

Interface for Observation Data Access Status Update

23 January 2020

Interface for Observation Data Access (IODA)



UFO/JEDI requires access to observation data

Separation of concerns: isolate science from data storage

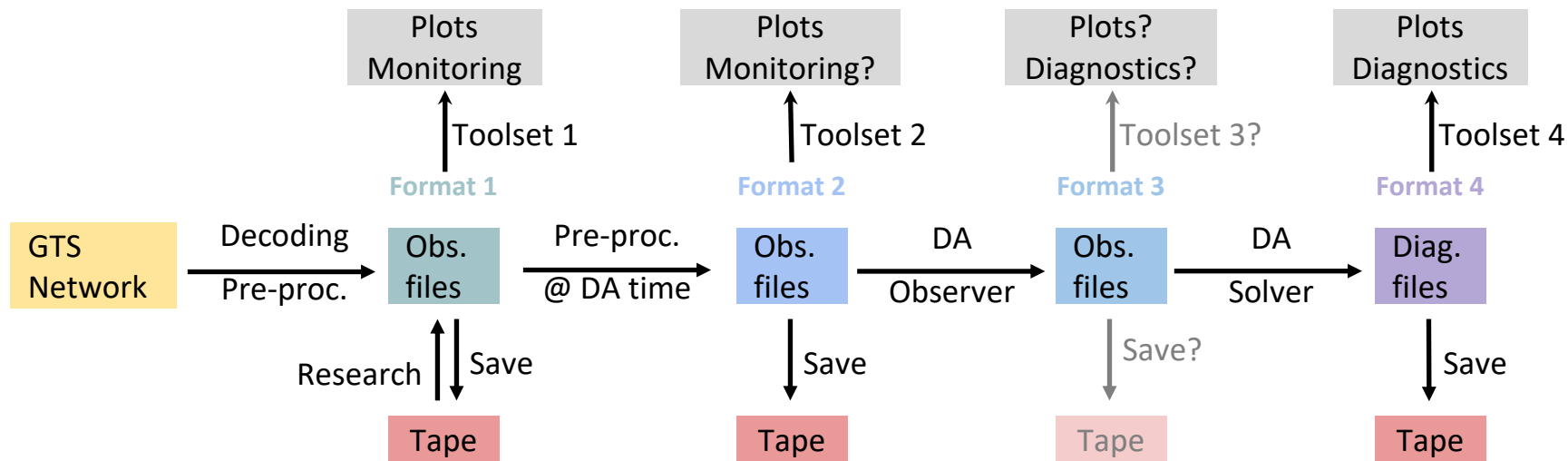
Three levels:

- Long term storage (historic database, R2D2)
- Files on disk (one DA cycle)
- In memory handling of observations

Two environments:

- Plotting, analyzing, verifying on workstation
- DA and other HPC applications (MPI, threads, GPUs...)

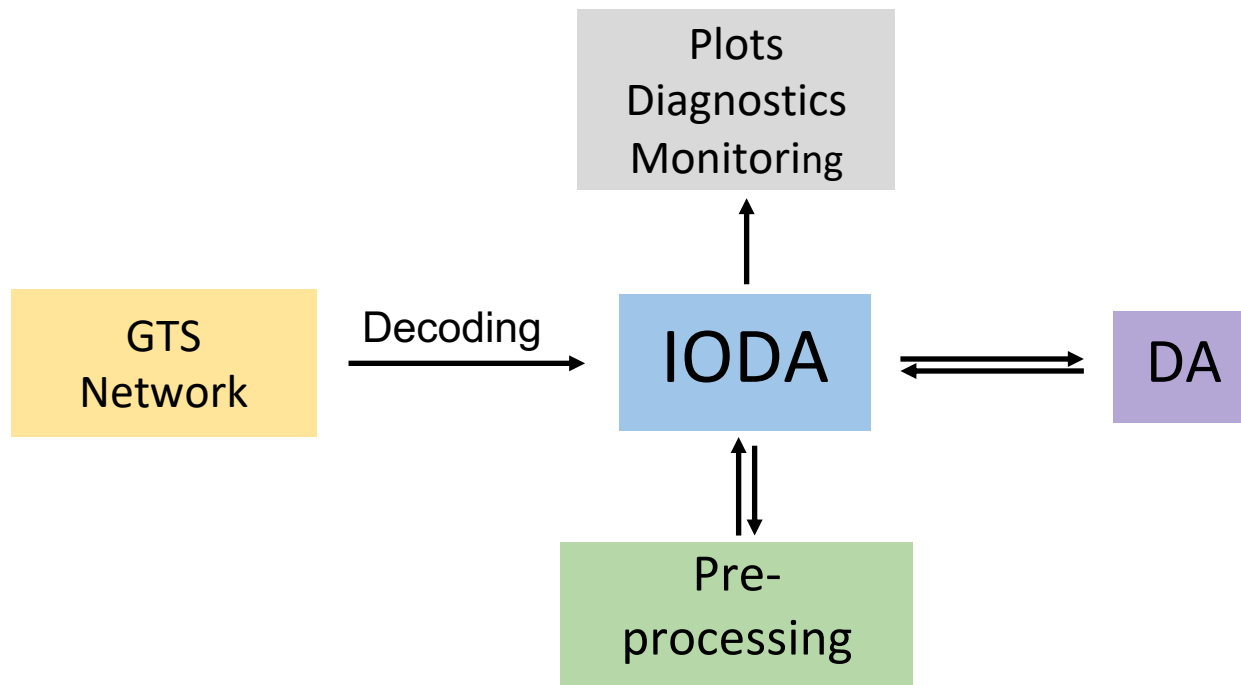
Typical Observation Data Flow



This is not sustainable:

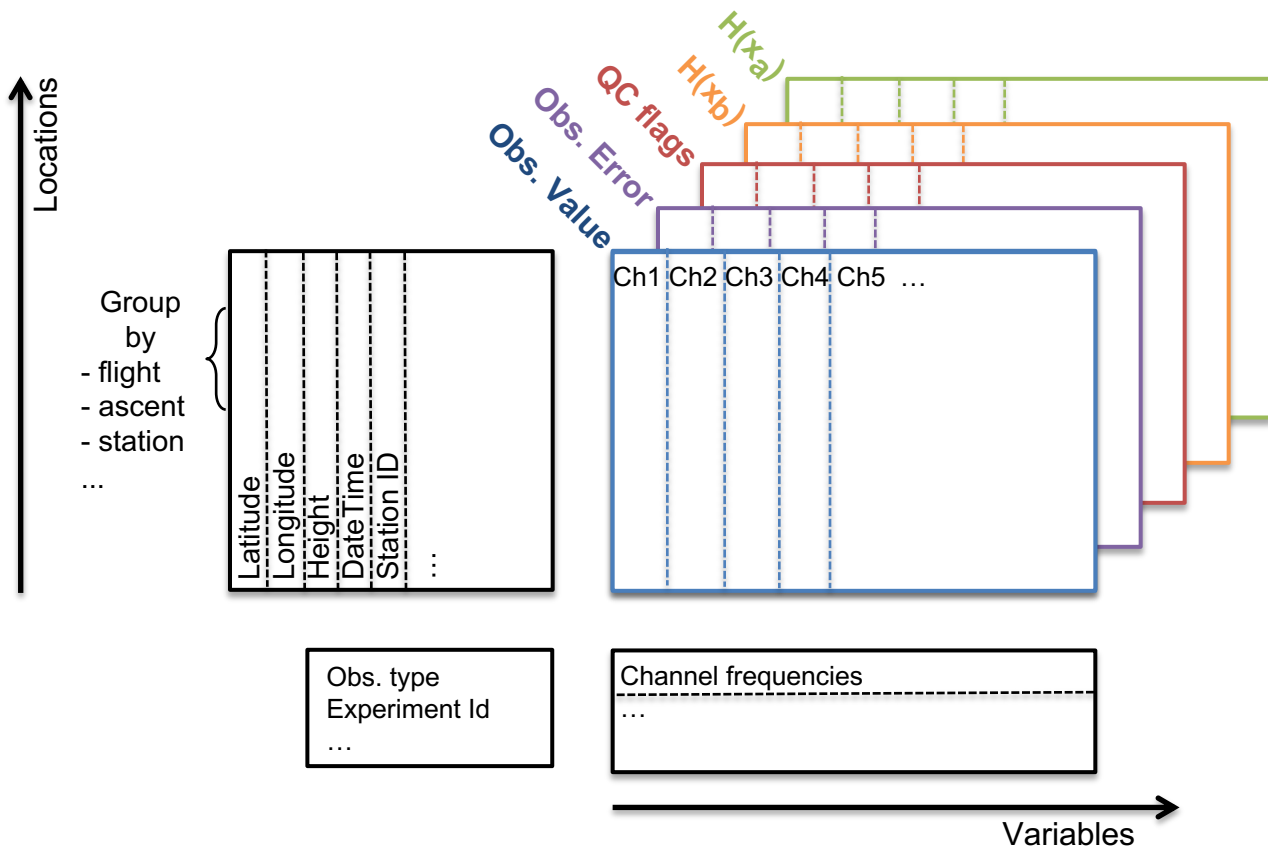
- maintenance and evolution are impossible
- does not fit possible evolution of DA methods (continuous assimilation)

Observation Data Flow with IODA



One observation data handling interface across the whole NWP chain

Access to Observations



Uniform view
and interface for
observation data access

Interface for Observation
Data Access based on
this abstract view

Abstract representation,
not necessarily actual
storage structure

Interface for Observation Data Access



The IODA API is used to access data:

```
std::vector<double> lats;  
ospace.get_db("MetaData", "latitude", lats);
```

C++

```
std::vector<float> viewing_angle;  
ospace.get_db("MetaData", "sensor_view_angle", viewing_angle);
```

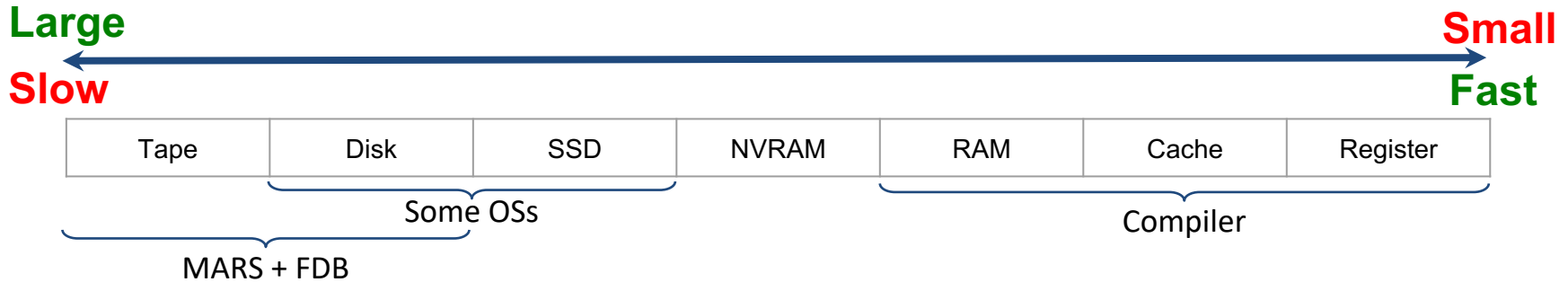
```
nlocs = obsspace_get_nlocs(obsdb)  
allocate(obsValue(nlocs))  
call obsspace_get_db(obsdb, "ObsValue", "bending_angle", obsValue)  
  
allocate(obselev(nlocs))  
call obsspace_get_db(obsdb, "MetaData", "station_elevation", obselev)
```

Fortran

The API is independent of the file format or internal structure

All applications (including plotting and diagnostics) should be based on this API

Observation Data Handling



Ideally, we want uniform and easy access to observation data

Can we hide the complexity of the memory hierarchy from scientists (separation of concerns)?

Files could become irrelevant in the future

Interface for Observation Data Access



One in-memory data structure has been implemented

Two I/O prototypes have been tested: NetCDF4 and ODB

Set of observations available and accessible from all platforms for testing

IODA will be a success if users don't need to know what the file format is.