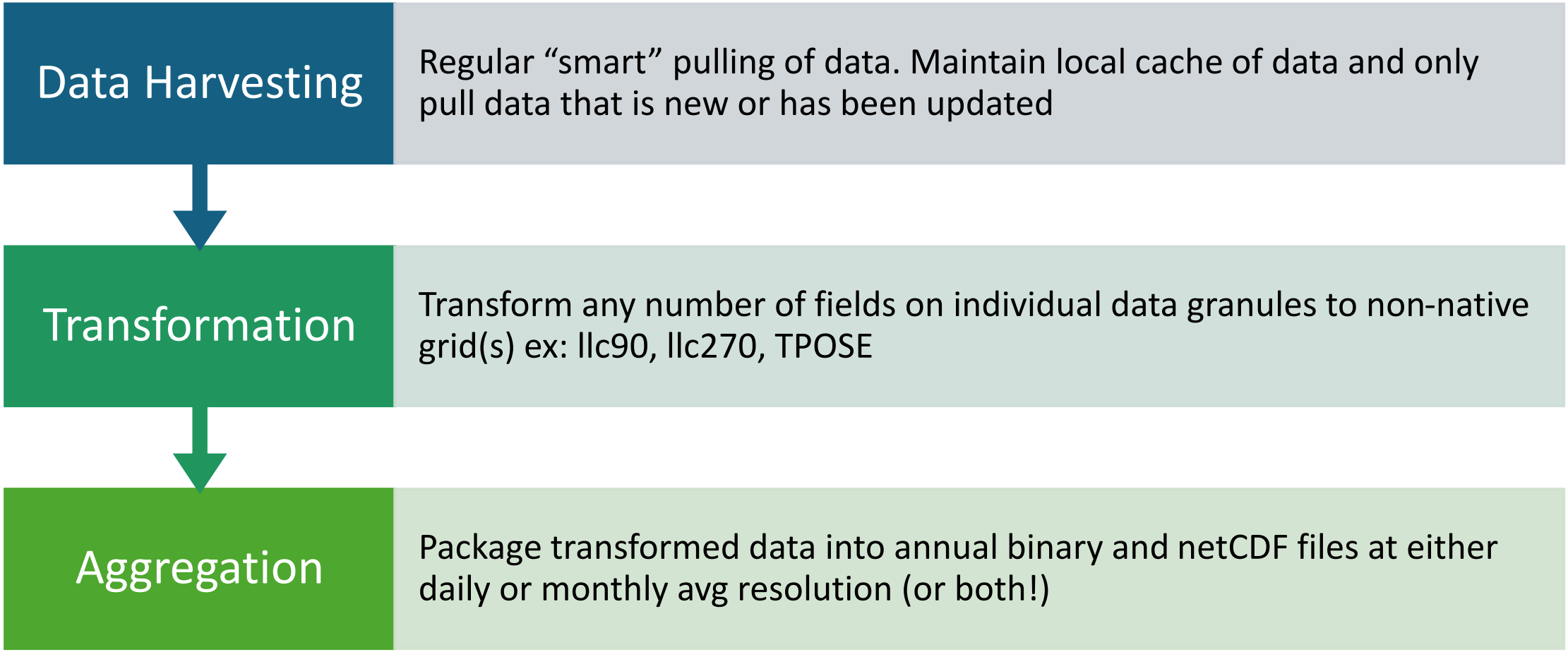
ECCO Observation Pipeline

Kevin Marlis, JPL

ECCO Observation Pipeline



Current supported datasets

SEA ICE CONCENTRATION

AMSR-2_OSI-408
G02202_V4
G10016_V2
SSMIS_OSI-430-a (daily and monthly)
SSMIS_OSI-450-a (daily and monthly)

SEA ICE THICKNESS

RDEFT4

SEA ICE TOTAL FREEBOARD

ATL20_V004 (daily and monthly)

SSH

SEA_SURFACE_HEIGHT_ALT_GRIDS_L4_2SATS_5DAY_6THDEG_V_JPL2205
ATL21_V003 (daily and monthly)

OBP

TELLUS_GRAC_L3_CSR_RL06_OCN_v04
TELLUS_GRAC-GRFO_MASCON_CRI_GRID_RL06.1_V3
TELLUS_GRFO_L3_CSR_RL06.2_OCN_v04

SST

AVHRR_OI-NCEI-L4-GLOB-v2.0 / v2.1
MODIS_AQUA_L3_SST_THERMAL_DAILY_9KM_DAYTIME_V2019.0

SSS

AQUARIUS_L3_SSS_SMI_MONTHLY_V5
L3_DEBIAS_LOCEAN_v8_q09 / q18
OISSS_L4_multimission_monthly_v2
SMAP_RSS_L3_SSS_SMI_MONTHLY_V4

Cases to consider…data sources

Source	Dataset	Harvester
PODAAC	AQUARIUS_L3_SSS_SMI_MONTHLY_V5	NASA CMR
	AVHRR_OI-NCEI-L4-GLOB-v2.0 / v2.1	
	MODIS_AQUA_L3_SST_THERMAL_DAILY_9KM_DAYTIME_V2019.0	
	OISSS_L4_multimission_monthly_v2	
	RDEFT4	
	SEA_SURFACE_HEIGHT_ALT_GRIDS_L4_2SATS_5DAY_6THDEG_V_JPL2205	
	SMAP_RSS_L3_SSS_SMI_MONTHLY_V4	
	TELLUS_GRAC_L3_CSR_RL06_OCN_v04	
	TELLUS_GRAC-GRFO_MASCON_CRI_GRID_RL06.1_V3	
	TELLUS_GRFO_L3_CSR_RL06.2_OCN_v04	
NSIDC	ATL20_V004_daily / monthly	
	ATL21_V003_daily / monthly	
	G02202_V4	NSIDC NOAA Scraper
	G10016_V2	
OSISAF	AMSR-2_OSI-408	OSISAF Thredds Scraper
	SSMIS_OSI-430-a_daily / monthly	
	SSMIS_OSI-450-a_daily / monthly	
CATDS	L3_DEBIAS_LOCEAN_v8_q09 / q18	CATDS Scraper

Cases to consider…data structure

- File formats (.nc, .h5, .gz)
- Groups in data
- Aggregated data
- Hemispherical data
- Variables being renamed midstream

Cases to consider…source projections

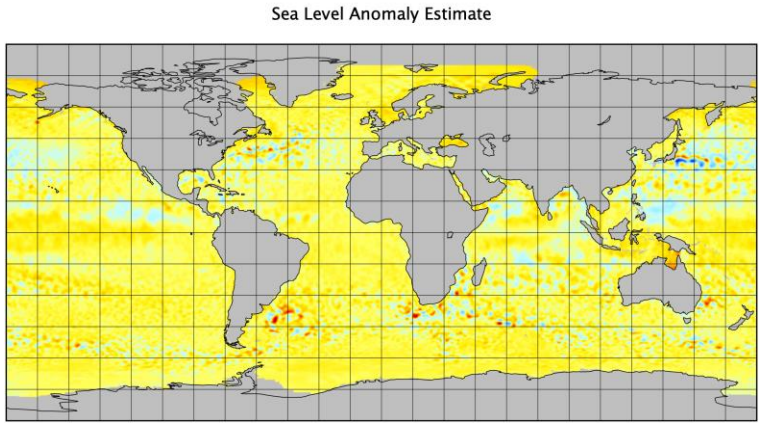
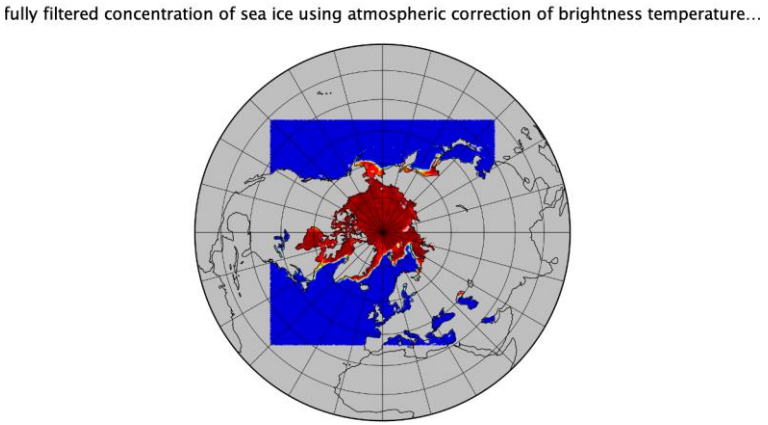
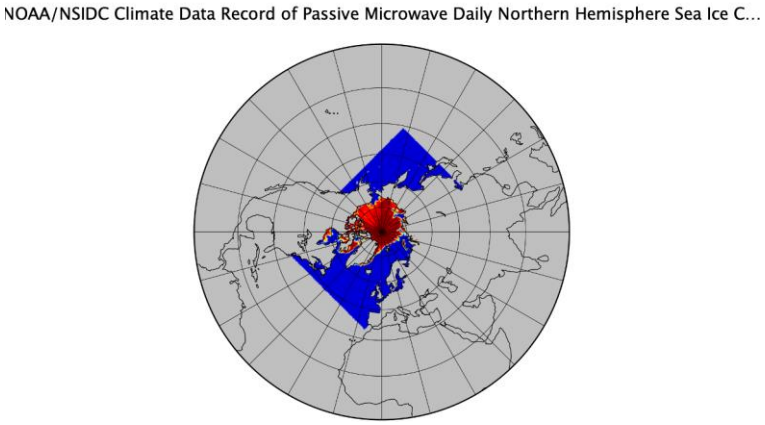


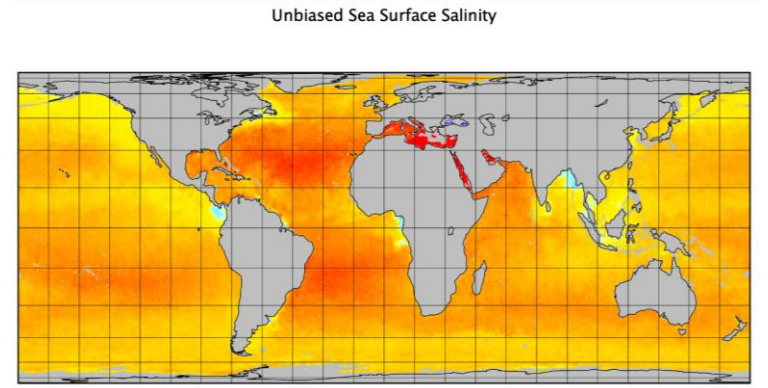
Plate Carree



NCIDS EASE-Grid 2.0



Polar Stereographic



Equal Area Cylindrical

Cases to consider…time

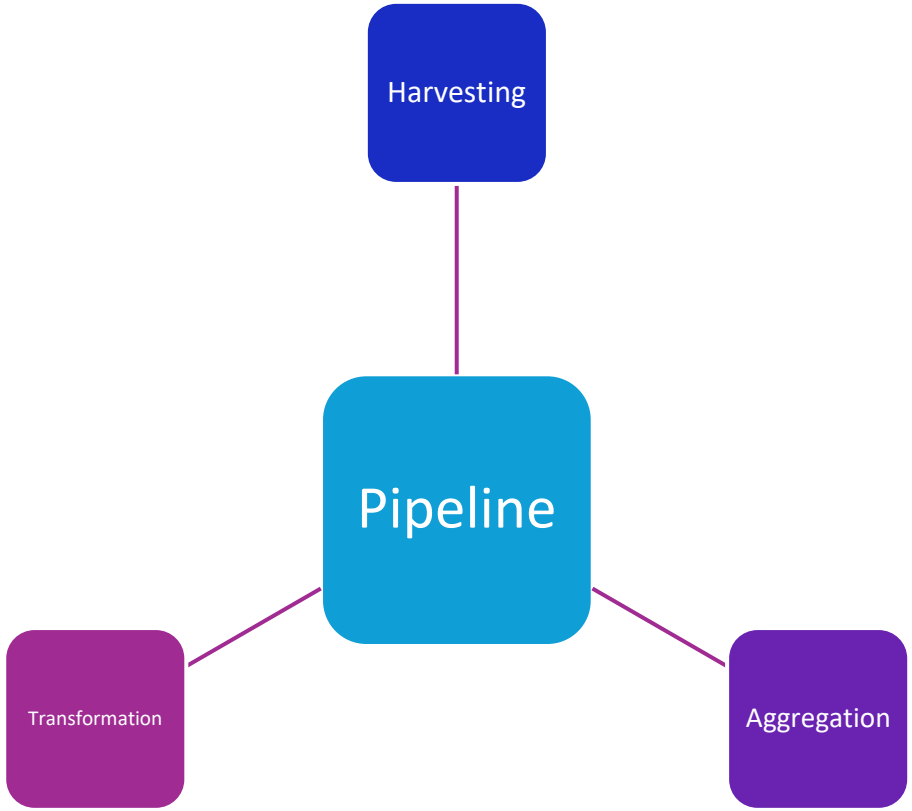
- Daily or Monthly resolution
- Daily resolution but not daily cadence
- Monthly resolution but daily cadence
- Date corresponds to *previous* 30 day average
- Date corresponds to +/- 30 days

Difficulties

- Building out software framework that supports all of this nuance *and* is extendable as *more* datasets get supported
- Handling scale of data:
  - Currently 119,323 individual data granules ingested in system with multiple fields per granule transformed to multiple grids
  - Currently 487,704 individual transformations in system
  - Avoid redundant work
  - Work in parallel where possible
- Building on top of work done by intern (me)

Part of the solution: modularizing the work

- Pipeline steps are generalized with specific implementations to account for individual unique cases
- Object oriented approach to framework design allows for strong amount of inheritance
- A solution for a specific case can be reused on any dataset where it is applicable
- Pipeline steps can be executed independently
- Easier to add support for unique situations
- Easier to fix bugs



Part of the solution: dataset configs

- Configs define the specifics for each generalized solution a dataset requires:
  - harvesting data
  - source grid projection
  - fields to transform and their metadata
  - set of pre or post processing functions to be applied to a field (ex: unit conversion, masking, etc)

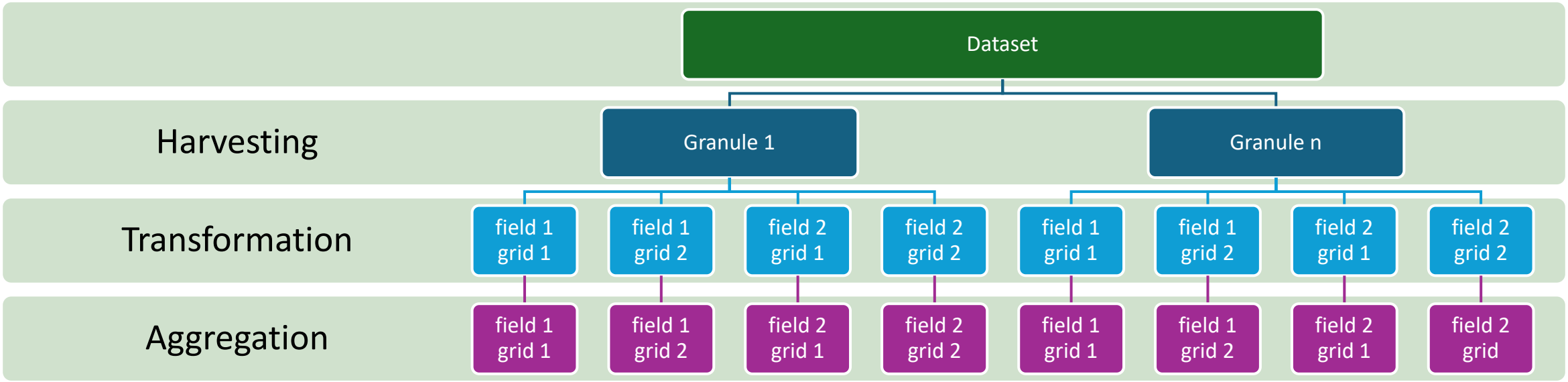
```yaml
ds_name: ATL21_V003_daily # Name for dataset
start: "19800101T00:00:00Z" # yyyymmddThh:mm:ssZ
end: "NOW" # yyyymmddThh:mm:ssZ for specific date or "NOW" for...now

# Provider specifications
harvester_type: cmr
cmr_concept_id: C2737912334-NSIDC_ECS
filename_date_fmt: "%Y%m%d" #20200701
filename_date_regex: '\d{8}'
provider: "n5eil01u.ecs.nsidc"

# Metadata
data_time_scale: "daily" # daily or monthly
mapping_operation: 'nanmean'
hemi_pattern:
  north: "ATL21-01"
  south: "ATL21-02"
fields:
  - name: mean_ssha
    long_name: Monthly mean sea surface height anomalies
    standard_name: mean_ssha
    units: "meters"
    pre_transformations: [] # List of functions to call on the DataSet before transfor
    post_transformations: [] # List of functions to call on the DataArrays after trans
```

Part of the solution: Solr

- Apache Solr search platform (metadata database)
- Tracks *state* of pipeline: what's been done and what needs doing
- Each step of pipeline wraps the work with Solr queries and updates
- Aggregation step produces "provenance" JSON files

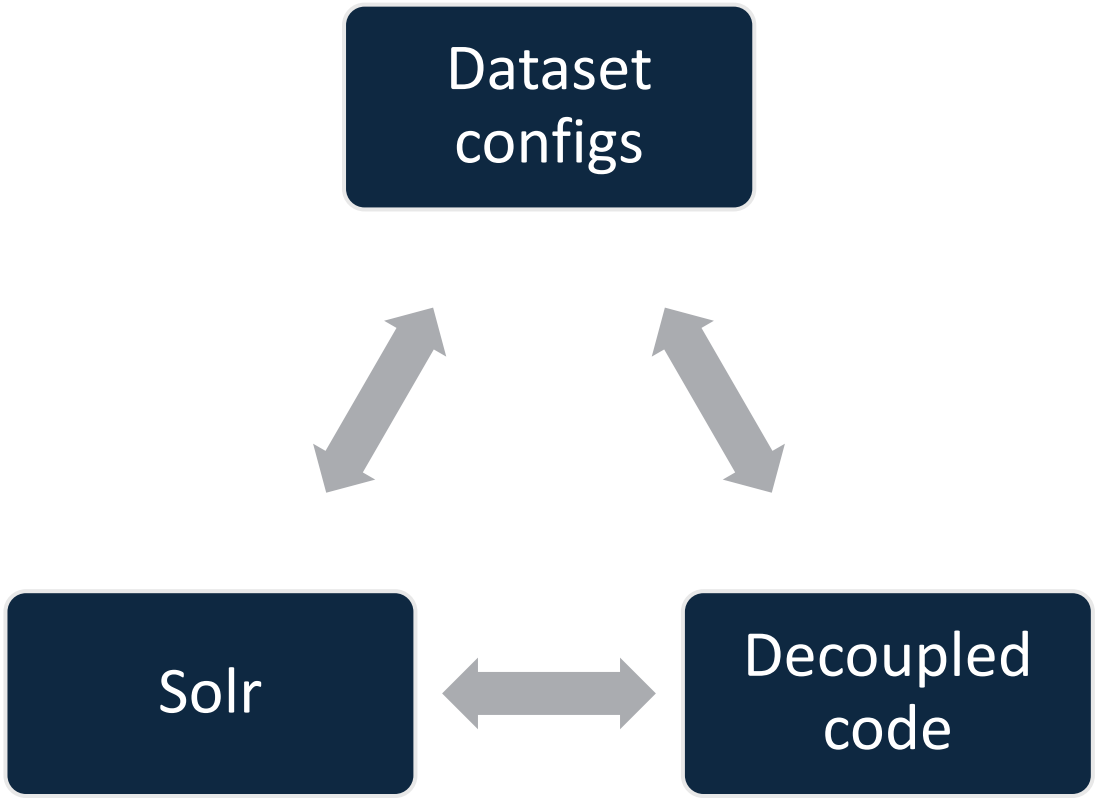


Part of the solution: Solr

- Apache Solr search platform (metadata database)
- Tracks *state* of pipeline: what's been done and what needs doing
- Each step of pipeline wraps the work with Solr queries and updates
- Aggregation step produces "provenance" JSON files


```
{
  "type_s":"granule",
  "date_s":"2023-08-01T00:00:00Z",
  "dataset_s":"ATL21_V003_monthly",
  "filename_s":"ATL21-02_20230801003256_06392001_003_01.h5",
  "source_s":"https://n5eil01u.ecs.nsidc.org/DP5/ATLAS/ATL21.003/2023.08.01/ATL21-02_20230801003256_06392001_003_01.h5",
  "modified_time_dt":"2024-03-08T00:00:00Z",
  "checksum_s":"5dcbdf19ab99b68d62236bee2904f98a",
  "pre_transformation_file_path_s":"/Users/marlis/Developer/ECCO/ecco_output/ATL21_V003_monthly/harvested_granules/2023/
  "harvest_success_b":true,
  "file_size_l":3625615,
  "download_time_dt":"2024-03-11T00:00:00Z",
  "id":"52a3cf64-1b4d-4d4b-888c-78799088d1da",
  "_version_":1793265829688639490},
```

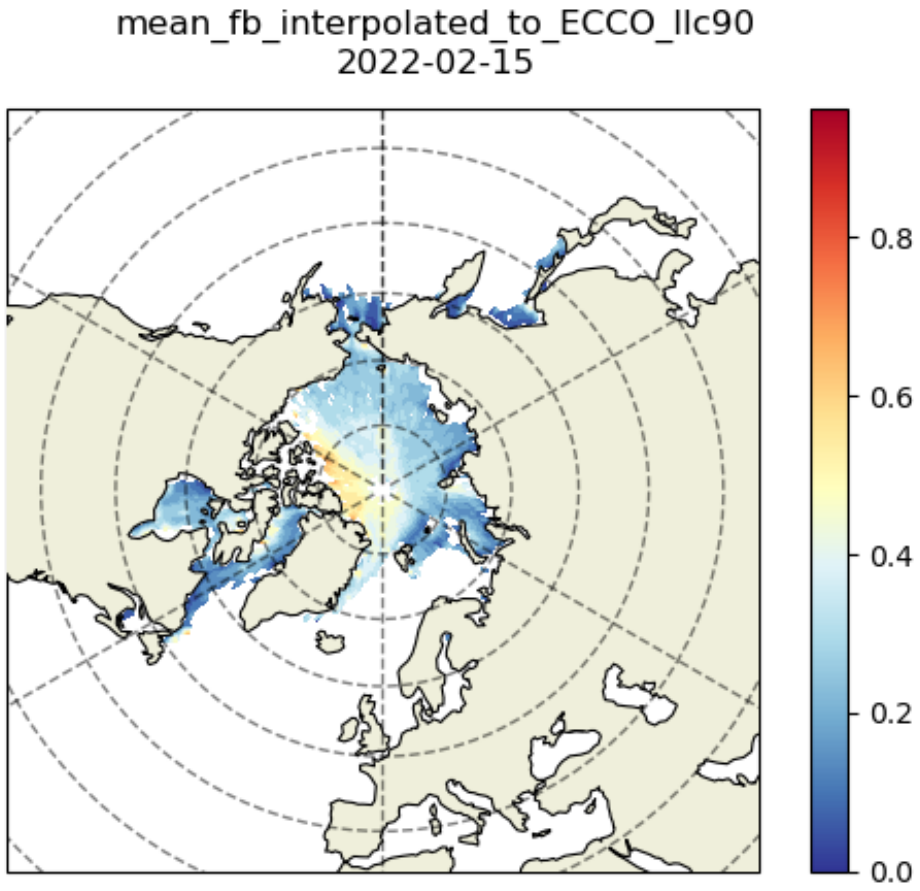
Putting it all together



- Use configs to control the specifics of a given dataset
- Use Solr to track the state of the pipeline both as a whole and for each individual granule
- User decides which steps to execute, Solr and dataset config efficiently handle the rest

Adding a new dataset

- Create a new config and fill in values
  - Looking at a sample granule
  - Looking at dataset documentation
  - Determining harvesting specifics
- Harvest! (But maybe start with a tight start/end date range in the config)
- Create a new test notebook for the dataset
  - Quick look at validating the transformation of a single granule
  - VERY handy for debugging projection information in configs
- If the results of the notebook look good, let it rip on everything!



State of the pipeline

- *In development*: an automated script to digest what is different about the state of the pipeline from week to week
  - Quick high level look at what work has been done
- Ongoing maintenance cycle:
  - Run EVERYTHING weekly
  - Deprecate older versions datasets as new versions are released
  - Add new grids (ASTE)
- Move to the cloud?

Try it out!

- https://github.com/ECCO-GROUP/ECCO-obs-pipeline
- Clone repo, download and setup Solr, try it out!
- Let Ian and I know of any issues or feature requests via github or email