

# Observation Handling with the IODA Subsystem of JEDI

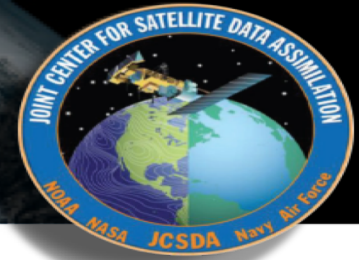
**Stephen Herbener**

Joint Center for Satellite Data Assimilation (JCSDA)

17<sup>th</sup> JCDSA Science and Technical Workshop  
May 29-31, 2019

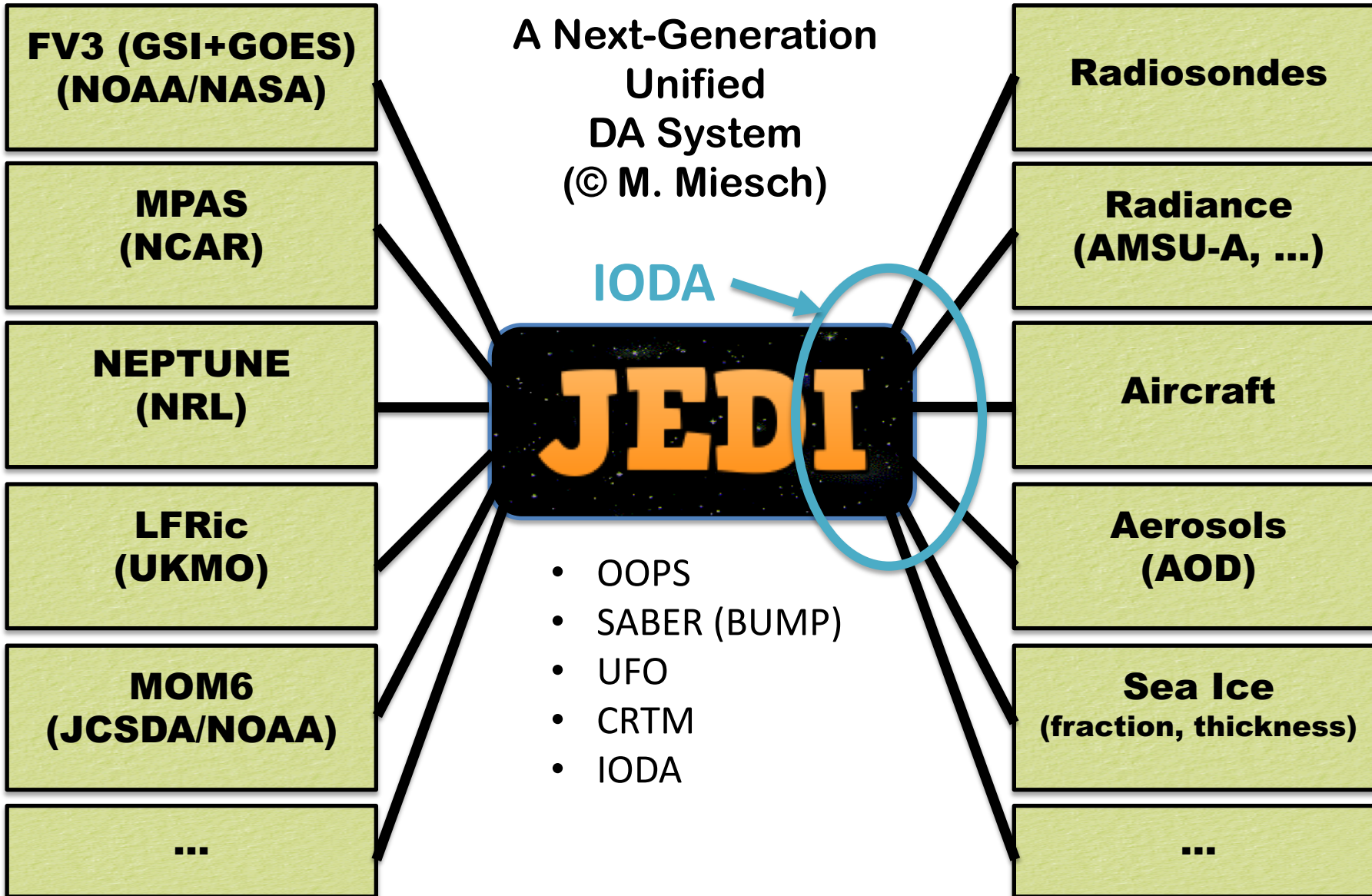
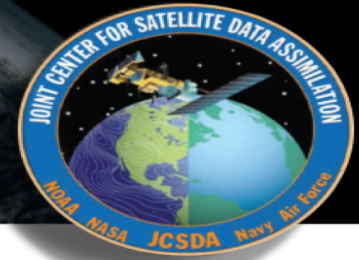


# What is IODA?



- IODA is the subsystem in JEDI that provides access to observation data
- Interface for **O**bservation **D**ata **A**ccess
- Three levels
  - Archive: long term storage, historic database
  - File: on disk, data for one DA Cycle
  - Memory
- Two environments
  - Plotting, analyzing, verifying on workstation or laptop
  - DA and other HPC applications (MPI, threads, GPUs, ...)

# JEDI Overview



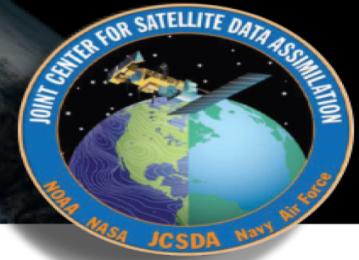
- Enables high leverage
- For example, add your model
- Then you have access to:
  - Obs data
  - Forward operators
  - DA flows
  - Etc.

# IODA Long Term Vision



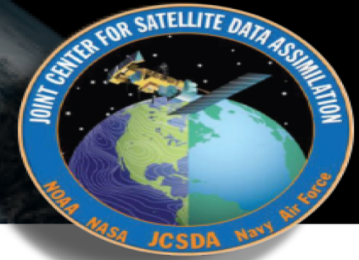
- Look and feel of a database
  - Select and filter data on various criteria
    - Select observations within a DA timing window
    - Filter on QC marks, horizontal locations, station id's, etc.
- Converge on a common file format for holding observation data
  - A common format would greatly facilitate the sharing of data and the exchange science results
- Likely that we will adopt an existing database solution
  - We will soon be evaluating ECMWF's ODB solution once the ODC API becomes available

# IODA Requirements



- IODA Workshop
  - February 2019 at NRL in Monterey, CA
  - Requirements gathering effort
  - First round of gathering (ala agile methodology)
- Categories of requirements include, but not limited to:
  - Access to Data and Meta-data
    - Data and meta-data are both important
    - Efficient query style access
  - Flexible
    - Wide variety of obs types
  - Reliable
    - Operational mode cannot break down
  - Portable
    - Across hardware platforms, programming languages and compilers
  - Security
    - Protected data and results

# Revisit IODA Levels



- Three levels
  - Archive: long term storage, historic database
  - File: on disk, data for one DA Cycle
  - Memory

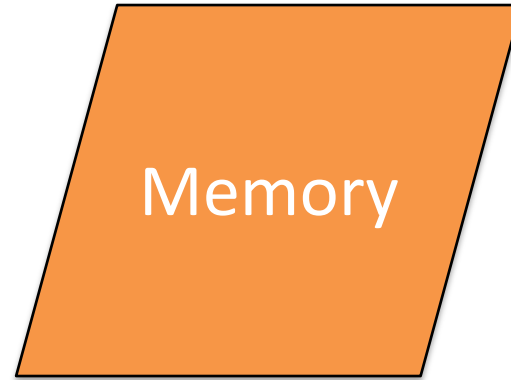
# IODA Levels: Capacity-Speed Tradeoff



Archive



File



Memory

- **Archive**
  - All obs types
  - All dates (decades)

- **File**
  - Specific obs types
  - DA cycle begin - end

- **Memory**
  - Specific obs types
  - Forecast begin - end



Capacity



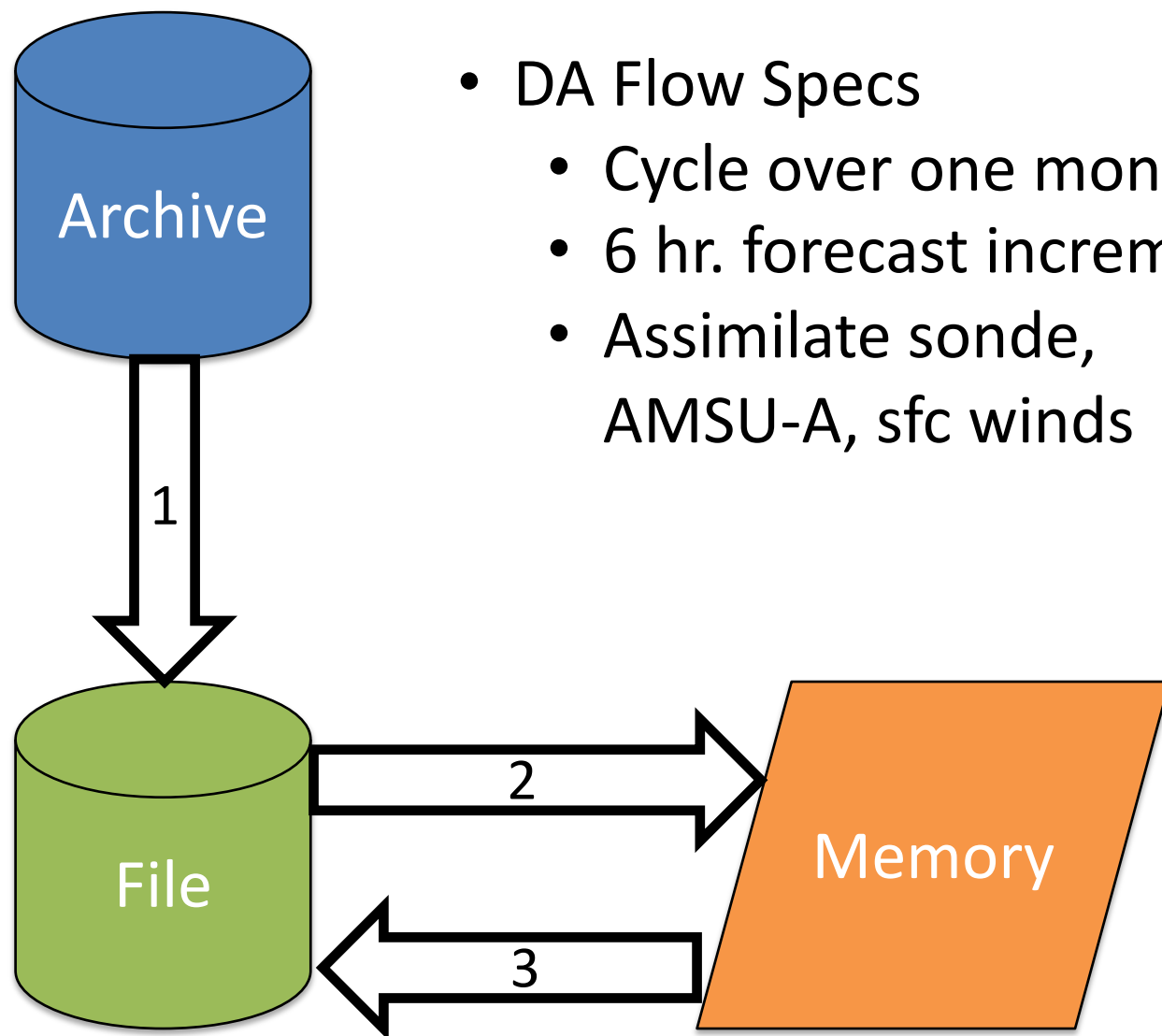
Speed

# DA Flow



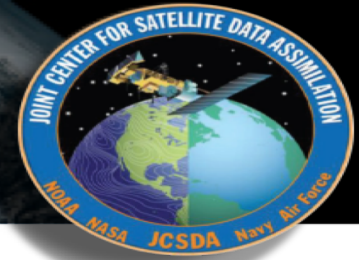
- DA Flow Specs
  - Cycle over one month
  - 6 hr. forecast increments
  - Assimilate sonde, AMSU-A, sfc winds

1. Retrieve all sonde, AMSU-A and sfc winds within the one month period
2. Loop over each 6-hr forecast window retrieving appropriate sonde, AMSU-A and sfc winds as needed
3. As DA flow progresses, store diagnostics into output files





# IODA Status



- IODA started as a simple prototype and is evolving toward the long term vision
- We are currently using pieces of existing systems to mimic the database style access to the three IODA levels
  - Archive
    - Data tanks from various data centers
    - Different file types (BUFR, netcdf, specialized binary)
    - Different methods of organizing data within the file
      - QC code semantics, internal table structure and layout, etc.
  - File
    - Netcdf
    - Unified organization within the file
  - Memory
    - C++ Standard Data Structures



# IODA Today

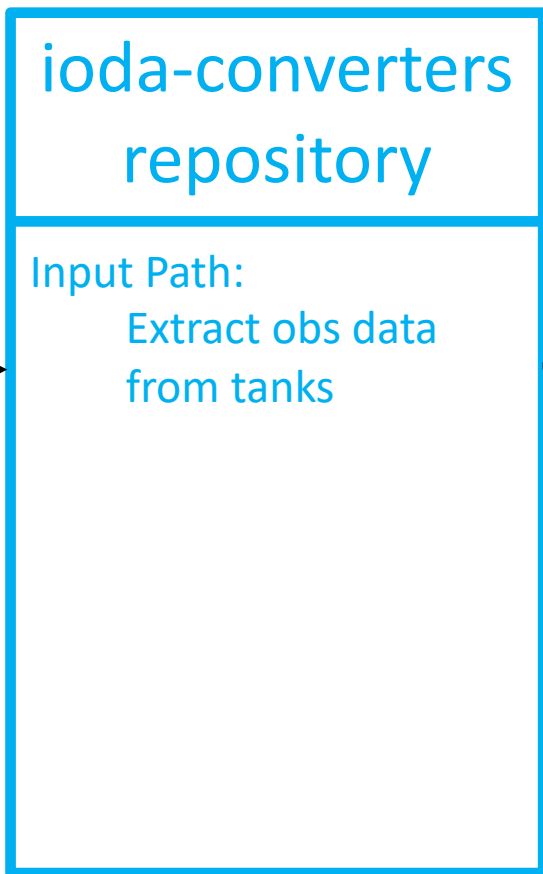


## Archive

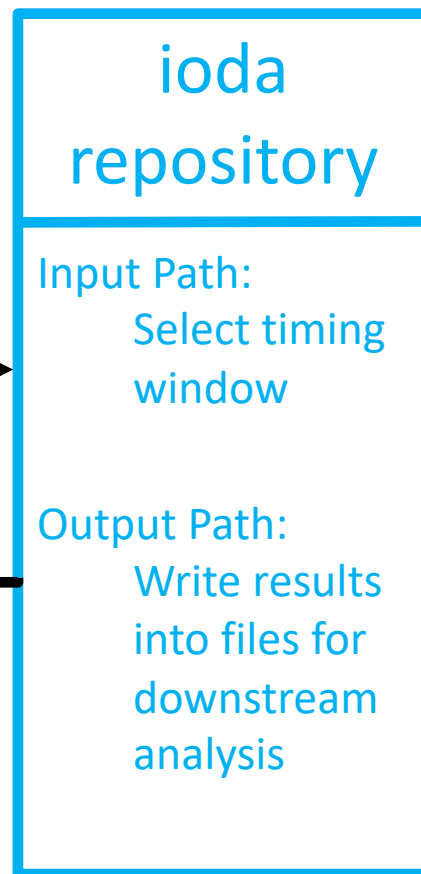


...

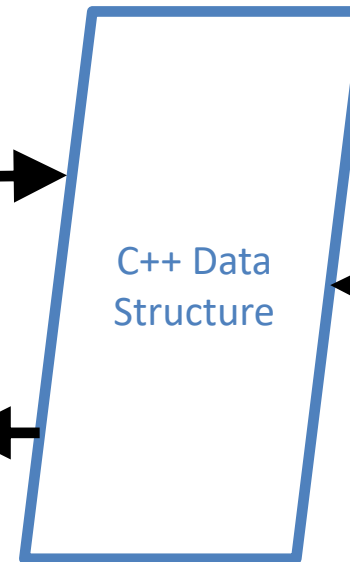
“Tanks”



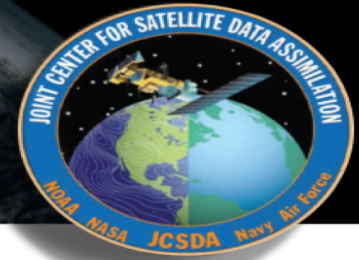
## File



## Memory



# IODA Next steps



- Complete the design of long term IODA subsystem
  - Database design
    - Select a database solution (ODC, other?)
    - Define how to organize data within the database file and memory structures
  - This task will determine the common file format for IODA
- Create the IODA Archive Level
  - Data storage strategy (cloud)
  - Interface for archiving and retrieving data
  - Tool to convert raw observation data to the IODA common file format