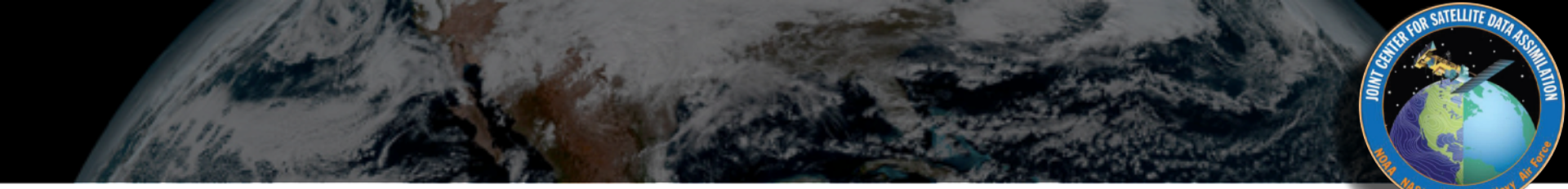


Joint Effort for Data assimilation Integration

Why OOPS/JEDI?

Joint Center for Satellite Data Assimilation (JCSDA)

13 - 15 November 2018, Boulder

Joint Center for Satellite Data Assimilation

A multi-agency research center created to improve the use of satellite data for analyzing and predicting the weather, the ocean, the climate and the environment.

Collaborative organization funded by

- NOAA/NWS
- NOAA/NESDIS
- NOAA/OAR
- NASA/GMAO
- US Navy
- US Air Force

Organized by projects:

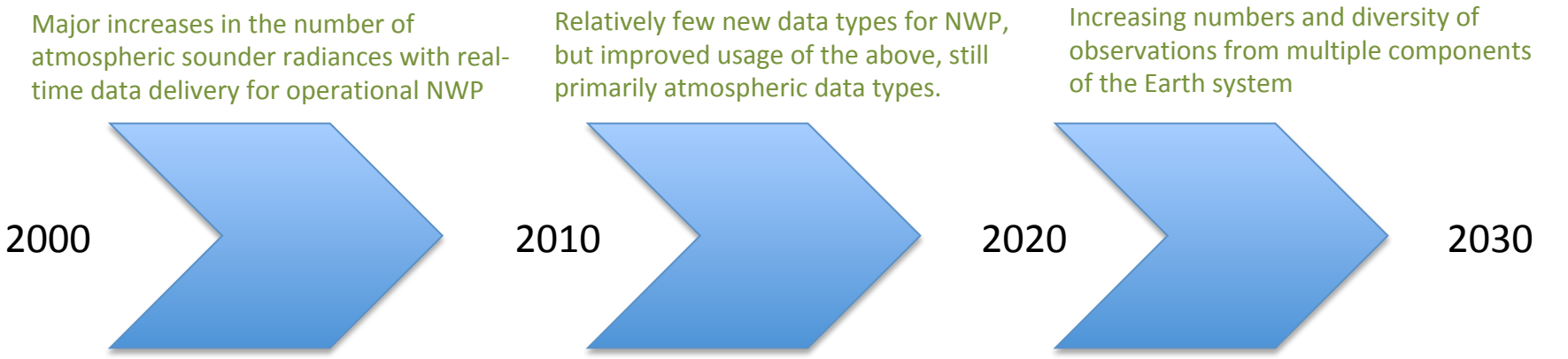
- CRTM (Community Radiative Transfer Model)
- JEDI (Joint Effort for Data assimilation Integration)
- SOCA (Sea-ice Ocean Coupled Assimilation)
- NIO (New and Improved Observations)
- IOS (Impact of Observing Systems)

Why JEDI?

Because we want better forecasts

JCSDA: Same Mission, New Approach

The JCSDA mission remains the same. However, as compared with the time of JCSDA's founding (c. 2000), the rapid evolution of operational NWP toward coupled models and assimilation systems, combined with the expected proliferation of new observations from components of the Earth system other than the atmosphere, demand **new and more unified approaches to algorithm development, observation processing, and maintenance of software.**



The JCSDA is taking a proactive approach to current and future satellite data assimilation challenges to reposition the research and operational communities in the US to take full advantage of the coming era of full Earth system prediction.

JEDI: Motivations and Objectives

The Joint Effort for Data assimilation Integration (JEDI) is a collaborative development between JCSDA partners.

Develop a unified data assimilation system:

- From toy models to Earth system coupled models
- Unified observation (forward) operators (UFO)
- For research and operations (including R2O/O2R)
- Share as much as possible without imposing one approach (one system, multiple methodologies/configurations)

Why JEDI?

- Reduce duplication of effort between JCSDA partners
  - Adding new observations (UFO and IODA)
  - Implementation of new DA algorithms (OOPS)

- Bring all components of Earth-system in one DA system
  - Develop DA once for all components (OOPS)
  - Enable future coupled DA developments (OOPS)

Why JEDI?

- Modern DA systems are too complex for one person to grasp
  - Collaborative developments
  - Separation of concerns

- Modernize software
  - Speed-up future developments
  - Ease maintenance
  - Increase portability and efficiency

Joint Effort for Data assimilation Integration

OOPS Approach

Model Space Interfaces

Observation Space Interfaces

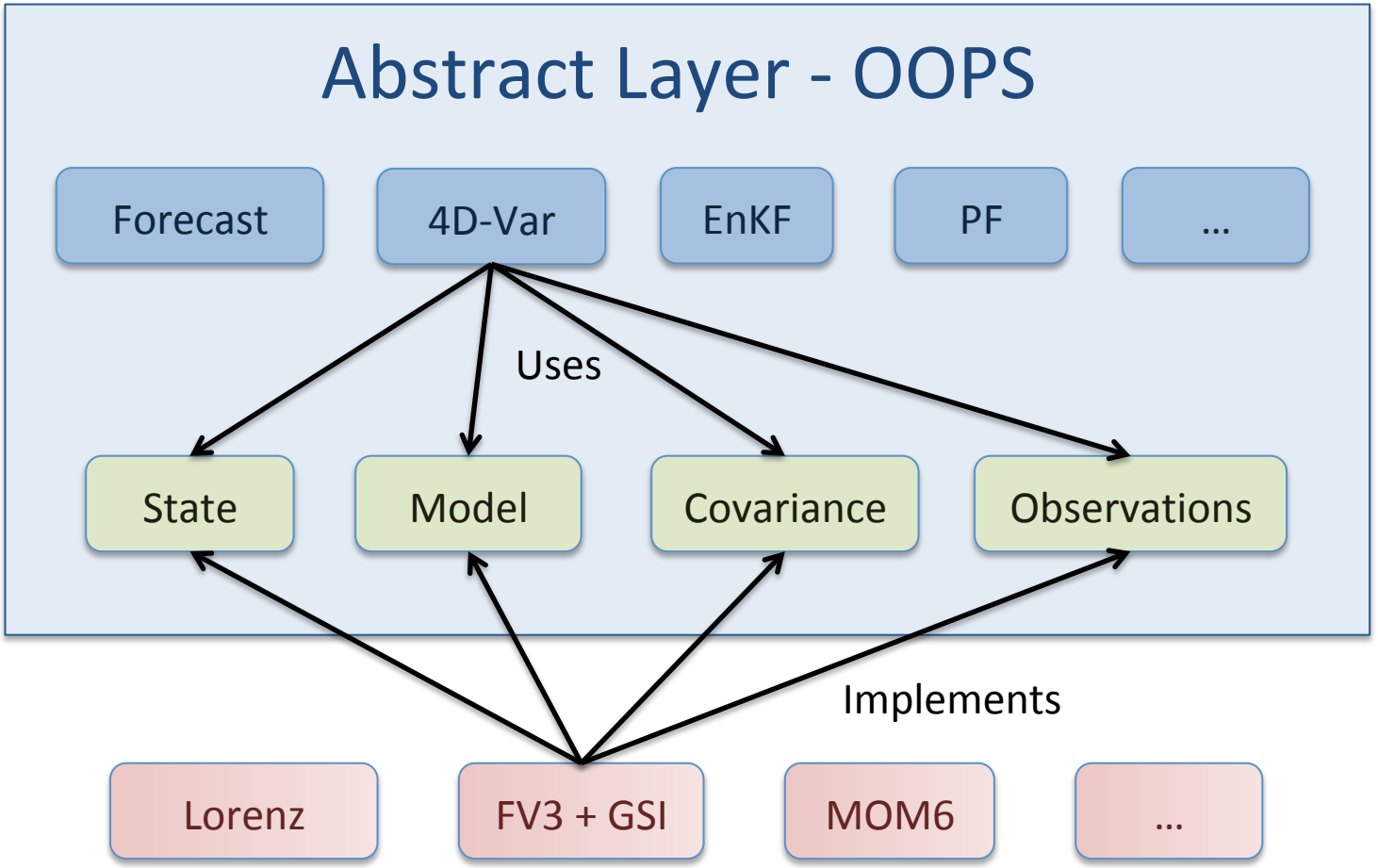
Infrastructure and working practices

Design: separation of concerns

Generic Applications

Abstract building blocks

Systems



Abstract interfaces are the most important aspect of the design

Joint Effort for Data assimilation Integration

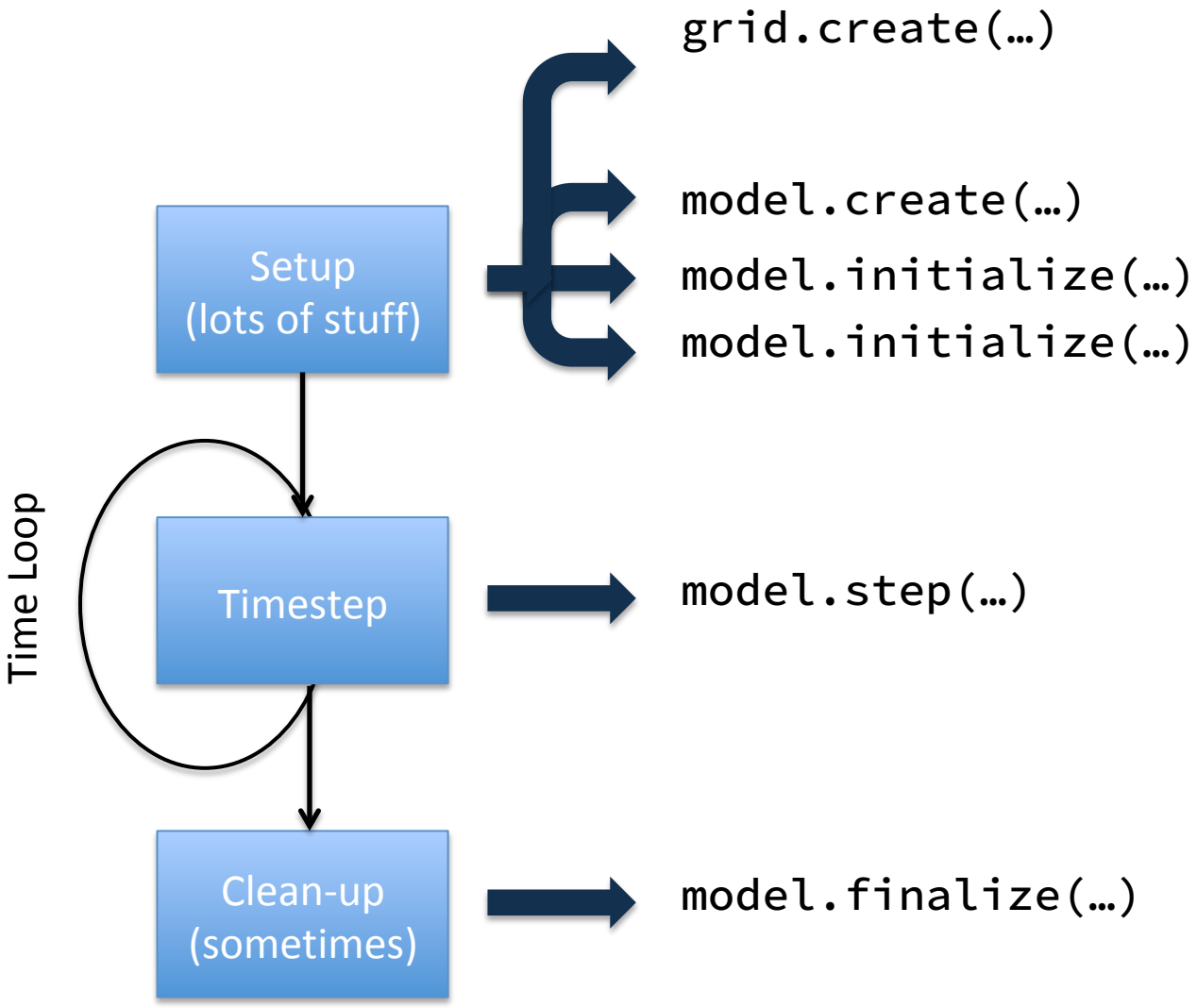
OOPS Approach

Model Space Interfaces

Observation Space Interfaces

Infrastructure and working practices

Model Design: $x_t=M(x_0)$



Models Interfacing Status

	State	3D H(x)	M(x)	4D H(x)	3D-Var	TL/AD	4D-Var
FV3-GFS (NOAA)	✔	✔	✔	✔	✔	✔*	✔
FV3-GEOS (NASA)	✔	✔	✔	✔	✔	✔*	✔
MPAS (NCAR)	✔	✔	✔		✔	N/A	
WRF (NCAR/NOAA)	✔		✔				
LFRic (UKMO)	✔	✔	✔		✔	?	
NAVGEM (NRL)	✔						
NEPTUNE (NRL)	✔					?	
CICE5 (JCSDA/NOAA)	✔	✔			✔	N/A	
MOM6 (JCSDA/NOAA)	✔	✔			✔	N/A	

✔ = technically working ✔ = in progress * Linearized physics in progress

The project started in January 2017, coding started in August 2017.

Joint Effort for Data assimilation Integration

OOPS Approach

Model Space Interfaces

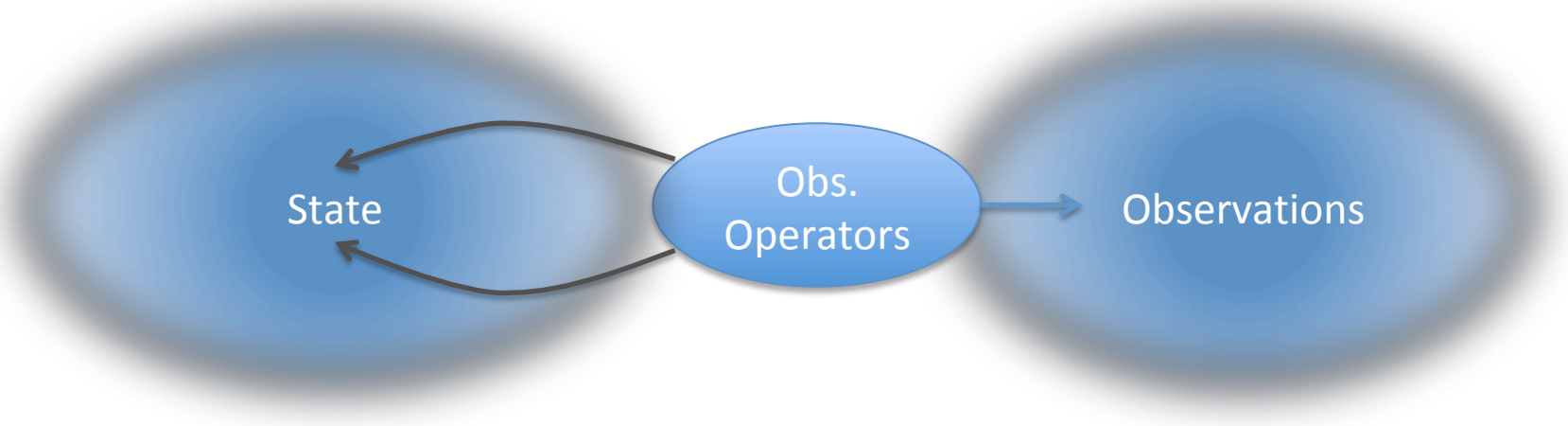
Observation Space Interfaces

Infrastructure and working practices

Observation Space Objectives

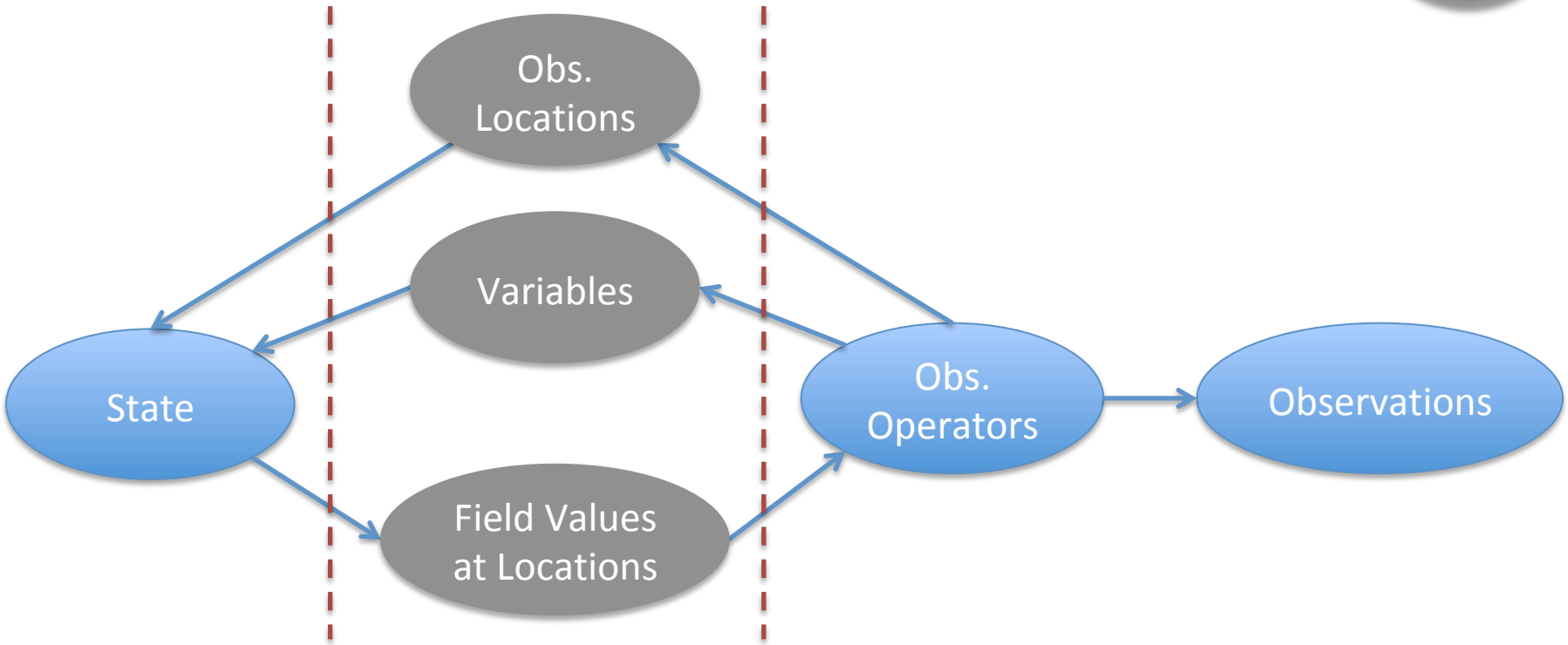
- Share observation operators between JCSDA partners and reduce duplication
  - Include instruments science teams
- Faster use of new observing platforms
  - Regular satellite missions are expensive
  - Cube-sat have short expected life time
- Unified Forward Operator (UFO)
  - Build a community app-store of observation operators (“op-store”)

Observation Operators



- In most existing systems, observation operators directly access state/model data
- Observation operators, and as a result DA systems, are very model specific

UFO: the interface advantage



- JEDI/UFO introduces standard interfaces between the model and observation worlds
- Observation operators are independent of the model and can easily be shared, exchanged, compared

UFO Observation “filters”

- JEDI/UFO calls abstract “observation filters” before and after the actual observation operator
- Filters can be written once and used with many observation types
- Observation filters are generic and have access to
  - Observation values and metadata
  - Simulated observation value (post-filter)
  - Their own private data
- Examples:
  - Quality control (background check, buddy check, cloud detection…)
  - Thinning

Interface for Observation Data Access (IODA)

Interface to isolate science code from data storage

Three levels:

- Long term storage (historic database)
- Files on disk (one DA cycle)
- In memory handling of observations (hardware specific?)

Two environments:

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

Goal: one interface, possibly several implementations?

Observation Space Interfaces Status

- Initial implementation of interface classes
  - Locations
  - Variables
  - GeoVaLs (Geophysical Values at Locations)
- Currently (Nov. 2018):
  - Radiosonde, aircraft
  - CRTM radiances (tested for AMSU-A) and AOD
  - GNSSRO
  - Marine observations
  - Interface for QC filters, background check implemented
- IODA v0 based on NetCDF4 or ODB-API

Joint Effort for Data assimilation Integration

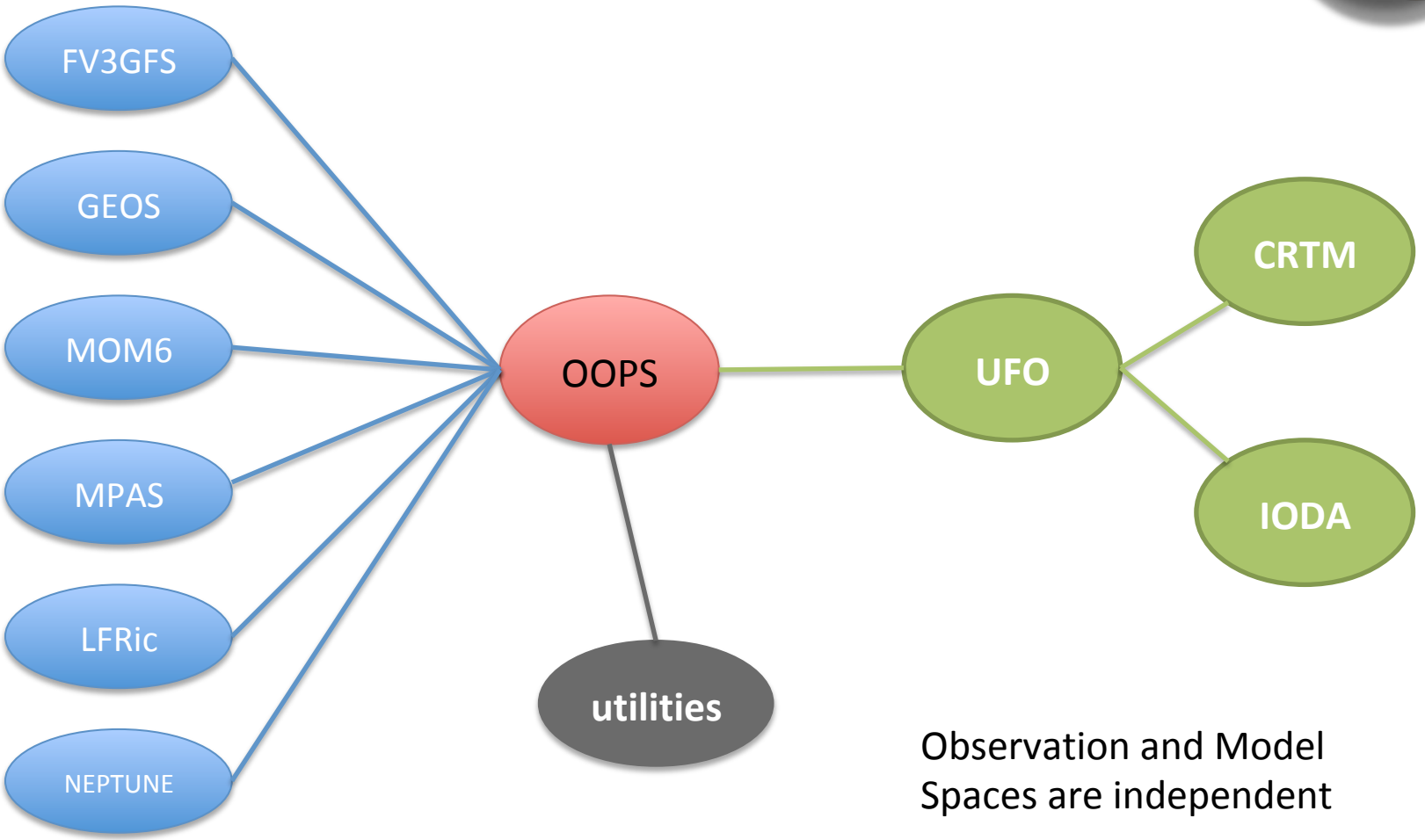
OOPS Approach

Model Space Interfaces

Observation Space Interfaces

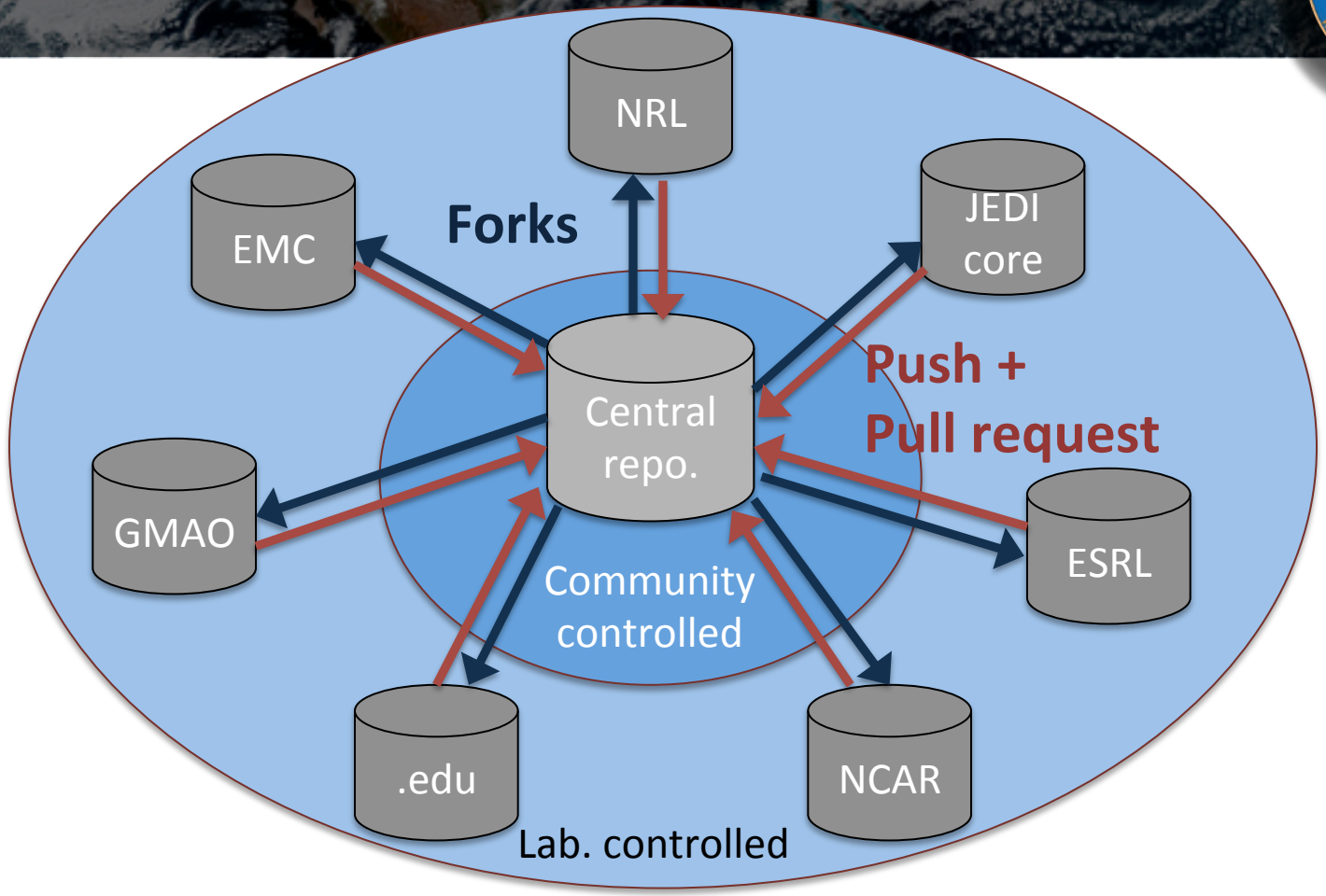
Infrastructure and working practices

Code and repositories



Observation and Model Spaces are independent

Collaborating: Repositories



Permission to fork repository are very easy to obtain

Contributing code is very controlled:

- Pushing a branch requires write permission on central repository
- Pull request triggers code review and approval for merging to higher level branch

Infrastructure, working practices

- Project methodology inspired by Agile/SCRUM
  - Adapted to distributed teams and part time members
- Collaborative environment
  - Easy access to up-to-date source code (github)
  - Easy exchange of information (Confluence, zenhub)
- Flexible build system (cmake-based)
- Documentation, tutorials, JEDI Academy

Infrastructure, working practices

- Continuous Integration, Testing framework (coming)
  - Toolbox for writing tests
  - Automated running of tests (on push to repositories)
- Effort on portability
  - Automatically run tests with several compilers
  - JEDI available in containers (singularity)
- Enforce software quality (correctness, coding norms, efficiency)
- Change in working practices take time…

Code Sprints

- Gather 8-10 people in a room for 2 weeks
  - B Matrix (Aug. 2017, Aug. 2018)
  - Observation Operators (Nov. 2017, Aug 2018)
- Efficient use of time, especially for part time contributors
- Involve people from partner institutions in project
- Very motivating (before, during, after)