Collaborative Tools 1

- ***Agile Project Management and Collaborative Workflow***
  - ✦***git/GitHub***
  - ✦***git-flow***
  - ✦***ZenHub***
- ***Documentation***
  - ✦***Sphinx/ReadTheDocs*** ***(high-level manuals, how-to's, etc)***
  - ✦***Doxygen*** ***(low-level code details)***
- ***Computing Environment***
  - ✦***Software containers***
  - ✦***Cloud Computing (AWS)***

Mark Miesch (JCSDA)

JEDI Academy – 13-16 Nov, 2018
College Park, MD

http://wookiepedia.com

Academy website

http://academy.jcsda.org/nov2018

Already has

- Instructions for accessing AWS
- Doxygen documentation for fv3-bundle

We will add further content throughout the week, including slides from presentations

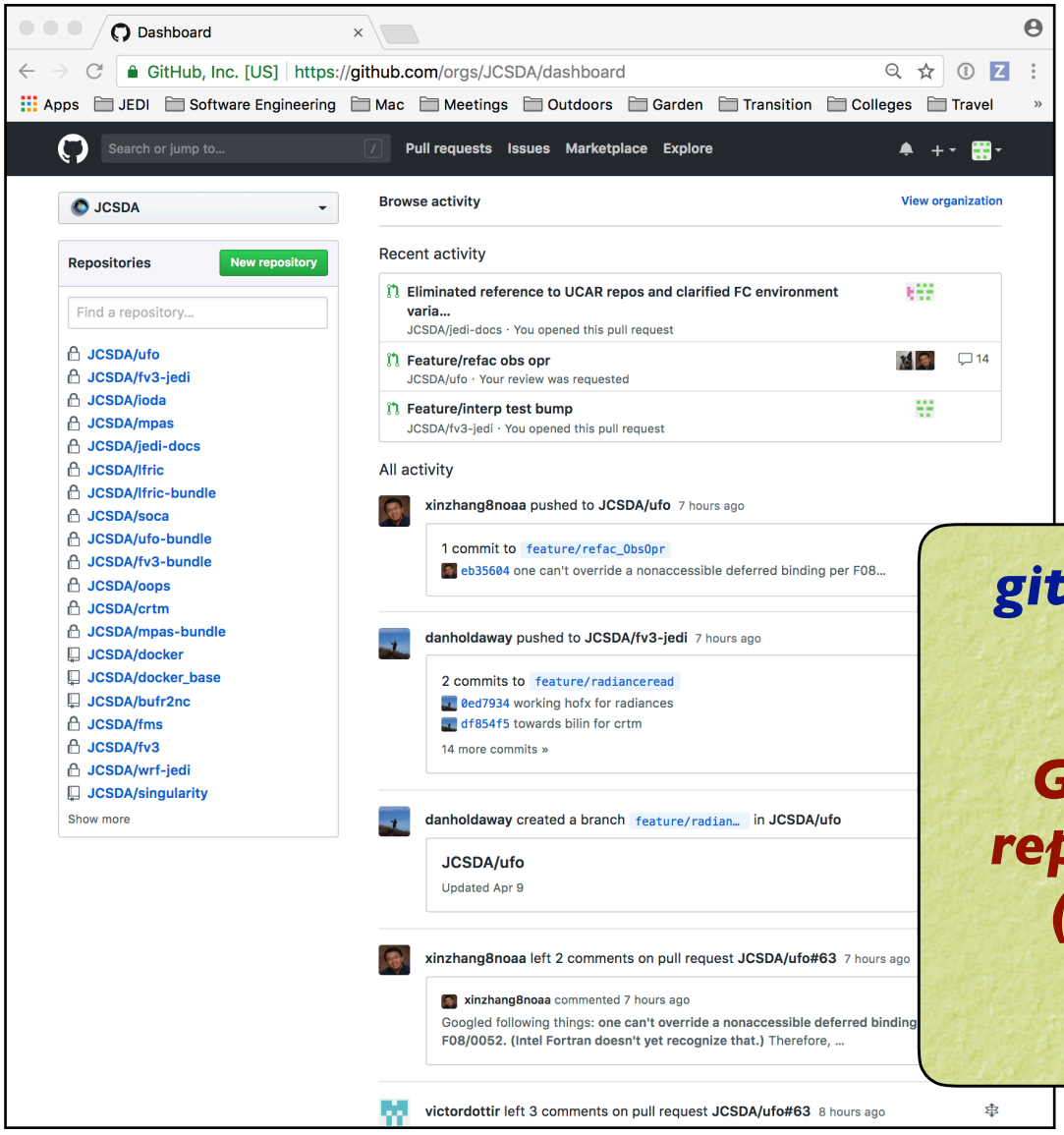
The Way of a JEDI

- *Collaborative*
  - *A Joint Center (JCSDA)*
    - *Partners, collaborators, stakeholders, community*
  - *A Joint Effort (JEDI)*
    - *Distributed team of software developers, with varying objectives and time commitments*
- *Agile*
  - *Innovative*
  - *Flexible (future-proof)*
  - *Responsive to users and developers*
  - *Continuous delivery of functional software*

Outline

- *git/GitHub*
  - *Version control*
  - *Enhancements and bug fixes immediately available to distributed community of developers*
  - *Code review, issue tracking*
  - *Community exports (Code distribution)*
    ...and imports (ecbuild, eckit, fckit)
- *Git-Flow*
  - *Innovation*
  - *Continuous Delivery*
- *ZenHub*
  - *Agile project management*
  - *Enhances GitHub's issue tracking and code review functionality*

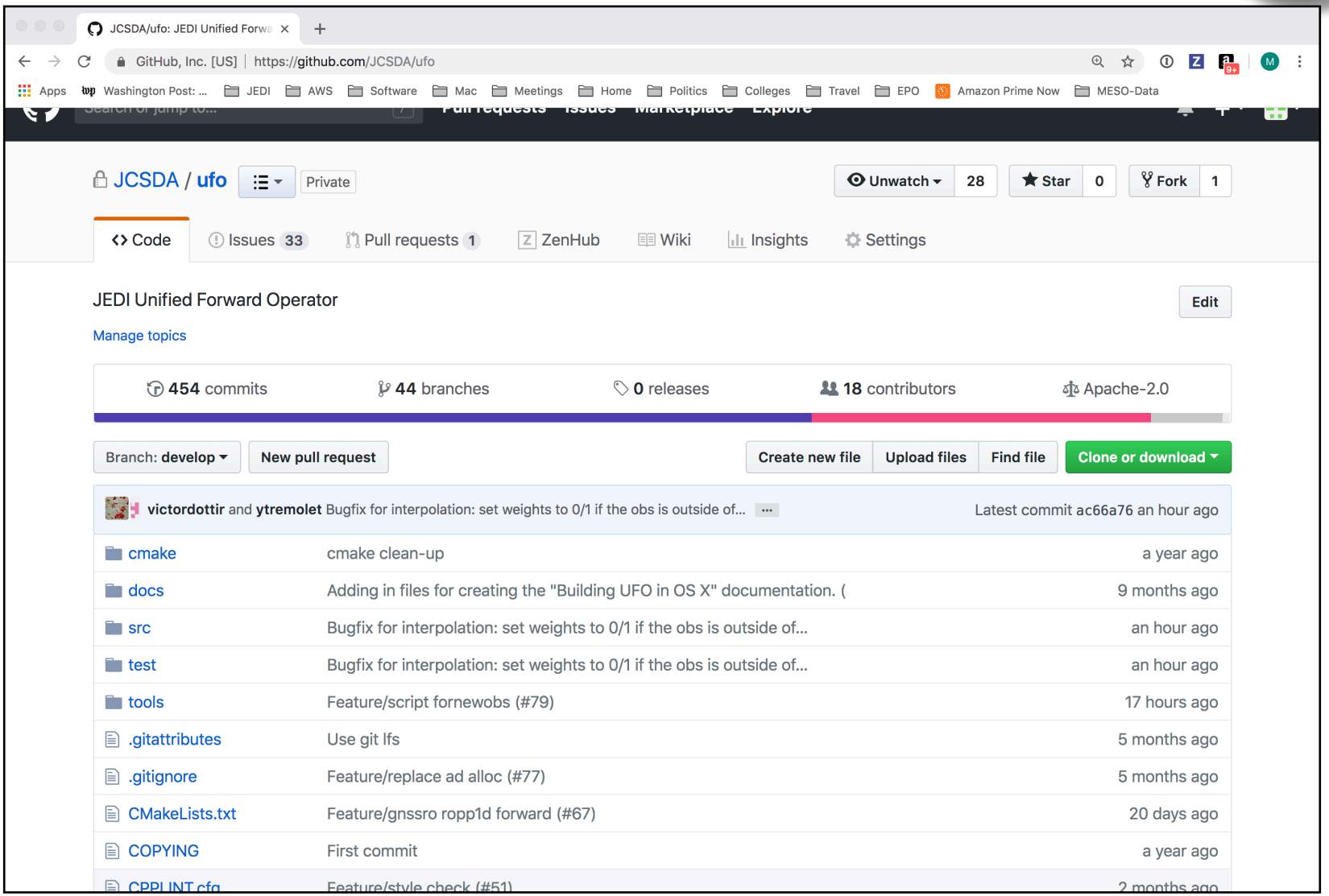
git/GitHub



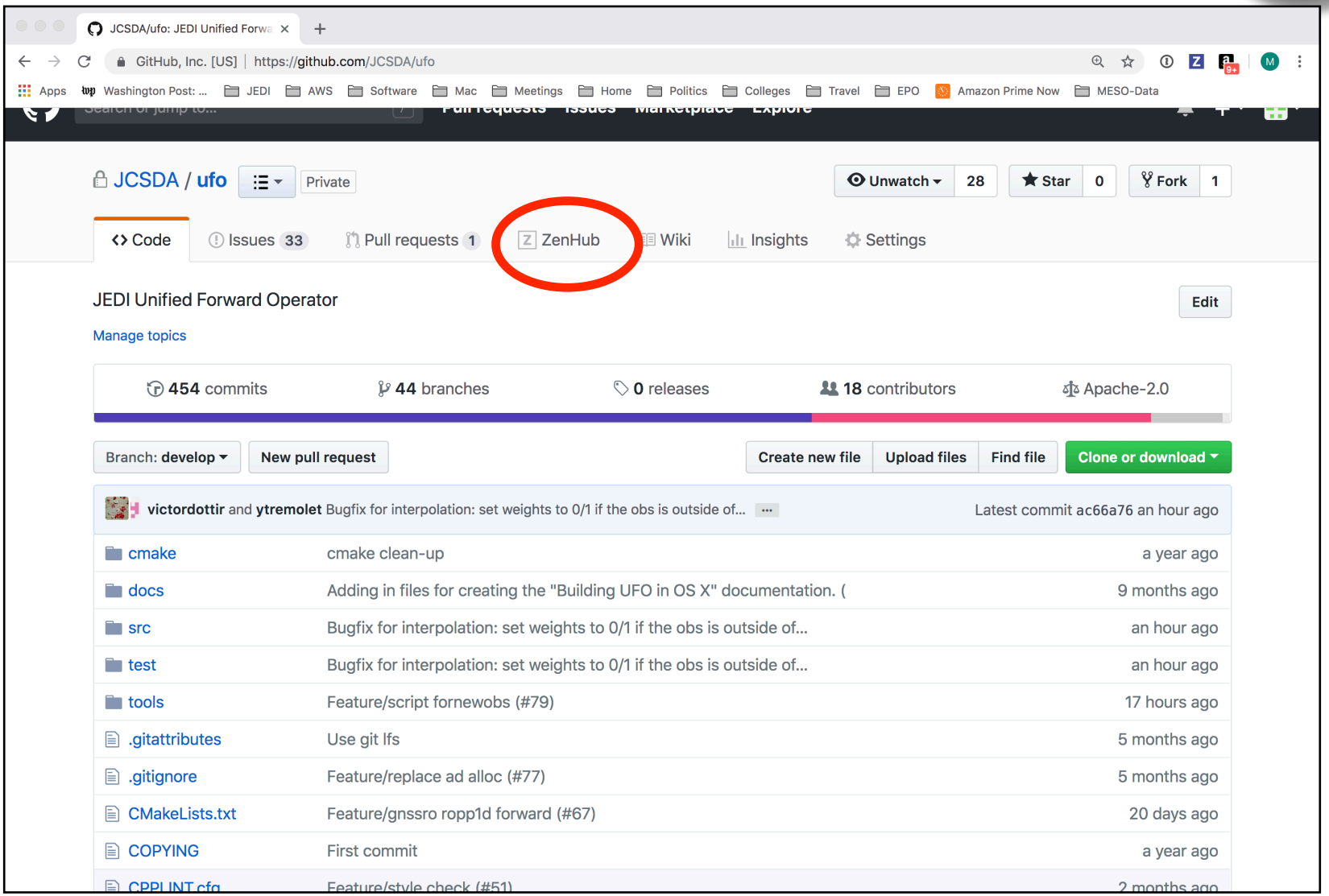
git - command line tool (version control)

GitHub - Web-based repository management (branches, releases, code reviews)

GitHub


JCSDA/ufo: JEDI Unified Forwa
GitHub, Inc. [US] | https://github.com/JCSDA/ufo
Apps
Washington Post: ...
JEDI
AWS
Software
Mac
Meetings
Home
Politics
Colleges
Travel
EPO
Amazon Prime Now
MESO-Data
JCSDA / ufo
Private
Unwatch 28
Star 0
Fork 1
Code
Issues 33
Pull requests 1
ZenHub
Wiki
Insights
Settings
JEDI Unified Forward Operator
Edit
Manage topics
454 commits
44 branches
0 releases
18 contributors
Apache-2.0
Branch: develop
New pull request
Create new file
Upload files
Find file
Clone or download
victordottir and ytremolet Bugfix for interpolation: set weights to 0/1 if the obs is outside of...
Latest commit ac66a76 an hour ago
cmake cmake clean-up a year ago
docs Adding in files for creating the "Building UFO in OS X" documentation. (9 months ago
src Bugfix for interpolation: set weights to 0/1 if the obs is outside of... an hour ago
test Bugfix for interpolation: set weights to 0/1 if the obs is outside of... an hour ago
tools Feature/script fornewobs (#79) 17 hours ago
.gitattributes Use git lfs 5 months ago
.gitignore Feature/replace ad alloc (#77) 5 months ago
CMakeLists.txt Feature/gnssro ropp1d forward (#67) 20 days ago
COPYING First commit a year ago
CPPLINT.cfg Feature/style check (#51) 2 months ago

GitHub


JCSDA/ufo: JEDI Unified Forwa
GitHub, Inc. [US] | https://github.com/JCSDA/ufo
Apps
Washington Post: ...
JEDI
AWS
Software
Mac
Meetings
Home
Politics
Colleges
Travel
EPO
Amazon Prime Now
MESO-Data
JCSDA / ufo
Private
Unwatch 28
Star 0
Fork 1
Code
Issues 33
Pull requests 1
ZenHub
Wiki
Insights
Settings
JEDI Unified Forward Operator
Edit
Manage topics
454 commits
44 branches
0 releases
18 contributors
Apache-2.0
Branch: develop
New pull request
Create new file
Upload files
Find file
Clone or download
victordottir and ytremolet Bugfix for interpolation: set weights to 0/1 if the obs is outside of...
Latest commit ac66a76 an hour ago
cmake cmake clean-up a year ago
docs Adding in files for creating the "Building UFO in OS X" documentation. (9 months ago
src Bugfix for interpolation: set weights to 0/1 if the obs is outside of... an hour ago
test Bugfix for interpolation: set weights to 0/1 if the obs is outside of... an hour ago
tools Feature/script fornewobs (#79) 17 hours ago
.gitattributes Use git lfs 5 months ago
.gitignore Feature/replace ad alloc (#77) 5 months ago
CMakeLists.txt Feature/gnssro ropp1d forward (#67) 20 days ago
COPYING First commit a year ago
CPPLINT.cfg Feature/style check (#51) 2 months ago

git/GitHub (JEDI tips)

- ***Work with JEDI bundles***
  - ✦***Clone bundle repo***
  - ✦***Let ecbuild do the rest***
  - ✦***If that doesn't work, read the README file***
  - ✦***Get in the habit of running make update after ecbuild***
  - ✦***Edit the CMakeLists.txt file to use your local version***

```
#ecbuild_bundle( PROJECT ufo    GIT "https://github.com/JCSDA/ufo.git"    BRANCH develop UPDATE )
ecbuild_bundle( PROJECT ufo SOURCE "~/jedi/src/ufo-bundle/ufo" )
```

- ***Cache your GitHub credentials***

```
git config --global credential.helper 'cache --timeout=3600'
```

Git-LFS

- *LFS = Large File service*
  - *Increases GitHub size limits for individual files from 100 MB to 2GB*
  - *Cumulative storage purchased in 50 GB data packs*
  - *Used for anything that isn't code (data files, restart files, etc)*
- *Transparent to the user*
  - *When you push to GitHub, any files that are tracked by LFS will go to a remote server (the LFS Store)*
  - *The GitHub repo will only contain a pointer to that file*
  - *When you fetch/pull/clone an LFS-enabled repo from GitHub, LFS will check to see if you have the large files on your computer (local LFS cache). If not, it will retrieve them from the LFS Store as needed.*

Git-Flow

Git Flow is:

- ***A Philosophy***
  - ***Optimal for Agile Software Development***
    - ***Innovation***
    - ***Continuous Delivery***
- ***A Working Principle***
  - ***Enforcement of branch naming conventions soon to come***
- ***An Application (extension to git)***
  - ***Already installed in AMI and Singularity Container***
  - **brew install git-flow-avh # (Mac)**
  - **sudo apt-get install git-flow # (linux)**
  - **https://github.com/petervanderdoes/gitflow-avh**

The Git-Flow Manifesto

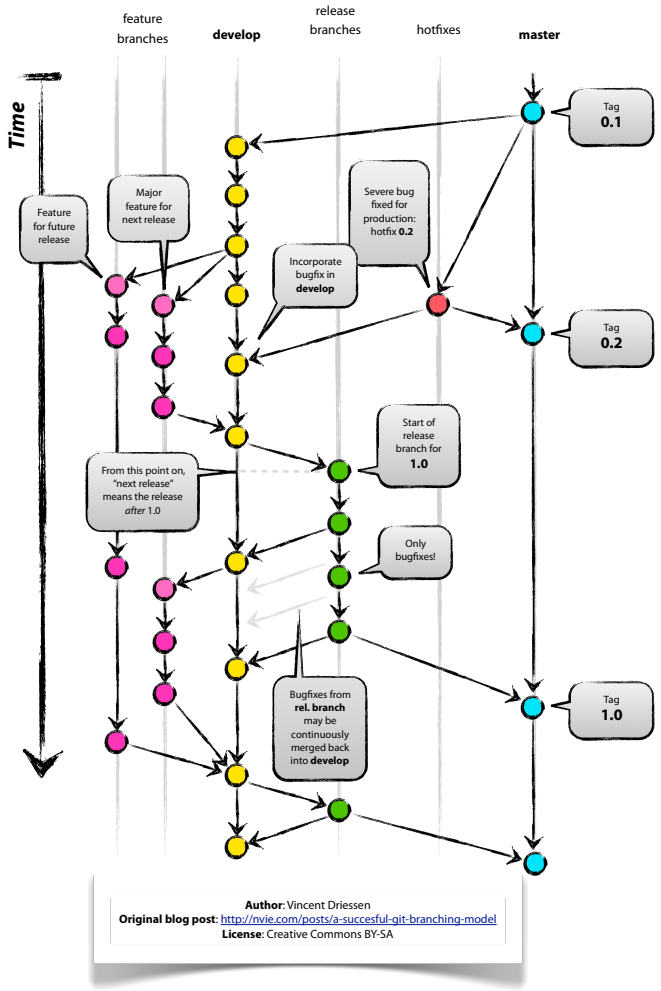
http://nvie.com/posts/a-successful-git-branching-model/

Vincent Driessen (2010)

Google

git branching model

Highly Recommended!

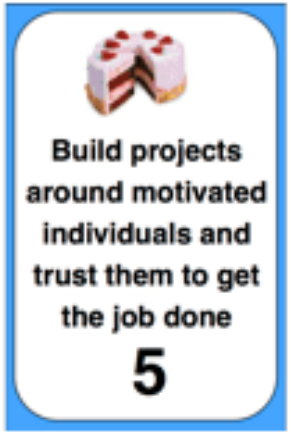


The Git-Flow Manifesto: Takaways

- *master is for releases only*
- *develop*
  - *Not ready for pubic consumption but compiles and passes all tests*
- *Feature branches*
  - *Where most development happens*
  - *Branch off of develop*
  - *Merge into develop*
- *Release branches*
  - *Branch off of develop*
  - *Merge into master and develop*
- *Hotfix*
  - *Branch off of master*
  - *Merge into master and develop*
- *Bugfix*
  - *Branch off of develop*
  - *Merge into develop*

Agile Software Development

- ***12 Agile Principles***



https://nomad8.com/

Agile Software Development

- *12 Agile Principles*

Git-Flow helps with many of these
For the rest, we have ZenHub

Early and continuous delivery of valuable software

1 ✔

Welcome changing requirements even late in development

2 ✔

Deliver working software frequently

3 ✔

Business people and developers working together daily

4

Build projects around motivated individuals and trust them to get the job done

5 ✔

The most effective method of conveying information is face-to-face conversation

6

Working software is the primary measure of progress

7 ✔

Sustainable development: maintain a constant pace indefinitely

8

Continuous attention to technical excellence

9 ✔

Simplicity: maximize the amount of work not done

10 ✔

Teams self-organize

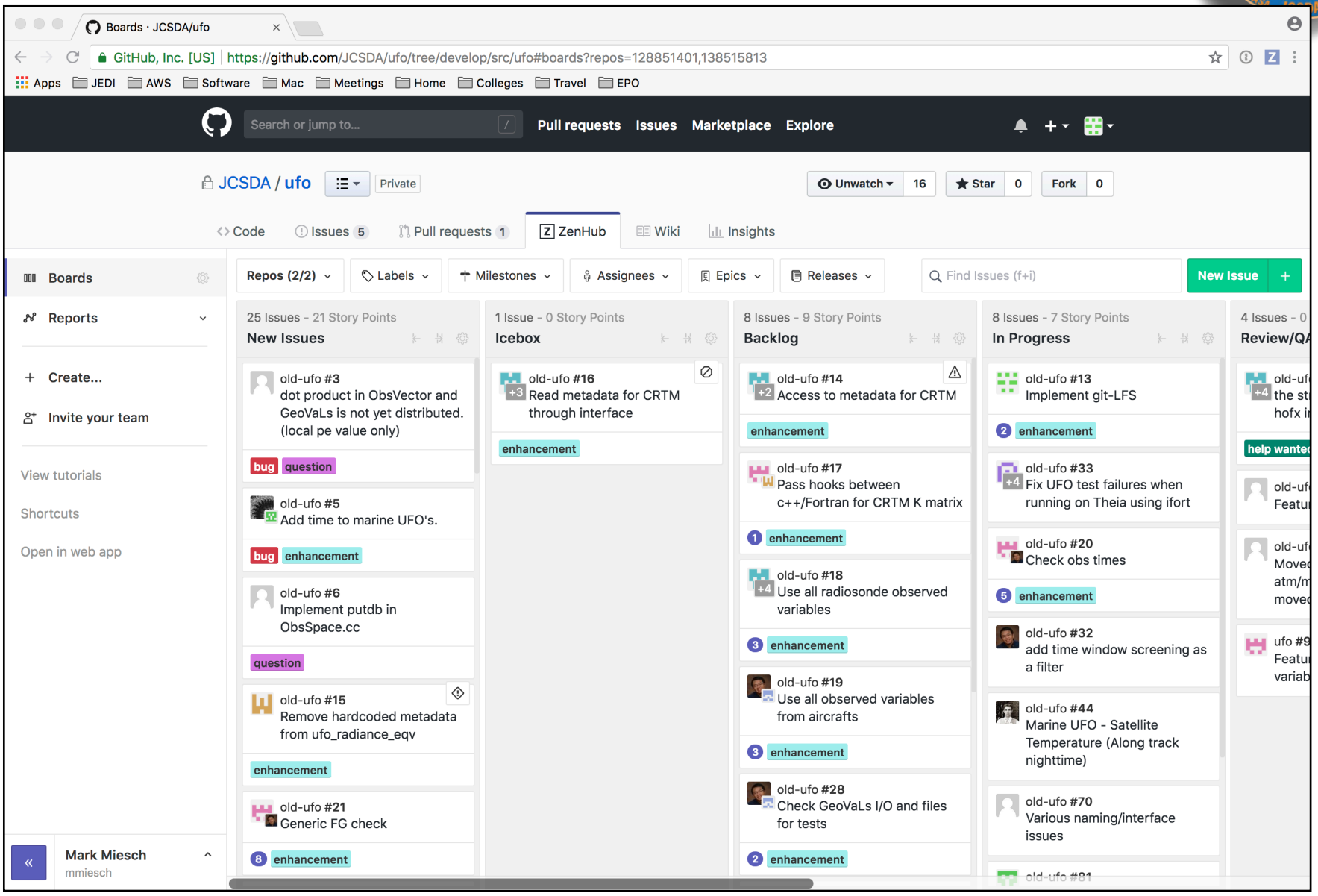
11

Teams regularly reflect and adjust behaviour

12 ✔

https://nomad8.com/

Agile workflows: ZenHub



ZenHub Features

- ***Customizable Project boards***
  - ✦ ***Prioritize and organize tasks***
  - ✦ ***Reviews/Feedback***
  - ✦ ***Sprints (Milestones) and Epics***
- ***Closely integrated with GitHub***
  - ✦ ***Access boards directly from GitHub repos***
  - ✦ ***ZenHub tasks are GitHub issues and vice versa***
- ***Tasks/Issues***
  - ✦ ***Assign up to 10 individuals***
  - ✦ ***Labels, difficulty estimates, etc.***
  - ✦ ***Can be linked to pull requests***
  - ✦ ***Markdown supported (boldface, checklists…)***
- ***Monitoring progress***
  - ✦ ***Burndown charts***
  - ✦ ***Velocity tracking***
  - ✦ ***Release reports***
    - ***Time estimate to deliver a specified set of features***

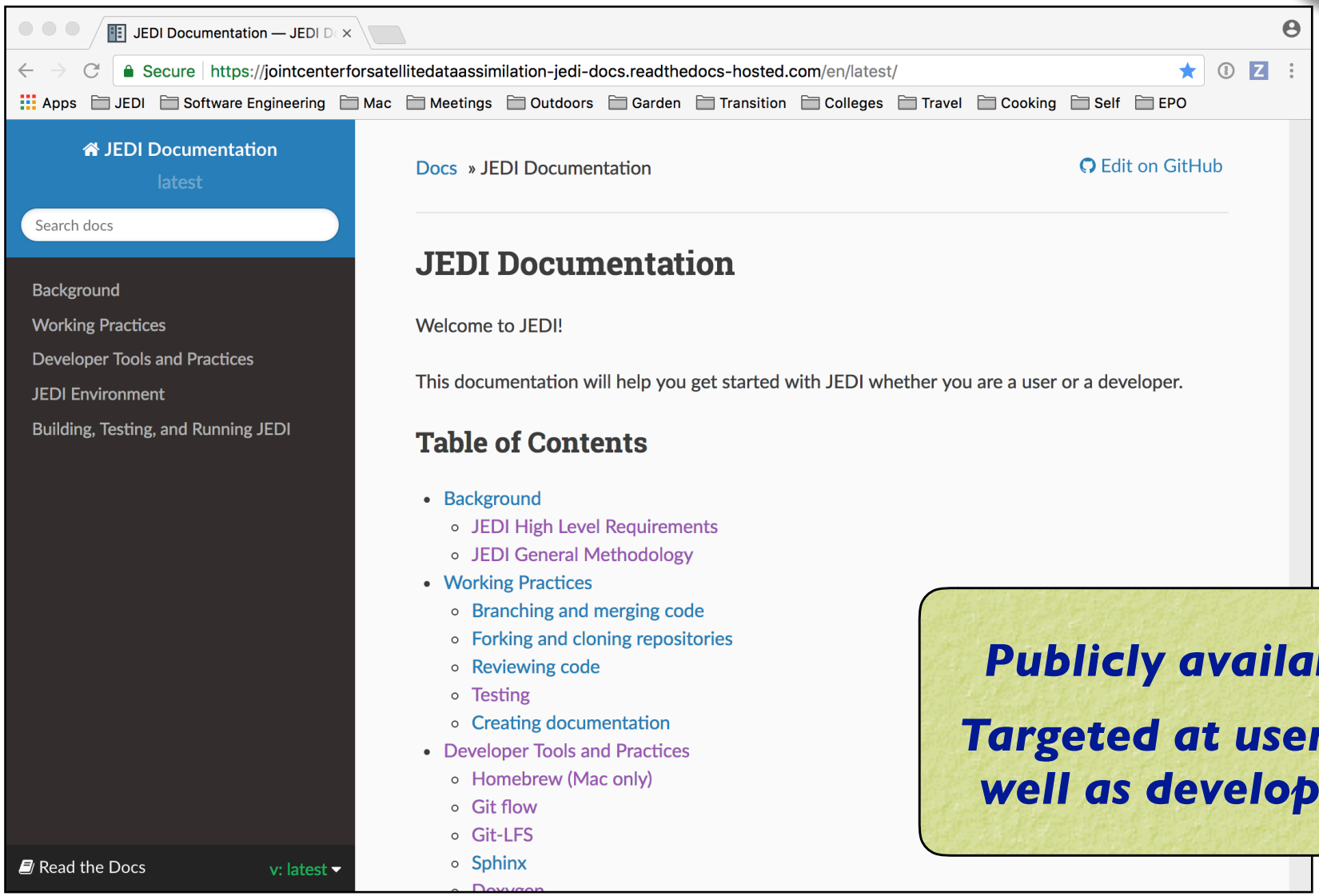
ZenHub Pipelines

- ***New Issues***
  - ***Default landing spot***
  - ***Issues should not stay here long***
- ***Backlog***
  - ***Main "To Do" List***
  - ***Arrange in order of priority (reviewed regularly by teams)***
- ***IceBox***
  - ***Low-priority items that should be done at some point but do not require immediate attention***
- ***In Progress***
  - ***Lets others know what you are doing to promote collaboration and avoid redundancy***
- ***Review/QA***
  - ***Solicit feedback before you mark something as…***
- ***Closed***

Documentation

- *Agile Project Management and Collaborative Workflow*
  - ✦*git/GitHub*
  - ✦*git-flow*
  - ✦*ZenHub*
- *Documentation*
  - ✦*Sphinx/ReadTheDocs (high-level manuals, how-to's, etc)*
  - ✦*Doxygen (low-level code details)*
- *Computing Environment*
  - ✦*Software containers*
  - ✦*Cloud Computing (AWS)*

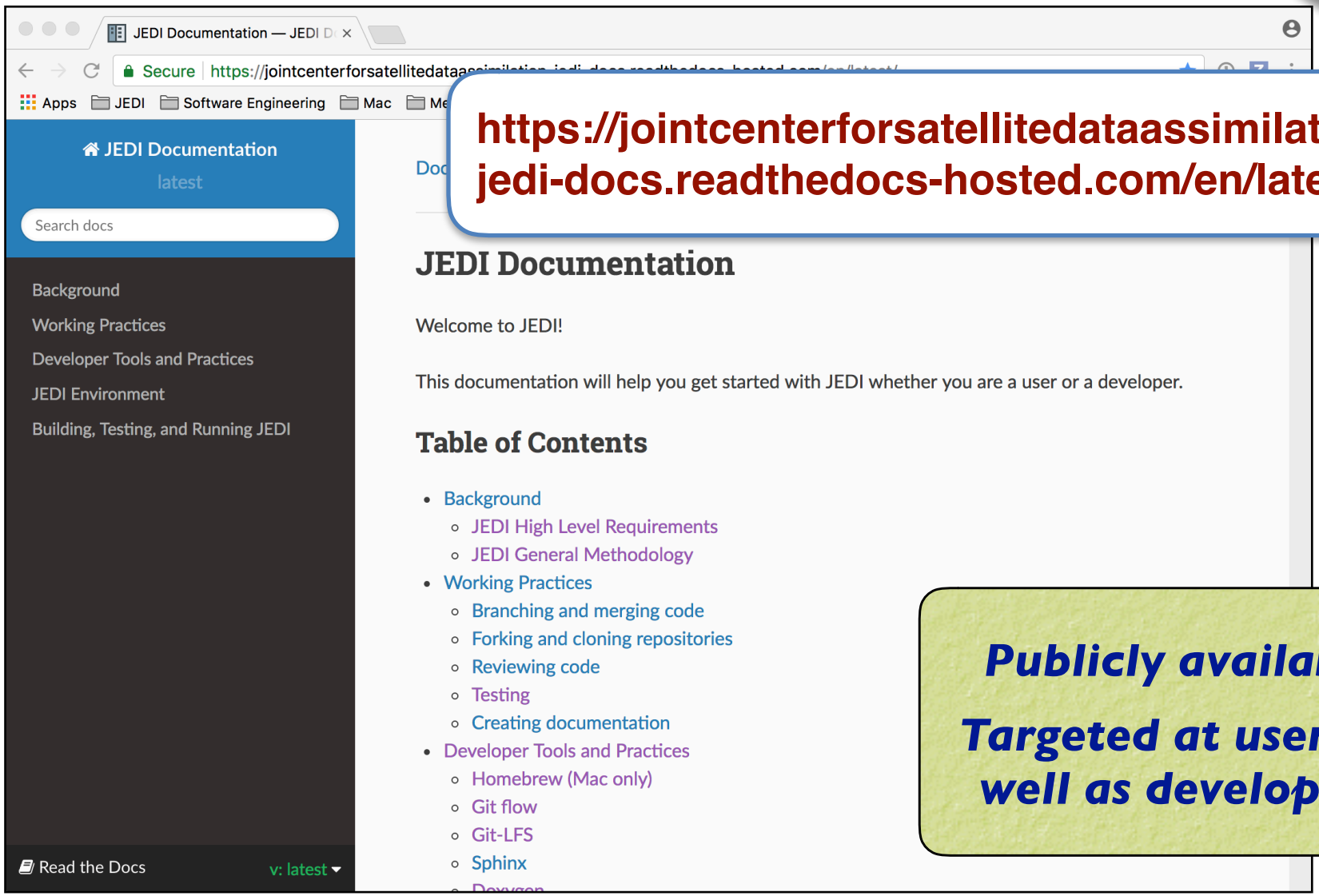
Sphinx/ReadtheDocs



Publicly available

Targeted at users as well as developers

Sphinx/ReadtheDocs

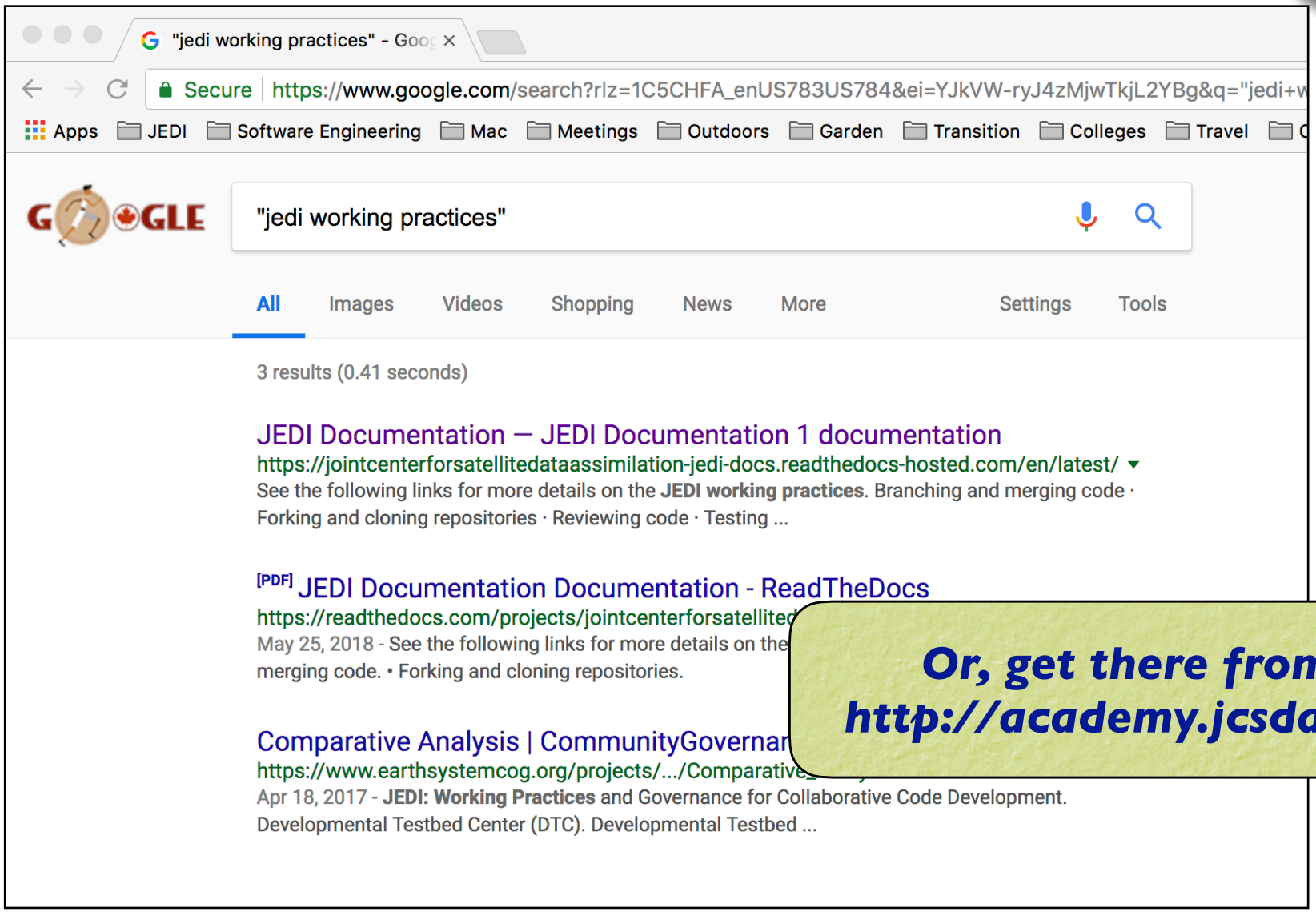


https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/

Publicly available

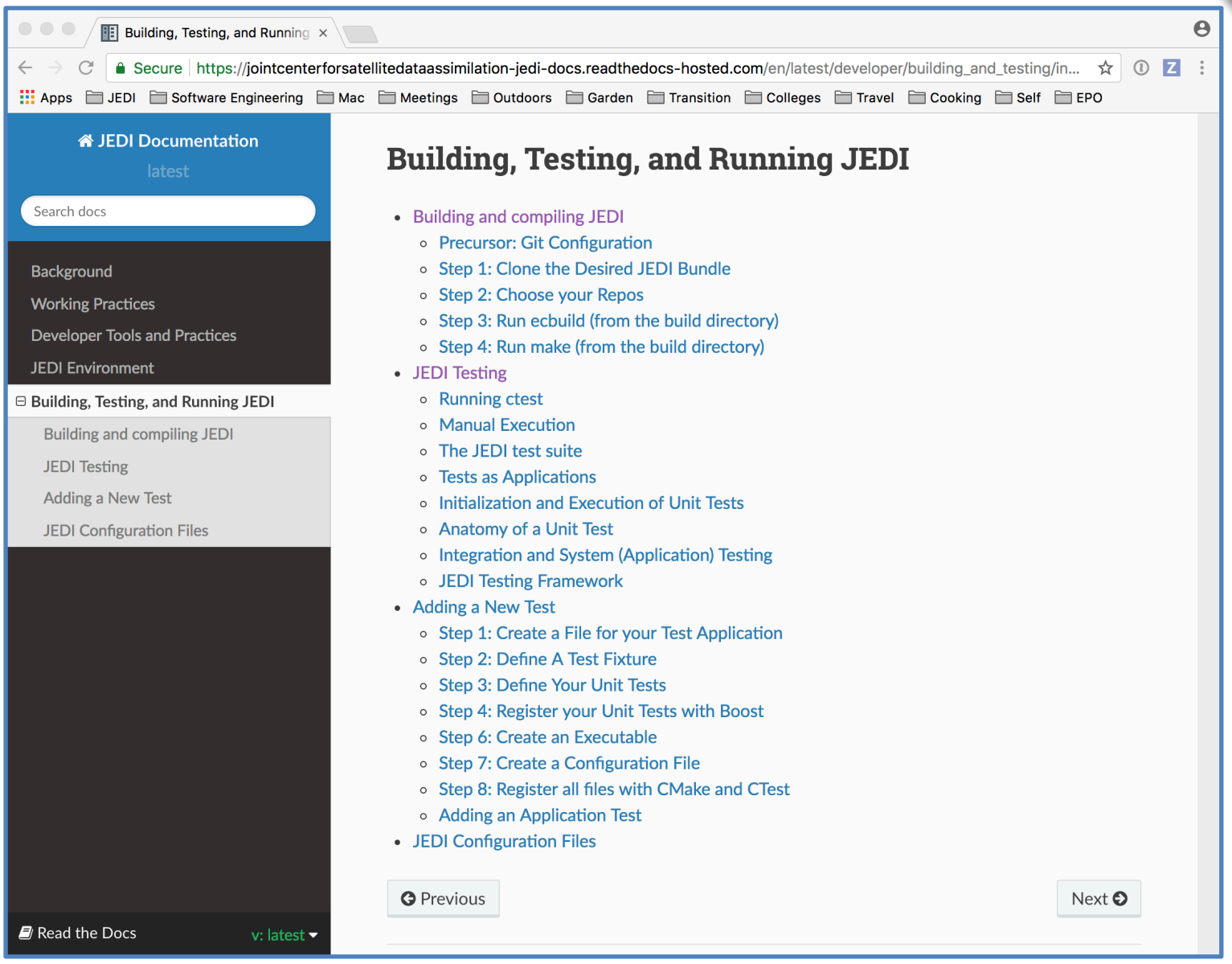
Targeted at users as well as developers

Sphinx/ReadtheDocs



Or, get there from http://academy.jcsda.org

Sphinx/ReadtheDocs



Sphinx

- *Sphinx*
  - *The real workhorse behind the documents*
  - *Python package*
  - *Source code written with Restructured text*

- *Distribution plan*
  - *ReadtheDocs for now to publish*
  - *Sphinx Source code on GitHub (jedi-docs)*
  - *Tagged versions of the doc repos will be linked to JEDI releases*

For more info on Sphinx see the corresponding page in the JEDI documentation, under Developer Tools and Practices

Doxygen

Used in JEDI for:

- ***Documenting functions and subroutines (C++ and F90)***
- ***Documenting classes and structures (C++ and F90)***
- ***Viewing namespaces and modules***
- ***Generating Class Hierarchies***
- ***Generating Call diagrams***
- ***Any other documentation that involves specific blocks of code***

For example Doxygen documentation (fv3-bundle)
See
https://github.com/nov2018

Sample output: “man page”

◆ testStateInterpolation()

template<typename MODEL >

void test::testStateInterpolation ( )

Interpolation test.

testStateInterpolation() tests the interpolation for a given model. The conceptual steps are as follows:

1. Initialize the JEDI **State** object based on idealized analytic formulae
2. Interpolate the **State** variables onto selected "observation" locations using the getValues() method of the **State** object. The result is placed in a JEDI **GeoVaLs** object
3. Compute the correct solution by applying the analytic formulae directly at the observation locations.
4. Assess the accuracy of the interpolation by comparing the interpolated values from Step 2 with the exact values from Step 3

The interpolated state values are compared to the analytic solution for a series of **locations** which includes values optionally specified by the user in the "StateTest" section of the config f
a randomly-generated list of **Nrandom** random locations. Nrandom is also specified by the user in the "StateTest" section of the config file, as is the (nondimensional) tolerence level (**inte**
to be used for the tests.

This is an equation:

$$\zeta = \left(\frac{x - x_0}{\lambda} \right)^{2/3}$$

Relevant parameters in the **State* section of the config file include

- **norm-gen** Normalization test for the generated **State**
- **interp_tolerance** tolerance for the interpolation test

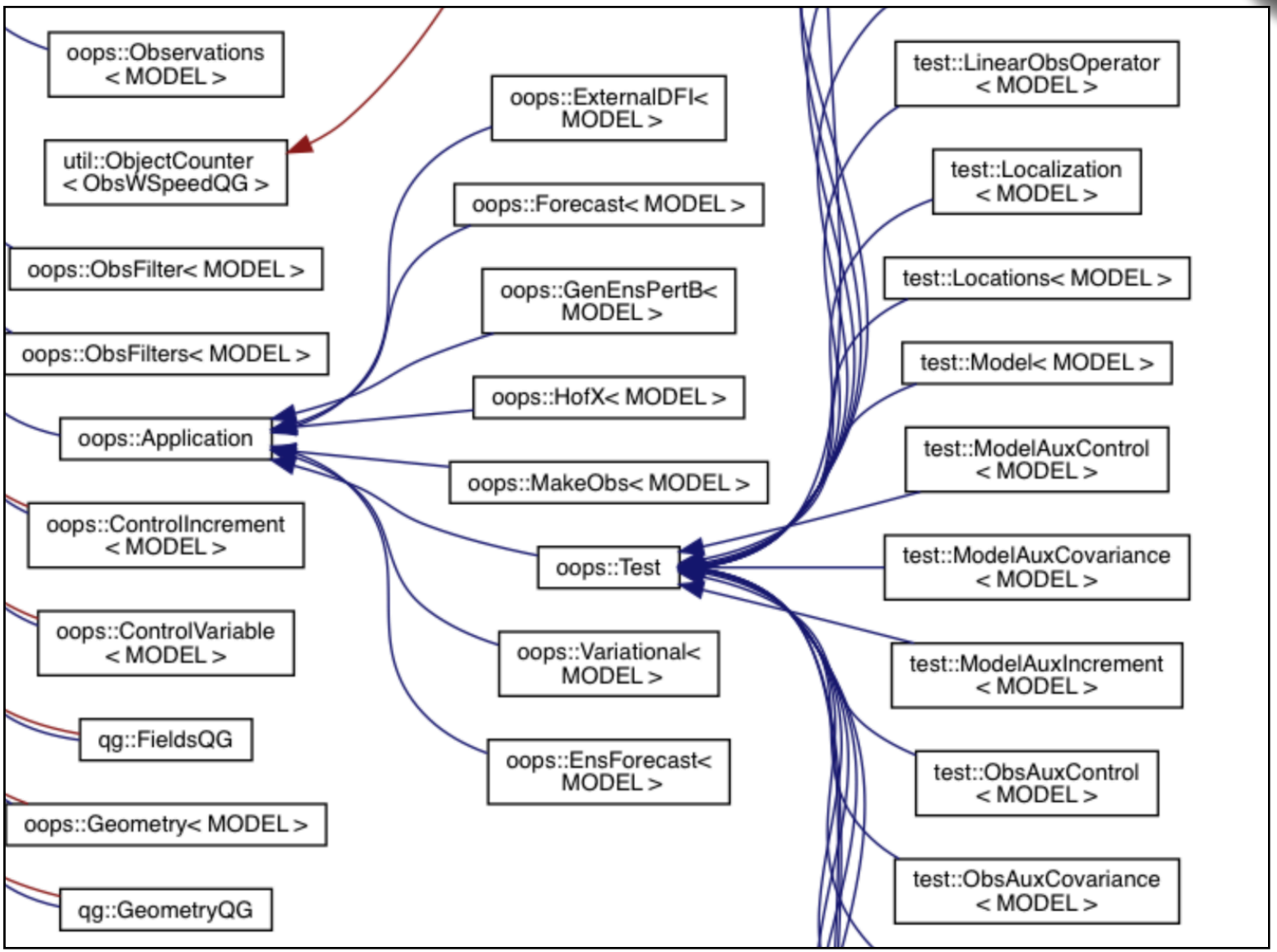
Date

April, 2018: M. Miesch (JCSDA) adapted a preliminary version in the feature/interp branch

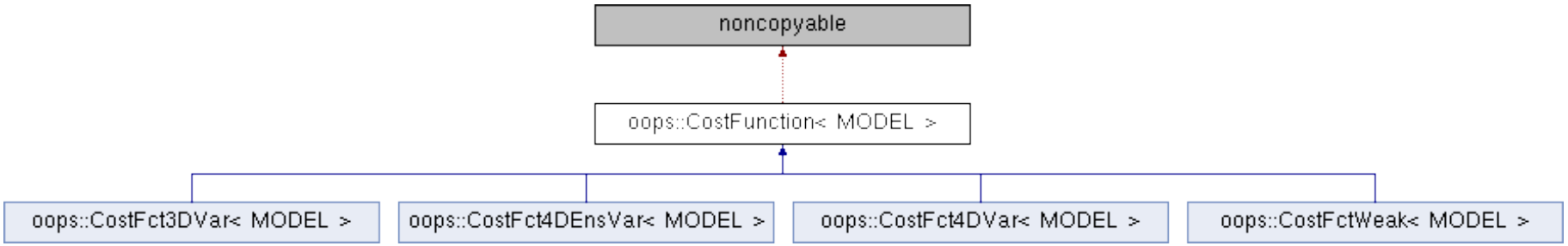
Warning

Since this model compares the interpolated state values to an exact analytic solution, it requires that the "analytic_init" option be implemented in the model and selected in the "State.StateGenerate" section of the config file.

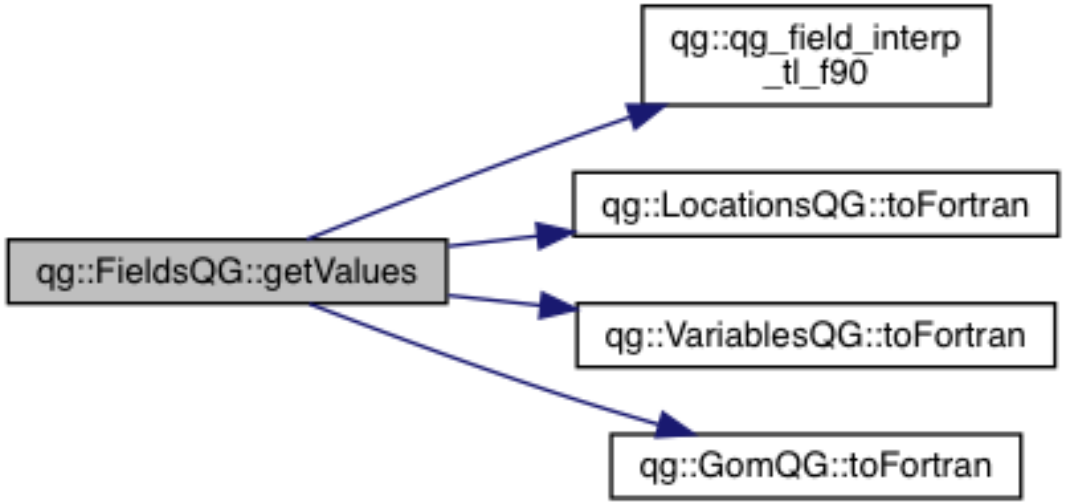
Sample output: class hierarchy



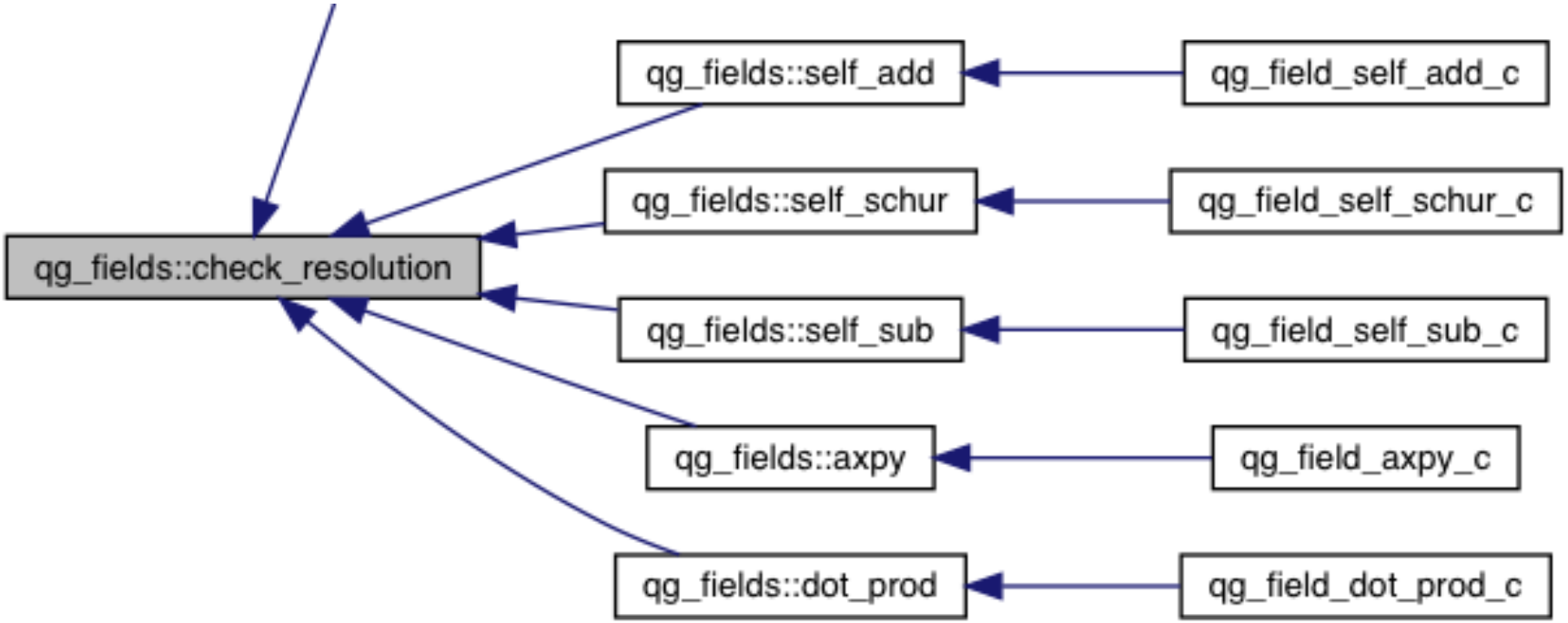
Sample output: inheritance, call graphs



Clickable boxes!

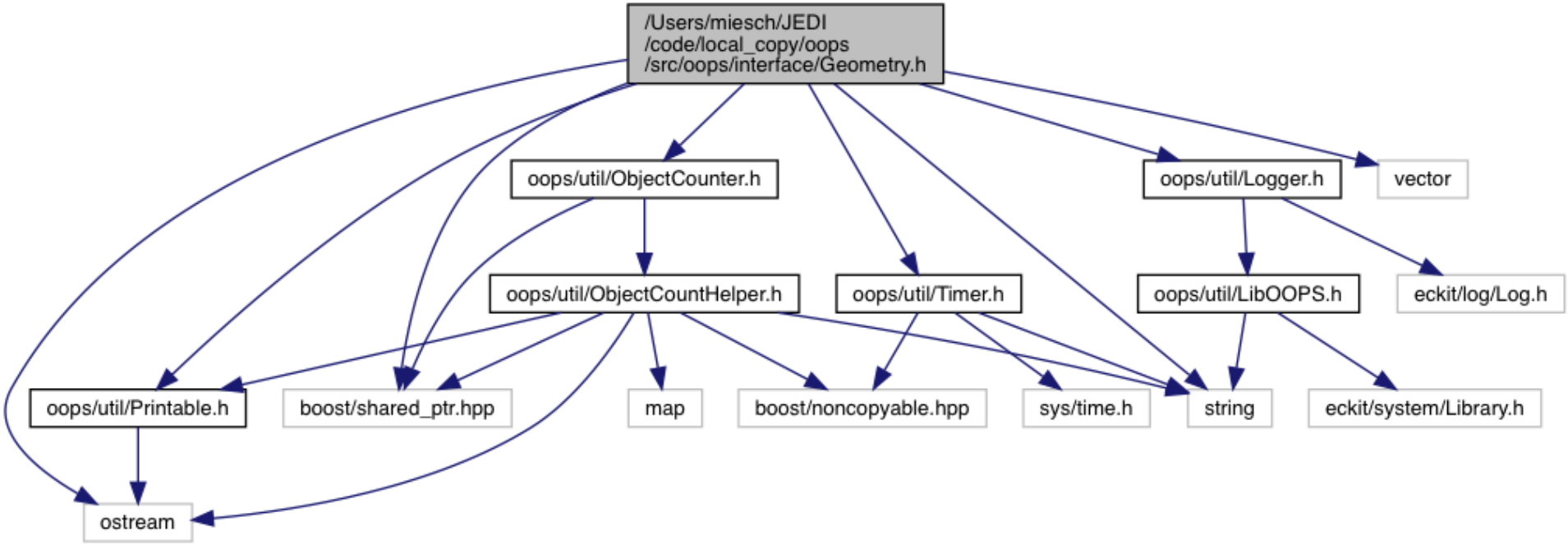


Sample output: caller graphs



Note that these traces end in _c (this is a Fortran routine)
Doxygen has trouble with C++ / Fortran binding
Look for corresponding _f90 routine to follow further

Sample output: include diagrams



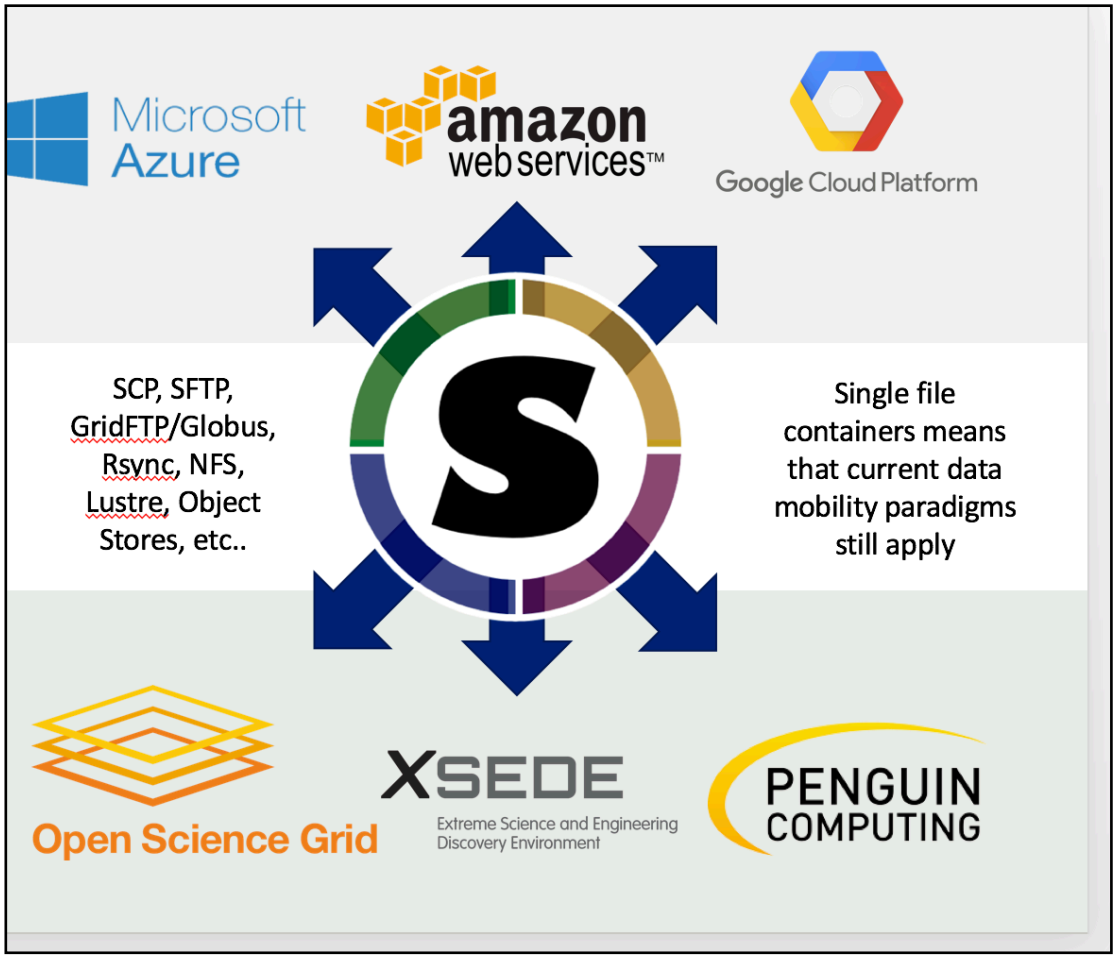
Can get complicated!

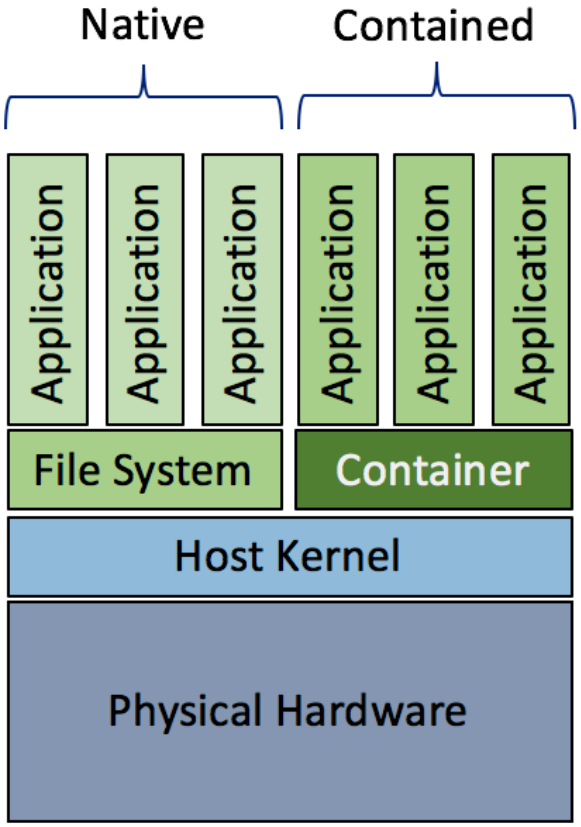
Computing Environment

- *Agile Project Management and Collaborative Workflow*
  - ✦*git/GitHub*
  - ✦*git-flow*
  - ✦*ZenHub*
- *Documentation*
  - ✦*Sphinx/ReadTheDocs (high-level manuals, how-to's, etc)*
  - ✦*Doxygen (low-level code details)*
- *Computing Environment*
  - ✦*Software containers (Singularity)*
  - ✦*Cloud Computing (AWS)*

Containers for Portability

Singularity, Docker





Container Architecture

JEDI Singularity Container

- *Pre-installed software*
  - ✦*Compilers (C++, Fortran)*
  - ✦*open-mpi*
  - ✦*LAPACK*
  - ✦*Eigen3*
  - ✦*HDF5*
  - ✦*NetCDF4*
  - ✦*Boost*
  - ✦*python*
  - ✦*…*
- *Self-contained in a single image file*
  - ✦*singularity pull shub://JCSDA/singularity*
  - ✦*singularity shell -e <image-file>*

Caveat: Singularity must be installed!

Amazon Web Services (AWS)

- *Computing Resources*
  - ✦*Variety of optimized compute instances*
  - ✦*State-of-the-art hardware (Intel Xeon)*
  - ✦ *Available on-demand or queued (spot instances)*
  - ✦*Pay only for what you use*
  - ✦*Community Applications (e.g. sagemaker, lambda)*
- *AWS in JCSDA*
  - ✦*Training (e.g. this week!)*
  - ✦*Sharing Data, applications*
  - ✦*Continuous Integration (JEDI)*
  - ✦*Applications*
    - *NWP*
    - *Machine Learning*
    - *FSOI*
    - *Web hooks*
    - *…*

amazon web services

Resources

JEDI Documentation - access link from
https://academy.jcsda.org

Extensive GitHub documentation & tutorials
https://help.github.com

Lots of Great Github Cheat Sheets
https://education.github.com/git-cheat-sheet-education.pdf
https://jan-krueger.net/git-cheat-sheet-extended-edition
https://patrickzahnd.ch/uploads/git-transport-v1.png

Doxygen
http://www.stack.nl/~dimitri/doxygen/manual/index.html

AWS
http://aws.com