Real-Time Ocean Monitoring at the Joint Center for Satellite Data Assimilation: A Testbed for Ice-Ocean DA Development and Evaluation

Travis Sluka [1,2], Guillaume Vernieres [1,2], Rahul Mahajan [3]   [1]UCAR/JCSDA, [2]NOAA, [3]NASA GMAO

Overview

The Joint Center for Satellite Data Assimilation (JCSDA) is working with NOAA and NASA to develop improved ocean data assimilation methods for the next-generation seasonal to sub-seasonal prediction systems. These systems will be based on JCSDA's **Joint Effort for Data Assimilation Integration (JEDI)**, a unified DA framework designed to work across various domains and models. As a demonstration and development testbed for the ocean component, JCSDA is creating a real-time daily ocean monitoring system based on **S**ea-ice **O**cean **C**oupled **A**ssimilation (SOCA) .

System Goals

- Real-time demonstration of functioning ocean DA within JEDI framework
- **agile development** – ensure an up-to-date working version can be released at any time, yet also automatically incorporating the latest changes from multiple developers and repositories
- **NOT** intended to compete with current operational monitoring systems

Agile Development

Development within JEDI projects is very fast paced as upstream repositories are often updated several times a day. It is not feasible for a human to check every change to every repository (**figure 1**) for their impacts on system performance. This process has been automated using common agile development practices used in the software development community.

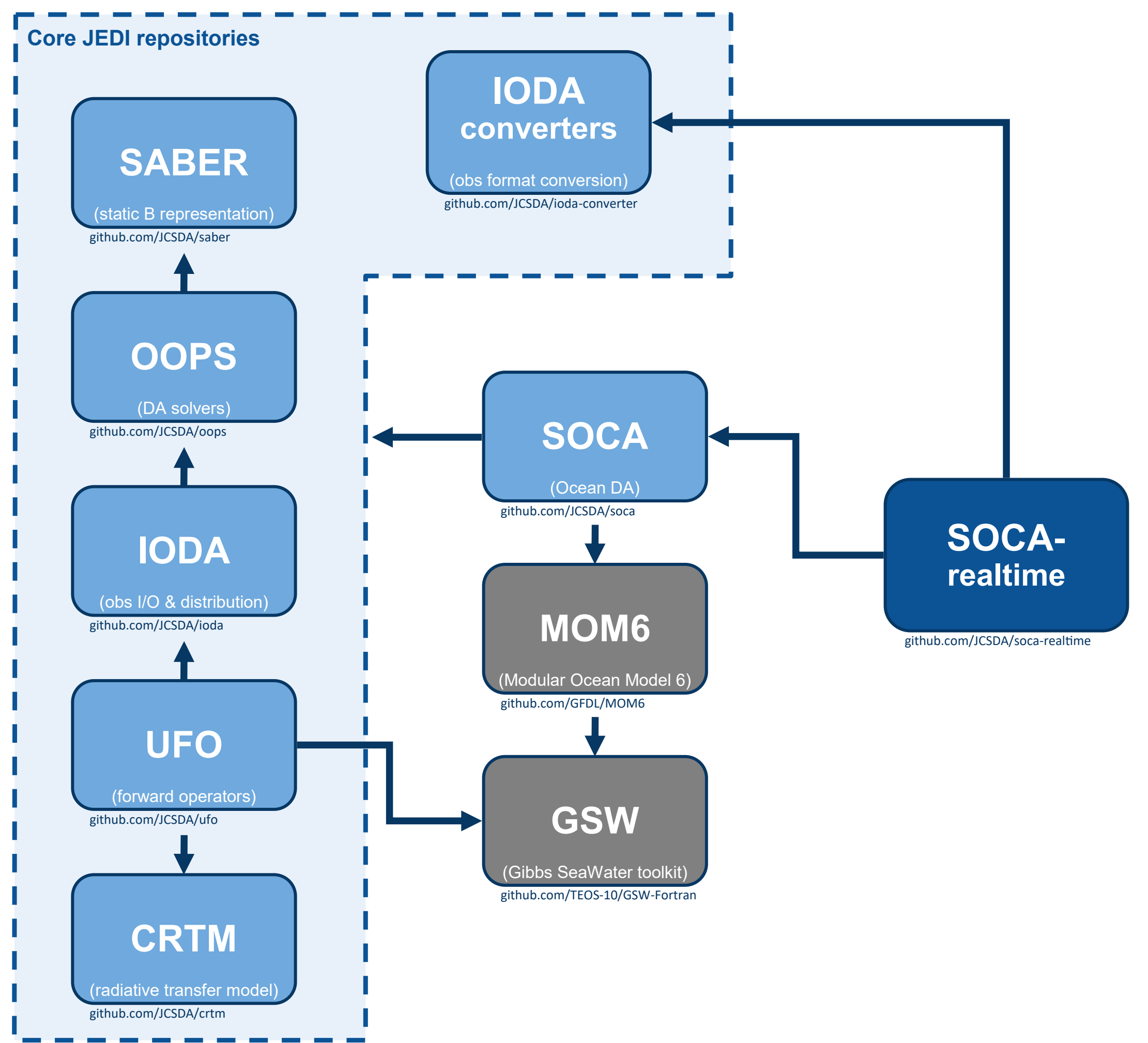


Figure 1 Some of the dependencies among the GitHub repositories used by the real-time system. The complex dependencies and frequent updates requires a robust continuous delivery pipeline to keep up with development.

Agile Development Terminology

- **Continuous integration** – automating the process of integrating code changes from multiple contributors, using GitHub, TravisCI, and AWS
- **Continuous delivery** – code changes are automatically prepared and tested for a release to the production code (although the push to production is still a manual task)
- **Continuous deployment** – continuous delivery, plus the use of metrics that if passed, automatically push code to production level *(not implemented here yet)*

Benefits of a continuous delivery pipeline

- Automation helps keep up with fast paced changes
- Building, testing, and releasing are less work for the humans (don't those humans have enough work to do already?)
- The latest version of the code always works.
- Code changes can easily make its way into the production run on the same day.

Configuration

Model

- MOM6/SIS2 ocean/ice model from NOAA/GFDL
- 75 hybrid vertical levels
- Horizontal resolution 1 degree
  (0.5, 0.25 and regional 0.4 degree are ready to test as well)

Observations

The latest observations from several near real-time operational streams are downloaded nightly, thinned, and converted to formats readable by JEDI, coverage from an example date is shown in **figure 2.**

- **satellite sea surface temperature** – from NESDIS Advanced Clear Sky Processor for Ocean (ACSPO). VIIRS, AVHRR, MODIS, AHI, and ABI platforms are available (though only AVHRR used for current initial tests)
- **absolute dynamic topography** – various altimeters (e.g. Jason-2/3, Sentinel-3a, Cryosat-2, SARAL) from NESDIS/RADS using XGM2016 geoid model
- **sea surface salinity** – from SMAP satellite
- **insitu T/S profiles** - ARGO, XBT, CTD, and MRB profiles from Fleet Numerical Meteorology and Oceanography Center (FNMOC)

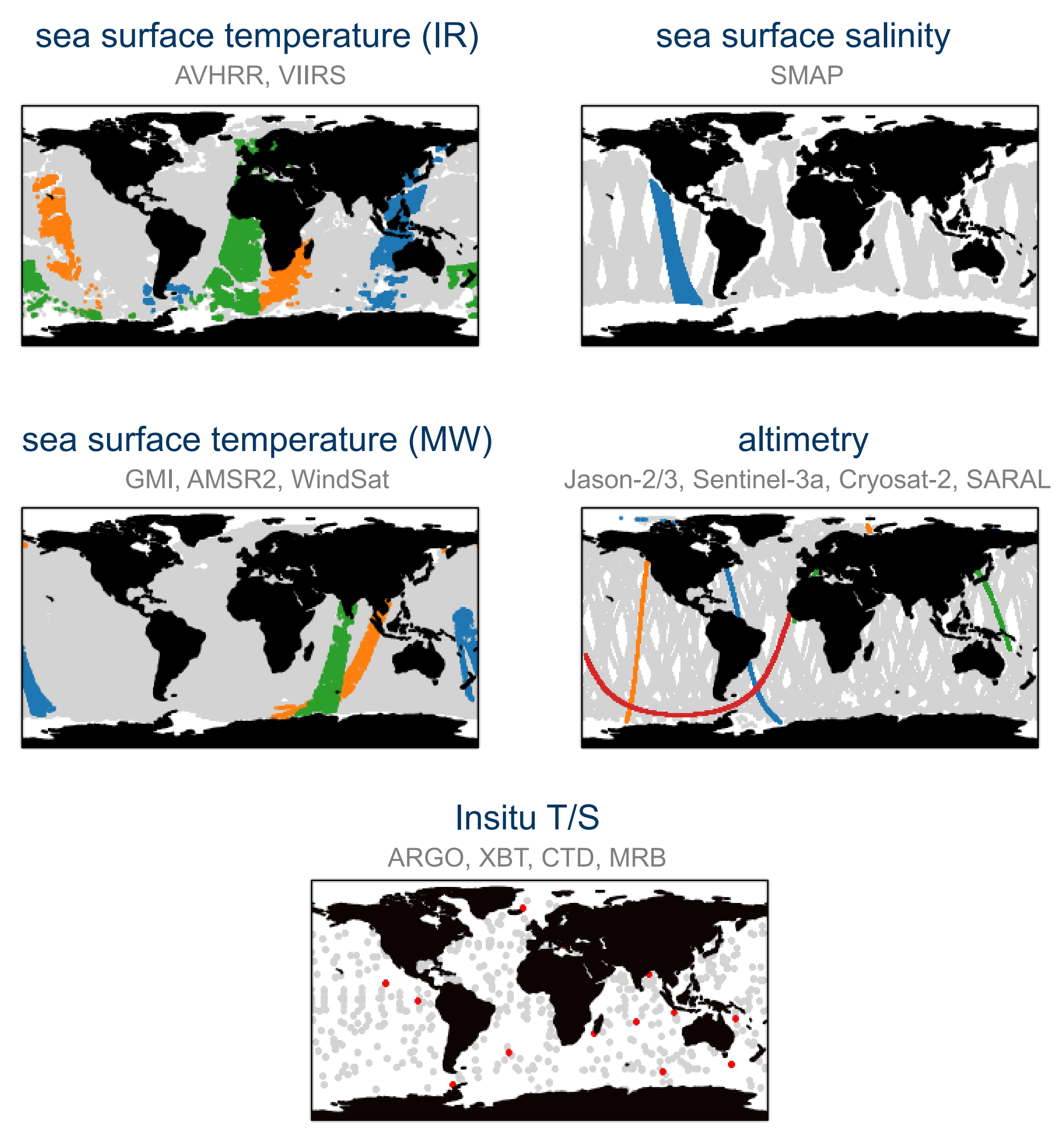

sea surface temperature (IR)
AVHRR, VIIRS
sea surface salinity
SMAP
sea surface temperature (MW)
GMI, AMSR2, WindSat
altimetry
Jason-2/3, Sentinel-3a, Cryosat-2, SARAL
Insitu T/S
ARGO, XBT, CTD, MRB

Figure 2 Coverage from a representative set of observations assimilated. One day of observations shown in gray, one hour shown in various colors for each platform

Data assimilation

- JEDI system used for unified forward operators, observation database IO, variational solvers, and static background error representation.
- Observation-space 3DVAR is currently used, though JEDI provides straightforward transition to 3DVAR-FGAT, EnKF, and EnVar as well
- Multivariate covariance between T/S/SSH (Weaver et al, 2006)

Continuous Delivery Pipeline

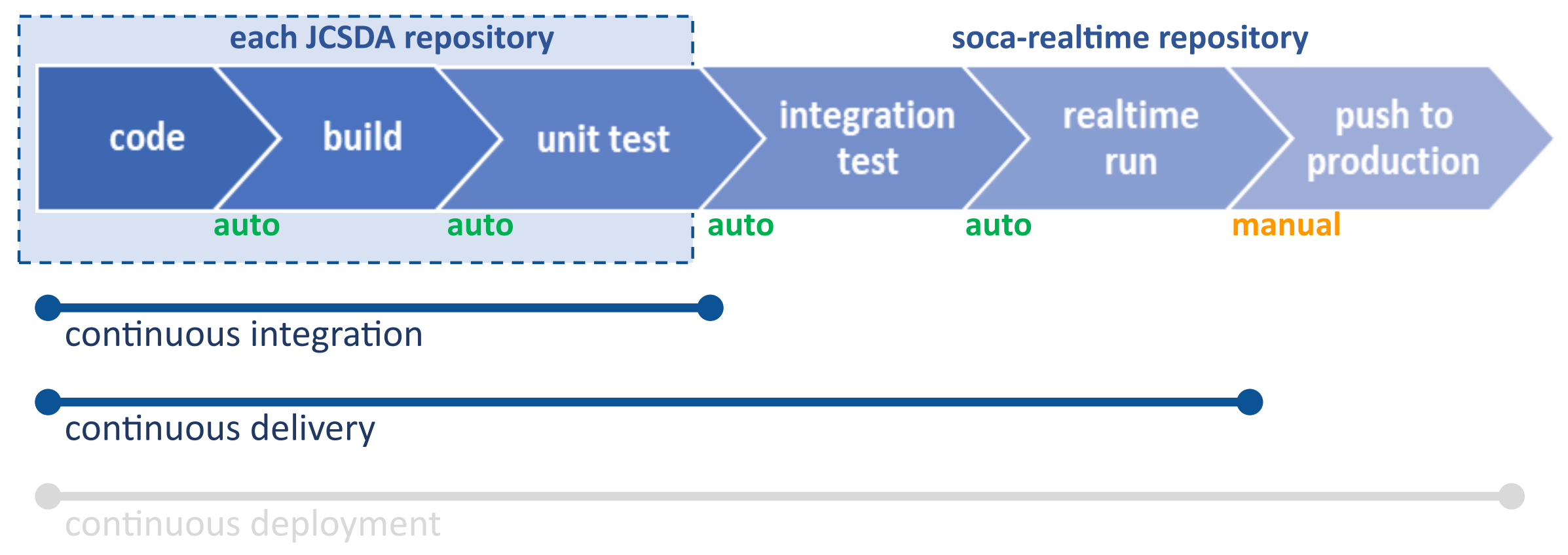


Figure 3 The continuous delivery pipeline. After code in any repository is pushed to GitHub it is automatically built and tested. Once the code changes are merged into the development branch, soca-realtime will use it for automatic tests and parallel production runs.

Continuous integration

Whenever any code is to be merged into any repository's development branch in the JEDI framework or SOCA, a pull request is issued on GitHub. This triggers the first level of automatic steps:

1) Code is built on TravisCI and AWS
2) Unit tests for the repository are run

Continuous Delivery

Only after the above automatic steps pass can code be merged into the main development branch. Then, on a daily basis the following steps are performed for soca-realtime

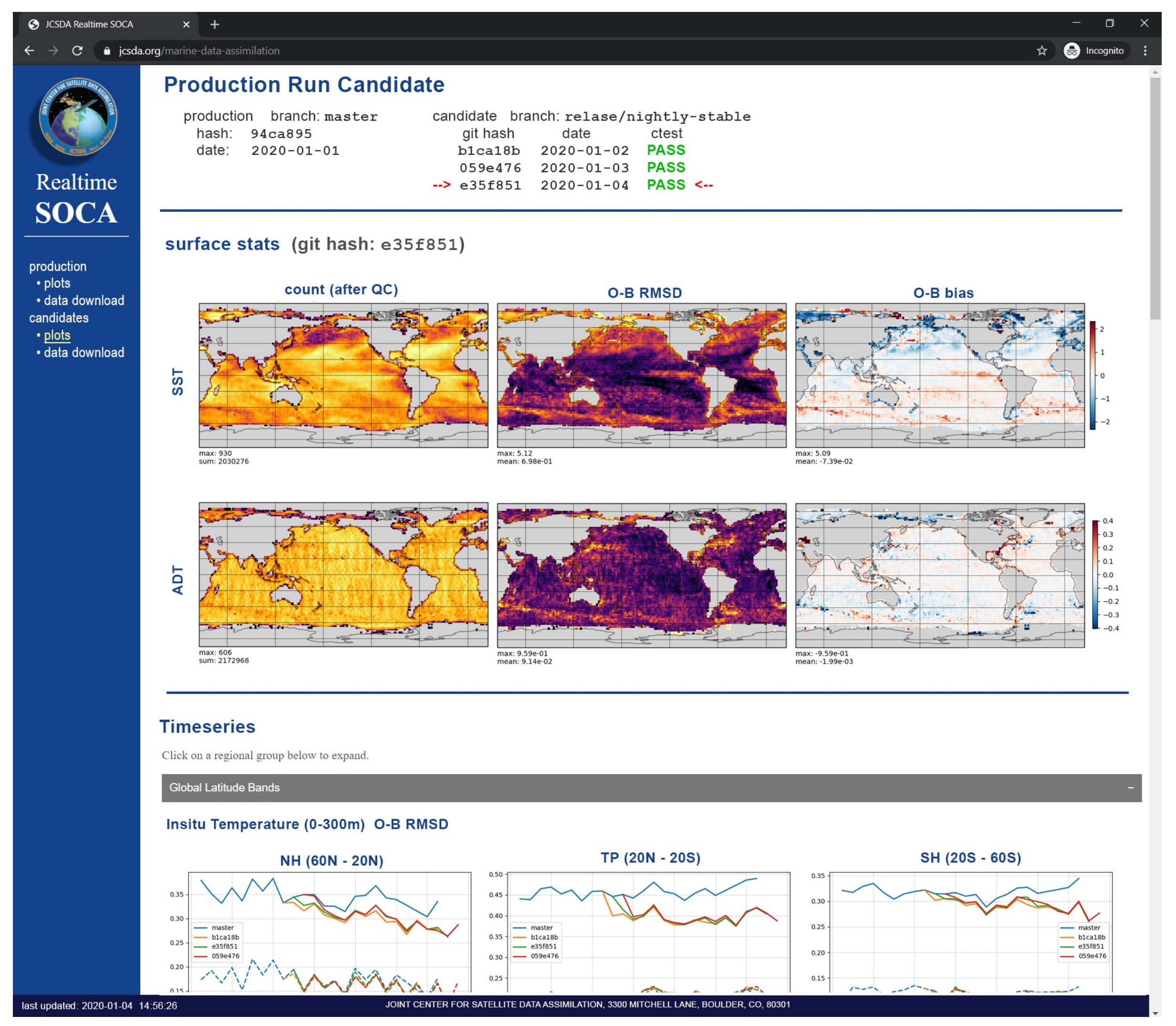
1) Integrated code build, using all the latest repositories in **figure 1**
2) Integration testing
3) Realtime and 14-day retrospective run
4) Evaluation of performance metrics

Having a useful continuous delivery/deployment pipeline relies on creating <u>good tests and useful validation metrics</u>. If these are done well, the entire pipleline can be completely automated.

Evaluation:

Every night the latest production and development streams are updated to a website for evaluation. Various observation space metrics are used (e.g. O-A and O-B rmsd and bias) **Figure 4**

- **Production level** — current master branch on GitHub
- **Nightly stable branch** — development branch on GitHub, assimilation is restarted from 2 weeks in the past whenever the code changes, in order to provide enough overlap with the production run for useful metrics



JCSDA Realtime SOCA
jcsda.org/marine-data-assimilation

Realtime SOCA
production
- plots
- data download
candidates
- plots
- data download

Production Run Candidate

```
production   branch: master          candidate  branch: relase/nightly-stable
   hash:   94ca895                        git hash     date        ctest
   date:   2020-01-01                     b1ca18b    2020-01-02   PASS
                                          059e476    2020-01-03   PASS
                                      --> e35f851    2020-01-04   PASS <--
```

surface stats (git hash: e35f851)

Timeseries

Click on a regional group below to expand.

Global Latitude Bands

Insitu Temperature (0-300m) O-B RMSD

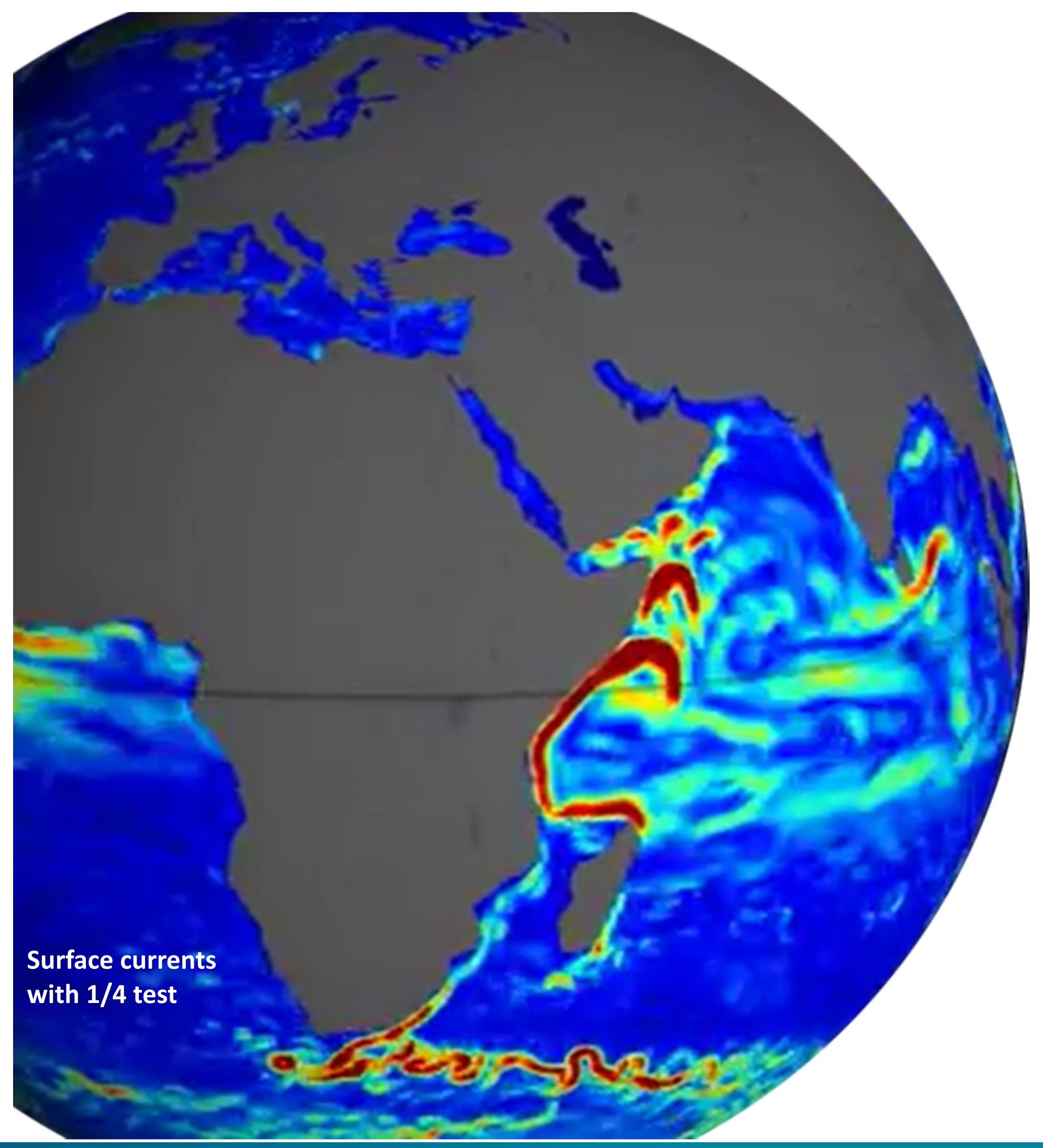
last updated: 2020-01-04 14:56:26   JOINT CENTER FOR SATELLITE DATA ASSIMILATION, 3300 MITCHELL LANE, BOULDER, CO, 80301

Figure 4 Example of the diagnostic evaluation website showing that updated code in the candidate branches performs better than the production version and ready for promotion to production.

Future Work

A robust continuous delivery pipeline provides rapid deployment of new code to the production version, providing immediate turnaround of performance metrics and quickly exposing unforeseen issues with "operational" cycling. It also reduces workload on developers by automating testing and ensuring there is always a latest working version of the code. Several upgrades are planned as the system evolves:

- Upgrade to global 1/4 degree, and 1/25th degree regional Gulf of Mexico
- Upgrade to ensemble DA methods (EnKF, 4DEnVar)
- Diagnostics and fields made publicly available at www.jcsda.org
- Metrics for automatic acceptance of production code (i.e. a transition from **continuous delivery** to **continuous deployment**)



Travis Sluka: tsluka@ucar.edu