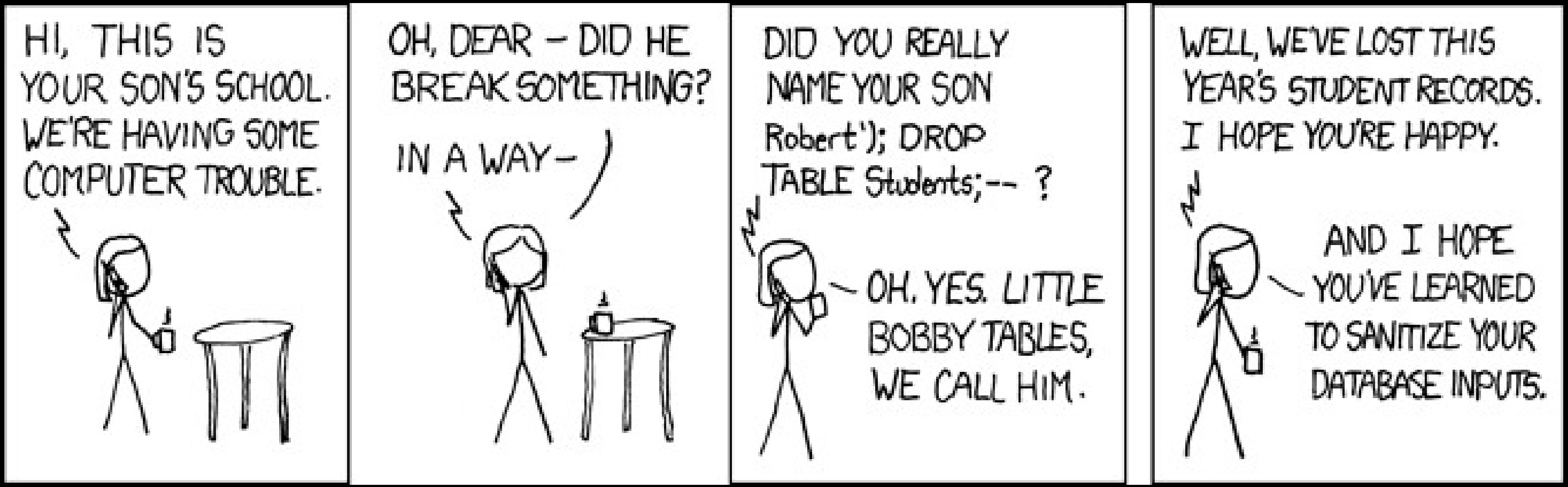
Embrace the database - manage, query and combine your datasets in the cloud with geoslurp

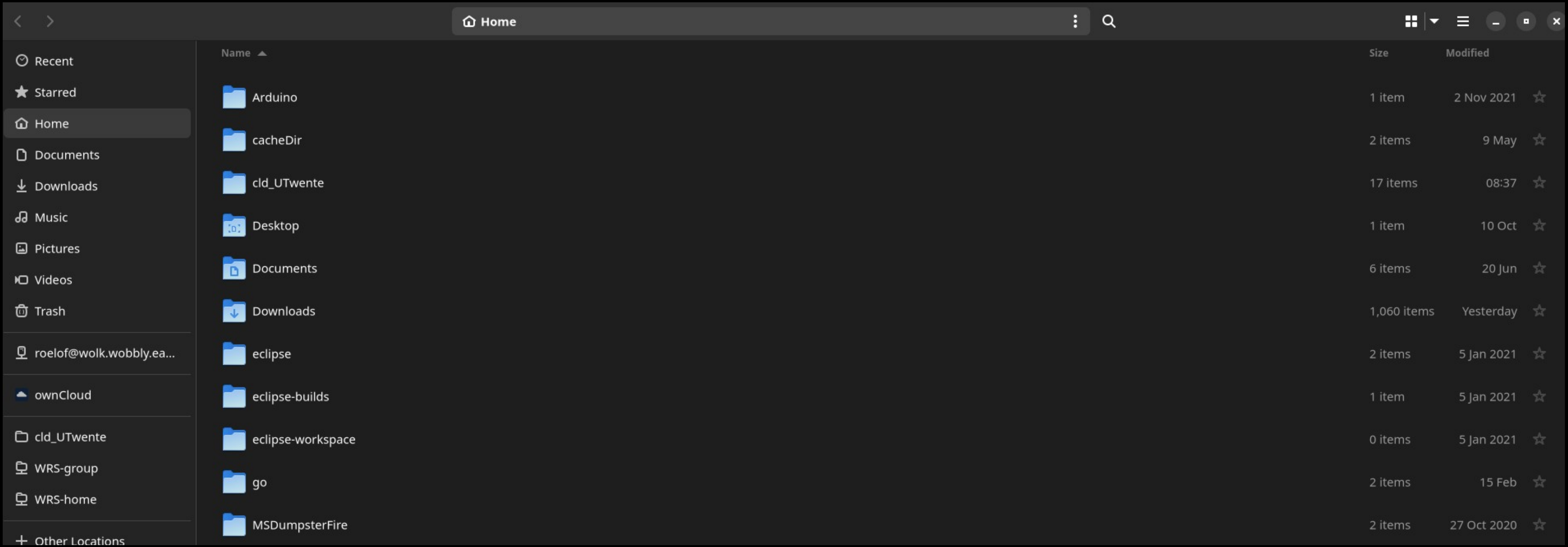

HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

source:XKCD

Roelof Rietbroek
Water Resources Department
Faculty of Geo-Information Science and Earth Observation

UNIVERSITY OF TWENTE. | ITC

Most of us probably use a file system as a data catalogue

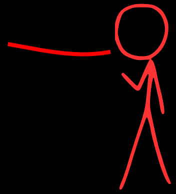


Name	Size	Modified
Arduino	1 item	2 Nov 2021
cacheDir	2 items	9 May
cld_UTwente	17 items	08:37
Desktop	1 item	10 Oct
Documents	6 items	20 Jun
Downloads	1,060 items	Yesterday
eclipse	2 items	5 Jan 2021
eclipse-builds	1 item	5 Jan 2021
eclipse-workspace	0 items	5 Jan 2021
go	2 items	15 Feb
MSDumpsterFire	2 items	27 Oct 2020

‘Geoslurp’ philosophy

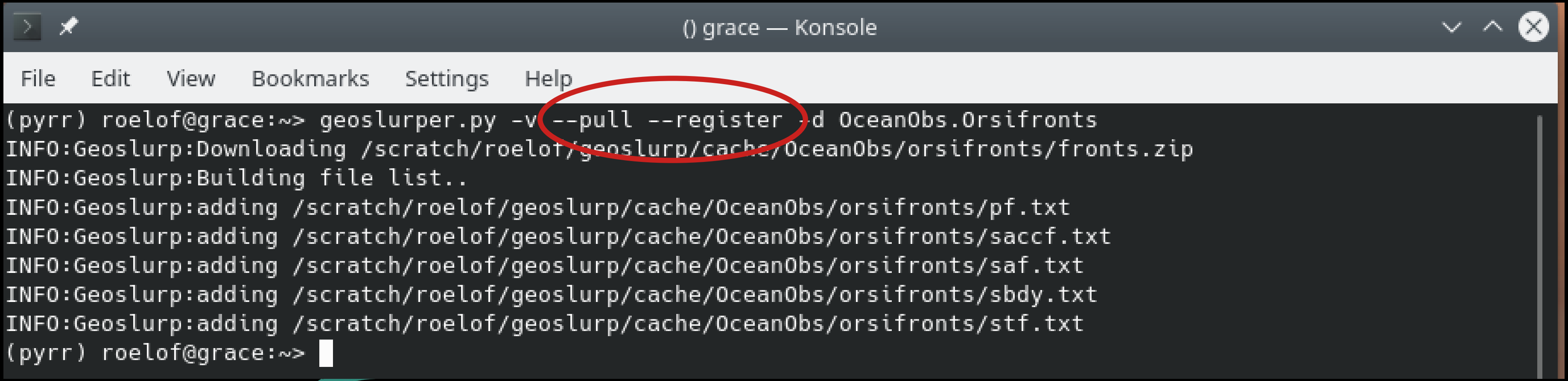
- Open source python module on github https://github.com/strawpants/geoslurp
  - Download/update datasets
  - Uses PostgreSQL+PostGIS as underlying database for META DATA
  - Sharing per default
- Other ‘Clients’ (not necessarily part of geoslurp) can then access the database
  - Use standard protocol
  - Make queries (spatial queries!)
  - Find storage-locations of datasets
  - Joins (e.g. mix and match datasets with overlapping time periods, locations,..)
  - …
- Target audience:
  - Scientists, small working groups

Does it makes sense to store a terabytes of science data in a database?



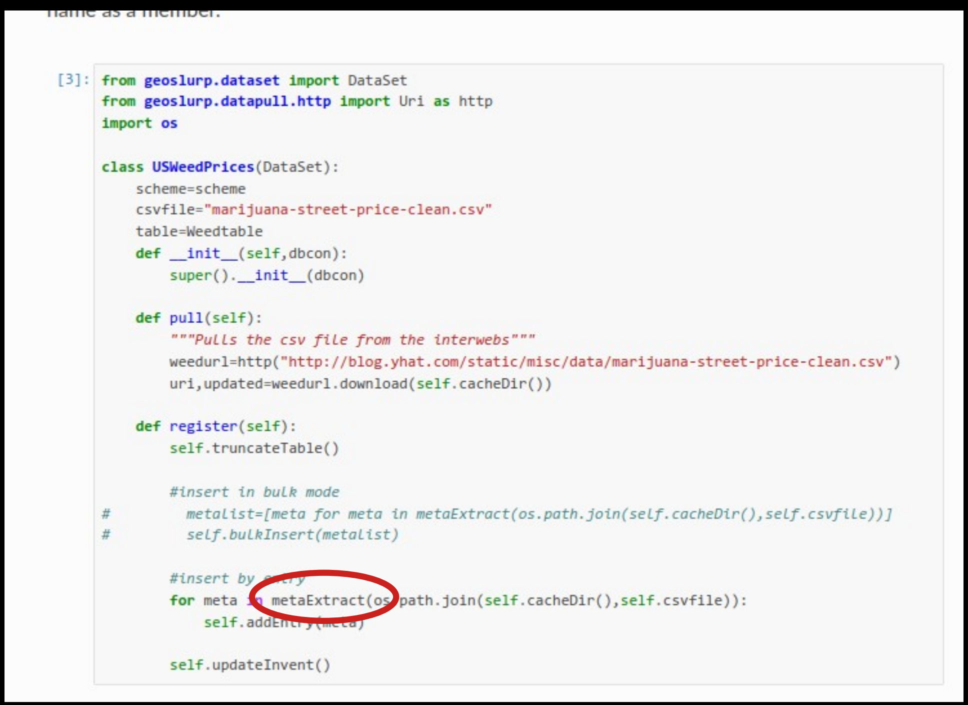
Make downloading easier..

- Download from public http/ftp/webdav/..
- Also allow users to store/reuse authentication (e.g. login data for copernicus cmems, lies encrypted on the server)
- Avoid superflous downloading (i.e. allow update)



```
(pyrr) roelof@grace:~> geoslurper.py -v --pull --register -d OceanObs.Orsifronts
INFO:Geoslurp:Downloading /scratch/roelof/geoslurp/cache/OceanObs/orsifronts/fronts.zip
INFO:Geoslurp:Building file list..
INFO:Geoslurp:adding /scratch/roelof/geoslurp/cache/OceanObs/orsifronts/pf.txt
INFO:Geoslurp:adding /scratch/roelof/geoslurp/cache/OceanObs/orsifronts/saccf.txt
INFO:Geoslurp:adding /scratch/roelof/geoslurp/cache/OceanObs/orsifronts/saf.txt
INFO:Geoslurp:adding /scratch/roelof/geoslurp/cache/OceanObs/orsifronts/sbdy.txt
INFO:Geoslurp:adding /scratch/roelof/geoslurp/cache/OceanObs/orsifronts/stf.txt
(pyrr) roelof@grace:~>
```

Dynamically create datasets in python

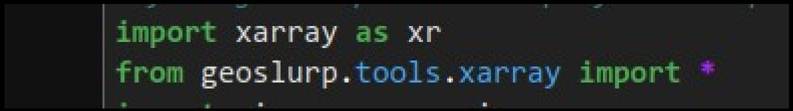


name as a member.

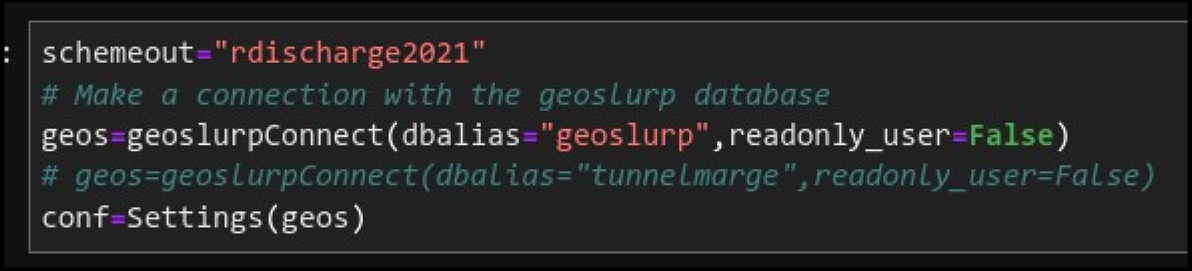
```
[3]: from geoslurp.dataset import DataSet
     from geoslurp.datapull.http import Uri as http
     import os

     class USWeedPrices(DataSet):
         scheme=scheme
         csvfile="marijuana-street-price-clean.csv"
         table=Weedtable
         def __init__(self,dbcon):
             super().__init__(dbcon)

         def pull(self):
             """Pulls the csv file from the interwebs"""
             weedurl=http("http://blog.yhat.com/static/misc/data/marijuana-street-price-clean.csv")
             uri,updated=weedurl.download(self.cacheDir())

         def register(self):
             self.truncateTable()

             #insert in bulk mode
     #         metalist=[meta for meta in metaExtract(os.path.join(self.cacheDir(),self.csvfile))]
     #         self.bulkInsert(metalist)

             #insert by entry
             for meta in metaExtract(os.path.join(self.cacheDir(),self.csvfile)):
                 self.addEntry(meta)

             self.updateInvent()
```

- Idea: derive from Base class
  - User implement pull & register member function
- More advanced derived classes exist (e.g. loading ogr files)

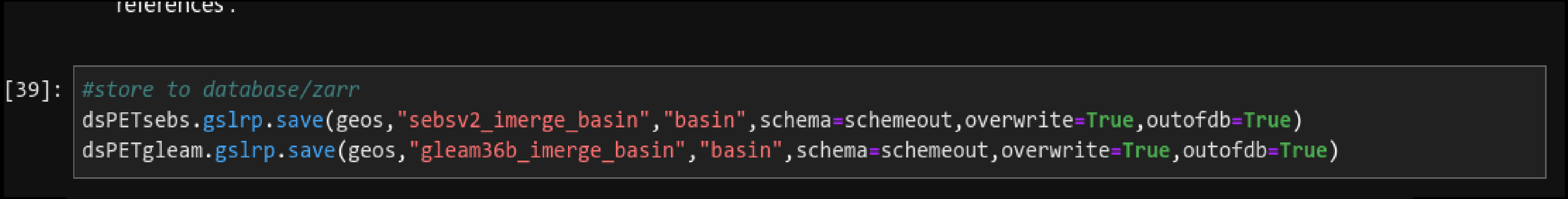
geoslurp has an xarray accessor



```
import xarray as xr
from geoslurp.tools.xarray import *
```

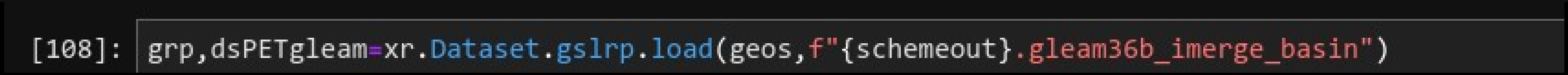


```
: schemeout="rdischarge2021"
  # Make a connection with the geoslurp database
  geos=geoslurpConnect(dbalias="geoslurp",readonly_user=False)
  # geos=geoslurpConnect(dbalias="tunnelmarge",readonly_user=False)
  conf=Settings(geos)
```



references :

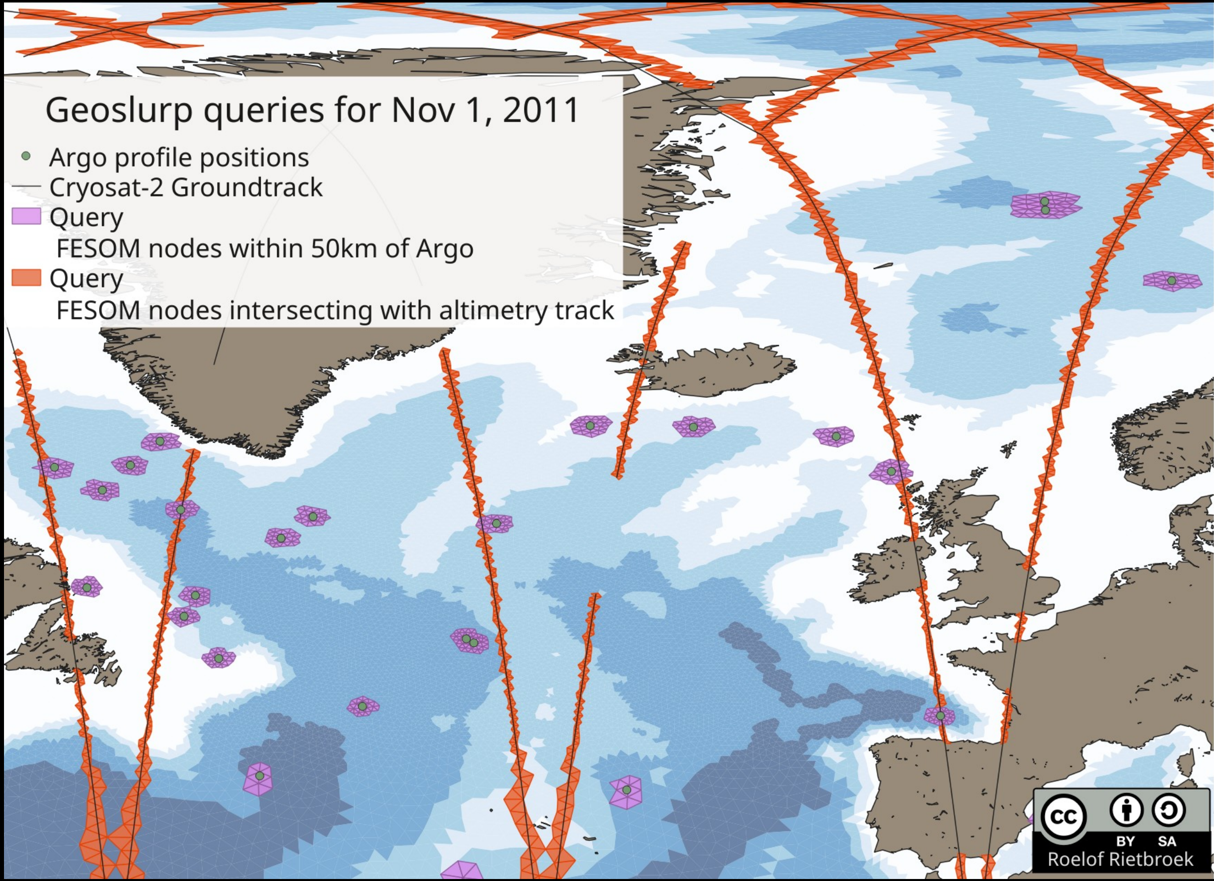
```
[39]: #store to database/zarr
      dsPETsebs.gslrp.save(geos,"sebsv2_imerge_basin","basin",schema=schemeout,overwrite=True,outofdb=True)
      dsPETgleam.gslrp.save(geos,"gleam36b_imerge_basin","basin",schema=schemeout,overwrite=True,outofdb=True)
```



```
[108]: grp,dsPETgleam=xr.Dataset.gslrp.load(geos,f"{schemeout}.gleam36b_imerge_basin")
```

Example: storing unstructured grids from models

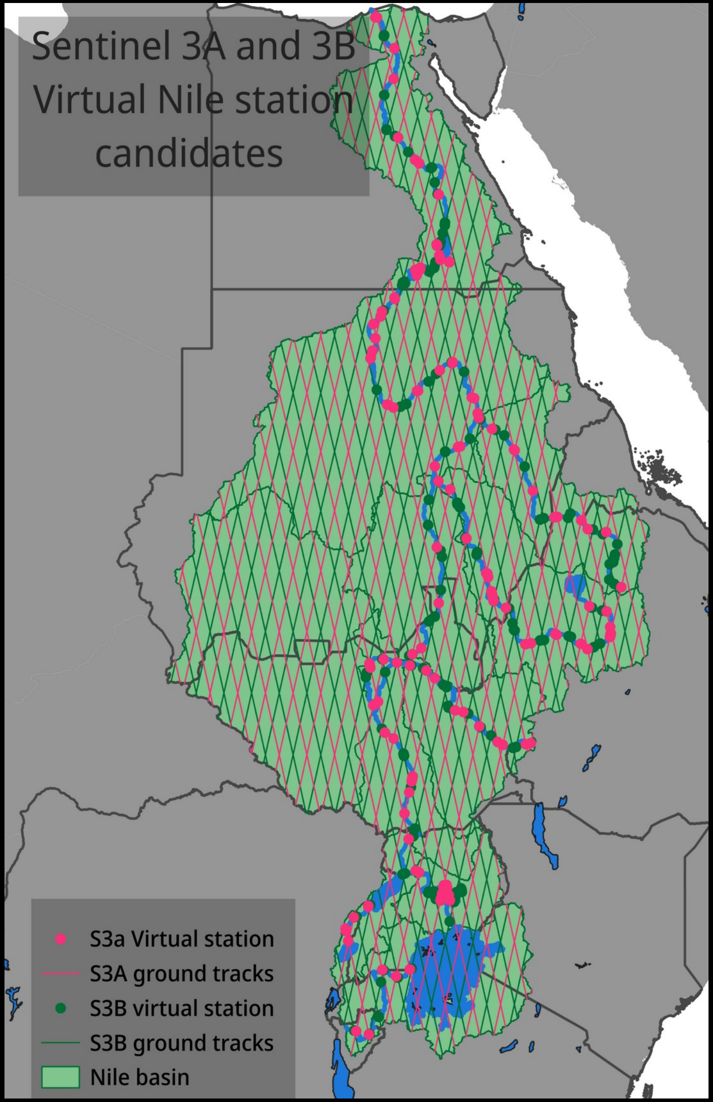
- FESOM model: triangular grid on different levels
- Intersecting elements with Radar altimetry, Argo


Geoslurp queries for Nov 1, 2011
● Argo profile positions
— Cryosat-2 Groundtrack
Query
FESOM nodes within 50km of Argo
Query
FESOM nodes intersecting with altimetry track
BY SA
Roelof Rietbroek

UNIVERSITY OF TWENTE. ITC

Example: virtual stations

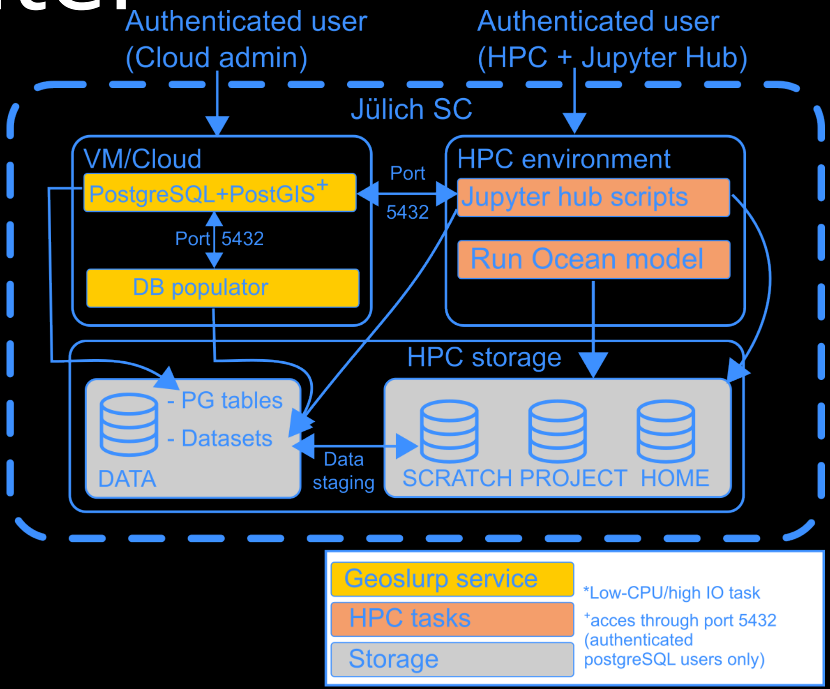
- Where does Sentinel 3A/B cross a river?
- More criteria possible (e.g. vicinity to discharge station,..)



UNIVERSITY OF TWENTE. | ITC

Deploy close to your computing-center

- We had a VM version on the HPC system in Jülich
- There is a version running on ITC crib



UNIVERSITY OF TWENTE. | ITC

Conclusion: geoslurp = a data catalogue on steroids

- Interested? Visit https://github.com/strawpants/geoslurp
- At ITC? Contact me to get connected to the ITC catalogue
- Future?
  - Deploy: geoslurp fed DB’s as a microservice
  - Wrap geonode, wfs servers