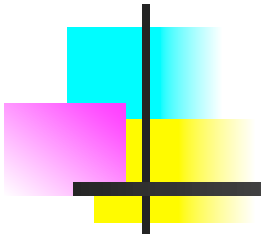
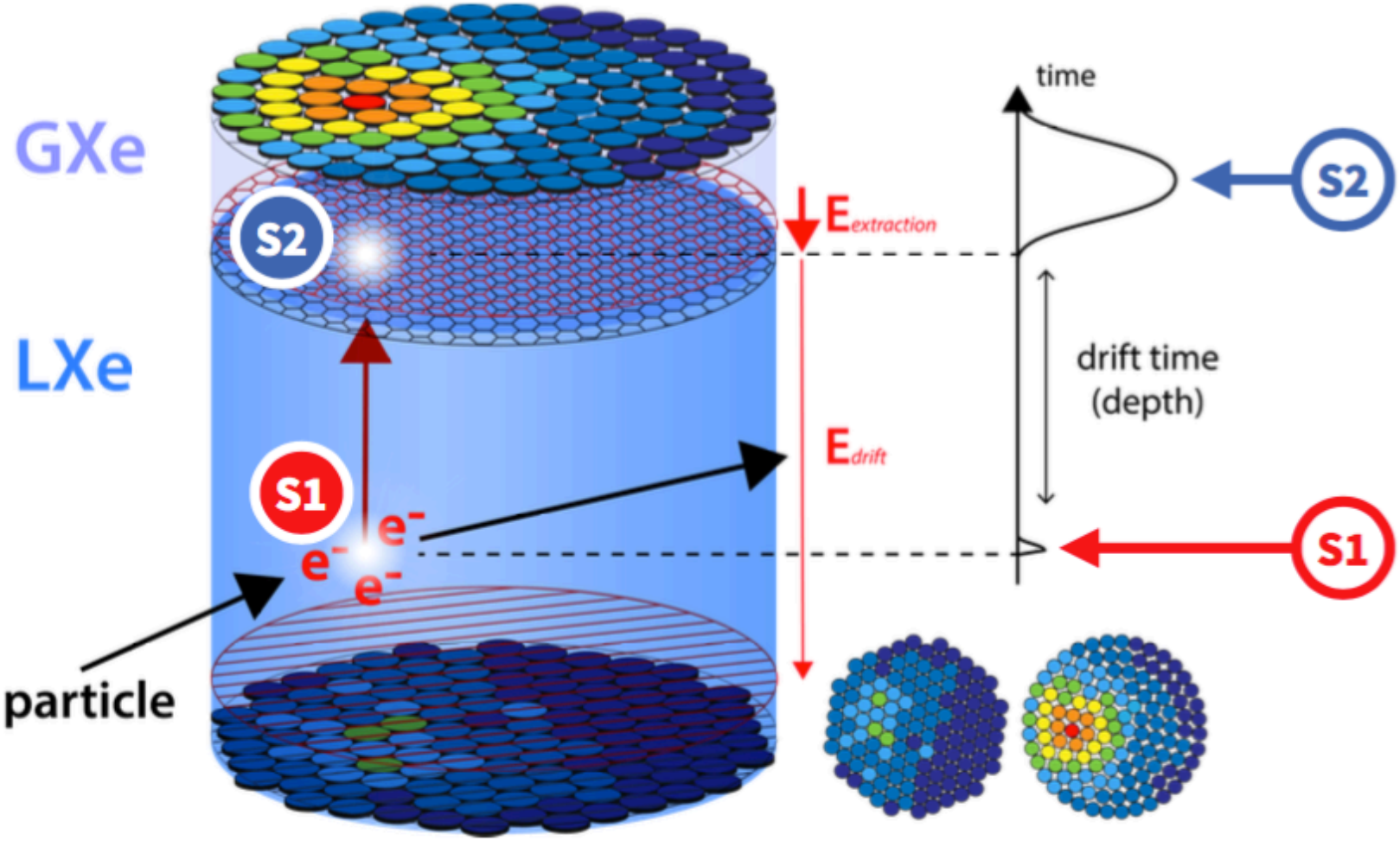


Distributed Computing Resources for XENONnT

B. Andrieu (LPNHE-Paris)

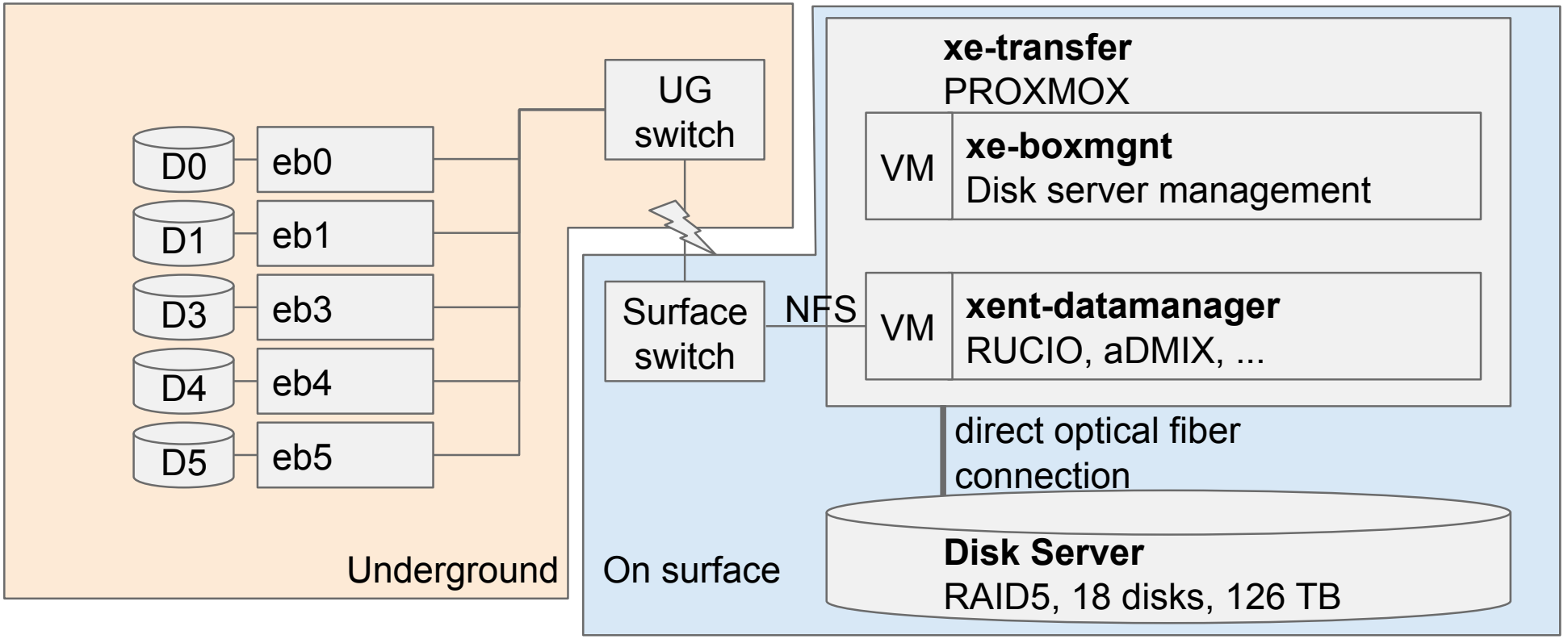
Data Flow



- "Trigger-less" DAQ mode
- 494 (TPC) + 84 (muon Veto) + 120 (neutron Veto) PMTs digitized at 100 MHz
- After online processing (0-suppression) on Event Builders → up to 300 MB/s
- Total volume of recorded data up to 1 PB/yr (1000 TB)

Data Acquisition & Online Processing

- 3 independent detectors (TPC, muon veto, neutron veto)
- DAQ system adapted from XENON1T
- Additional TPC channels (~ #PMTs x2 w.r.t. XENON1T) easily included by increasing the level of parallel readout chains (6 →10)
- Event rate throughput constant



Data Acquisition & Online Processing

- TPC PMT signal split by amplifiers in low-gain (×0.5) / high-gain channel (×10), each with their own readout.
- All PMT pulses above threshold recorded independently (no hardware trigger)
- Digitized, time-stamped & baseline-suppressed WaveForms stored in MongoDB
- Common clock (also for muon and neutron veto) for digitizers Time Sync
- Online Real-Time Software Trigger decided on Event Builders in DAQ room
  - Group WFs based on Time Coincidences
  - Classifies pulses as light- (S1) or charge-like (S2)
  - Associate S1-S2 to define full events from particle interactions
  - Store all (in event time range) available raw data from all PMTs in DB
  - Discard pulses outside of events (keep meta-information, e.g.rates,...)

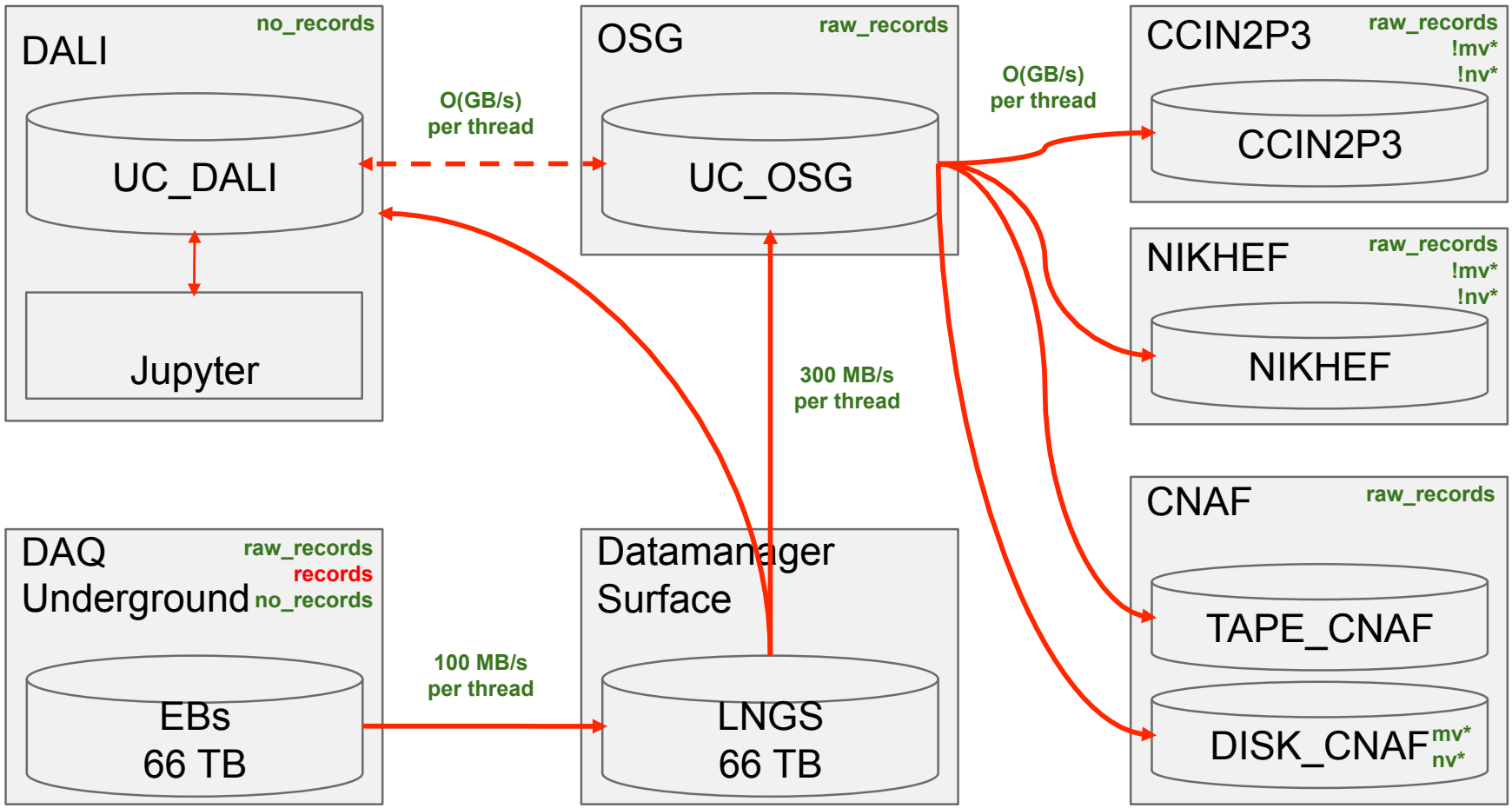
Data Types

At the end of data processing, different data types classified in four categories:

- **raw_records** : heavy raw data
  - `["raw_records", "raw_records_he", "raw_records_mv", "raw_records_nv", "raw_records_coin_nv", "lone_raw_records_nv", "lone_raw_record_statistics_nv"]`
- **light_raw_records** : light raw data
  - `["raw_records_aqmon","raw_records_aqmon_nv","raw_records_aux_mv"]`
- **records :** data produced directly by raw data and as heavy as them
  - `["records","records_he", "records_nv", "records_mv"]`
- **no_records** : high level data (priority for analysis)
  - `["peak_basics", "peak_positions_mlp","lone_hits", "pulse_counts", "peaklets", "merged_s2s", "peaklet_classification", "peaklet_classification_he", "veto_regions", "pulse_counts_he", "peaklets_he","led_calibration","hitlets_mv","hitlets_nv", "event_basics", "events", "peak_proximity", "peak_positions", "peak_positions_cnn", "peak_positions_gcn"]`

Data Transfer and Management

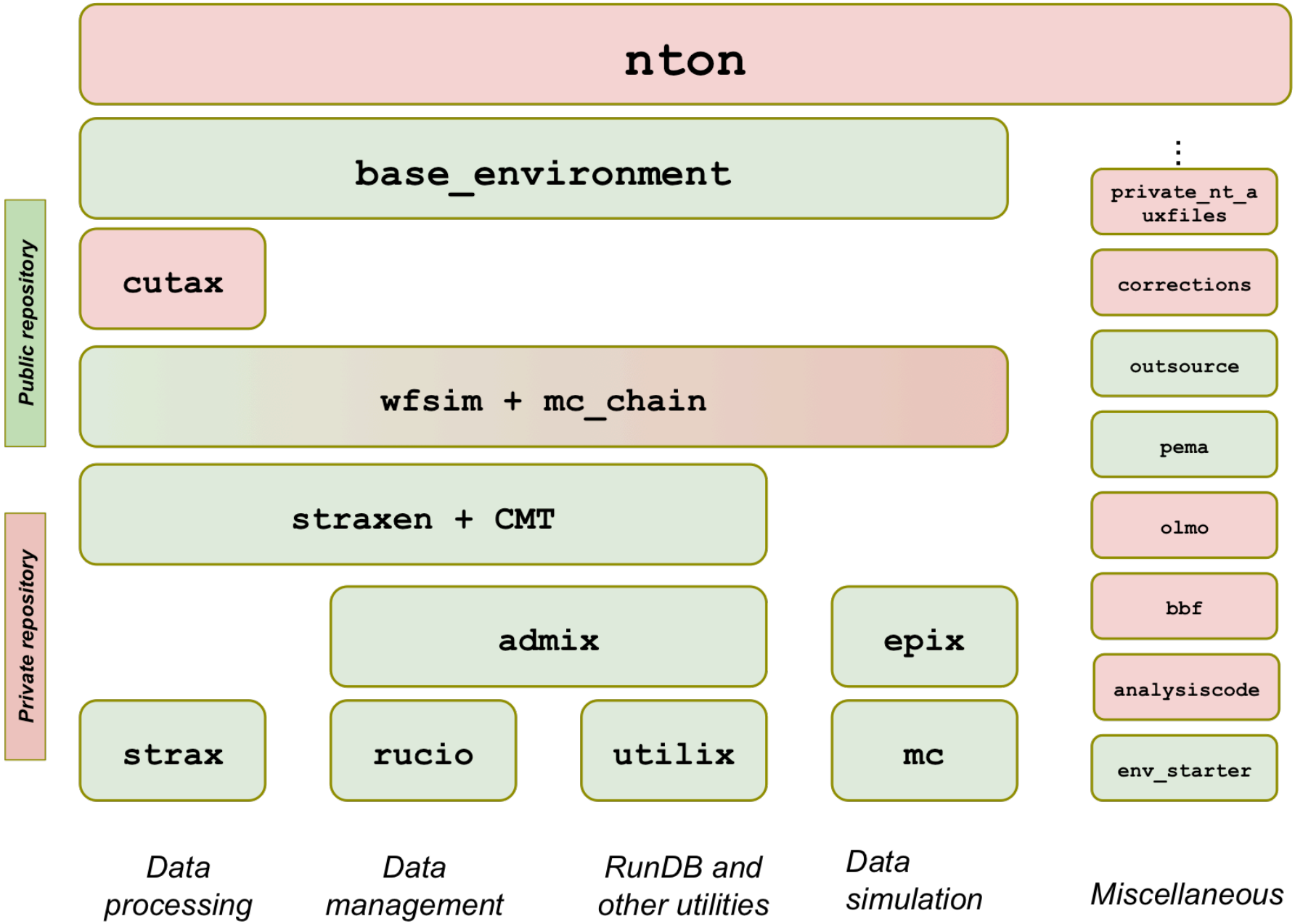
How data would travel



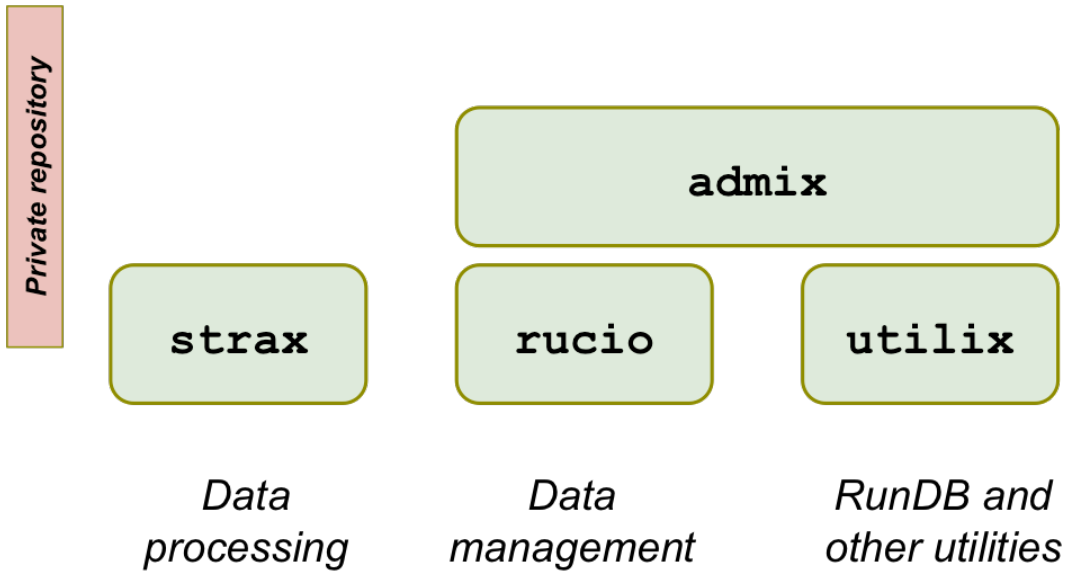
Three main tasks

- Pull data quickly from DAQ, assuring DAQ buffer as clean as possible
- Distributing data on GRID efficiently and with high redundancy
- Ease the data access to end users

Software Tools



Data Management Tools



GRID Tools

`rucio`

Rucio - Scientific Data Management

- Data Grid management software used also by ATLAS, IceCube, LIGO, DUNE, and many others

- Data spread across several *Rucio Storage Elements (RSEs) via replication rules*

- Rucio used to
  - **Upload** new data into our grid storage
  - **Transfer** data to our various RSEs
  - **Download** data from an RSE to a local machine

rucio nomenclature in XENONnT

Run: set of data collected within well defined experimental conditions during a given amount of time (usually 1 hour for stable running conditions). A new entry in the database (run DB) is added for each new run ("rundoc" in jargon, stands for run document). A run can be composed of several datasets. Defined by a unique run number Nrun (e.g. 010135).

Dataset: a collection of files of the same data type defined in strax (e.g. *raw_records*) produced by the same strax plugin for a given run Nrun. Named in the form *data_type-plugin_hash* (e.g. *raw_records-rfzvpzj4mf*) and stored by DAQ as a directory *Nrun-dataset_name* (e.g. *010135-raw_records-rfzvpzj4mf*) containing one metadata file and several chunk files.

Metadata: a json file located in its dataset directory retaining informations about dataset and its chunks. Named in the form *dataset_name-metadata.json* (e.g. *raw_records-rfzvpzj4mf-metadata.json*).

Chunk: a single file. Datasets too big to be contained in a single file are splitted in several chunks, numbered at the end with a progressive number *Nchunk* and named in the form *dataset_name-Nchunk* (e.g. *raw_records-rfzvpzj4mf-000270* for data chunk 270).

Scope: a Rucio concept designed to contain multiple datasets of a given run. Each run then corresponds to a single scope named in the form *xnt_Nrun* (e.g. *xnt_010135*), which contains one dataset for each particular data type.

DID (Data IDentifier): a string used by Rucio to identify a particular dataset of a given scope. It is named in the form *scope_name:dataset_name* (e.g. *xnt_010135:raw_records-rfzvpzj4mf*).

RSE (Rucio Storage Element): a logical abstraction used by Rucio to indicate a storage system located somewhere in the GRID, e.g. for RSEs used in XENON: LNGS_USERDISK, UC_DALI_USERDISK, CCIN2P3_USERDISK,…

Replication rule: an instruction which tells Rucio to replicate data of a given DID from one RSE to another one. For instance, to copy data of a given DID from LNGS_USERDISK (our disk storage in LNGS) to UC_DALI_USERDISK (our disk storage in UChicago used for data analysis), a new rule has to be created, which Rucio will interpret to automatically transfer all data in that DID from LNGS_USERDISK to UC_DALI_USERDISK. Replication rules can have lifetimes, so that a rule can be read by Rucio as, e.g. : "Put this dataset on UC_OSG_USERDISK for 1 week, then remove it". If no lifetime is passed, it is infinite.

GRID Sites

List of **RSEs** (Rucio Storage Elements) for XENON VO:

RSE	USAGE	LIMIT	QUOTA LEFT	
CCIN2P3_USERDISK	1.005 PB	2.500 PB	1.495 PB	Lyon
CNAF_TAPE2_USERDISK	189.324 TB	3.000 PB	2.811 PB	CNAF tape (old schema)
CNAF_TAPE3_USERDISK	426.941 GB	1.000 PB	999.573 TB	CNAF tape (new schema)
CNAF_TAPE_USERDISK	782.936 TB	3.000 PB	2.217 PB	
CNAF_USERDISK	40.532 TB	300.000 TB	259.468 TB	CNAF
LNGS_USERDISK	3.981 TB	50.000 TB	46.019 TB	LNGS (disk for uploading)
NIKHEF2_USERDISK	370.220 TB	400.000 TB	29.780 TB	Nikhef
NIKHEF_USERDISK	53.319 TB	275.000 TB	221.681 TB	Nikhef (old)
SDSC_USERDISK	1.632 MB	50.000 TB	50.000 TB	San Diego
SURFSARA_USERDISK	299.781 TB	500.000 TB	200.219 TB	Surfsara (Nikhef)
UC_DALI_USERDISK	42.663 TB	100.000 TB	57.337 TB	Dali (UChicago)
UC_OSG_USERDISK	845.096 TB	980.000 TB	134.904 TB	UChicago
WEIZMANN_USERDISK	79.998 TB	80.000 TB	2.075 GB	Weizmann (not used in nT)

rucio commands examples in XENONnT

rucio list-rules *xnt_031803:raw_records-rfzvpzj4mf*

```
(XENONnT_2021.11.5) [ershockley@login-el7 ~]$ rucio list-rules xnt_031803:raw_records-rfzvpzj4mf
ID                                ACCOUNT     SCOPE:NAME                          STATE[OK/REPL/STUCK]
RSE_EXPRESSION       COPIES  EXPIRES (UTC)    CREATED (UTC)
--------------------------------  ----------  ----------------------------------  --------------------
-----------------  --------  ---------------  -------------------
e4e452f0c1ef4620b16bafeafd58a8fa  production  xnt_031803:raw_records-rfzvpzj4mf  OK[85/0/0]
SURFSARA_USERDISK         1                  2021-11-09 17:45:02
8c7ac38c0eac4fb29496b1598bdb465c  production  xnt_031803:raw_records-rfzvpzj4mf  OK[85/0/0]
UC_OSG_USERDISK           1                  2021-11-09 17:45:01
```

rucio list-rules *xnt_031803:hitlets_nv-tbvnulr7cb*

```
(XENONnT_2021.11.5) [ershockley@login-el7 ~]$ rucio list-rules xnt_031803:hitlets_nv-tbvnulr7cb
ID                                ACCOUNT     SCOPE:NAME                          STATE[OK/REPL/STUCK]
RSE_EXPRESSION       COPIES  EXPIRES (UTC)    CREATED (UTC)
--------------------------------  ----------  ----------------------------------  --------------------
----------------  --------  ---------------  -------------------
8feaf0b69f2948ffabc2440ca9700dfd  production  xnt_031803:hitlets_nv-tbvnulr7cb  OK[2/0/0]
UC_DALI_USERDISK          1                  2021-12-01 00:03:16
```

Advanced Data Management In XENONnT

admix

XENONnT data too large to be stored at any one site, must be distributed on grid storage at several grid sites (RSE) → **aDMIX** developed at LPNHE

- main task of **aDMIX** : perform offline real-time data transfer from DAQ to grid
  → need to free limited size DAQ buffer disks for smooth data taking
- other **aDMIX** task: allow analysts to download specific data
  → easy access also to low-level data (not permanently available on Dali)
- **aDMIX** is a XENONnT wrapper around **rucio**, using **utilix** and the **RunDB**
- **aDMIX** is installed and always running on xent-datamanager to:
  - upload on the fly data of each completed run from Event Builder onto datamanager
    → create new rucio datasets
  - transfer data from datamanager to other grid sites
    → use rucio replication rules
  - update our runDB and free Event Builder disk when data transfer is completed
    → in parallel, rucio also keeps track of data location
- **aDMIX** is also installed and running permanently on Dali
  → get rucio information of any dataset for data downloading from grid

aDMIX Manager and Tasks

aDMIX Upload Manager controls 3 tasks:

- **Upload**
  - Uploads a dataset when the Upload Manager assigns it
  - Creates replication rules for exporting data to grid
- **CheckTransfers**
  - Runs every 10 minutes
  - Updates DB to flag successfully transferred datasets and runs
- **CleanEB**
  - Runs every 10 minutes
  - Cleans data from EB and datamanager disk as soon as there are enough redundant copy elsewhere

aDMIX Upload Manager

- Assigns tasks to the Upload jobs
- Assigns priorities for jobs between high and low level data.
- Reports which **aDMIX** tasks are running

```
Run
Datasets still to upload: 0

--------------------------------------------------------------------
There are currently 10 active threads, 0/2 high level and 2/8 low level:
Task: Upload, Screen: upload1 , not yet assigned
Task: Upload, Screen: upload10, not yet assigned
Task: Upload, Screen: upload2 , not yet assigned
Task: Upload, Screen: upload3 , Run: 25454, Type raw_records, Hash rfzvpzj4mf, EB eb4, Priority 3, Since
00:08:40 ago
Task: Upload, Screen: upload4 , not yet assigned
Task: Upload, Screen: upload5 , not yet assigned
Task: Upload, Screen: upload6 , Run: 25454, Type raw_records_nv, Hash rfzvpzj4mf, EB eb4, Priority 3, Since
00:08:40 ago
Task: Upload, Screen: upload7 , not yet assigned
Task: Upload, Screen: upload8 , not yet assigned
Task: Upload, Screen: upload9 , not yet assigned
--------------------------------------------------------------------
```

aDMIX Data Redundancy

aDMIX ensures redundancy:

- define clear policy of which data types should stay 1-3 days on EB before being deleted, and which data types will never be deleted from EB.
- at least two copies of **raw_records** (including light ones)
- one copy of **raw_records** should be in tapes
- all **no_records** (high-level data for analysts) on Dali disks
- remove **records** (as heavy as **raw_records**, can be reproduced if needed)
- All of that tracked by Rucio catalogue and runDB
- Each site no more than 1-write and 1-read operation
- **Copy on Dali must have the highest priority**

aDMIX Daily Operation

aDMIX is designed to run at all time and manage transfer and storage of XENONnT data as soon as they are processed by Event Builders.

It can happen nevertheless that one or more of **aDMIX** Tasks needed for that purpose may halt for various reasons, e.g. if rucio server stops or if there is a connexion problem between LNGS and Chicago, etc..

NO automatic recovery of interrupted data transfers
→ Human daily survey still needed
→ Takes 30” to check everything is fine (usually the case)
→ Once a week (average over last 7 months) takes ~1h to reinitiate data transfers

aDMIX Next Steps

- **Monte Carlo** simulation requiring a treatment similar to what has been done for data (collaboration with MC team)
- **UC Dali** storage capacity not infinite and not even as large as **UC OSG**. Need to develop an automatic caching system that will push/pull data to/from Dali.
- Data flow from DAQ to datamanager reasonable but not enough in case of high flows. Need to optimize data transfer by hardware upgrade (SSD disks) and/or reduced online processing (ideally both of them)?
- Problem with huge number of small files. Need to develop a solution based on tarballing raw data, proposed flow chart:
    - **Raw copy (rsync) from EB to datamanager (faster than upload, unsecure?)**
    - **Tarballing directly on datamanager → two copies (tarball and not tarball)**
    - **upload non-tarball and send to UC_OSG**
    - **upload tarball and sent to tape-based storages (CNAF, CCIN2P3, NIKHEF)**
    - In case of reprocessing and absence of non-tarball data in UC_OSG:
        - **Download the tarball (where? Dali?)**
        - **Untar it**
        - **Upload it on Rucio or starting strax directly?**

Summary and Outlook

aDMIX Data Management software has been developed at LPNHE

It is working well since beginning of XENONnT data taking

Much remains to be done (in priority for integrating Monte Carlo and reducing number of files on tape)

LPNHE is committed to data management (co-leading with Rice University)

Backup

Data Management Tools : Analysis Framework Utilities

strax

AxFoundation / strax Public

Streaming analysis for xenon experiments

DOI 10.5281/zenodo.1340632 | tests passing | docs passing | coverage 86% | pypi v1.1.2 | codefactor A | gitter join chat

Strax is an analysis framework for pulse-only digitization data, specialized for live data reduction at speeds of 50-100 MB(raw) / core / sec. For more information, please see the strax documentation.

Strax' primary aim is to support noble liquid TPC dark matter searches, such as XENONnT. The XENON-specific algorithms live in the separate package straxen. If you want to try out strax, you probably want to start there. This package only contains the core framework and basic algorithms any TPCs would want to use.

Strax is the framework on which straxen is based.

This is where you find the base classes for things like contexts and plugins.

Strax is *not* XENON-specific software

Data Management Tools : XENONnT Utilities

utilix

- Utilix is a *utility* package specific to XENON. It currently has 3 main features.
- Simplify access and use of our Mongo Database.
  - Tracks crucial information about each run taken by the DAQ
  - Tracks our corrections via **CMT** (more on this later)
  - Tracks important information about our production contexts stored in **cutax**
  - Tracks/stores input datafiles used by **straxen**, like correction maps
  - Many others
- Allow for a flexible configuration file that can be used by other packages.
  - For example, our Mongo credentials
  - Can set environment variable XENON_CONFIG to specify your config
- Simplify job submission to the Midway batch queue
  - For example, see the `Analysis` base class in nton here.

Data Management Tools : GRID Utilities

rucio

Rucio - Scientific Data Management

UC_DALI_USERDISK is a special RSE. It is mounted on our dali space and accessible for analysis. Our high-level data goes here.

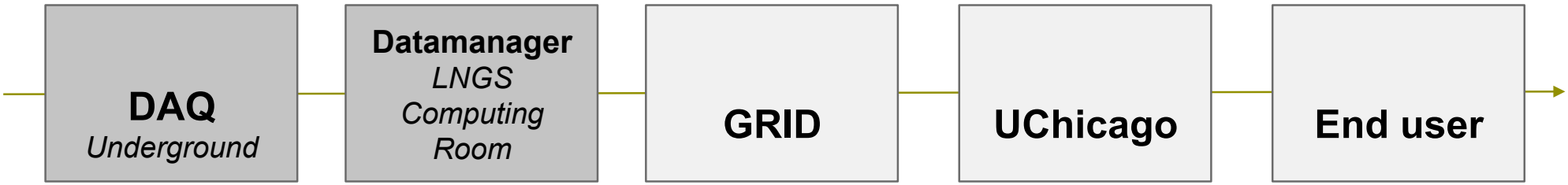
```
(XENONnT_2021.11.5) [ershockley@login-el7 ~]$ rucio list-file-replicas xnt_031803:hitlets_nv-tbvnulr7cb
+------------+------------------------------+------------+-----------
+-------------------------------------------------------------------------------------------------------------------
------------+
| SCOPE      | NAME                         | FILESIZE   | ADLER32    | RSE: REPLICA
|
|------------+------------------------------+------------+-----------
+-------------------------------------------------------------------------------------------------------------------
------------|
| xnt_031803 | hitlets_nv-tbvnulr7cb-000000 | 405.154 MB | e0edd8ab   | UC_DALI_USERDISK: gsiftp://
sdm06.rcc.uchicago.edu:2811/dali/lgrandi/rucio/xnt_031803/b6/a8/hitlets_nv-tbvnulr7cb-000000 |
| xnt_031803 | hitlets_nv-tbvnulr7cb-000001 | 151.816 MB | 995a2cf6   | UC_DALI_USERDISK: gsiftp://
sdm06.rcc.uchicago.edu:2811/dali/lgrandi/rucio/xnt_031803/07/02/hitlets_nv-tbvnulr7cb-000001 |
+------------+------------------------------+------------+-----------
+-------------------------------------------------------------------------------------------------------------------
------------+
```

```
[ershockley@dali-login1 ~]$ ls /dali/lgrandi/rucio/xnt_031803/b6/a8/hitlets_nv-tbvnulr7cb-000000
/dali/lgrandi/rucio/xnt_031803/b6/a8/hitlets_nv-tbvnulr7cb-000000
```

XENONnT data flow: from DAQ to DataManager

a) **Pull data quickly from DAQ, assuring DAQ buffer as clean as possible**
b) Distributing data on GRID efficiently and with high redundancy
c) Ease the data access to end users

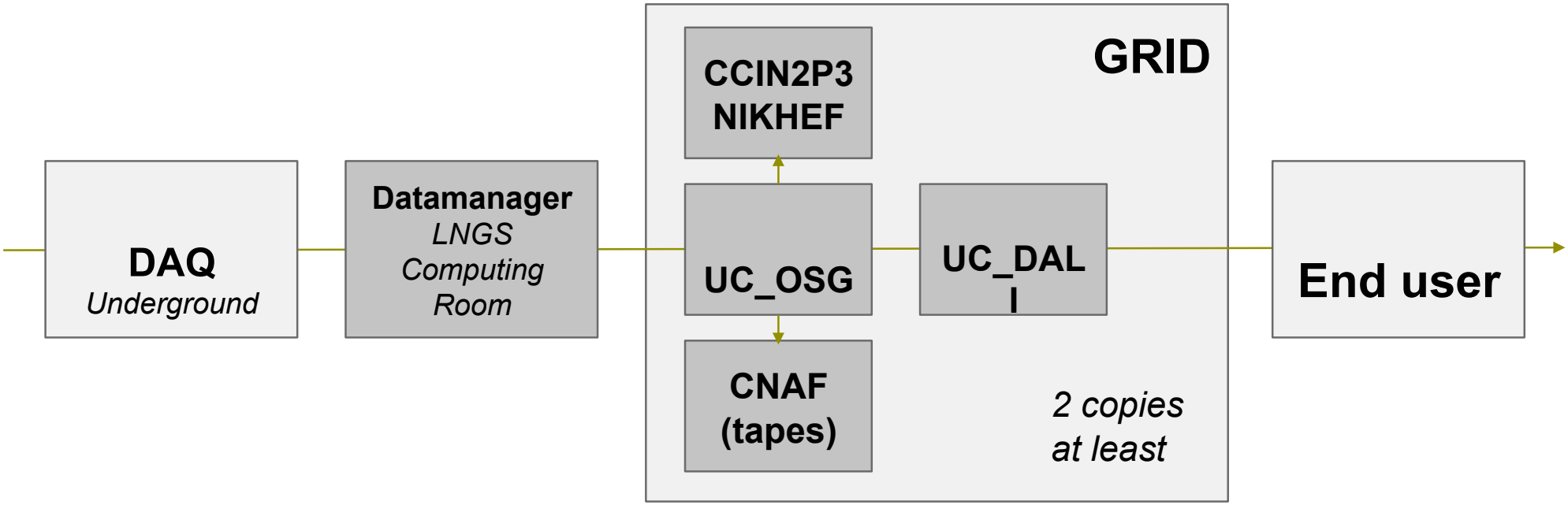
Interaction with DAQ crew very positive with continuous feedback. Admix takes care of pulling data and clean them from the DAQ Event Builder and from Datamanager as soon as enough redundancy is obtained on GRID. Also, we defined a clear policy of which data types should stay 1-3 days on EB before being deleted, and which data types will never be deleted from EB.



XENONnT data flow: from DataManager to GRID

a) Pull data quickly from DAQ, assuring DAQ buffer as clean as possible
b) Distributing data on GRID efficiently and with high redundancy
c) Ease the data access to end users

Thanks to Admix (interfaced with Rucio and Run DB), data flow is optimized to take the best from each site by minimizing the need of human intervention/support. Any action on data is registered at same time by Rucio Catalogue and Run DB. In case of issues (network, disks, DB or Rucio failures), we have ready a backup solution based on a simple multi-thread rsync script (already Admix commissioning).



XENONnT data flow: from GRID to users

a) Pull data quickly from DAQ, assuring DAQ buffer as clean as possible
b) Distributing data on GRID efficiently and with high redundancy
c) Ease the data access to end users

Dali disk storage is at same time a GRID node (handled by Rucio) and the analysis hub for users. Strax/ Straxen has been modified in such a way that analysts can access data in transparent way to the Rucio Dali Storage directly (before, you had to download them). Advantages: 1) we do not waste space 2) easier bookkeeping