ELAsTiCC Tutorial

Rob Knop

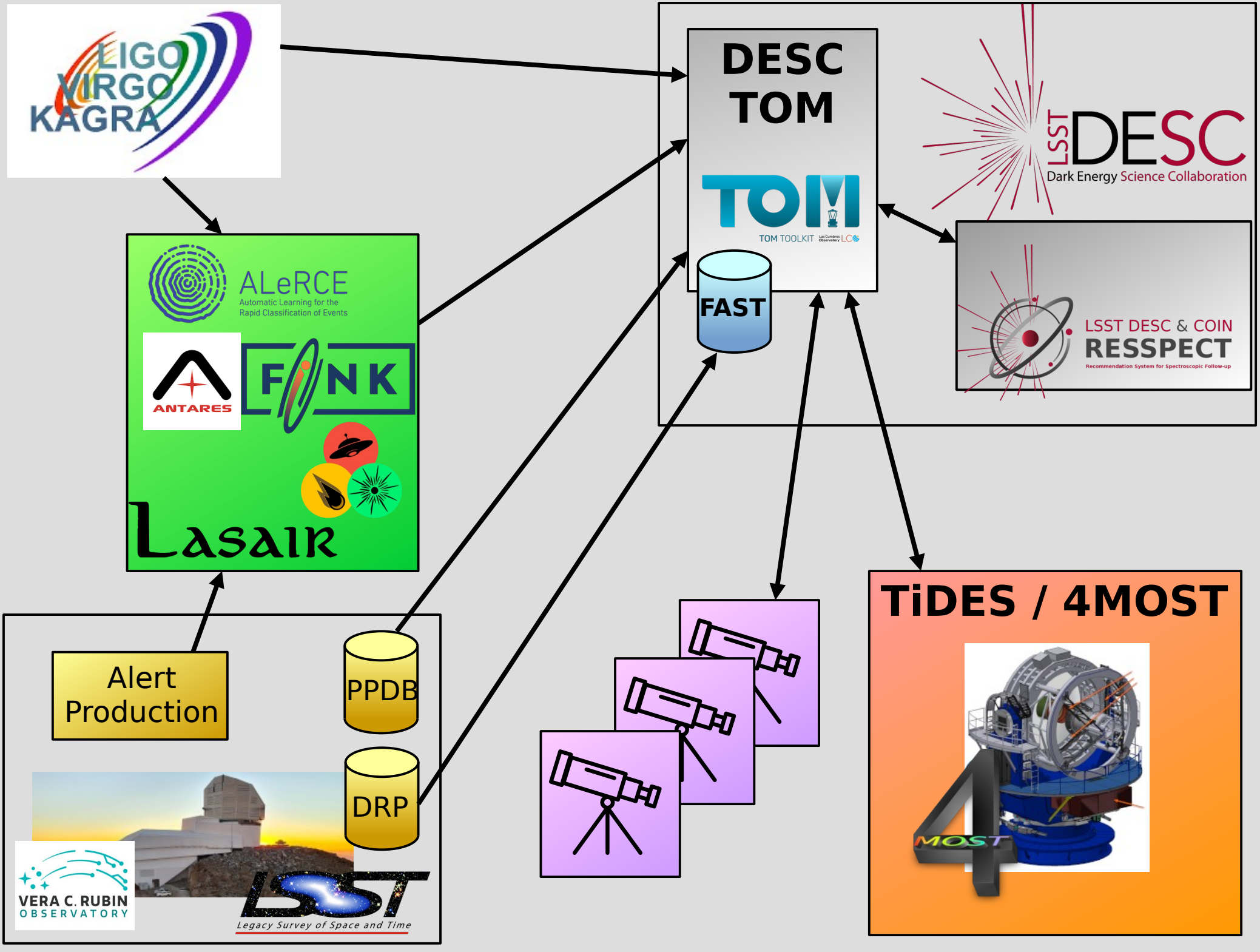
DESC Sprint Week

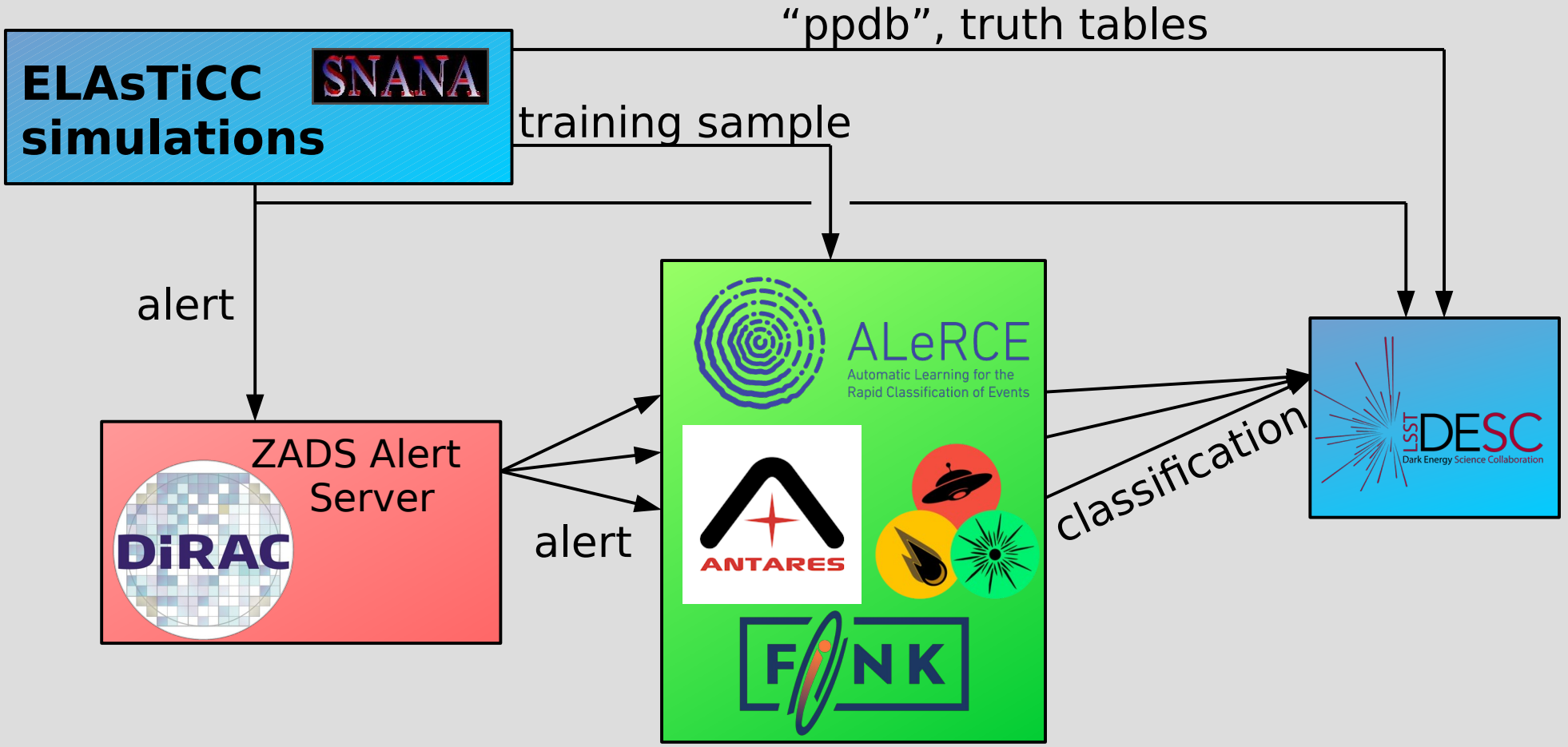
2024-10-28

What is ELAsTiCC?

- Catalog-level simulation of 4 million transients and variables
  - SNANA simulations (R. Kessler)
  - 37 SNANA models*
  - 62 million “sources” (photometry points), 43 million with S/N≥5
  - 990 million “forced sources” (forced photometry points)

- Campaign to test LSST/DESC alert cycle (LSST Alerts → Brokers → DESC)
  - Trial run: ELAsTiCC1, 2022 Sep–2023 Jan
  - ELAsTICC2, 2023 Nov–Dec
  - Includes broker classification data in addition to SNANA simulations from 4–5 brokers.


LIGO
VIRGO
KAGRA
DESC
TOM
TOM
TOM TOOLKIT
FAST
LSST DESC
Dark Energy Science Collaboration
ALeRCE
Automatic Learning for the
Rapid Classification of Events
ANTARES
FINK
Lasair
LSST DESC & COIN
RESSPECT
Recommendation System for Spectroscopic Follow-up
Alert
Production
PPDB
DRP
VERA C. RUBIN
OBSERVATORY
LSST
Legacy Survey of Space and Time
TiDES / 4MOST
4MOST


ELAsTiCC simulations
SNANA
“ppdb”, truth tables
training sample
alert
ZADS Alert Server
DiRAC
alert
ALeRCE
Automatic Learning for the Rapid Classification of Events
ANTARES
FINK
classification
LSST DESC
Dark Energy Science Collaboration

Data Formats Available

- SNANA FITS files (+CSV for object truth)
- Parquet files
- Tarballs of AVRO alerts
- Web API / PostGreSQL Database on the DESC TOM

SNANA FITS Files

- Use if you already know or want to learn the SNANA format.
- Use with the `lib_elasticc2` python library.

On NERSC: `/global/cfs/cdirs/desc-td/ELASTICC2`

- One subdirectory for each model
- 40 `_HEAD.FITS` and `_PHOT.FITS` (gzipped) files in each subdirectory
- CSV file in subdirectory with object truth: `ELASTICC2_FINAL_{model}.DUMP`

Parquet Files

On NERSC:
`/global/cfs/cdirs/desc-td/ELASTICC2_parquet`

Parquet schema includes many (but not all) of the fields from the HEAD file and object truth file as scalars, and several fields from the PHOT files as lists. (See demo.)

Files are divided by SNANA model. SNID is the “key” field that identifies a given object.

DESC Tom Web API / SQL Database

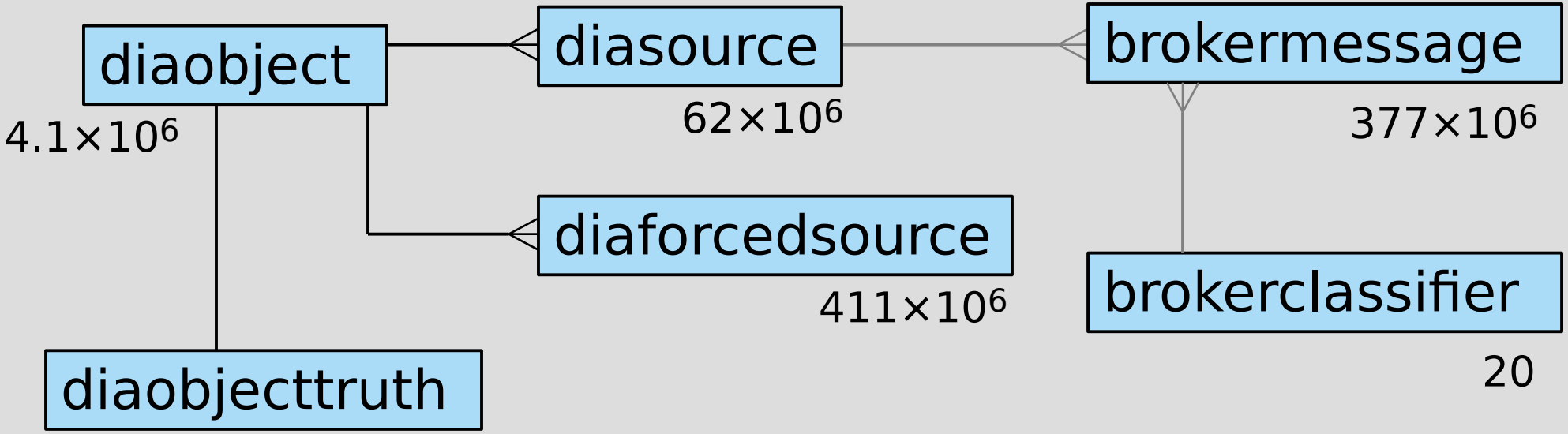
https://desc-tom.lbl.gov

- You need a username/password on the TOM. (Ask Rob; give him the username you want and the email address associated with the TOM account.)
- There are some web API calls to get information (demo shortly). If there are others that you think would be useful, talk to Rob.
- There is an interface that lets you query the PostgreSQL database directly:

  https://github.com/LSSTDESC/tom_desc/blob/main/sql_query_tom_db.ipynb

Structure of ELAsTiCC2 database

Note: all tables have `elasticc2_` prepended



gentypeofclassid

classidofgentype

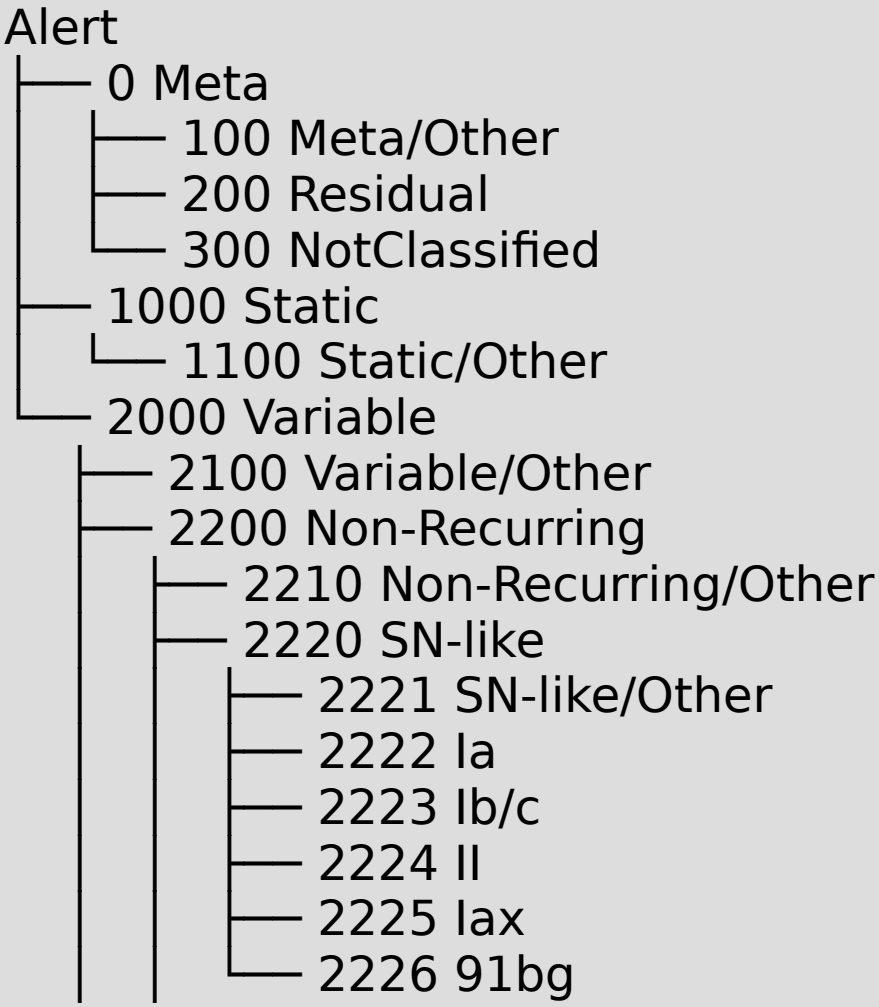
- Brokers provided classifications as classID, using the ELAsTiCC2 taxonomy.
- Truth tables have SNANA gentype.
- The *of* tables provide the mapping.

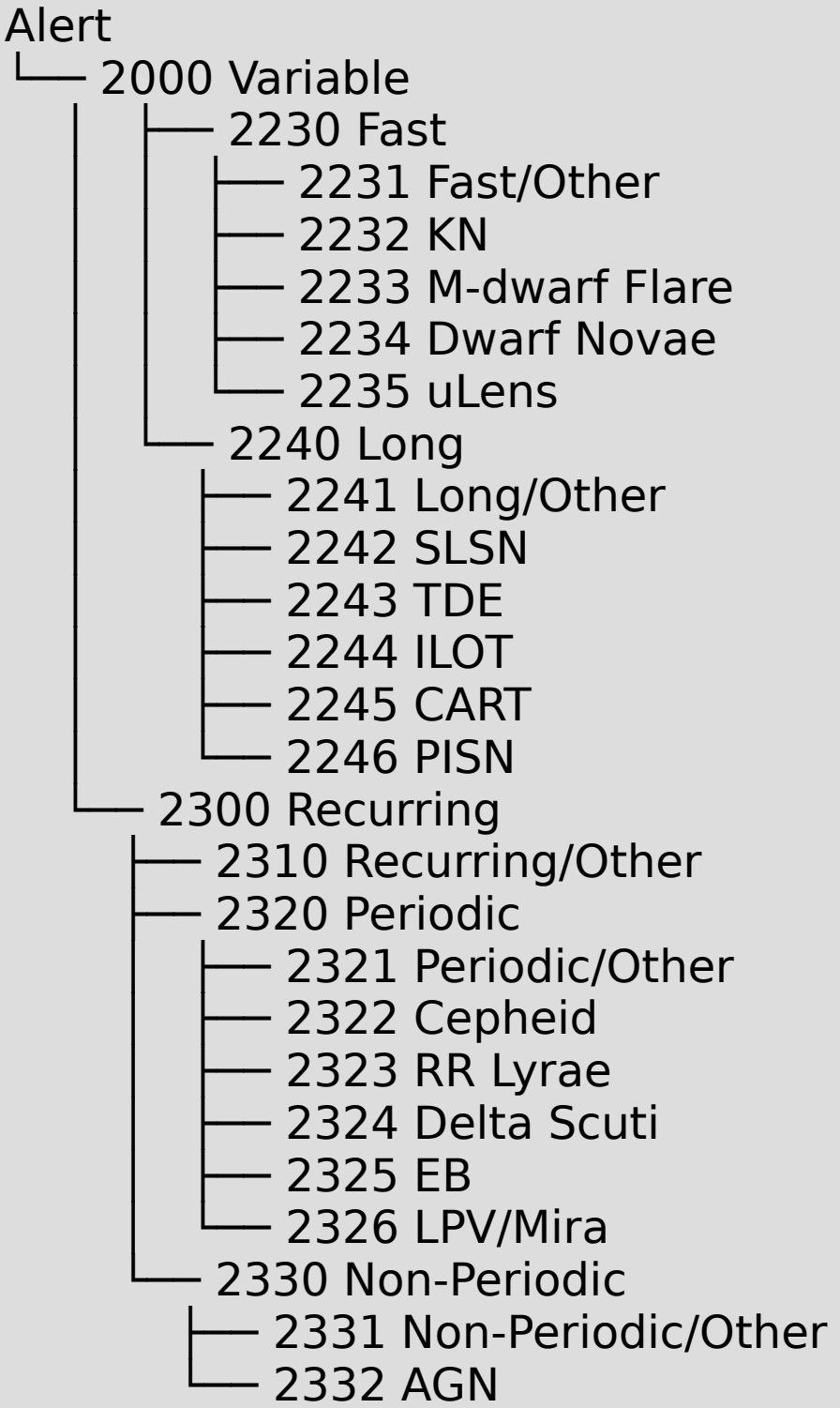
Table definitions:

https://github.com/LSSTDESC/elasticc/blob/lib_elasticc2/jupyter/sprint_week_2024oct/elasticc2_schema.txt

ELAsTiCC2 Taxonomy

https://github.com/LSSTDESC/elasticc/blob/main/taxonomy/taxonomy.ipynb





Tarballs of AVRO Alerts

- This is a terrible format for reading the data: redundant, inefficient.
- Only use these if you are **specifically** dealing with realtime processing of actual alerts.
- **No, really, don't use this format.**

Alerts tarred up by MJD (all object types mixed):
`/global/cfs/cdirs/desc-td/ELASTICC2_ALERTS`

You will need the ELAsTiCC2 alert schema:

https://github.com/LSSTDESC/elasticc/tree/elasticc2/alert_schema

`elasticc.v0_9_1.alert.avsc` is the schema of an alert; it refers to other files in that directory.

Getting set up for the tutorials at NERSC

Setting yourself up to run the DESC Jupyter kernels:

```
source /global/common/software/lsst/common/miniconda/kernels/setup.sh
```

→ This will set up the standard DESC kernels for use in jupyter. You may want to rerun this every so often to get updates!

(It creates directories and files underneath ~/.local/share/jupyter/kernel)

Verify that you have them with:

```
jupyter kernelspec list
```

→ You should see a number of **desc-*** kernels, including the most basic **desc-python** and **desc-td-env-dev**

https://confluence.slac.stanford.edu/display/LSSTDESC/Using+Jupyter+at+NERSC#UsingJupyteratNERSC-setup

Running the desc-td environment at NERSC

Jupyter is great for tutorials and demos and mucking about, but is not where you want to be doing serious development.

Get yourself into the desc-td environment on your shell at NERSC by running:

```
source /global/common/software/lsst/common/miniconda/setup_current_python.sh
```

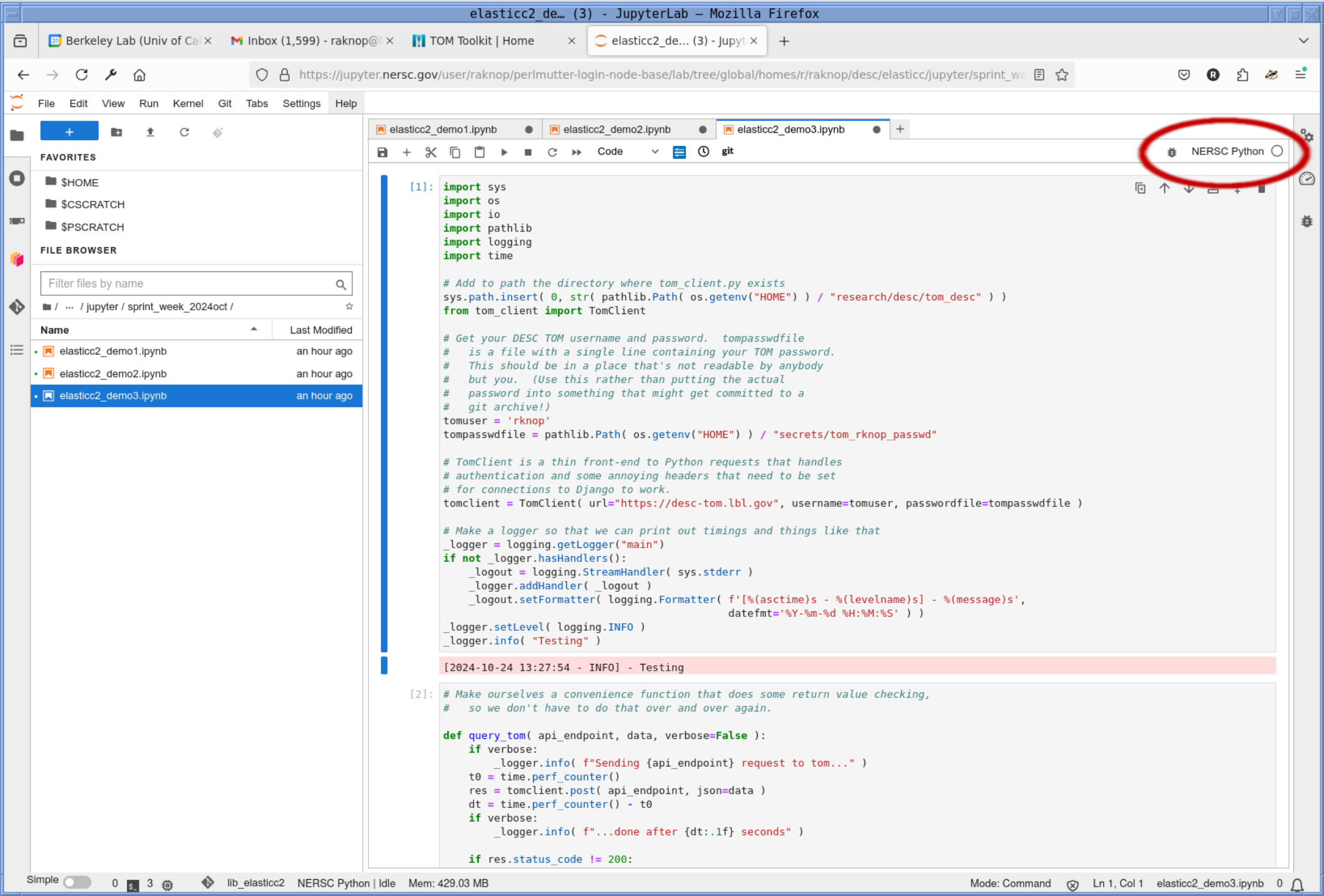
or (for the Time Domain environment):

```
source /global/cfs/cdirs/lsst/groups/TD/setup_td.sh
```

or

```
source /global/cfs/cdirs/lsst/groups/TD/setup_td_dev.sh
```

Change your kernel to desc-td-env-dev



Finding information about ELAsTiCC:

- ELAsTiCC public web page, ideally with full documentation:
  https://portal.nersc.gov/cfs/lsst/DESC_TD_PUBLIC/ELASTICC/
- Slack: #elasticc-comms (ping me, Rick, Alex M., etc.)
- The DESC TOM:
  - https://desc-tom.lbl.gov (ping me for an acocunt)
  - https://github.com/LSSTDESC/tom_desc
  - ELAsTiCC Metrics:
    - https://desc-tom.lbl.gov/elasticc2/
    - API:

https://github.com/LSSTDESC/elasticc_metrics/blob/main/elasticc2_rest_metric_demo.ipynb

    - SQL:

https://github.com/LSSTDESC/tom_desc?tab=readme-ov-file#accessing-the-tom
https://github.com/LSSTDESC/tom_desc/blob/main/sql_query_tom_db.ipynb