COSMO Development at MeteoSwiss

Pascal Spörri
pascal.spoerri@env.ethz.ch

About me

- Maintainer of the C++ Dynamical Core
- Successor of Andrea Arteaga
- Primarily work at MeteoSwiss
- POMPA project contributor

Why?

- Development process has grown organically
- Developing against multiple architectures creates overhead
- Goal
  - Decrease developer friction
  - Reduce the number of introduced bugs
  - Make the development process more open: Students, industry, external collaborators

Overview

- Source code management
- Development process
- Testing and code validation

Source code management

Git - All the way

- SVN
  - Need access to a server
  - Cumbersome with a lot of users
  - Difficult to move code and features around
- Git
  - Easy to branch and merge code
  - Lots and lots of free tools available that make life easier
  - Entire history can be stored locally and moved around
  - Flexible workflows

Definitions

- Commit: A changeset
- Tree: A set of commits forming a parent child relationship
- Repository: Contains the git tree
- Branch: A dynamic pointer to a commit (similar to SVN)
- Tag: A static pointer to a commit (similar to SVN)
- Clone: To copy of a repository to your workspace

C2SM
Center for Climate
Systems Modeling

Migration to git

- Finished since December 2015
- Relatively painless
  - Git and SVN share the same concepts
  - Users needed only one tutorial to get started
- No vendor lock-in

New team member!

GitHub

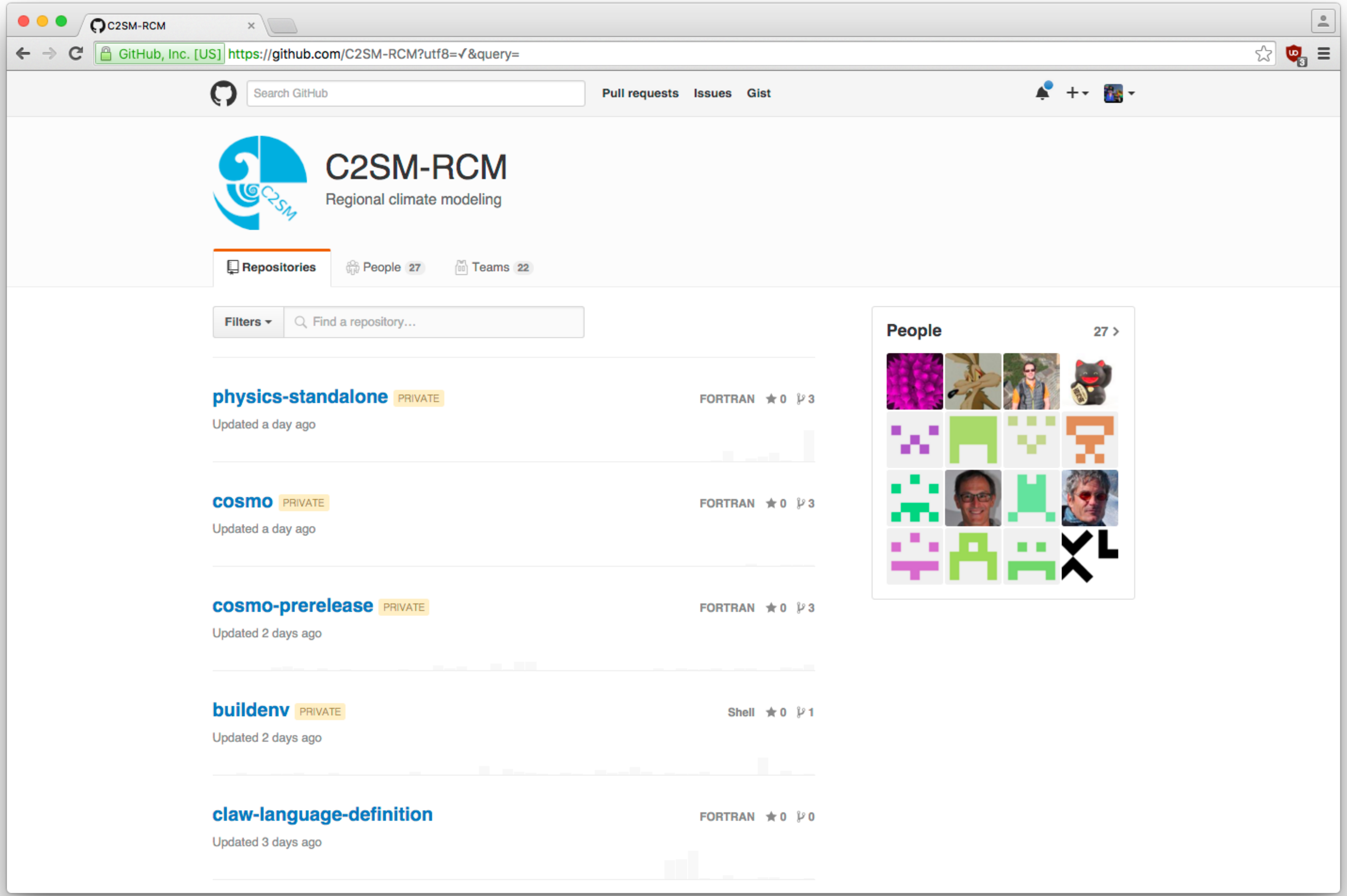
Github for source code management

- Facilitates easier collaboration between MeteoSwiss and external partners
- Popular with a lot of companies and the OpenSource community
- Easy to use web interface to browse code and commits
- Encourages an interactive style of software development

Github Definitions

- Fork: A copy of a repository from one Github account to another
- Pull-Request: An interactive tool for **reviewing code** and **merging** branches automatically

Setup on Github


C2SM-RCM

https://github.com/C2SM-RCM?utf8=✓&query=

Search GitHub | Pull requests | Issues | Gist

C2SM-RCM

Regional climate modeling

Repositories | People 27 | Teams 22

Filters | Find a repository…

physics-standalone PRIVATE — FORTRAN ★ 0 3
Updated a day ago

cosmo PRIVATE — FORTRAN ★ 0 3
Updated a day ago

cosmo-prerelease PRIVATE — FORTRAN ★ 0 3
Updated 2 days ago

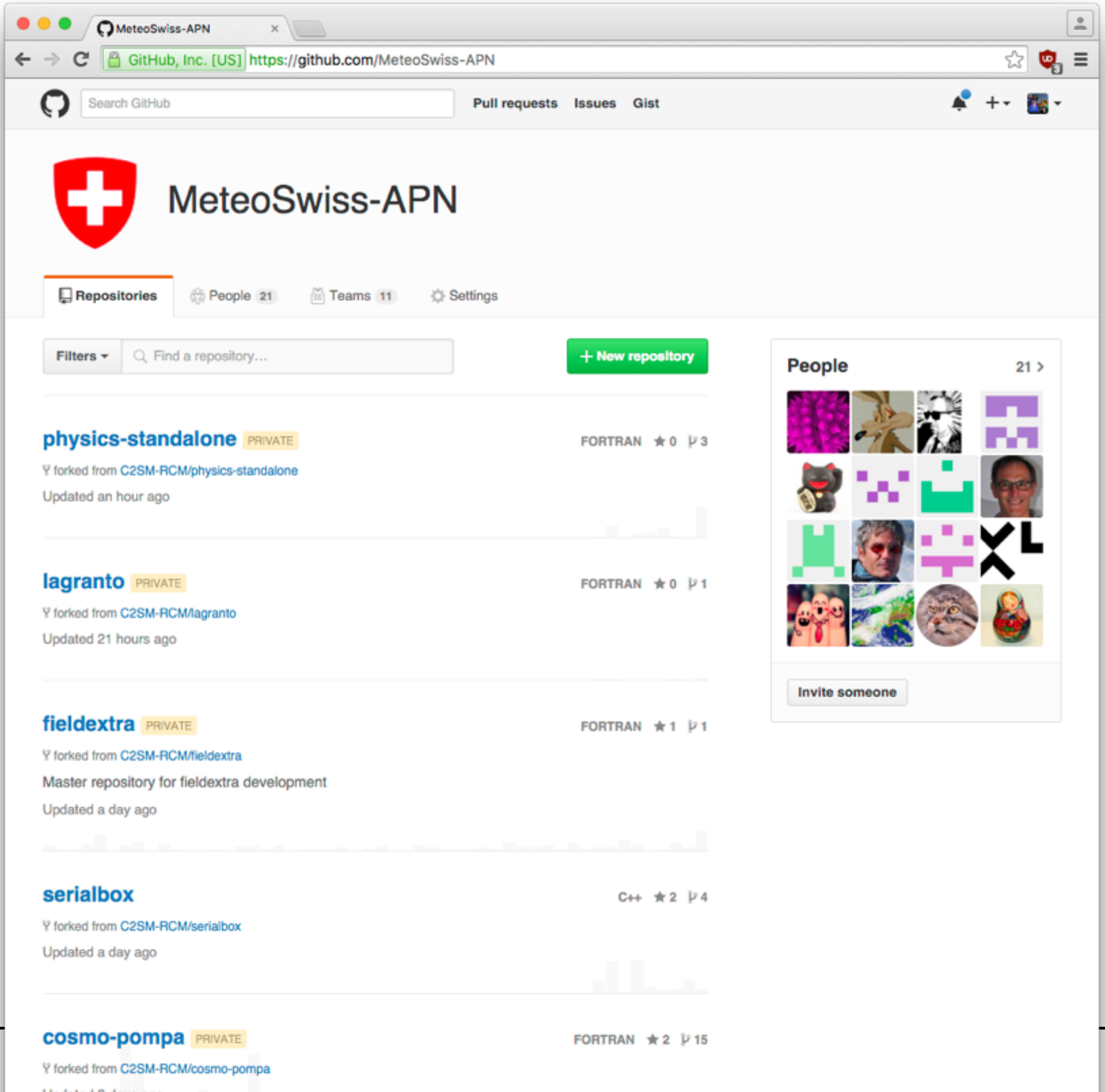
buildenv PRIVATE — Shell ★ 0 1
Updated 2 days ago

claw-language-definition — FORTRAN ★ 0 0
Updated 3 days ago

People 27

Source Code Management

- MeteoSwiss maintains its own fork (copies)
  - Facilitate internal development
  - Prepare releases for production
- Regular releases with production code to base repositories
- Great for collaboration: Easy to migrate changes



MeteoSwiss-APN

GitHub, Inc. [US] https://github.com/MeteoSwiss-APN

Search GitHub | Pull requests | Issues | Gist

MeteoSwiss-APN

Repositories | People 21 | Teams 11 | Settings

Filters | Find a repository… | + New repository

physics-standalone PRIVATE — FORTRAN ★ 0 3
forked from C2SM-RCM/physics-standalone
Updated an hour ago

lagranto PRIVATE — FORTRAN ★ 0 1
forked from C2SM-RCM/lagranto
Updated 21 hours ago

fieldextra PRIVATE — FORTRAN ★ 1 1
forked from C2SM-RCM/fieldextra
Master repository for fieldextra development
Updated a day ago

serialbox — C++ ★ 2 4
forked from C2SM-RCM/serialbox
Updated a day ago

cosmo-pompa PRIVATE — FORTRAN ★ 2 15
forked from C2SM-RCM/cosmo-pompa

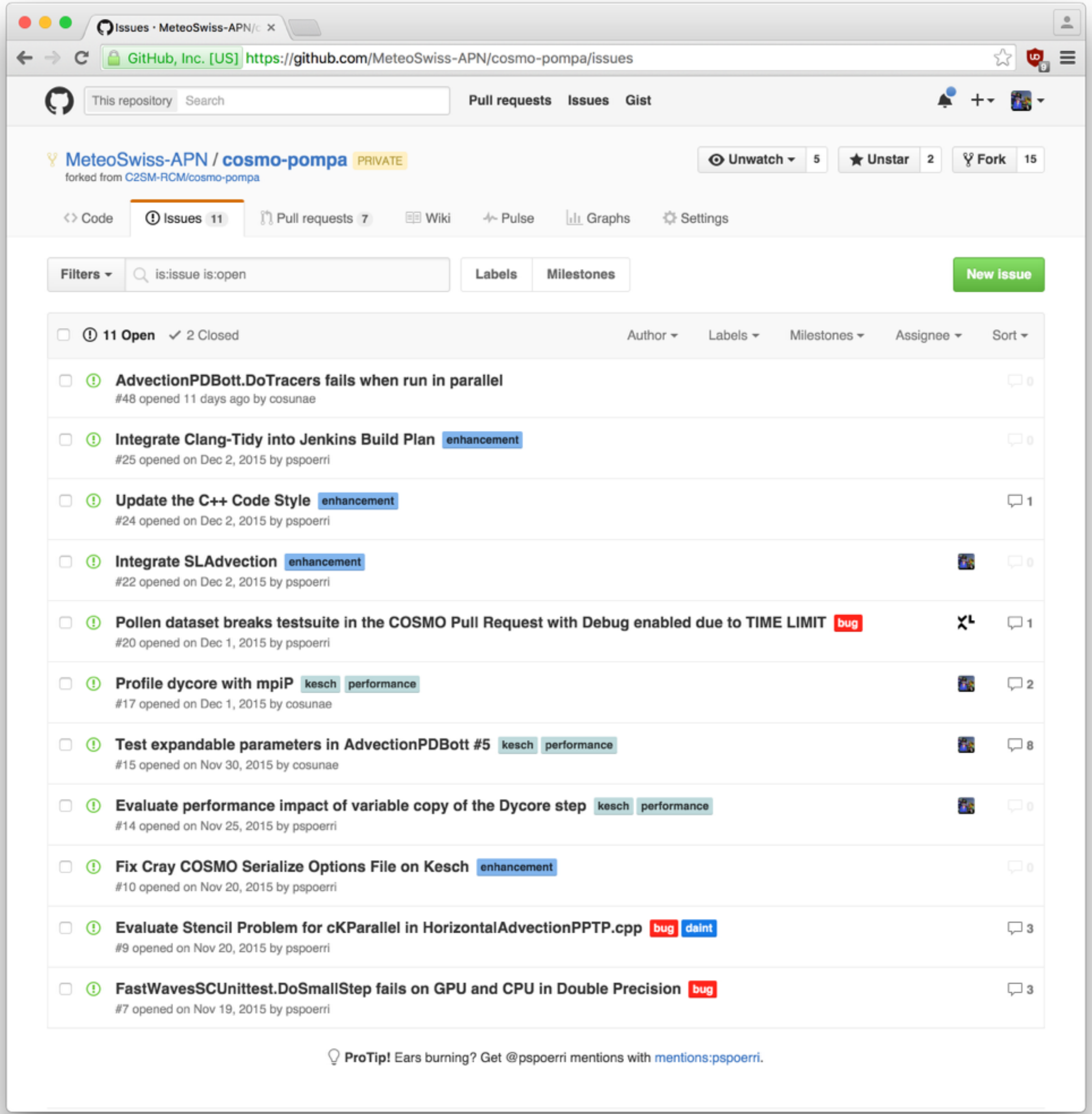
People 21 >

Invite someone

C2SM
Center for Climate
Systems Modeling

Issue Tracking

- Track Issues
- Make feature requests visible
- Plan releases


Issues · MeteoSwiss-APN/c
GitHub, Inc. [US] https://github.com/MeteoSwiss-APN/cosmo-pompa/issues
This repository Search Pull requests Issues Gist

MeteoSwiss-APN / cosmo-pompa PRIVATE
forked from C2SM-RCM/cosmo-pompa
Unwatch 5 Unstar 2 Fork 15

Code | Issues 11 | Pull requests 7 | Wiki | Pulse | Graphs | Settings

Filters is:issue is:open Labels Milestones New issue

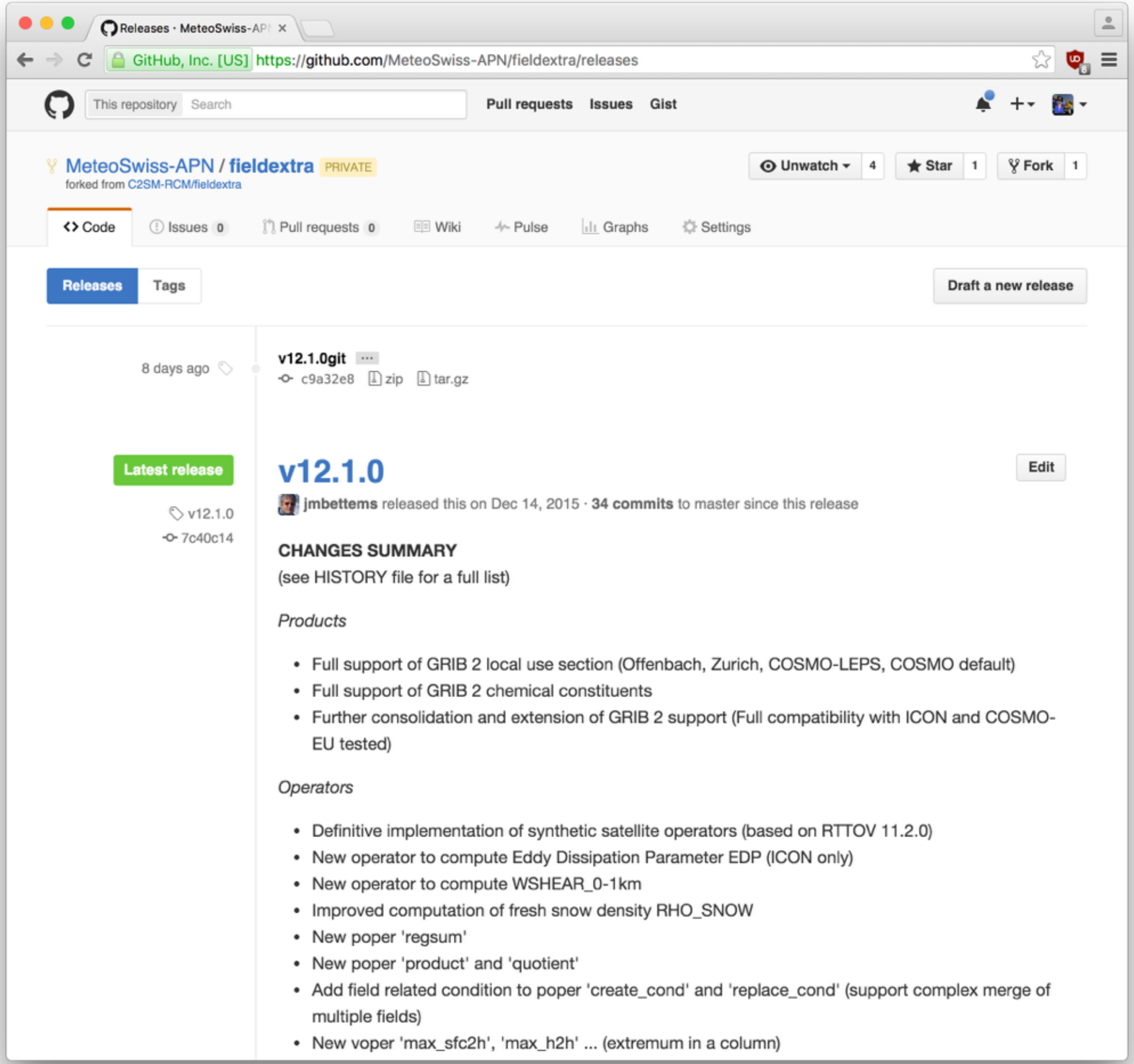
11 Open ✓ 2 Closed Author Labels Milestones Assignee Sort

- AdvectionPDBott.DoTracers fails when run in parallel
  #48 opened 11 days ago by cosunae — 0
- Integrate Clang-Tidy into Jenkins Build Plan `enhancement`
  #25 opened on Dec 2, 2015 by pspoerri — 0
- Update the C++ Code Style `enhancement`
  #24 opened on Dec 2, 2015 by pspoerri — 1
- Integrate SLAdvection `enhancement`
  #22 opened on Dec 2, 2015 by pspoerri — 0
- Pollen dataset breaks testsuite in the COSMO Pull Request with Debug enabled due to TIME LIMIT `bug`
  #20 opened on Dec 1, 2015 by pspoerri — 1
- Profile dycore with mpiP `kesch` `performance`
  #17 opened on Dec 1, 2015 by cosunae — 2
- Test expandable parameters in AdvectionPDBott #5 `kesch` `performance`
  #15 opened on Nov 30, 2015 by cosunae — 8
- Evaluate performance impact of variable copy of the Dycore step `kesch` `performance`
  #14 opened on Nov 25, 2015 by pspoerri — 0
- Fix Cray COSMO Serialize Options File on Kesch `enhancement`
  #10 opened on Nov 20, 2015 by pspoerri — 0
- Evaluate Stencil Problem for cKParallel in HorizontalAdvectionPPTP.cpp `bug` `daint`
  #9 opened on Nov 20, 2015 by pspoerri — 3
- FastWavesSCUnittest.DoSmallStep fails on GPU and CPU in Double Precision `bug`
  #7 opened on Nov 19, 2015 by pspoerri — 3

ProTip! Ears burning? Get @pspoerri mentions with mentions:pspoerri.

C2SM Center for Climate Systems Modeling

Releases


Releases · MeteoSwiss-APN

GitHub, Inc. [US] https://github.com/MeteoSwiss-APN/fieldextra/releases

This repository Search Pull requests Issues Gist

MeteoSwiss-APN / fieldextra PRIVATE
forked from C2SM-RCM/fieldextra

Unwatch 4 Star 1 Fork 1

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

Releases Tags Draft a new release

8 days ago
v12.1.0git
c9a32e8 zip tar.gz

Latest release
v12.1.0
7c40c14

v12.1.0

Edit

jmbettems released this on Dec 14, 2015 · 34 commits to master since this release

CHANGES SUMMARY
(see HISTORY file for a full list)

Products

- Full support of GRIB 2 local use section (Offenbach, Zurich, COSMO-LEPS, COSMO default)
- Full support of GRIB 2 chemical constituents
- Further consolidation and extension of GRIB 2 support (Full compatibility with ICON and COSMO-EU tested)

Operators

- Definitive implementation of synthetic satellite operators (based on RTTOV 11.2.0)
- New operator to compute Eddy Dissipation Parameter EDP (ICON only)
- New operator to compute WSHEAR_0-1km
- Improved computation of fresh snow density RHO_SNOW
- New poper 'regsum'
- New poper 'product' and 'quotient'
- Add field related condition to poper 'create_cond' and 'replace_cond' (support complex merge of multiple fields)
- New voper 'max_sfc2h', 'max_h2h' ... (extremum in a column)

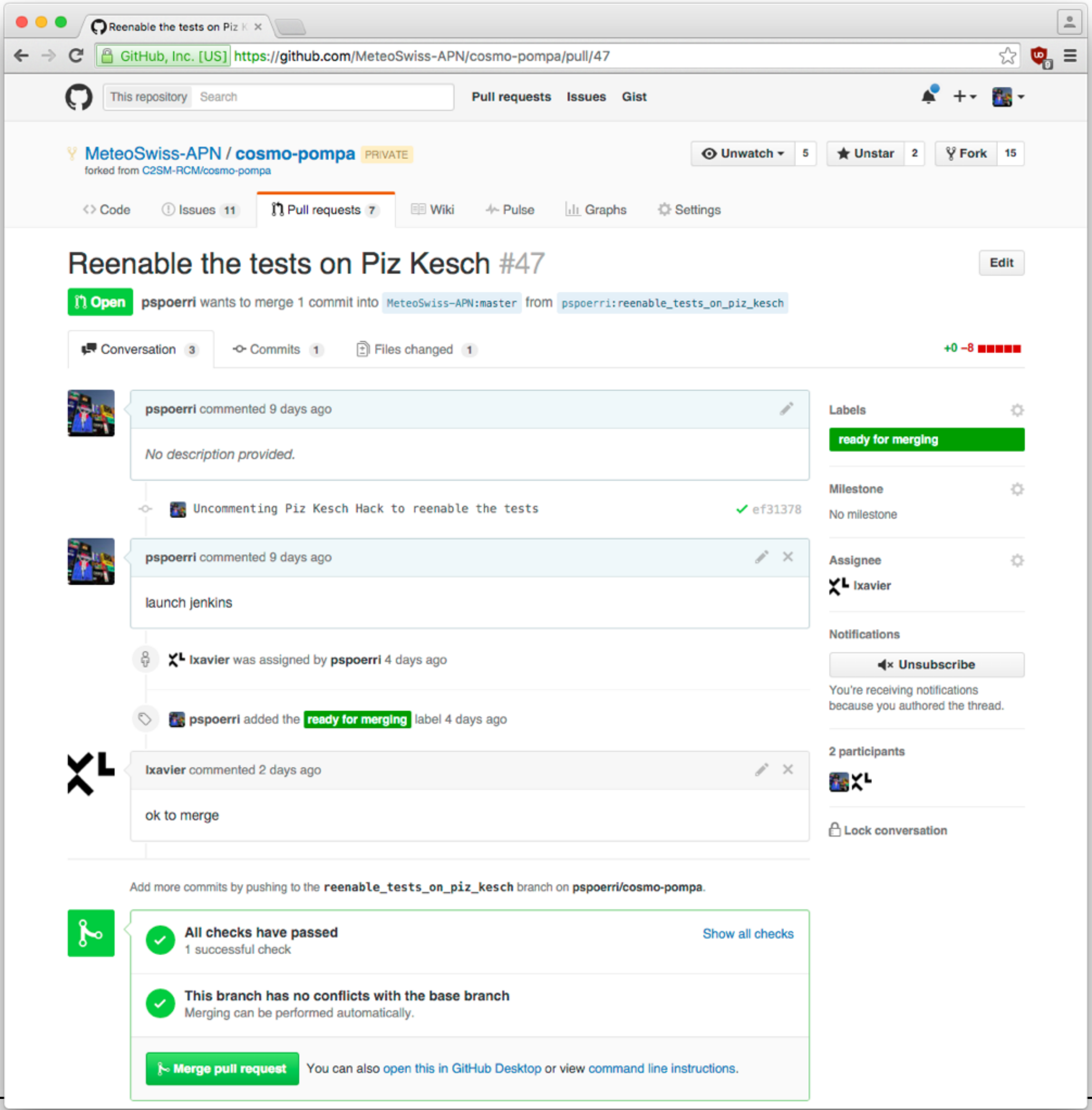
Development Workflow

Developer Workflow

1. Developer creates a fork of $REPO on Github
2. Developer creates a $FEATUREBRANCH
3. Developer opens a pull-request on Github to reintegrate $FEATUREBRANCH
4. Code review/verification
    1. Code Owner reviews code
    2. Automatic testing with Jenkins
    3. Developer fixes the issues reported by Code Owner and Jenkins
5. The code is merged into the master

C2SM Center for Climate Systems Modeling

A typical pull request


Reenable the tests on Piz K

GitHub, Inc. [US] https://github.com/MeteoSwiss-APN/cosmo-pompa/pull/47

This repository Search Pull requests Issues Gist

MeteoSwiss-APN / cosmo-pompa PRIVATE
forked from C2SM-RCM/cosmo-pompa

Unwatch 5 Unstar 2 Fork 15

Code Issues 11 Pull requests 7 Wiki Pulse Graphs Settings

Reenable the tests on Piz Kesch #47

Edit

Open **pspoerri** wants to merge 1 commit into `MeteoSwiss-APN:master` from `pspoerri:reenable_tests_on_piz_kesch`

Conversation 3 Commits 1 Files changed 1 +0 −8

pspoerri commented 9 days ago

No description provided.

`Uncommenting Piz Kesch Hack to reenable the tests` ✓ ef31378

pspoerri commented 9 days ago

launch jenkins

lxavier was assigned by **pspoerri** 4 days ago

pspoerri added the ready for merging label 4 days ago

lxavier commented 2 days ago

ok to merge

Add more commits by pushing to the **reenable_tests_on_piz_kesch** branch on **pspoerri/cosmo-pompa**.

All checks have passed
1 successful check
Show all checks

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Labels
ready for merging

Milestone
No milestone

Assignee
lxavier

Notifications
Unsubscribe
You're receiving notifications because you authored the thread.

2 participants

Lock conversation

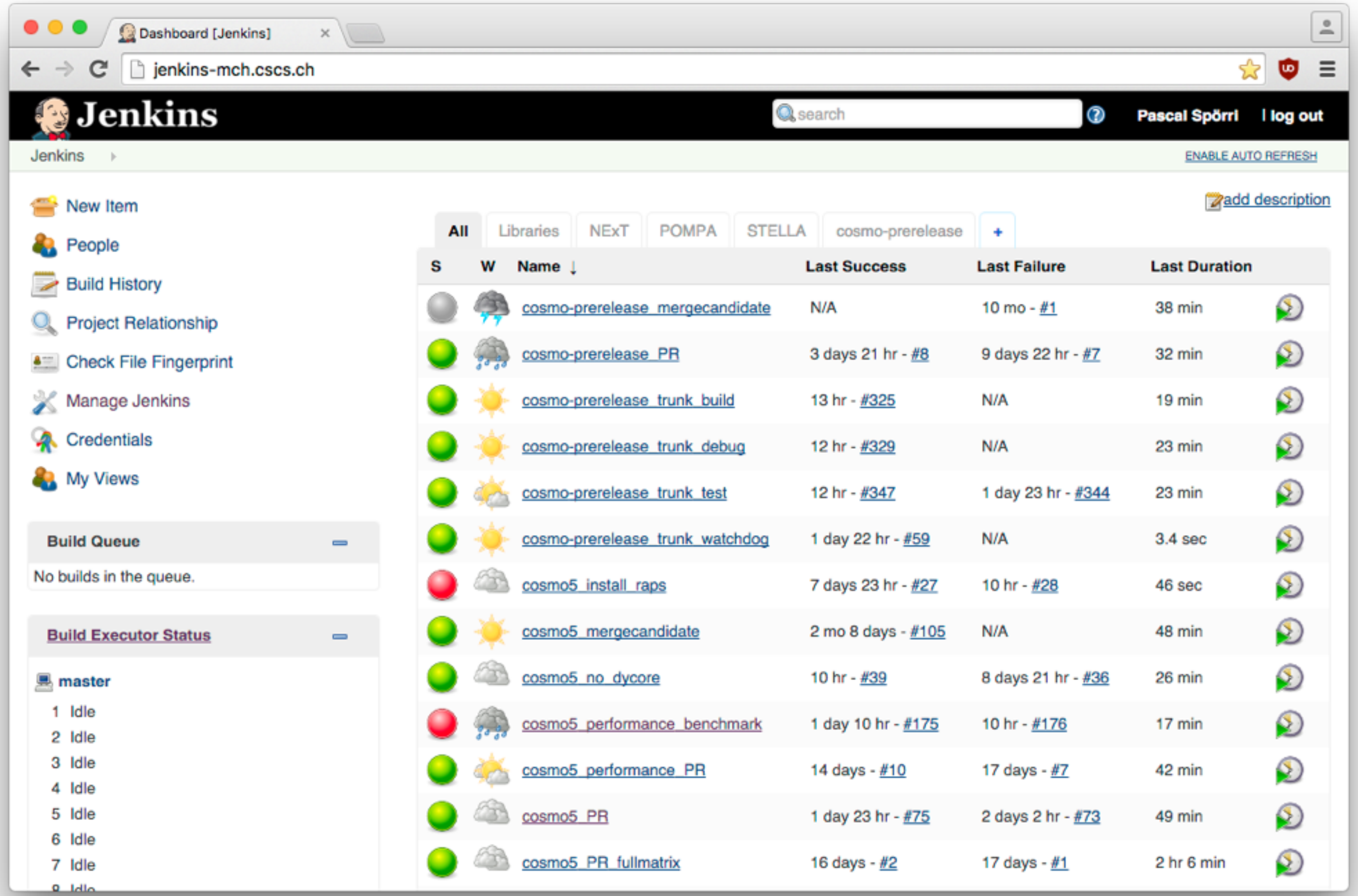
Benefits of the Pull-Request Workflow

- Less code breakages
  - The code is tested automatically, reduces friction with developer
- Developer is responsible to adapt his changes to the code base
- Smaller change sets
- High developer interaction
- Each developer works on his own clone of the repository

Testing and Validation

Jenkins

- Jenkins instance running at CSCS
- Daily builds of the latest version at $repository/master
- Daily tests of the build version



Dashboard [Jenkins]

jenkins-mch.cscs.ch

Jenkins — search — Pascal Spörri | log out

Jenkins › ENABLE AUTO REFRESH

- New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- Credentials
- My Views

Build Queue

No builds in the queue.

Build Executor Status

master
1 Idle
2 Idle
3 Idle
4 Idle
5 Idle
6 Idle
7 Idle

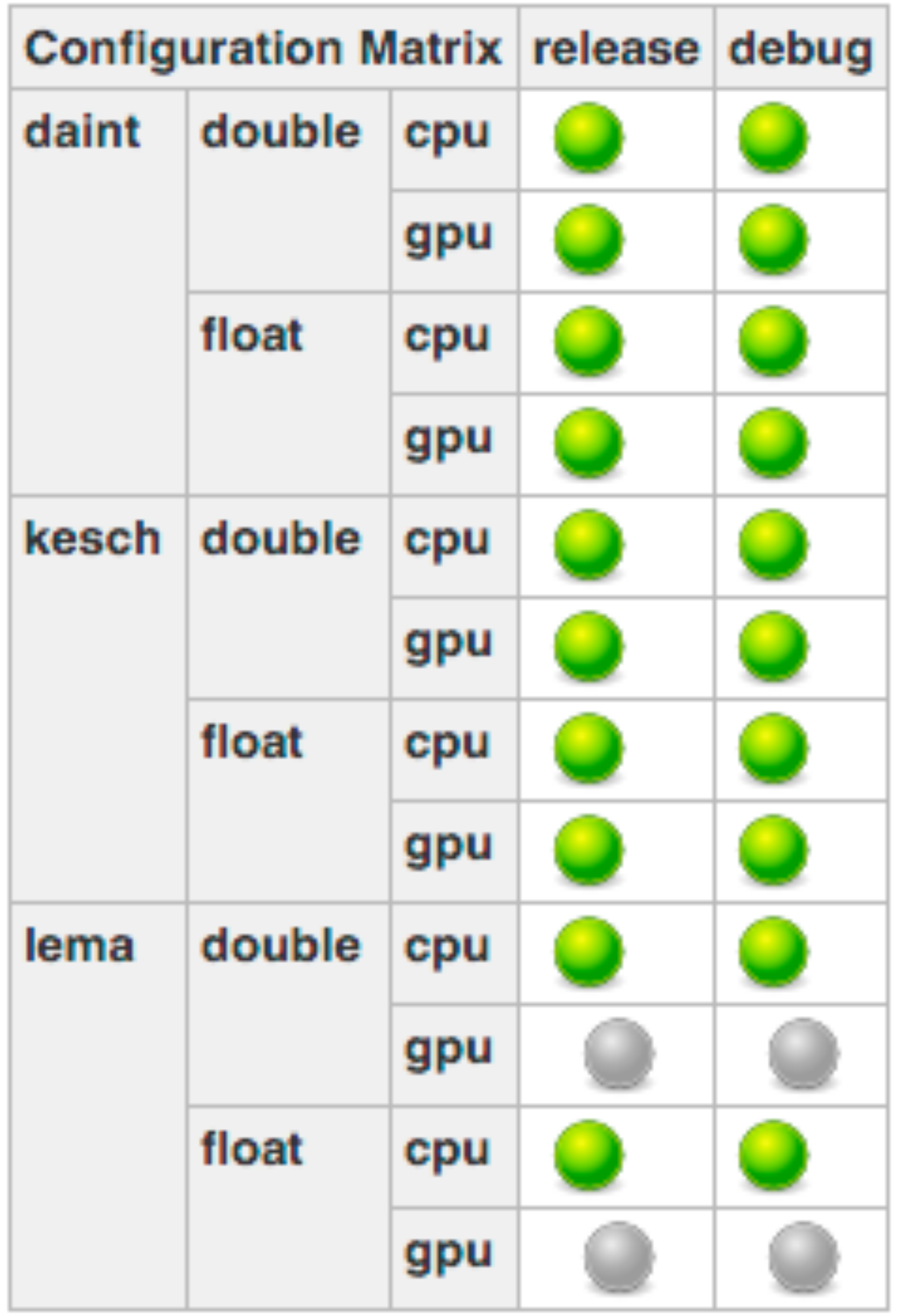
add description

All | Libraries | NExT | POMPA | STELLA | cosmo-prerelease | +

S	W	Name ↓	Last Success	Last Failure	Last Duration
⚪	⛈	cosmo-prerelease_mergecandidate	N/A	10 mo - #1	38 min
🟢	🌧	cosmo-prerelease_PR	3 days 21 hr - #8	9 days 22 hr - #7	32 min
🟢	☀	cosmo-prerelease_trunk_build	13 hr - #325	N/A	19 min
🟢	☀	cosmo-prerelease_trunk_debug	12 hr - #329	N/A	23 min
🟢	⛅	cosmo-prerelease_trunk_test	12 hr - #347	1 day 23 hr - #344	23 min
🟢	☀	cosmo-prerelease_trunk_watchdog	1 day 22 hr - #59	N/A	3.4 sec
🔴	☁	cosmo5_install_raps	7 days 23 hr - #27	10 hr - #28	46 sec
🟢	☀	cosmo5_mergecandidate	2 mo 8 days - #105	N/A	48 min
🟢	☁	cosmo5_no_dycore	10 hr - #39	8 days 21 hr - #36	26 min
🔴	🌧	cosmo5_performance_benchmark	1 day 10 hr - #175	10 hr - #176	17 min
🟢	⛅	cosmo5_performance_PR	14 days - #10	17 days - #7	42 min
🟢	☁	cosmo5_PR	1 day 23 hr - #75	2 days 2 hr - #73	49 min
🟢	☁	cosmo5_PR_fullmatrix	16 days - #2	17 days - #1	2 hr 6 min

Test Matrix

- Target machines
  - Piz Daint (Research, Fall-Back)
  - Piz Lema/Albis (Production)
  - Piz Kesch/Es-Cha (Next-Gen)
- Single and Double Precision
- GPU and CPU testing



Configuration Matrix			release	debug
daint	double	cpu	🟢	🟢
		gpu	🟢	🟢
	float	cpu	🟢	🟢
		gpu	🟢	🟢
kesch	double	cpu	🟢	🟢
		gpu	🟢	🟢
	float	cpu	🟢	🟢
		gpu	🟢	🟢
lema	double	cpu	🟢	🟢
		gpu	⚪	⚪
	float	cpu	🟢	🟢
		gpu	⚪	⚪

Testsuite

- Executed as a shell script
- 38 test cases
  - Split into cosmo1, cosmo2, cosmo7, flake, kenda, pollen
- Prerlease
  - DWD tests
- Results are validated against reference run
  - Mitigate development errors
  - Need to be recomputed when results are changed

C++ Dynamical Core unit testing

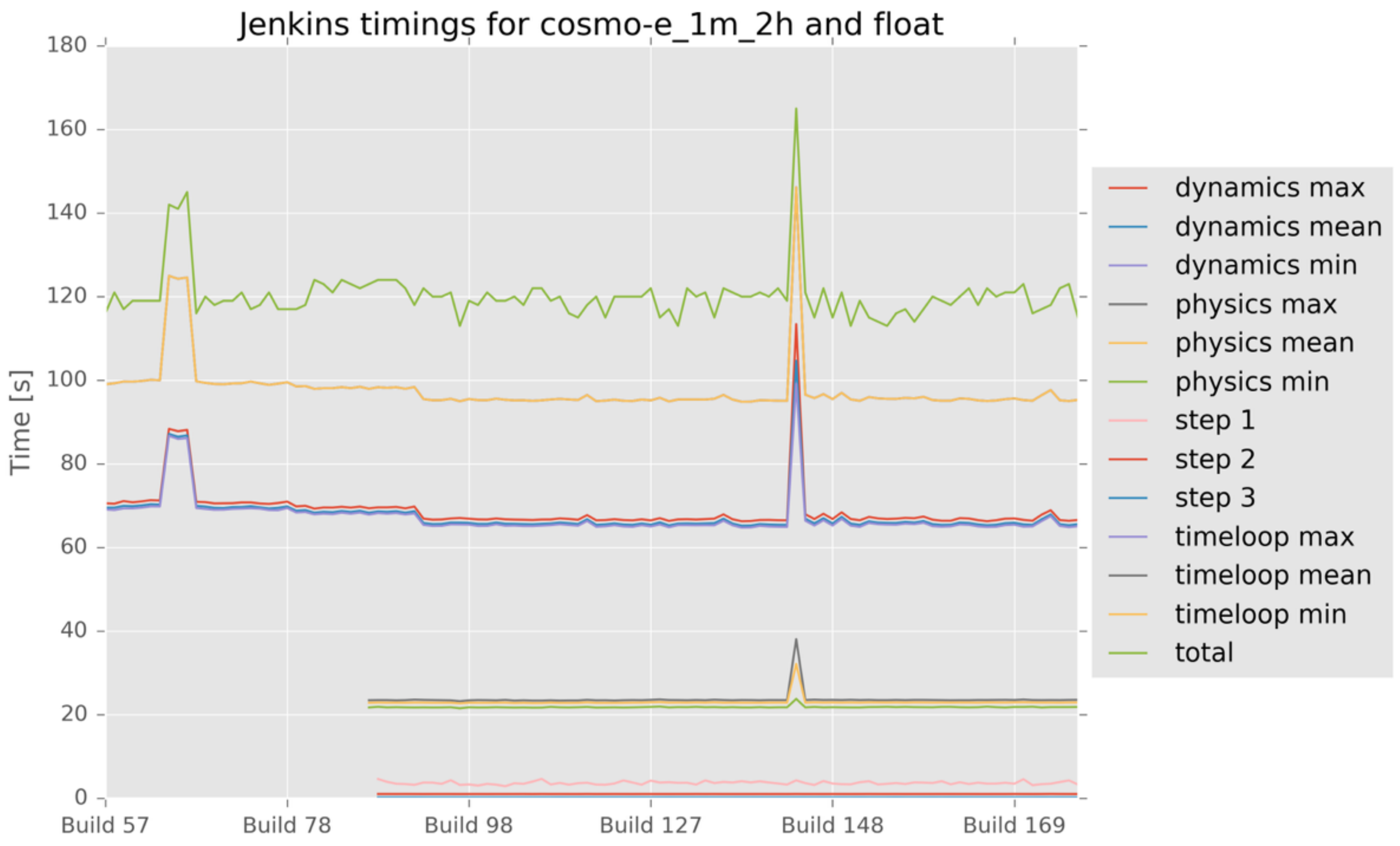
- Special COSMO serialize build is created

```
!$ser savepoint VerticalDiffusionUnittest.PrepareStep-in LargeTimeStep=ntstep

!$ser data u_nnow=u(:,:,:,nnow) v_nnow=v(:,:,:,nnow)
```

- Serialize build is run against test case
    - Generates serialize data of specified fields at each tag
    - Typically 10 time steps
- Each component is tested individually (40 test cases)

Performance Testing

- Small test runs that represent our operational configuration
- Detect performance regressions
- Observe a lot of fluctuations depending on the utilization



Validation for the Future

- Currently not sufficient
- Components often require expert knowledge to validate
  - Automation needed
  - Testsuite not sufficient for correctness
- Testing values in production good for safety but bad for cycles
  - Reductions are expensive on GPUs
- First experiments with analytic tests

New tool for testing

Serialbox

Pascal Spörri
pascal.spoerri@env.ethz.ch

Serialbox

- Serialization framework originally developed for the C++ Dycore
- Developed by MeteoSwiss APND
- Serialize and deserialize Fortran, STELLA, Python Numpy fields
- OpenSource, BSD Clause 2 License
- Available on Github: https://github.com/C2SM-RCM/serialbox

Definitions and Use Cases

- Purpose: Write and read fields from and to the disk at any point in time
- Usefulness
  - Testing
  - Validation of small components

Producer

```
PROGRAM serialbox_producer

  IMPLICIT NONE

  REAL, DIMENSION(5,5,5) :: a

  a = 5.0

  PRINT *, 'Serialize with sum(a)=', sum(a)

  !$ser init directory='.' prefix='SerialboxTest'
  !$ser savepoint sp1
  !$ser mode write
  !$ser data a=a

END PROGRAM serialbox_producer
```

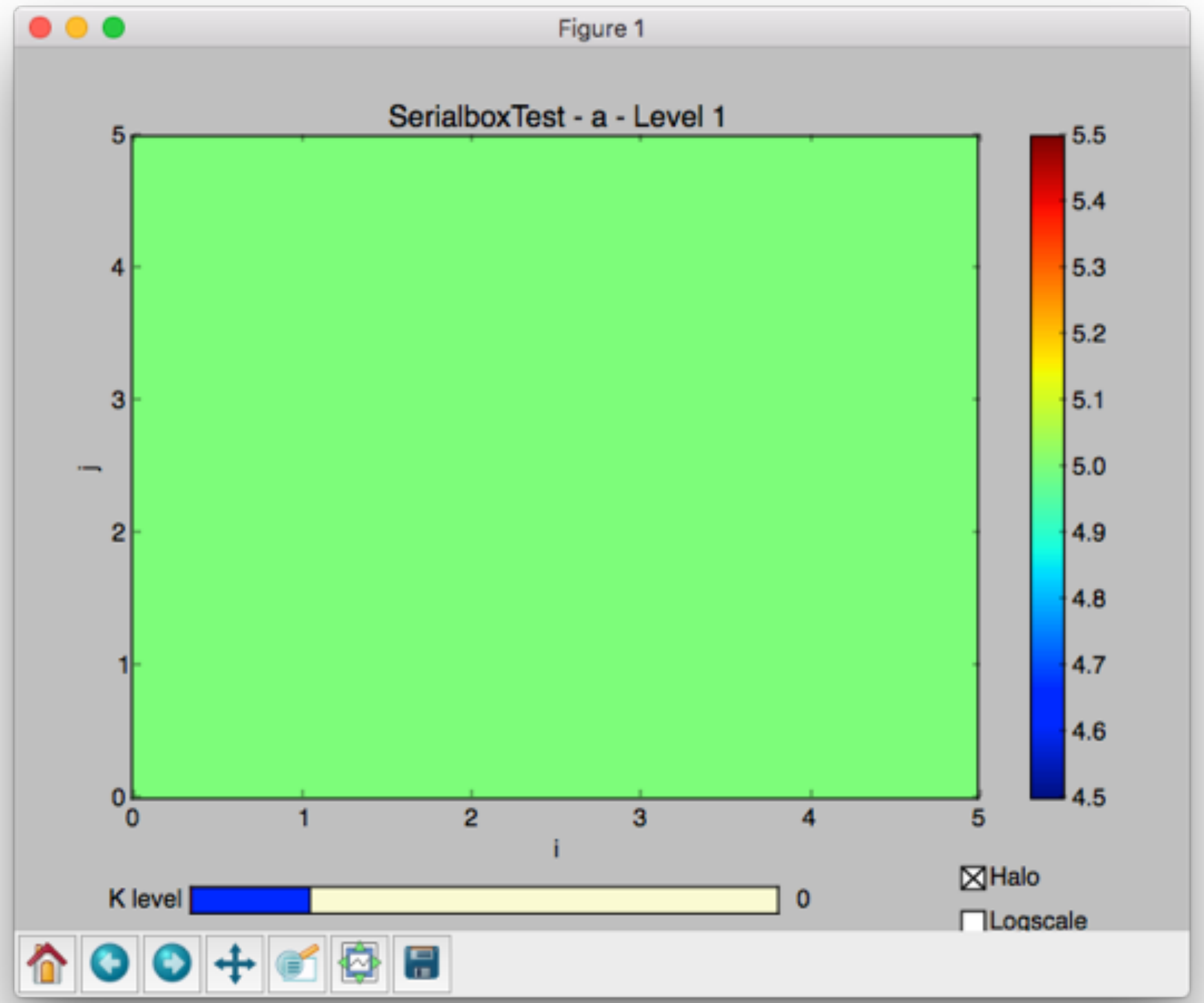
Consumer

```
PROGRAM serialbox_consumer

  IMPLICIT NONE

  REAL, DIMENSION(5,5,5) :: a

  a = 0.0

  !$ser init directory='.' prefix_ref='SerialboxTest'
  !$ser savepoint sp1
  !$ser mode read
  !$ser data a=a

  PRINT*,'After read from serializer: sum(a)=', sum(a)

END PROGRAM serialbox_consumer
```

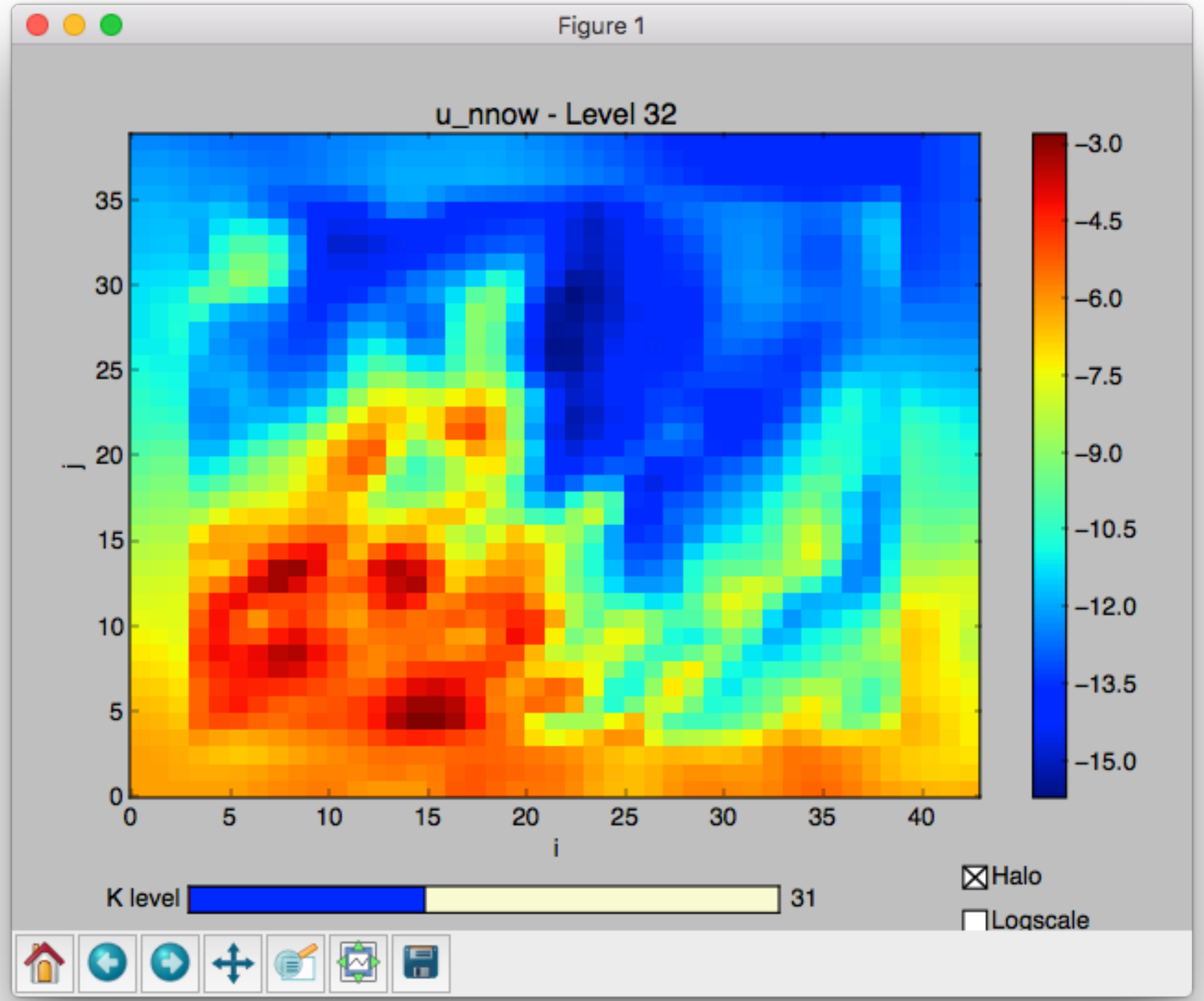
Mixed Read/Write is also possible

Python

```
from serialbox import *
ser = Serializer('.', 'SerialboxTest')
print(ser)
# { 'sp1' = [...] }
print(ser['sp1']['a'])
# array([[[ 5.,  5.,  5.,  5.,  5.],
#          [ 5.,  5.,  5.,  5.,  5.],
#          [ 5.,  5.,  5.,  5.,  5.],
#          [ 5.,  5.,  5.,  5.,  5.],
#          [ 5.,  5.,  5.,  5.,  5.]], ... , dtype=float32)
Visualizer(ser['sp1']['a'], 'SerialboxTest - a')
```



Visualizer Demo



Usage sample

COSMO

Metainformation

```
!$ser savepoint AdvectionPDBottUnittest.DoTracers-in LargeTimeStep=ntstep
!$ser data u=u(:,:,:,nnew) u_nnow=u(:,:,:,nnow)                             &
!$ser&     v=v(:,:,:,nnew) v_nnow=v(:,:,:,nnow)                             &
!$ser&     w=w(:,:,:,nnew) w_nnow=w(:,:,:,nnow)                             &
!$ser&     rho=rho(:,:,:)
!$ser tracer %all@nnow
CALL advection_pd(u_half(:,:,:), v_half(:,:,:), w_half(:,:,:), nnow, dt, &
                  im, ip, j2dim, ny_2dim)
!$ser savepoint AdvectionPDBottUnittest.DoTracers-out LargeTimeStep=ntstep
!$ser data rho=rho
!$ser tracer %all@nnew
```

Conventions

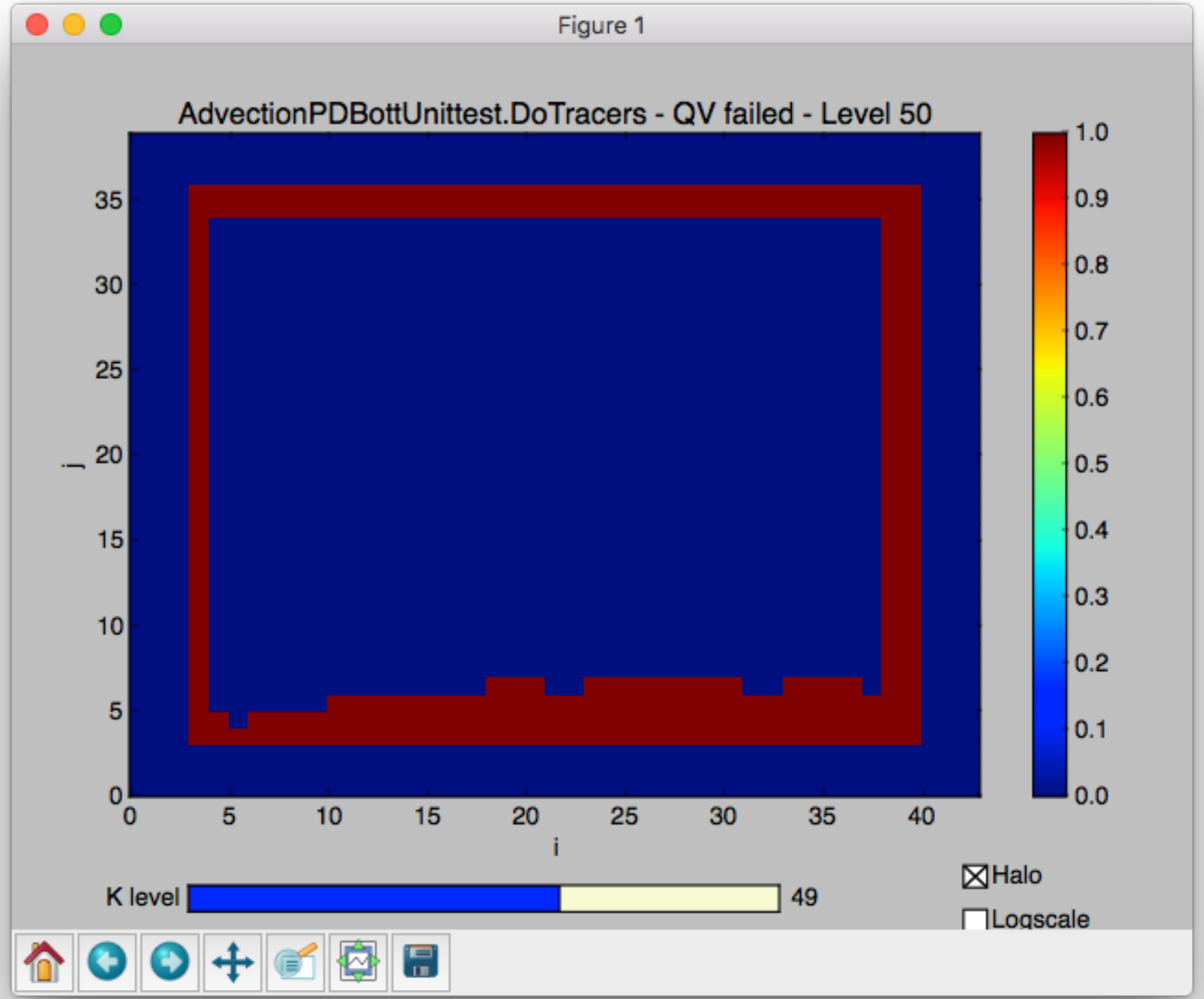
- The name determines the unit test and the function
  - AdvectionPDBottUnittest — The unit test
  - DoTracers — The function
- The -in and -out postfix determines unit test in/output
- The meta information stores the current iteration

C2SM
Center for Climate
Systems Modeling

C++ Dycore unit tests

```
TEST_F(AdvectionPDBottUnittest, DoTracers)
{
        Specify input data

        Specify reference data

    for(int i = 0; i < iterations; ++i)
    {
            Call Advection PD

            Verify Result
    }
}
```

Direct error visualization



Most likely a problem with the boundary conditions

Status

- Support for COSMO fields
- Floating point precision agnostic
- Fortran module support: Work in progress
- Unit tests: Work in progress
- Documentation: Planned

C2SM
Center for Climate
Systems Modeling

Questions?

Git:
https://wiki.c2sm.ethz.ch/C2SM/Git

Serialbox:
https://github.com/C2SM-RCM/serialbox

Source: xkcd.com/1597

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.