Using JEDI for Atmospheric Composition applications: A technical introduction

Jérôme Barré[1], Maryam Abdi-Oskouei[1]
[1] NASA, GMAO

JEDI-AC tutorial
February 2026

Today

What this tutorial covers?

1. **Introduction**: Understand the basic concepts in JEDI. The object-oriented benefits
2. **Lecture 1**: Understand the IODA observation space format - learn to use and develop IODA converters
3. **Lecture 2**: Use observations in JEDI. Monitoring demonstration H(x) experiment example
4. **Lecture 3**: Assimilate observations in JEDI. EnVar experiment example

What this tutorial does not cover (yet)?

1. How to build/compile your own JEDI system
2. How to add new feature in JEDI
3. How to use data assimilation workflows

GODDARD EARTH SCIENCES

Today: Agenda

Introduction: Understand the basic concepts in JEDI. The object-oriented benefits 5 mins

Questions 5 mins

Lecture 1: IODA observation space format 20mins

Questions 10mins

Lecture 2: Using observations and H(x) experiment 30 mins

Questions 10mins

Lecture 3: Assimilate observations and EnVar experiments 30mins

Closing remarks 5mins

Break 15mins

Open hour, Q&A, trying yourself! 45 mins

Please save your questions for the dedicated question times

GODDARD EARTH SCIENCES

Intro: JEDI – Design Principles and Notional Status

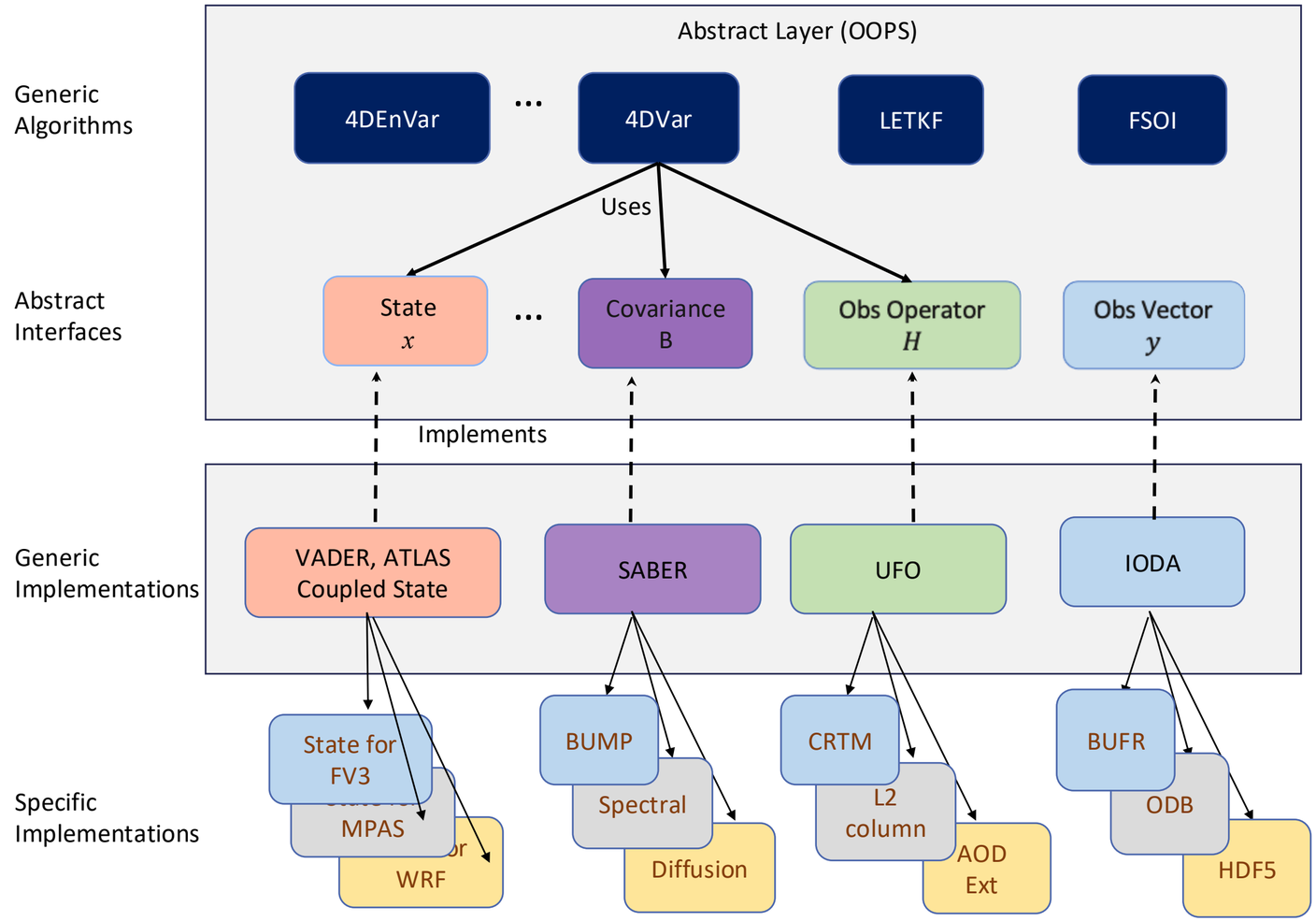
$$\Delta x_a = \mathbf{B}\mathbf{M}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}}\left(\mathbf{H}\mathbf{M}\mathbf{B}\mathbf{M}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}} + \mathbf{R}\right)^{-1}\left(y_o - H(M(x_b))\right)$$

JEDI design cleanly separates the concepts of:

- DA algorithms
- model specifics
- observation specifics

Abstract interfaces are used by DA algorithms and implemented by specific models and observations.

Generic code eliminates duplication of development, testing and maintenance effort, across teams & Earth-system components.



Intro: Do I need to know C++ to work with JEDI?

Not really… but if you do, that's a ++

If you are using JEDI there's a high chance that what you need is already coded there. You often don't need to modify core code.

You need to understand the flow of information in JEDI but LLMs (e.g. chat-GSFC) can help you with that!

During this session we will go over concept of C++ classes and manipulation of YAMLs in JEDI

Intro: Object-Oriented JEDI (OOPS) and YAML driven execution

OOPS in JEDI:

- DA concepts are **classes**: e.g. Geometry, State, Model, ObsSpace, CostFunction
- Abstract base classes define required **methods,** which are concrete implementations selected at runtime: ModelInterfaces, ObsOperators, BackgroundError, etc…
- **Algorithms** compose and drive objects inside specific implementations

YAMLs controls the DA "objects":

- YAML "wires" which objects exist in the execution (model, obsop, filters, covariance, solver, etc..) and their options
- **Same YAML patterns** and keys across applications.
- **Same executable** with different experiments: edit YAML to change wiring, functionality and tuning

```yaml
cost function:
  cost type: 3D-Var
  time window:
    begin: 2020-09-03T15:00:00Z
    length: PT6H
  analysis variables:
  - volume_mixing_ratio_of_no2
  geometry:
    fms initialization:
      namelist filename: Data/fv3files/fmsmpp.nml
    akbk: Data/inputs/fv3files/akbk72.nc4
    npx: 13
    npy: 13
    npz: 72
  background:
    datetime: 2020-09-03T18:00:00Z
    filetype: cube sphere history
    datapath: Data/inputs/geos_c12
    filename: geos_cf.bkg.%yyyy%mm%dd_%hh%MM%ssz.nc4
    state variables:
    - air_pressure_thickness
    - volume_mixing_ratio_of_no2
    - volume_mixing_ratio_of_no
    - volume_mixing_ratio_of_o3
    - air_pressure_at_surface
    - water_vapor_mixing_ratio_wrt_moist_air
    field io names: &field_io_names
      volume_mixing_ratio_of_no2: NO2
      air_pressure_thickness: DELP
      volume_mixing_ratio_of_no: "NO"
      volume_mixing_ratio_of_o3: O3
      air_pressure_at_surface: PS
```

GODDARD EARTH SCIENCES

Intro: Object-Oriented JEDI (OOPS) and YAML driven execution

DA methods require the same limited number of entities. The development of these entities should be reusable.
Entities = basic (abstract) classes that define the system.
A typical Variational DA YAML for example has general classes (see below). Inside each of those you will have more specific implementations and YAML options. DA abstract classes generally are:

```
cost function:
  cost type:
  time window:
  analysis variables:
  geometry:
  background:
  background error:
  observations:
output:
variational:
```

```yaml
cost function:
  cost type: 3D-Var
  time window:
    begin: 2020-09-03T15:00:00Z
    length: PT6H
  analysis variables:
  - volume_mixing_ratio_of_no2
  geometry:
    fms initialization:
      namelist filename: Data/fv3files/fmsmpp.nml
    akbk: Data/inputs/fv3files/akbk72.nc4
    npx: 13
    npy: 13
    npz: 72
  background:
    datetime: 2020-09-03T18:00:00Z
    filetype: cube sphere history
    datapath: Data/inputs/geos_c12
    filename: geos_cf.bkg.%yyyy%mm%dd_%hh%MM%ssz.nc4
    state variables:
    - air_pressure_thickness
    - volume_mixing_ratio_of_no2
    - volume_mixing_ratio_of_no
    - volume_mixing_ratio_of_o3
    - air_pressure_at_surface
    - water_vapor_mixing_ratio_wrt_moist_air
    field io names: &field_io_names
      volume_mixing_ratio_of_no2: NO2
      air_pressure_thickness: DELP
      volume_mixing_ratio_of_no: "NO"
      volume_mixing_ratio_of_o3: O3
      air_pressure_at_surface: PS
```

GODDARD EARTH SCIENCES

Intro: Object-Oriented JEDI (OOPS) and YAML driven execution

DA methods require the same limited number of entities. The development of these entities should be reusable.
Entities = basic (abstract) classes that define the system.
A typical Variational DA YAML for example has general classes (see below). Inside each of those you will have more specific implementations and YAML options. DA abstract classes generally are:

```
cost function:
  cost type:
  time window:
  analysis variables:
  geometry:
  background:
  background error:
  observations:
output:
variational:
```

The goal of those lectures (2 and 3 particularly) is to go over the YAMLs and understand how they are constructed to perform a DA application

```
1   cost function:
2     cost type: 3D-Var
3     time window:
4       begin: 2020-09-03T15:00:00Z
5       length: PT6H
6     analysis variables:
7     - volume_mixing_ratio_of_no2
8     geometry:
                                                    nsmpp.nml
17        filetype: cube sphere history
18        datapath: Data/inputs/geos_c12
19        filename: geos_cf.bkg.%yyyy%mm%dd_%hh%MM%ssz.nc4
20        state variables:
21        - air_pressure_thickness
22        - volume_mixing_ratio_of_no2
23        - volume_mixing_ratio_of_no
24        - volume_mixing_ratio_of_o3
25        - air_pressure_at_surface
26        - water_vapor_mixing_ratio_wrt_moist_air
27        field io names: &field_io_names
28          volume_mixing_ratio_of_no2: NO2
29          air_pressure_thickness: DELP
30          volume_mixing_ratio_of_no: "NO"
31          volume_mixing_ratio_of_o3: O3
32          air_pressure_at_surface: PS
```

Lecture 1: Handling Observations

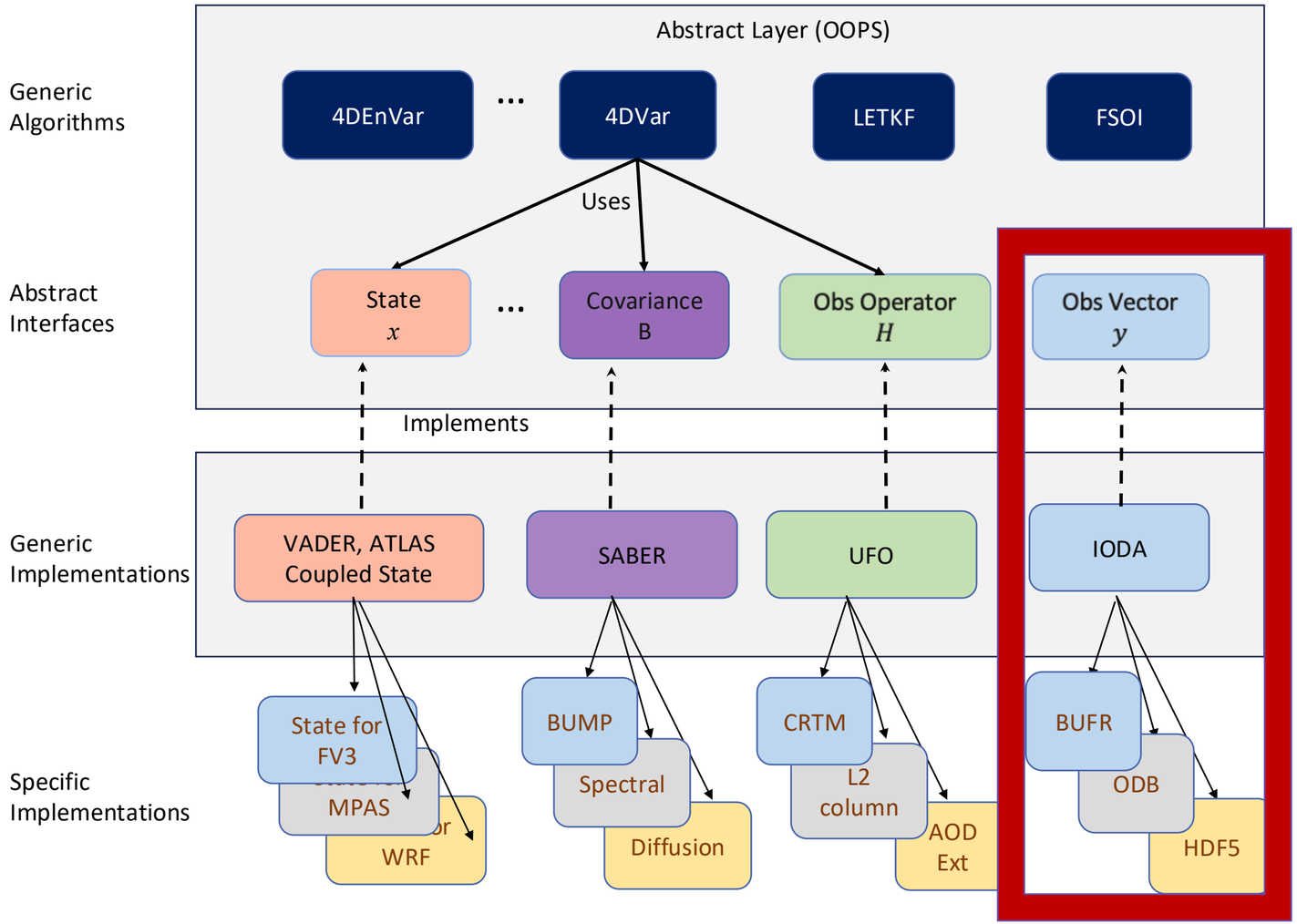
- Unified interface to observations (IODA) with multiple file-format backends

- IODA naming convention for observations & meta-data

YAML parts are something like that

```
observations:
   observers:
   - obs space:
       name: NO2
       obsdatain:
         engine:
           type: H5File
           obsfile: Data/obs/tropomi_no2.nc4
       obsdataout:
         engine:
           type: H5File
           obsfile: Data/hofx/tropomi_no2.nc
       simulated variables:
       - nitrogendioxideColumn
```

$$\Delta x_a = \mathbf{B}\mathbf{M}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}}\left(\mathbf{H}\mathbf{M}\mathbf{B}\mathbf{M}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}} + \mathbf{R}\right)^{-1}\left(y_o - H(M(x_b))\right)$$



Lecture 1: The IODA format: observation and feedback files

```
netcdf airnow_fb_example {
dimensions:
Location = UNLIMITED ; // (8755 currently)
variables:
int64 Location(Location) ;

group: EffectiveError0 {}
group: EffectiveError1 {}
group: EffectiveQC0 {}
group: EffectiveQC1 {}
group: MetaData {}
group: ObsBias0 {}
group: ObsBias1 {}
group: ObsError {}
group: ObsValue {}
group: PreQC {}
group: hofx0 {}
group: hofx1 {}
group: oman {}
group: ombg {}

}
```

In Observation file

Added as Feedback file after H(x)

Added as Feedback file after DA

```
group: MetaData {
  variables:
       int airQualityClassification(Location) ;
              airQualityClassification:_FillValue = -2147483643 ;
       int64 dateTime(Location) ;
              dateTime:_FillValue = -9223372036854775801LL ;
              string dateTime:units = "seconds since 1970-01-01T00:00:00Z" ;
       float latitude(Location) ;
              latitude:_FillValue = -3.368795e+38f ;
              string latitude:units = "degrees_north" ;
       float longitude(Location) ;
              longitude:_FillValue = -3.368795e+38f ;
              string longitude:units = "degrees_east" ;
} // group MetaData
```

For example:

```
group: hofx0 {
  variables:
       float carbonmonoxideInsitu(Location) ;
              carbonmonoxideInsitu:_FillValue = -3.368795e+38f ;
       float nitrogendioxideInsitu(Location) ;
              nitrogendioxideInsitu:_FillValue = -3.368795e+38f ;
       float ozoneInsitu(Location) ;
              ozoneInsitu:_FillValue = -3.368795e+38f ;
  } // group hofx0
```

Practical Lecture 1

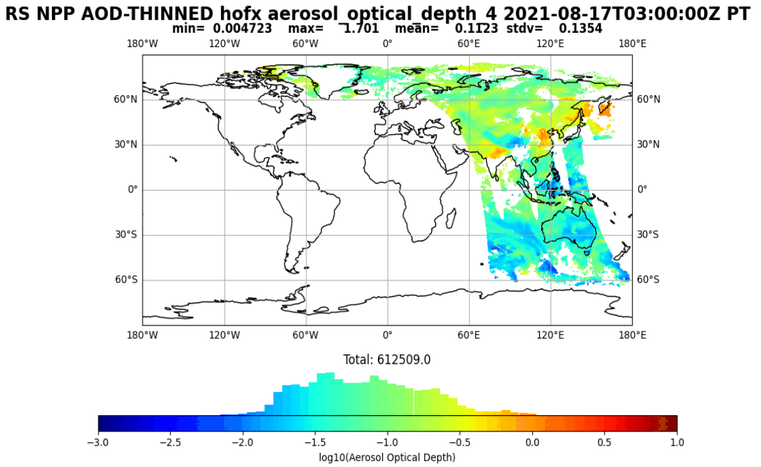
- https://mer-a-o.github.io/howtojedi/lecture1/

- Understanding the IODA format
- Use IODA converters

Lecture 1: Tested and assimilated AC observation products in JEDI

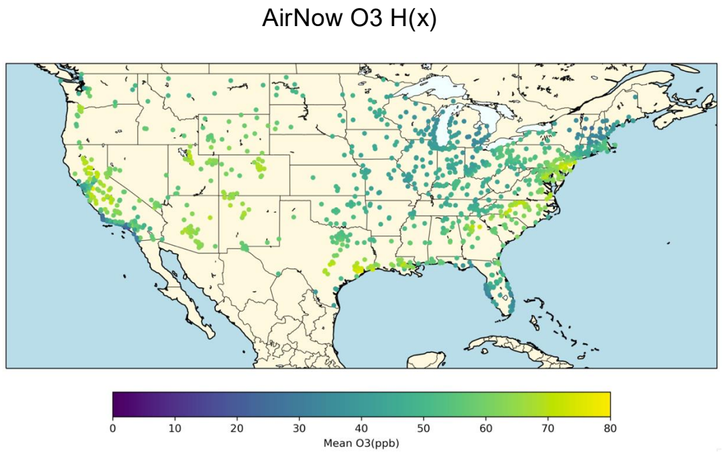
Aerosols:

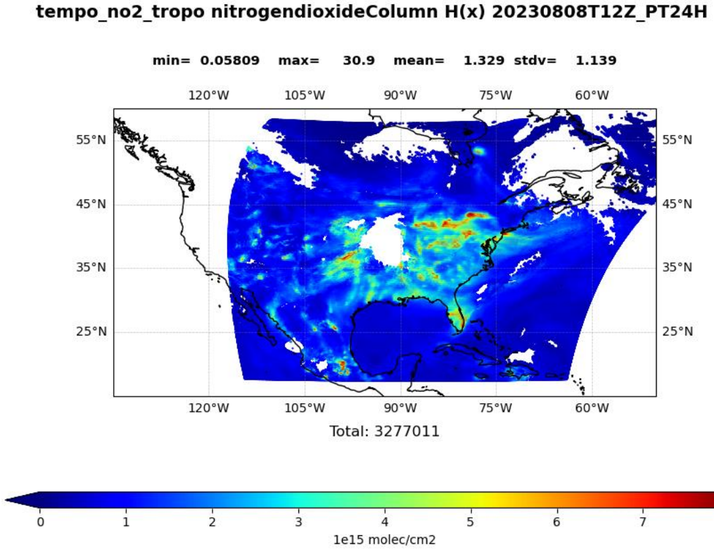
- VIIRS NPP AOD (DT,DB) and L1b
- VIIRS NOAA 20 AOD (DT,DB) and L1b
- MODIS Aqua AOD
- MODIS Terra AOD
- PACE AOD
- AeroNet
- CALIOP Lidar
- Surface: Airnow, Purple Air, Airbase



Trace gas:

- TropOMI NO2 (tropospheric and total columns)
- TropOMI CO total column
- MOPITT CO total column
- TEMPO NO2 and HCHO columns
- MLS O3 limb profile
- OMPS LP, NP and TC O3 (limb and nadir)
- PANDORA NETWORK
- Surface: Airnow, Airbase
- Aircraft campaigns





GODDARD EARTH SCIENCES

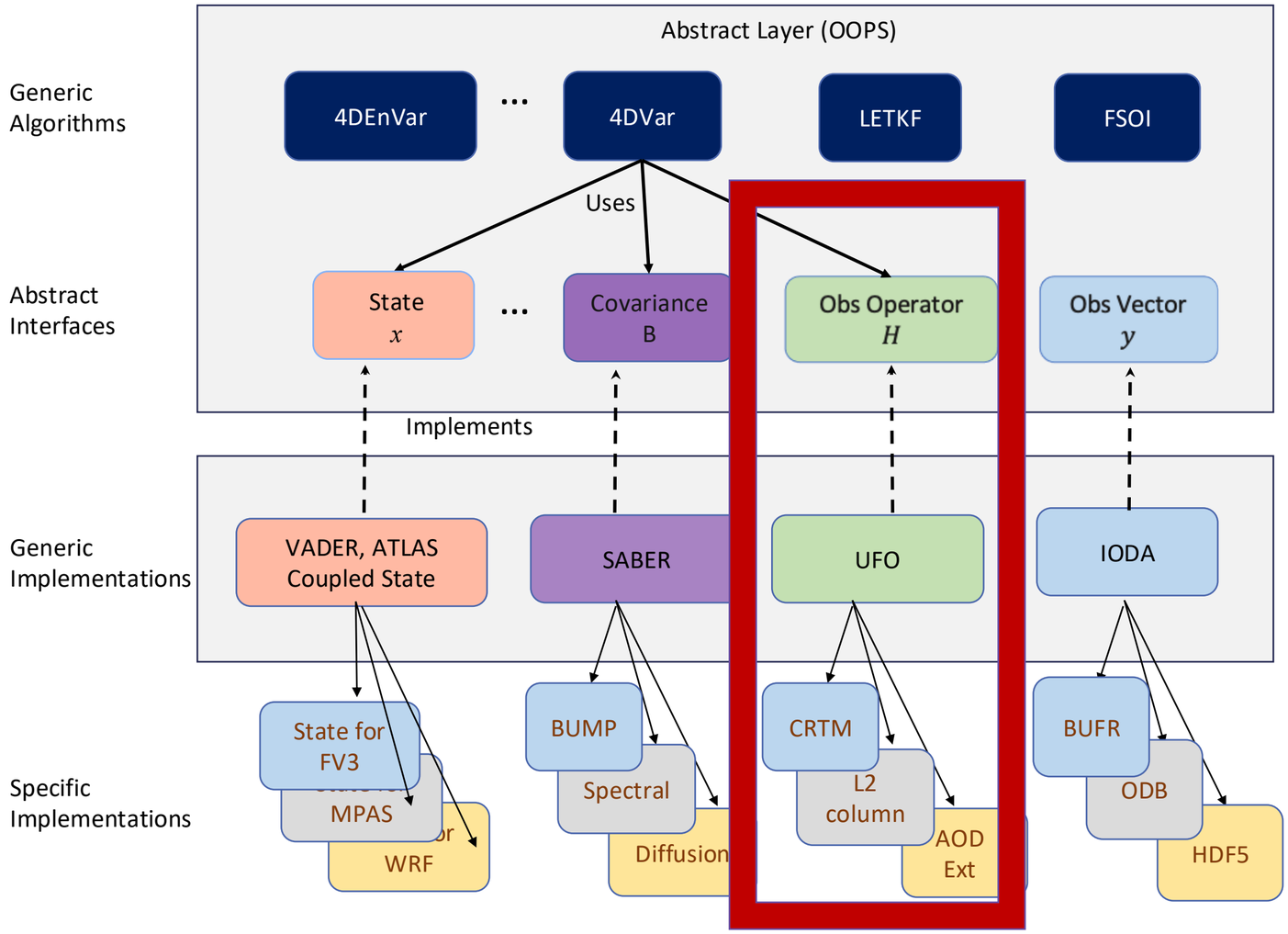
Lecture 2: Using the Observation operators

Large suite of Obs Operators & Quality Control

- CRTM, RTTOV satellite radiances
- Conventional atmospheric obs
- Aerosol optical depth
- Column retrievals
- And many more for NWP, Ocean, Space Weather…

```
observations:
   observers:
   - obs space:
   <the ioda part that we've just looked at>
   obs operator:
       name: ColumnRetrieval
       nlayers_retrieval: 34
       tracer variables:
       - volume_mixing_ratio_of_no2
       isApriori: false
       isAveragingKernel: true
       stretchVertices: topbottom
     obs error:
       covariance model: diagonal
```

$$\Delta x_a = \mathbf{B}\mathbf{M}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}}\left(\mathbf{H}\mathbf{M}\mathbf{B}\mathbf{M}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}} + \mathbf{R}\right)^{-1}\left(y_o - H(M(x_b))\right)$$

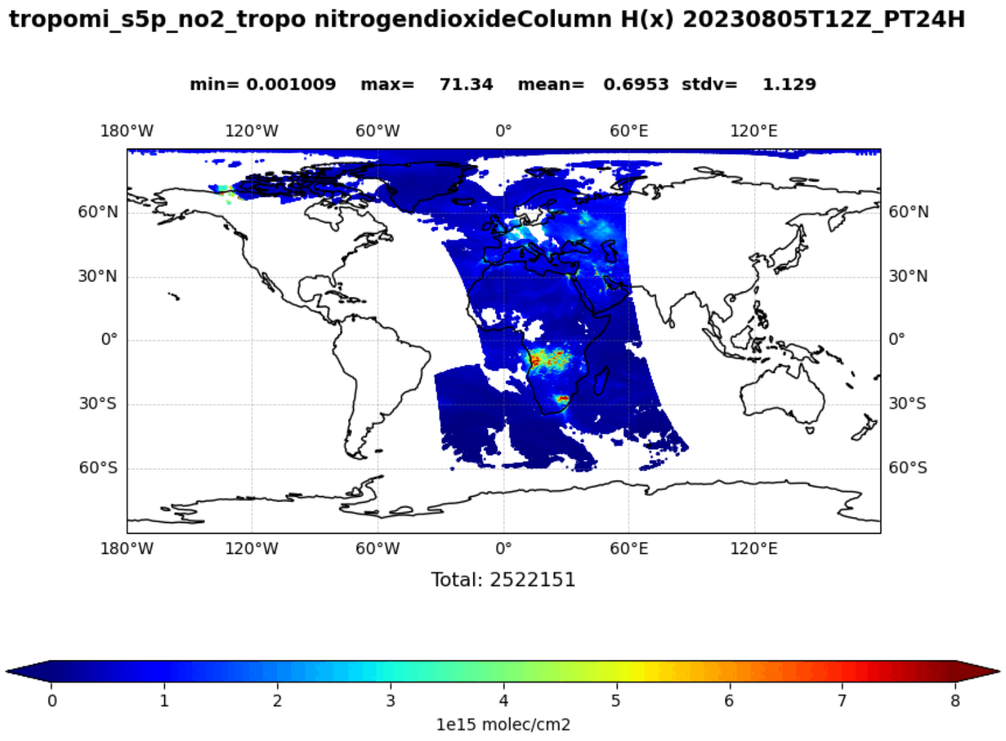


Lecture 2: Column retrieval operator

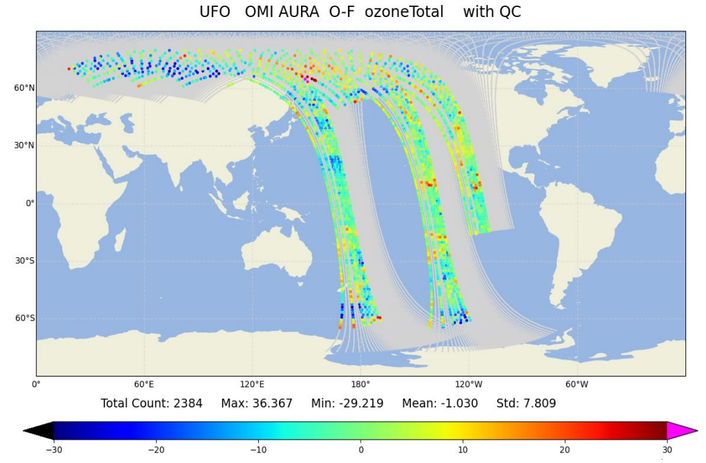
It computes the model equivalent to the observation using the retrieval equation as follows:

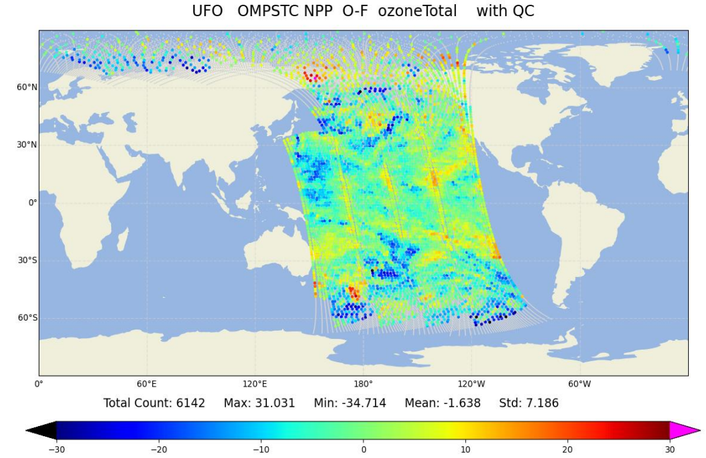
$$\mathbf{X}_c = \mathbf{A}\mathbf{X}_{\text{background}} + (\mathbf{I} - \mathbf{A})\mathbf{X}_{\text{apriori}}$$

The operator computes partial columns $\mathbf{X}_{\text{background}}$ on the retrieval vertical grid and then applies, if needed, the averaging kernel or scattering weights information $\mathbf{A}$. If needed as well, it can apply an apriori term $\mathbf{X}_{\text{apriori}}$. For each geoval profile the operator computes the partial or total column of any given atmospheric molecule concentration $\mathbf{X}_c$.



Operator adopted for NWP O3 operational use at NOAA NWS EMC





Lecture 2: Column retrieval operator

TropOMI CO

```
obs operator:
    name: ColumnRetrieval
    nlayers_retrieval: 50
    tracer variables: [volume_mixing_ratio_of_co]
    isApriori: false
    isAveragingKernel: true
    stretchVertices: topbottom
    model units coeff: 1e-6
```

OMPS O3

```
- obs operator:
    name: ColumnRetrieval
    nlayers_retrieval: 1
    tracer variables: [mole_fraction_of_ozone_in_air]
    isApriori: false
    isAveragingKernel: false
    totalNoVertice: true
    stretchVertices: topbottom
    model units coeff: 2.240013904035E-3
```

etrieval equation as

$\mathbf{x}_{apriori}$

trieval vertical grid

needed, the averaging kernel or scattering weights information $\mathbf{A}$. If needed as

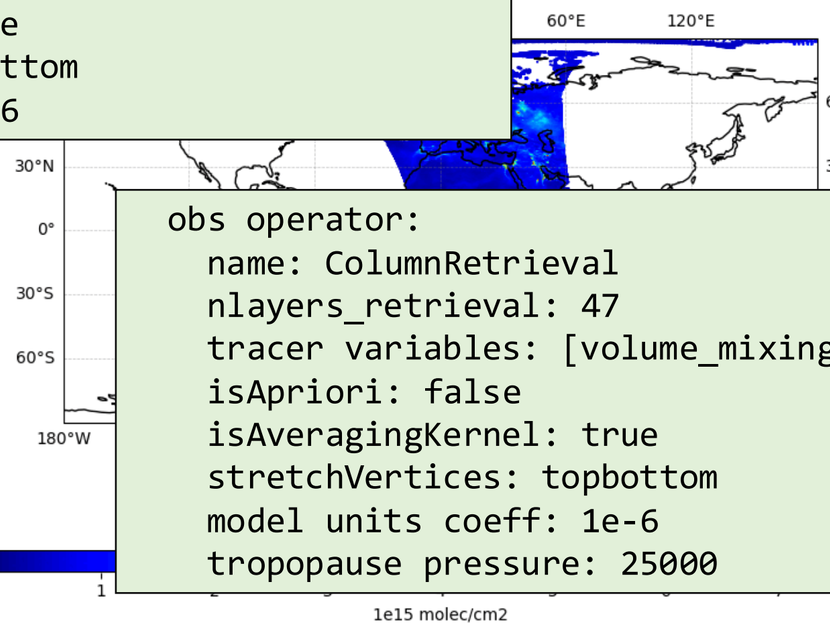
apriori term $\mathbf{X}_{apriori}$. For each geoval profile the operator computes the partial or total column of any

TropOMI NO2

```
obs operator:
    name: ColumnRetrieval
    nlayers_retrieval: 34
    tracer variables: [volume_mixing_ratio_of_no2]
    isApriori: false
    isAveragingKernel: true
    stretchVertices: topbottom
    model units coeff: 1e-6
```

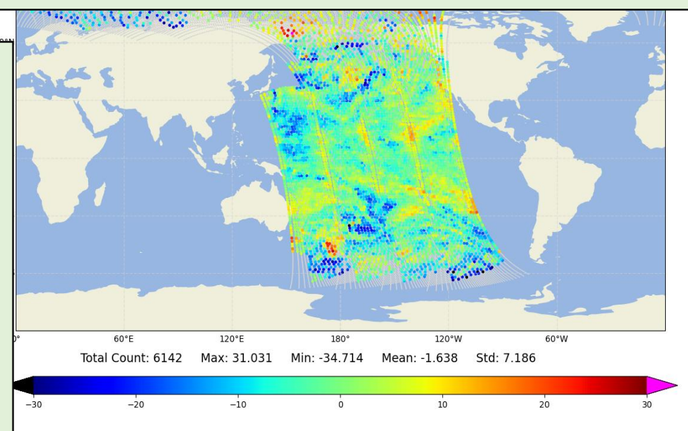
MOPITT CO

```
obs operator:
    name: ColumnRetrieval
    nlayers_retrieval: 10
    tracer variables: [volume_mixing_ratio_of_co]
    isApriori: true
    isAveragingKernel: true
    stretchVertices: topbottom
    model units coeff: 1e-6
```



TEMPO NO2

```
obs operator:
  name: ColumnRetrieval
  nlayers_retrieval: 47
  tracer variables: [volume_mixing_ratio_of_no2]
  isApriori: false
  isAveragingKernel: true
  stretchVertices: topbottom
  model units coeff: 1e-6
  tropopause pressure: 25000
```



Lecture 2: Aerosols operators

$$\Delta x_a = \mathbf{B}\mathbf{M}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}}\left(\mathbf{H}\mathbf{M}\mathbf{B}\mathbf{M}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}} + \mathbf{R}\right)^{-1}\left(y_o - H(M(x_b))\right)$$

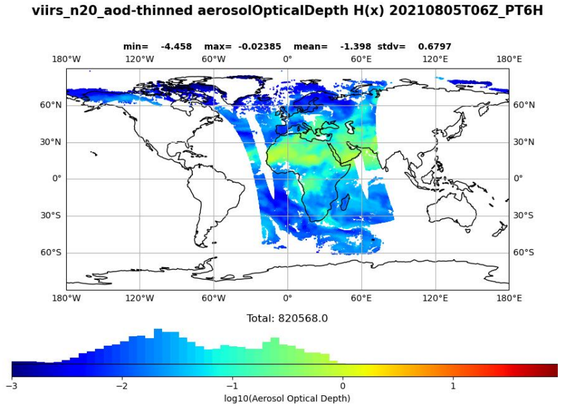
AOD CRTM:

- Extinction calculation within JEDI as part of the observation operator **H**
- Requires all aerosols variables in the state vector to compute AOD with RTM, could be expensive with EnVar at high resolution
- Prognostic variables directly updated to initialize the forecast.
- Different LUTs for optical properties can be loaded and tested within JEDI

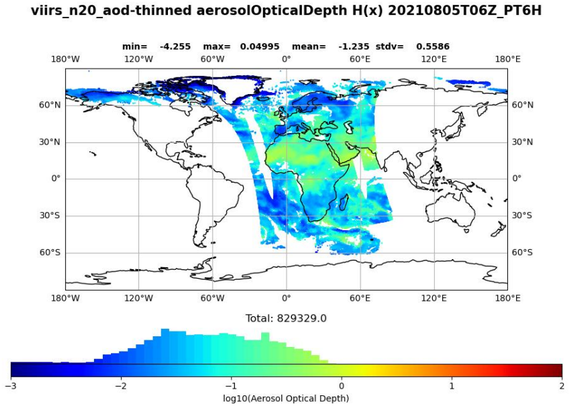
AOD Extinction:

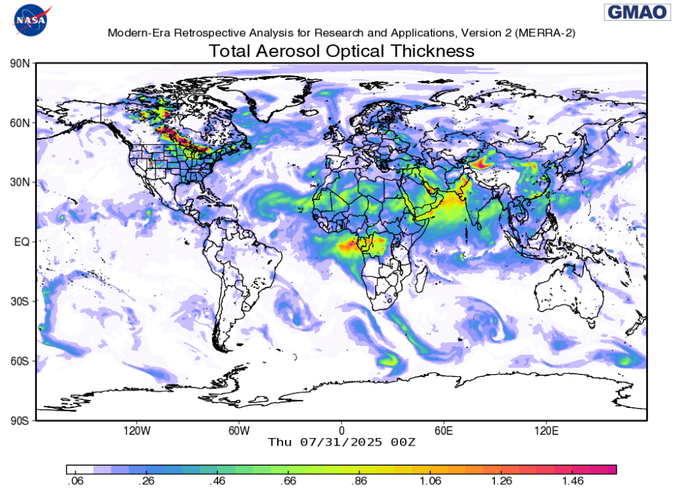
- Extinction calculation within GEOS (or other) as part of the forecast model **M**
- Doesn't requires all aerosols variables in the state vector therefore much cheaper for the JEDI part
- Prognostic variables are not updated through JEDI but a posteriori of the analysis

H(x) with default CRTM LUT



H(x) with NASA-GEOS LUT





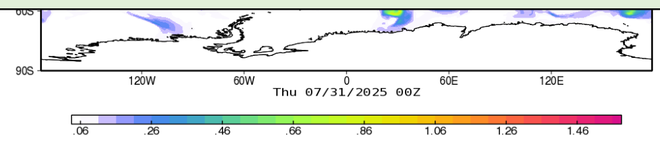
Lecture 2: Aerosols operators

VIIRS AOD with AODCRTM

```
observations:
   observers:
   - obs space:
       name: Aod
       obsdatain:
         engine:
           type: H5File
           obsfile: aod_viirs_npp_obs.nc4
       obsdataout:
         engine:
           type: H5File
           obsfile: aod_viirs_npp_3dvar.nc4
       simulated variables:
       - aerosolOpticalDepth
       channels: 4
     obs operator:
       name: AodCRTM
       Absorbers:
       - H2O
       - O3
       obs options:
         Sensor_ID: v.viirs-m_npp
         EndianType: little_endian
         CoefficientPath: Data/crtm/
         AerosolOption: aerosols_gocart_default
```

MODIS NNR AOD with AODExt

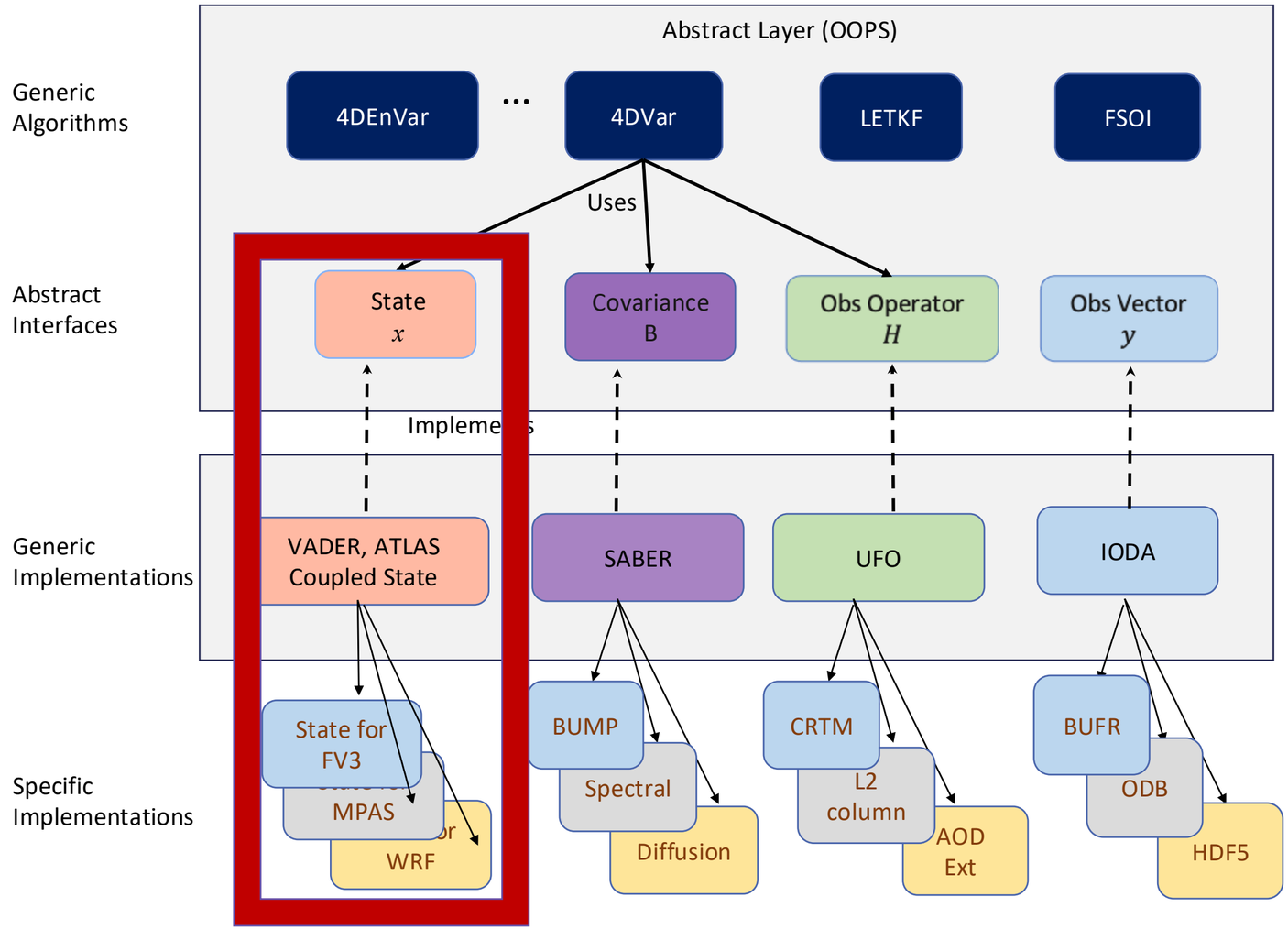
```
observations:
   observers:
   - obs space:
       name: Aod
       obsdatain:
         engine:
           type: H5File
           obsfile: nnr_aod_3ch_obs.nc4
       obsdataout:
         engine:
           type: H5File
           obsfile: nnr_aod_3ch_obs_out.nc4
       simulated variables:
       - aerosolOpticalDepth
       channels: 1, 3
     obs operator:
       name: AodExt
       nprofiles: 2
       bkg_wavelengths: [470.0,870.0]
       doing_log_transform_aod: true
       eps_for_log_transform_aod: 0.01
```



Lecture 2: Model interface

MODEL	TYPE
UFS	Global Atmosphere
GEOS	Global Atmosphere
NEPTUNE	Global atmosphere
MPAS	Global atmosphere
LFRic	Global atmosphere
Unified Model	Global atmosphere
UFS CAM	Regional atmosphere
MPAS-Regional	Regional atmosphere
WRF	Regional atmosphere
UFS GSDChem	Global constituents
GEOS-AERO	Global aerosols
UFS CAM-CMAQ	Regional air quality
MOM6	Global ocean
ROMS	Regional ocean
SIS2	Sea-ice
CICE6	Sea-ice
WW-III	Wave
NOAH-MP	Land and Snow
QG	Toy model
Lorenz 95	Toy model
Shallow Water	Toy model

$$\Delta x_a = \mathbf{B}\mathbf{M}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}}\left(\mathbf{H}\mathbf{M}\mathbf{B}\mathbf{M}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}} + \mathbf{R}\right)^{-1}\left(y_o - H(M(x_b))\right)$$

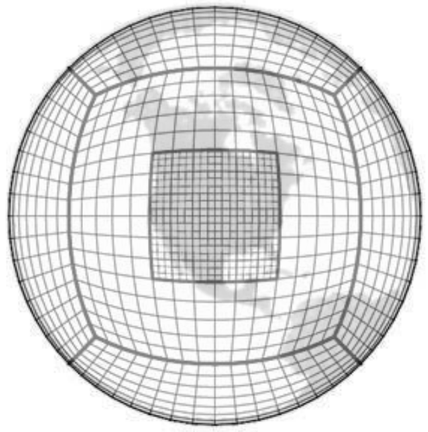


Lecture 2: Model interfaces

JEDI atmospheric composition modeling configurations for DA

NOAA UFS-Chem

Dycore/grid: FV3 – nested domain

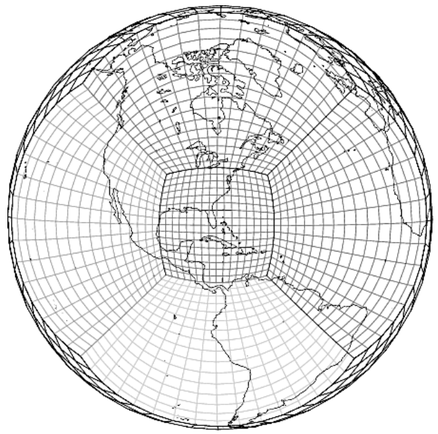


FV3-JEDI interface

JEDI AC Status: Tested and used with JEDI for regional and global AC applications (pre-operational)

NASA GEOS/GEOS-CF

Dycore/grid: FV3 – stretch grid



FV3-JEDI interface

JEDI AC Status: Tested and used with JEDI for global AC applications – tested with stretch grid

NCAR MPAS

Dycore/grid: MPAS – refined grid

MPAS-JEDI interface

JEDI Status: ongoing progress to develop JEDI-MPAS-GOCART for atmospheric composition
Reactive chemistry to come

GODDARD EARTH SCIENCES

Model interfaces

FV3-JEDI interface section

```
geometry:
  fms initialization:
    namelist filename: Data/fv3files/fmsmpp.nml
  akbk: Data/inputs/fv3files/akbk72.nc4
  npx: 13
  npy: 13
  npz: 72

state:
  datetime: 2020-09-03T18:00:00Z
  filetype: cube sphere history
  datapath: Data/inputs/geos_c12
  filename: geos_cf.bkg.%yyyy%mm%dd_%hh%MM%ssz.nc4
  state variables:
  - air_pressure_thickness
  - volume_mixing_ratio_of_no2
  - volume_mixing_ratio_of_no
  - volume_mixing_ratio_of_o3
  - air_pressure_at_surface
  - water_vapor_mixing_ratio_wrt_moist_air
  field io names: &field_io_names
    volume_mixing_ratio_of_no2: NO2
    air_pressure_thickness: DELP
    volume_mixing_ratio_of_no: "NO"
    volume_mixing_ratio_of_o3: O3
    air_pressure_at_surface: PS
    water_vapor_mixing_ratio_wrt_moist_air: SPHU
```

MPAS-JEDI interface section

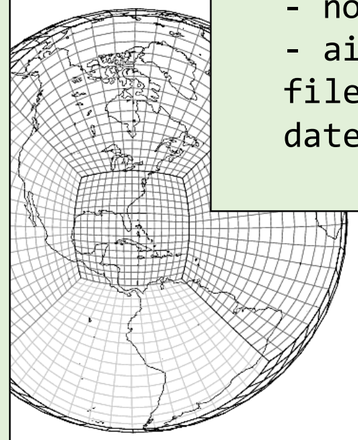
```
geometry:
  nml_file: "namelist.atmosphere_2018041500"
  streams_file: "streams.atmosphere”

state:
  state variables:
  - air_temperature
  - water_vapor_mixing_ratio_wrt_moist_air
  - eastward_wind
  - northward_wind
  - air_pressure_at_surface
  filename: “restart.2018-04-15_00.00.00.nc"
  date: '2018-04-15T00:00:00Z’
```

sition mod

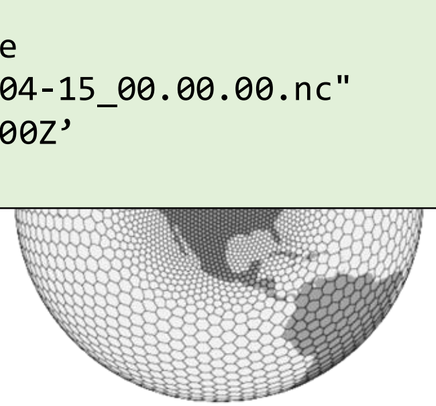
A GEOS/GE

grid: FV3 – s



3-JEDI interface

atus: Tested and used
r global AC applications
d with stretch grid



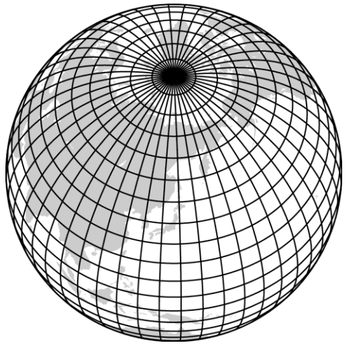
MPAS-JEDI interface

JEDI Status: ongoing progress to develop JEDI-MPAS-GOCART for atmospheric composition
Reactive chemistry to come

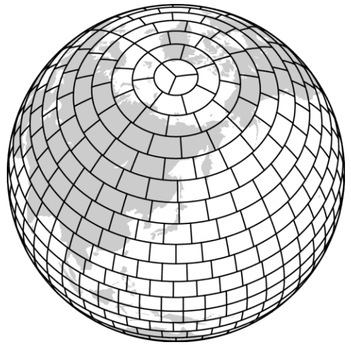
GODDARD EARTH SCIENCES

Lecture 2: Model interfaces: Versatile Implementation for Native Data (VIND)

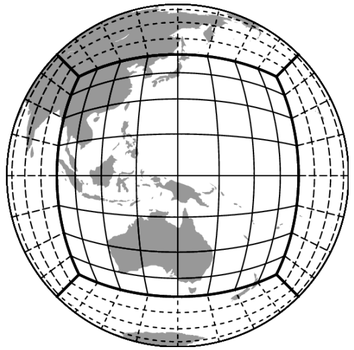
Regular LatLon



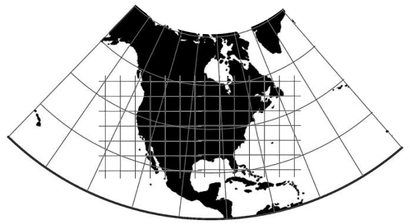
Gaussian



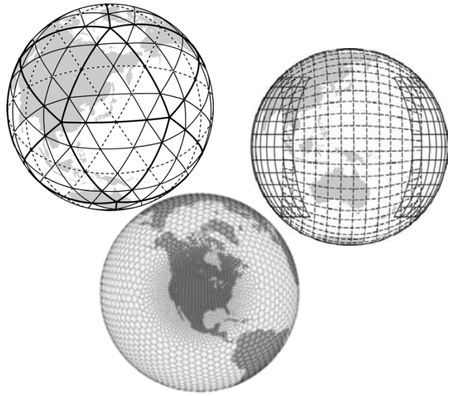
Cubed Sphere



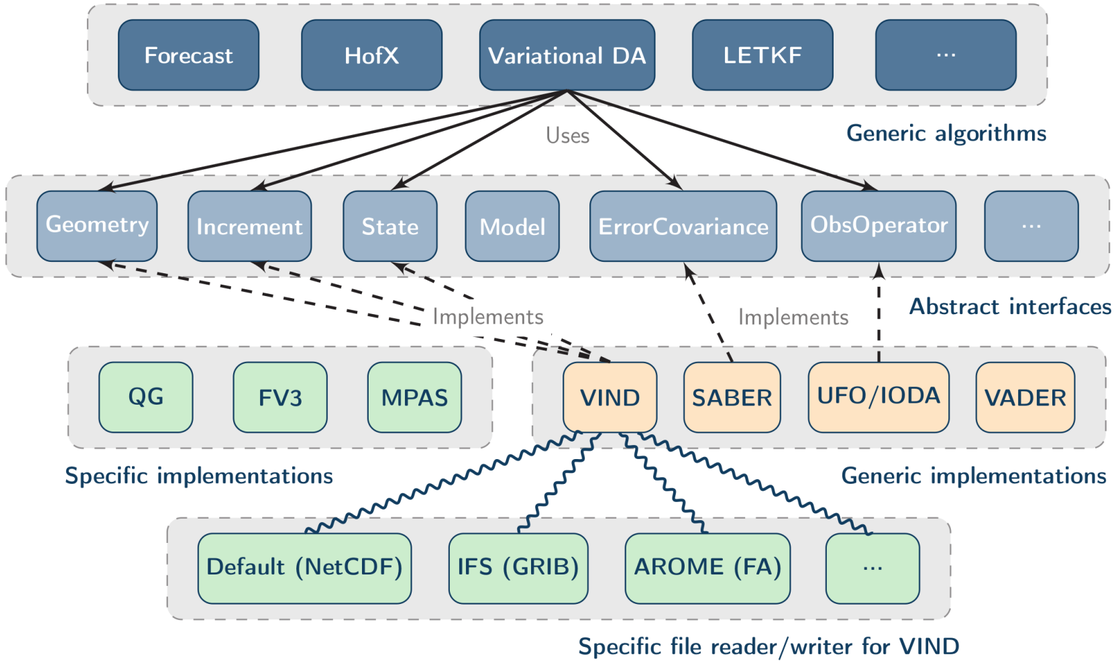
Limited areas models
Including Lambert Conformal



And many other geometries:
Structured and unstructured



VIND builds on ATLAS, an open-source C++ library from ECMWF, which provides data structures for handling fields on a wide range of global and regional grid geometries.



Courtesy: Benjamin Menetrier

Lecture 2: Model interfaces: Versatile Implementation for Native Data (VIND)

VIND WRF-Chem geometry

```
geometry:
  function space: StructuredColumns
  partitioner: "checkerboard"
  grid:
    type : "regional"
    nx : 222
    ny : 198
    dx : 4000.0
    dy : 4000.0
    lonlat(centre) : [-75.1618,42.90771]
    #lonlat(xmin,ymin): [-81.634705,38.214104]
    projection :
        type : "lambert_conformal_conic"
        longitude0 : -97.0
        latitude0 : 39.0
        latitude1  : 30.0
        latitude2  : 60.0
        radius: 6370000
```

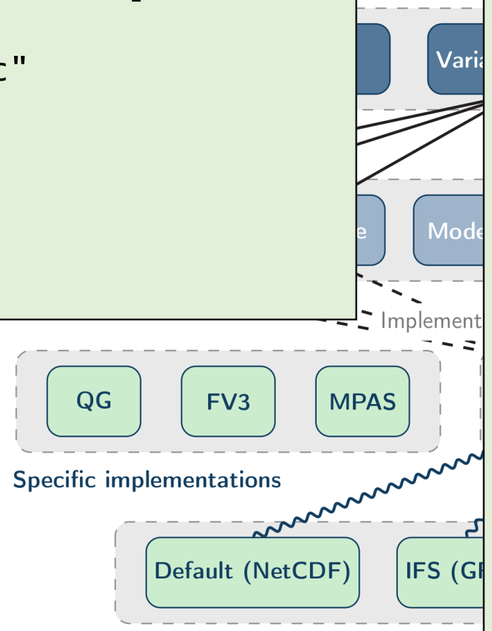
VIND LatLon geometry

```
geometry:
  function space: StructuredColumns
  grid:
    type: regular_lonlat
    nx: 30
    ny: 20
```

VIND regional rotated LatLon geometry

```
geometry: &geom
  function space: StructuredColumns
  grid:
    type: structured
    xspace:
      type: linear
      N : 351
      start : -35.0
      end :  35.0
    yspace:
      type: linear
      N: 271
      start: -27.0
      end : 27.0
    projection :
      type: rotated_lonlat
      north_pole: [190, 39]
    y_numbering: -1
```

on a wide range of global and regional grid geometries.



GODDARD EARTH SCIENCES

Lecture 2: Model interfaces: Versatile Implementation for Native Data (VIND)

VIND WRF-Chem geometry

```
geometry:
  function space: StructuredColumns
  partitioner: "checkerboard"
  grid:
    type : "regional"
    nx : 222
    ny : 198
    dx : 4000.0
    dy : 4000.0
    lonlat(centre) : [-75.1618,42.90771]
    #lonlat(xmin,ymin): [-81.634705
    projection :
        type : "lambert
        longitude0 : -9
        latitude0 : 39.0
        latitude1  : 30.0
        latitude2  : 60.0
        radius: 6370000
```

VIND LatLon geometry

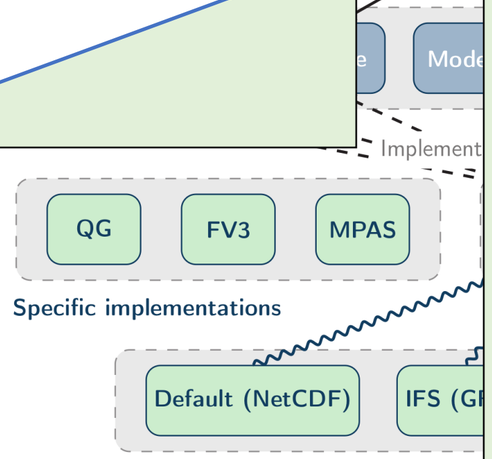
```
geometry:
  function space: StructuredColumns
  grid:
    type: regu                    t
    nx:
```

Not covered in this tutorial

onal rotated LatLon geometry

```
                pace: StructuredColumns
    type: structured
    xspace:
      type: linear
      N : 351
      start : -35.0
      end :  35.0
    yspace:
      type: linear
      N: 271
      start: -27.0
      end : 27.0
    projection :
      type: rotated_lonlat
      north_pole: [190, 39]
    y_numbering: -1
```

on a wide range of global and regional grid geometries.



GODDARD EARTH SCIENCES

Practical demonstration 2

- https://mer-a-o.github.io/howtojedi/lecture2/

- Understand how observation operators and model interfaces are combined
- Running an H(x) experiment
- Looking at the results

Lecture 3: Data assimilation algorithms

DA algorithms available:

- 3DVar, 3DVar-FGAT, 4DEnVar, 4DVar, …
- LETKF, LGETKF

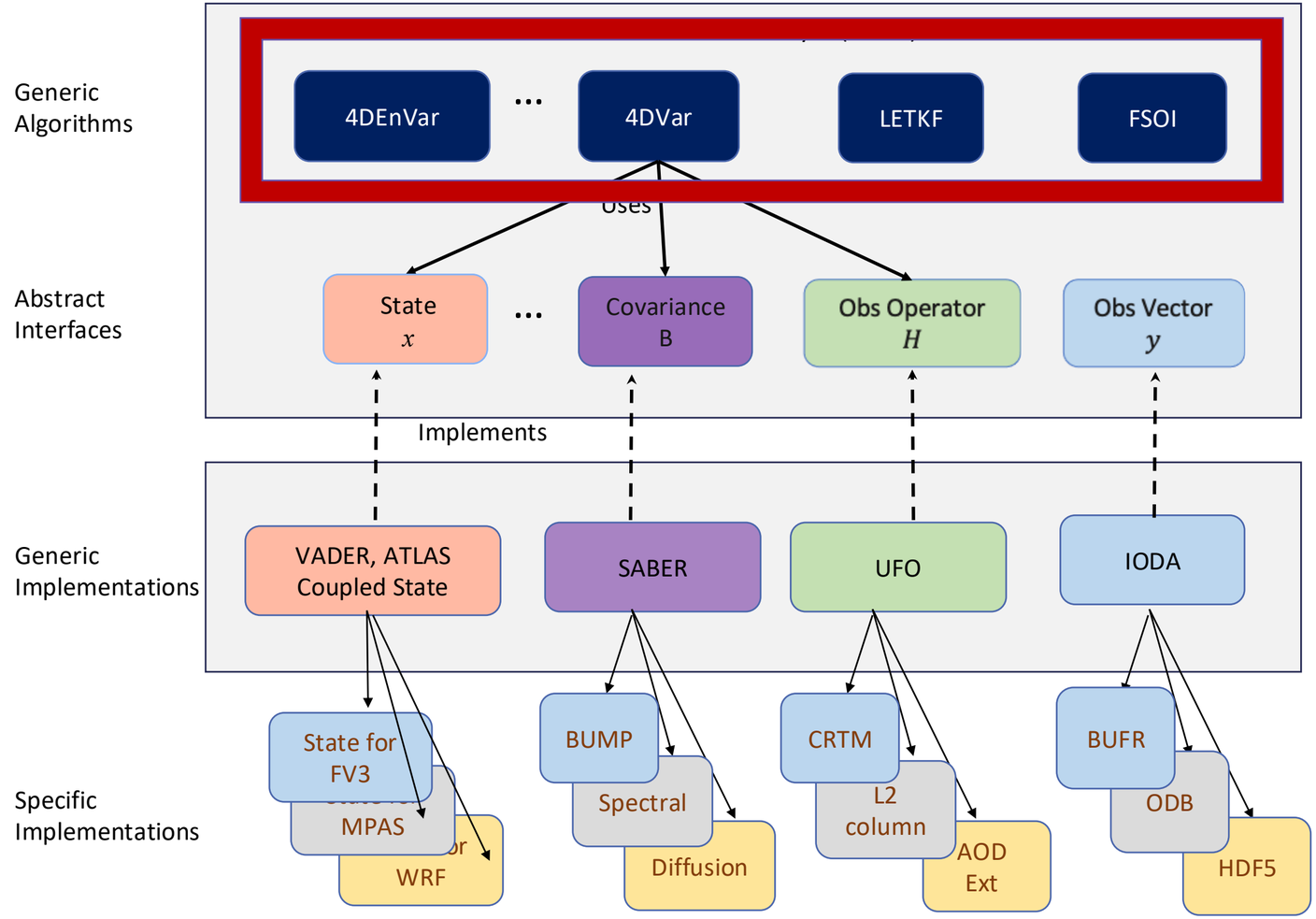
```
cost function:
  cost type: 3D-Var

  analysis variables:
  - volume_mixing_ratio_of_no2

  time window:
    begin: 2020-09-03T15:00:00Z
    length: PT6H

variational:
  minimizer:
    algorithm: DRPCG
  iterations:
  - ninner: 10
    gradient norm reduction: 1e-10
```

$$\Delta x_a = \mathbf{B}\mathbf{M}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}}\left(\mathbf{H}\mathbf{M}\mathbf{B}\mathbf{M}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}} + \mathbf{R}\right)^{-1}\left(y_o - H(M(x_b))\right)$$

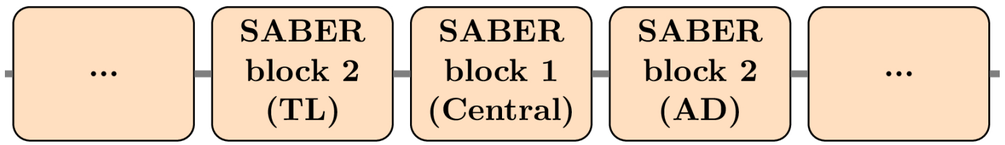


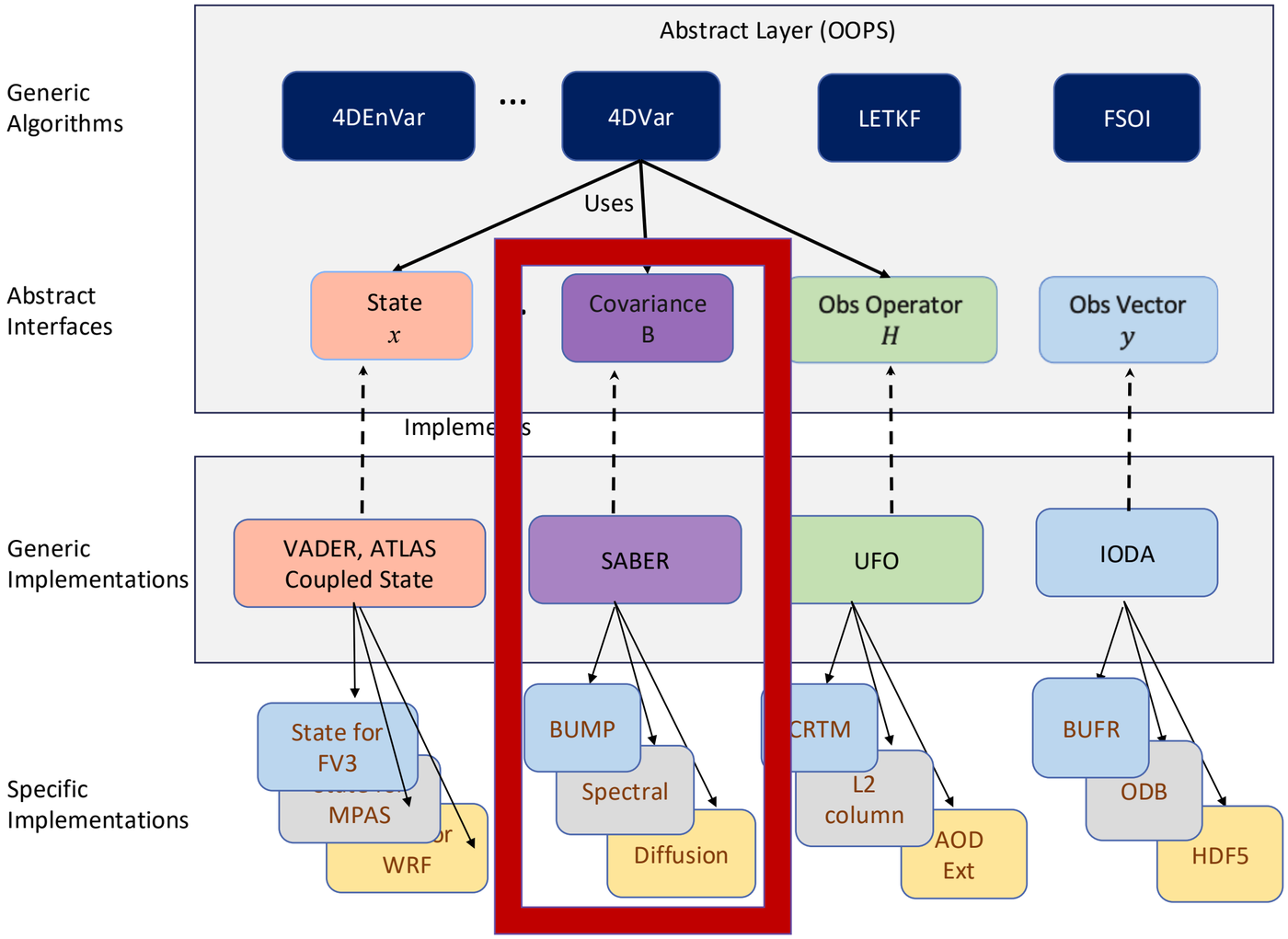
Lecture 3: Background error

Background Error Covariances represented as a series of generic blocks:

- Balance operator, change of variables
- Auto-correlation operator (BUMP, spectral, diffusion…)
- Hybrid static-ensemble
- Option to import GSI static B

$$\Delta x_a = \mathbf{B}\mathbf{M}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}}\left(\mathbf{H}\mathbf{M}\mathbf{B}\mathbf{M}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}} + \mathbf{R}\right)^{-1}\left(y_o - H(M(x_b))\right)$$





Lecture 3: Background error

SABER with BUMP Covariances

```
background error:
   covariance model: SABER

   saber central block:
     saber block name: BUMP_NICAS
     read:
       io:
         files prefix: Data/bump/
         alias:
         - in code: volume_mixing_ratio_of_no2
           in file: fixed_2500km_0.3
       drivers:
         multivariate strategy: univariate
         read local nicas: true

   saber outer blocks:
   - saber block name: StdDev
     read:
       model file:
         datetime: 2020-09-03T18:00:00Z
         set datetime on read: true
         filetype: cube sphere history
         datapath: Data/inputs/geos_c12
         filename: geos_cf.stddev.nox_50_o3_25.nc4
         field io names: *field_io_names
```

$\Delta x_a =$

SABER with Diffusion Covariances

```
background error:
  covariance model: SABER

  saber central block:
    saber block name: diffusion
    calibration:
      normalization:
        iterations: 1000
      groups:
      - horizontal:
          fixed value: 2500e3
      - vertical:
          levels: 72
          fixed value: 1.0
          as gaussian: true

  saber outer blocks:
    - saber block name: StdDev
      read:
        model file:
          datetime: 2020-09-03T18:00:00Z
          set datetime on read: true
          filetype: cube sphere history
          datapath: Data/inputs/geos_c12
          filename: geos_cf.stddev.nox_50_o3_25.nc4
          field io names: *field_io_names
```

Lecture 3: Background error

SABER with BUMP Covariances

```
background error:
   covariance model: SABER

   saber central block:
     saber block name: BUMP_NICAS
     read:
       io:
         files prefix: Data/bump/
         alias:
         - in code: volume_mixing_ratio_of_no2
           in file: fixed_2500km_0.3
       drivers:
         multivariate strategy: univariate
         read local nicas: true

   saber outer blocks:
   - saber block name: StdDev
     read:
       model file:
         datetime: 2020-09-03T18:00:00Z
         set datetime on read: true
         filetype: cube sphere history
         datapath: Data/inputs/geos_c12
         filename: geos_cf.stddev.nox_50_o3_25.nc4
         field io names: *field_io_names
```

SABER with Identity matrix

```
background error:
   covariance model: SABER
   saber central block:
     saber block name: ID
```

SABER with Diffusion Covariances

```
background error:
  covariance model: SABER

  saber central block:
                                    ffusion
                                    500e3
      - vertical:
          levels: 72
          fixed value: 1.0
          as gaussian: true

  saber outer blocks:
    - saber block name: StdDev
      read:
        model file:
          datetime: 2020-09-03T18:00:00Z
          set datetime on read: true
          filetype: cube sphere history
          datapath: Data/inputs/geos_c12
          filename: geos_cf.stddev.nox_50_o3_25.nc4
          field io names: *field_io_names
```

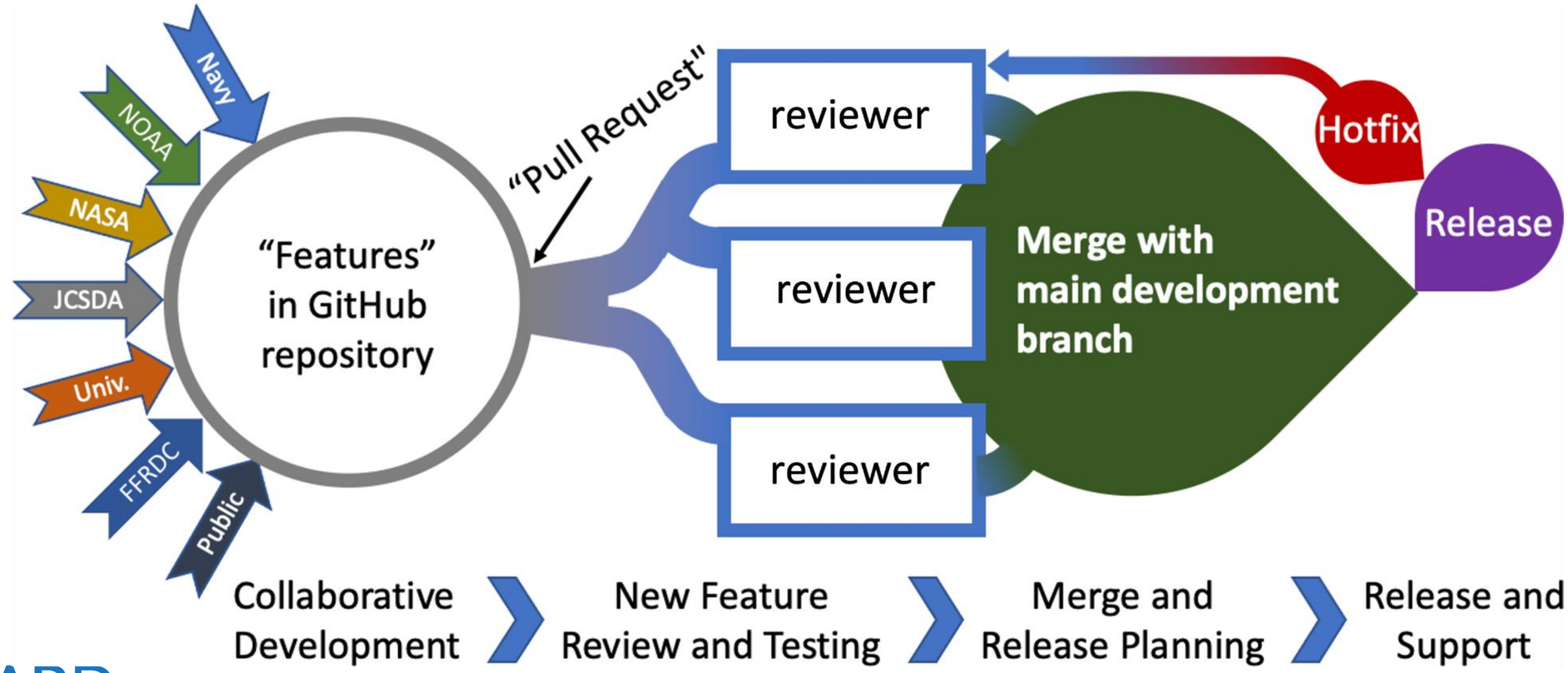
Practical demonstration 3

- https://mer-a-o.github.io/howtojedi/lecture3/

- Understand the additional elements needed to go from H(x) to Data Assimilation in JEDI
- Running a variational Data Assimilation experiment
- Looking at the results

Closing remarks: CI testing and improving JEDI

- **Test every feature** (generic and specific) or it can **silently break** as other parts evolve.
- **CI tests** prevent regressions and act as a **catalog + runnable examples** (YAML + expected input/outputs).
- **Merge rule:** CI must pass **and** a **human reviewer** approves.
- **Next tutorial?:** *JEDI for developers* (adding features + writing tests + CI/review workflow).



Cheng Dang (JCSDA)

Using existing JEDI builds on Discover

- Because of the JEDI modularity, you don't necessarily need to build you own JEDI system to use JEDI

  see https://mer-a-o.github.io/howtojedi/discover/

- You can point to your colleagues builds or the SI team builds:

/discover/nobackup/projects/gmao/advda/jedi_bundles_sles15/current_pinned_jedi_bundle/

- This allows better coordination for joint testing, deployment and then developments.

Closing remarks

- Bi-weekly JEDI office hours starting next week.

Microsoft Teams Need help?
Join the meeting now
Meeting ID: 294 485 870 585 94
Passcode: 4vS7kb6q

Dial in by phone
+1 256-715-9946,,539909751# United States, Huntsville
Find a local number
Phone conference ID: 539 909 751#

GODDARD EARTH SCIENCES