

CMS & Git

Giulio Eulisse

Reality check

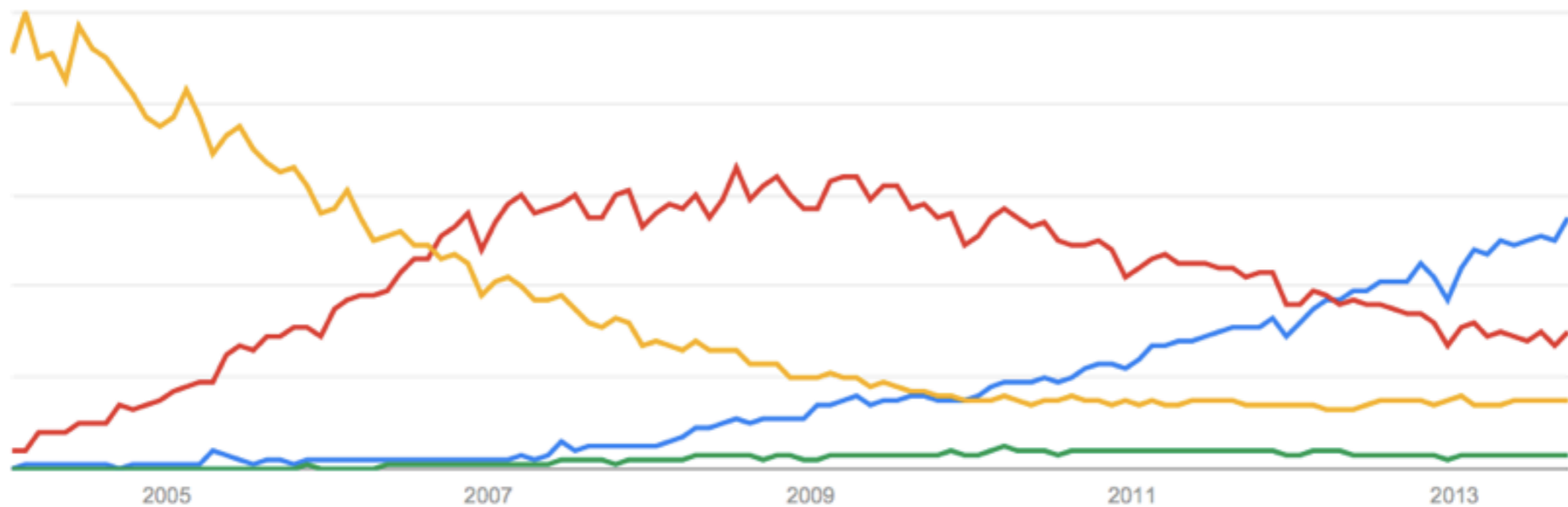
- Every few years or so physics needs to realign to reality:
- The atom is not one big pudding of protons and electrons
- No Technicolor
- I will not comment about SUSY...
- The damn cat is either dead or alive...

Reality check

- The same is true for HEP software:
 - Fortran vs. C++
 - Grid vs. Cloud
 - Quattor vs. Puppet
 - In house tools vs. industry standards
- CVS was designed in 1986, no new release since 2008. The rest of the world has long moved on...

Popularity*

git ● svn ● cvs ● hg ●



* as defined by Google Trends

A brave new world

- Long shut down as an opportunity to align to 2013:
 - New Version Control Systems
 - New Development Model
- The goal is to minimize the potential barrier required for newcomers in the next 5-10 years and to minimize the extra custom layers required.

CMS CVS usage

- 10+ years of experience with CVS

For over the last 10 years CMS has been using CVS as their VCS, initially on our own server, then on CERN/IT ones, AFS based (sigh).

- 10+ years of workarounds and bad practices due to ill-fated CVS design

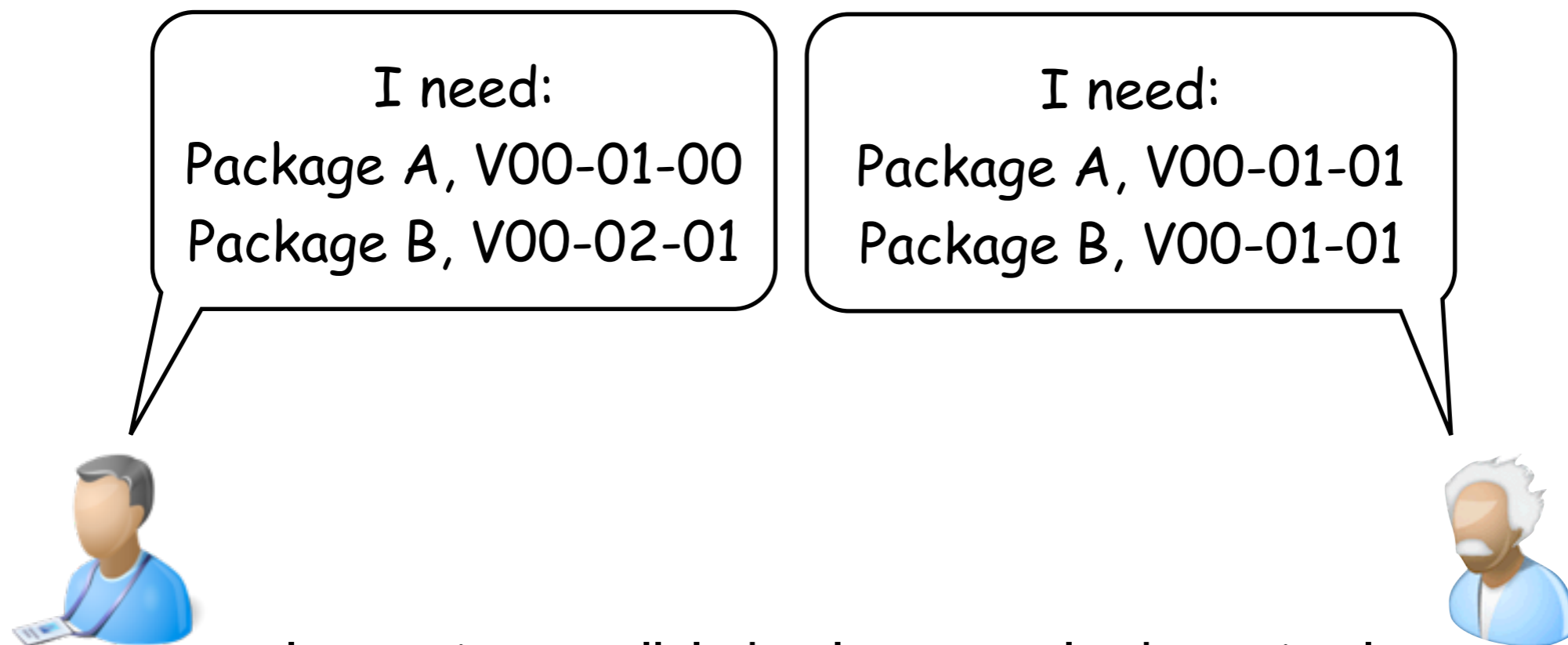
Massive hierarchy of folders (“packages”) to minimize concurrent development

The HEAD of the repository was technically a “junk-yard”

Per package tags

CMS Tag Collector, rich-man release management tool

Typical integration issue



- Integrating parallel developments lead to circular dependencies, dropping bug fixes, VXX-YY-ZZ-UU-WW tags, usage of “-j” option in CVS...
- To avoid there was a proliferation of packages in order to avoid parallel development altogether.
- Sometimes the only solution was to keep a wiki page with 20 or so file / revision pairs...

Picking up a better development model

- Continuous Integration:
 - HEAD of release branches must be always in a releaseable state (apart from bugs, of course). No more unfinished developments in the HEAD.
- Topic Branches:
 - New developments / bug-fixes need to happen in separate branches and will be merged in the release branch once ready and tested.
- Easy integration of unrelated changes:
 - Fixing a warning and changing a cut in the same package should not require 3 different tags in order to be done
- Easy traceability of changes:
 - It must be easy to track the history of a project globally, not only on a per-file basis.

Enter git

- git matches extremely well such a development model
 - Easy branching
 - Easy merging of changes
 - Disconnected operations
 - Designed for distributed development

git based workflow

- Still one authoritative repository: **<https://github.com/cms-sw/cmssw>**

Includes full per file history of the old CVS repository, at least for files which entered some release. All release tags ported to the new repository.

- One branch per active release

CMSSW_7_0_X, CMSSW_6_2_X, CMSSW_5_3_X, CMSSW_4_4_X, etc..

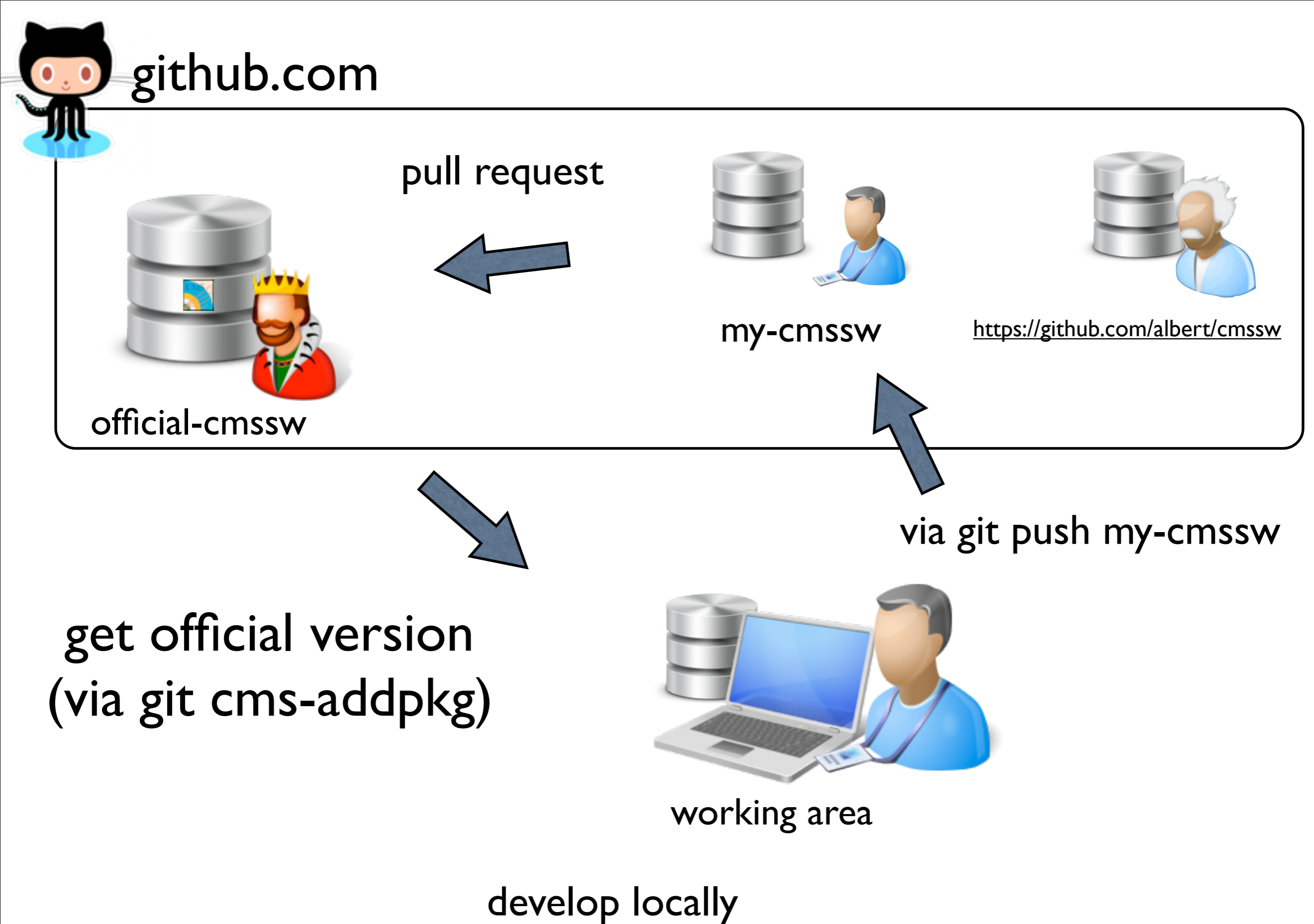
- Topic branches

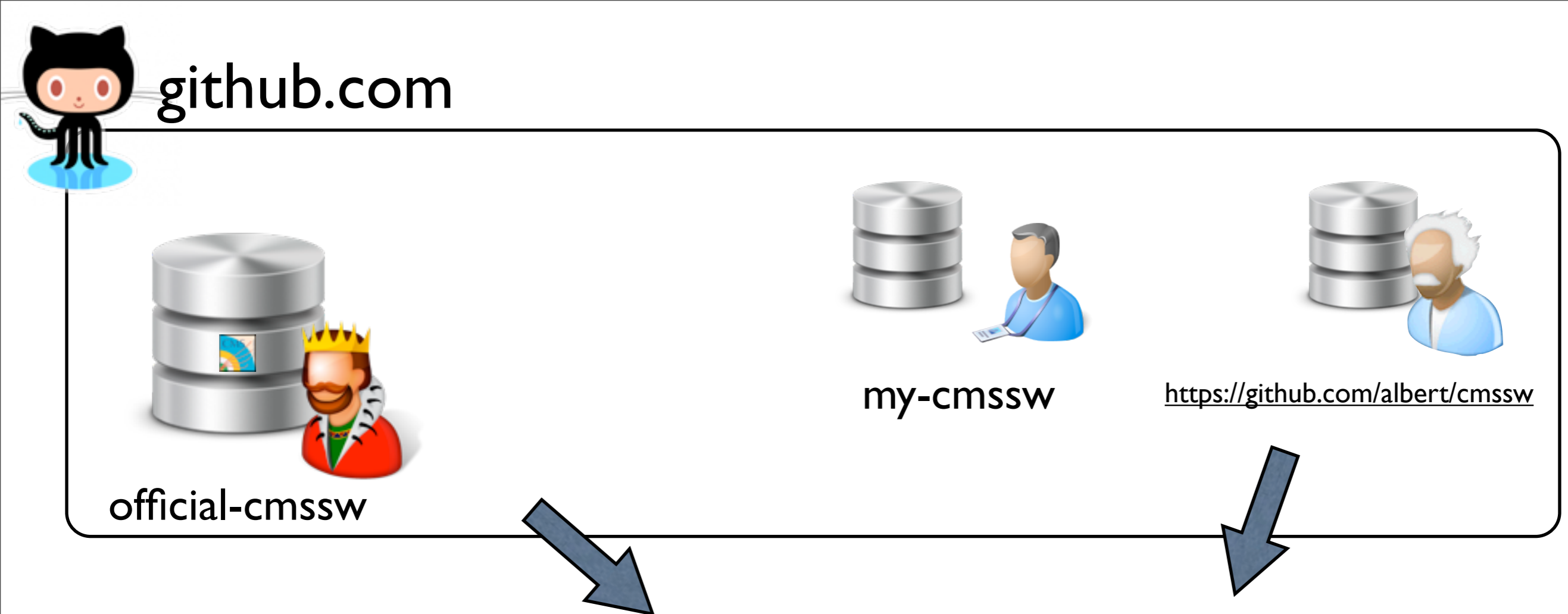
Developers use their private repositories to prepare additions in form of a branch and submit them for integration via the Github “Pull Requests” mechanism.

Topic branches can also be shared among peers to ease collaboration.

- Sign-off workflow

Pull Requests are validated by various Offline Software conveners (RECO, SIM, CORE, HLT, etc) and merged into the main branch. Automated signature handling by a bot.





get unmerged pull request / official
topic branches
(`git cms-merge-topic <pullrequest-id>`)

via `git cms-merge-topic albert:his-branch`



working area

`git cms-merge-topic` to fetch
other people branches

Merged

ktf merged 1 commit into cms-sw:CMSSW_7_0_X from ktf:remove-friend... 2 days ago

4

#824

Discussion

 Commits 1

 Files Changed 4



ktf opened this pull request 2 days ago

Edit

Remove friendship relations to removed DaqSource.

No one is assigned 

Milestone: CMSSW_7_0_0_pre4 

No description given.

Merged

+ 0 additions

- 4 deletions

4 participants    



 **cmsbuild** commented

6 days ago



A new Pull Request was created by [@ptcox](#) for CMSSW_7_0_X.

Consumes to CSCValidation

It involves the following packages:

RecoLocalMuon/CSCValidation

RecoLocalMuon/CSCSegment


RecoLocalMuon/CSCRecHitD

[@thspeer](#), [@slava77](#) can you please review it and eventually sign? Thanks.

You can sign-off by replying to this message having '+1' in the first line of your reply.

You can reject by replying to this message having '-1' in the first line of your reply.



 **thspeer** commented

4 days ago



-1

Changes are already contained in [#743](#) (already signed) and [#763](#)



slava77 commented

3 days ago



+1

(If this is still making it to pre4, please merge, otherwise, we can take a bit of time to solve CPU/timing degradation)

tested [a311925](#) in CMSSW_7_0_X_2013-09-10-0200

As expected, observe significant improvement (mostly in endcaps) of reco-true matching in phi (mainly) and a bit in eta. This is visible in electron and photon gun samples, nothing is really visible in high-PU or QCD samples, as it should be (no real stuff to do matching here).

I observe no degradation in MET.

Supposedly, there should also be an improvement in tau reco (mostly a guess, nothing visible in the test samples, which don't really have clean taus)

Main regression is in CPU performance: increase of a factor of 4-6 in timing of particleFlowClusterECAL.

Physics improvements would benefit more from a full validation cycle now.



cmsbuild commented

3 days ago



This pull request is fully signed and it will be integrated in one of the next IBs unless changes or unless it breaks tests.
@ktf can you please take care of it?



ktf referenced this pull request from a commit

3 days ago



ktf

Merge pull request #706 from lgray/fix_ECALpfclusterpos ...

[f98e088](#)

Merged



ktf merged commit [f98e088](#) into [cms-sw:CMSSW_7_0_X](#) from [lgray:fix_ECALpfclusterpos](#) 3 days ago

Closed



ktf closed the pull request 3 days ago

Browse IssuesMilestones


New Issue

Everyone's Issues41

Assigned to you1

Created by you6

Mentioning you16

Milestone: CMSSW_7_0_0_pre5 

41 open Issues · Due in 9 days

- Labels
- alca-approved1

alca-pending3


analysis-pending2

core-approved2


core-pending2

core-rejected2


- ⊗ Clear milestone and label filters
- 41 Open1 ClosedSort: Recently updated ▾
- ◀12▶
- CloseLabel ▾Assignee ▾Milestone ▾

☐  Multithread Framework -- Update MessageLogger to use new thread-safe Service signals analysis-pending core-rejected visualization-rejected pending-signatures #827


Opened by Dr15Jones ⌚ updated 2 hours ago 💬 9 comments

☐  Consumes to CSCOffline monitor - fix 1 dqm-pending pending-signatures #832


Opened by ptcx ⌚ updated 3 hours ago 💬 1 comment

☐  Fix chimney hole 70x geometry-pending pending-signatures #836


Opened by TaiSakuma ⌚ updated 3 hours ago 💬 1 comment

☐  Fix particleFlowClusterECAL timing. reconstruction-pending pending-signatures #837

Opened by lgray ⌚ updated 3 hours ago 💬 1 comment

☐  DataFormats -- changes for large object passed by value found by clang s... l1-pending analysis-pending core-approved reconstruction-approved generators-rejected pending-signatures #612

Opened by gartung ⌚ updated a day ago 💬 52 comments

☐  Migration to consumes -- EGamma DQM dqm-pending pending-signatures #812



slava77 started a discussion in the diff 3 days ago



RecoEcal/EgammaCoreTools/interface/PositionCalc.h

[View full changes](#)

```
...  ... @@ -55,4 +61,151 @@ class PositionCalc
64 +template<typename HitType>
65 +math::XYZPoint
66 +PositionCalc::Calculate_Location( const std::vector< std::pair<DetId, float> >&      iDetIds ,
67 +      const edm::SortedCollection<HitType>*      iRecHits ,
68 +      const CaloSubdetectorGeometry* iSubGeom ,
69 +      const CaloSubdetectorGeometry* iESGeom  )
70 +{
71 + typedef edm::SortedCollection<HitType> HitTypeCollection;
72 + math::XYZPoint returnValue ( 0, 0, 0 ) ;
73 +
74 + // Throw an error if the cluster was not initialized properly
75 +
76 + if( 0 == iRecHits ||
77 +     0 == iSubGeom  )
78 + {
79 +     throw(std::runtime_error("\n\nPositionCalc::Calculate_Location called uninitialized or wrong i
```

1



slava77

repo collab

3 days ago



I think this should be a cms::Exception

Add a line note

Impedance mismatch between HEP and git

Git design assumptions

Disk space is cheap

Developer knows how to build the software from scratch

Developer will happily have a single release area on his laptop SSD disk

CMSSW development workflow

We tend to work in terms of a base release (hosted centrally) on top of which we checkout a few packages in a so called development area.

We tend to create many development areas, one per “TODO” item.

Working around the mismatch -- CMS way

- Sparse Checkout

Feature of git allows to have most of the capabilities of single package checkouts back.

- Reference Cloning (--reference)

A local mirror of the Github repository reduces to few megabytes the overhead of a given work-area. Objects in the git object-storage which are present in the reference repository will be picked up from there, and not copied in the workarea.

What's the status?

CMSSW is 100% migrated to git

We have done a number of releases (both production and development ones) already and we have twice per day integration builds which happily do not compete for CVS locks on AFS.

Rough start, mostly due to broken tools (mea culpa!)

The lesson learned is that if you are finding yourself writing a wrapper or a custom tool, you should probably think twice about it, and then think again if you still think it's a good idea.

Avoid as much as possible the idea of mapping CVS / SVN to git

People are starting to see the light

Even some of the most reluctant developers are starting to appreciate the new development model.

Time scale

From 0 to “Migration Done” in ~6 months, with 50% of an average developer (myself).

By far the most challenging part is discussing with people, preparing documentation and answering questions.

Preparing tools was also a large part of the work, but in many cases the best solution is just to resist, keep it simple and do what everyone else is doing with standard git command

Bad ideas

Some of the helper scripts are overly complicated just to hide things to users which they should probably learn

Migrating the per-file history was probably not the correct thing to do. We should have migrated the per release history, interpolating the per file commits in between

The CVS keywords caused a lot of problems in the beginning. I wish there was a way to nuke them completely from the repository

Bad ideas

I initially wanted to have our own custom interface on top of Github API. This is doable, but it requires dedicated manpower for the development and would have required us to maintain a MemCached (or similar) service, which is really missing the whole point (going from CERN supported Oracle to CMS supported Memcached is no improvement, at all).

Good ideas

Using Github

In rough seas you need a safe harbor which “just works”

Great hosting and support

Pull requests

Ability to comment on pull request and keep review and approval process together

Some people adore the possibility of refining the same topic branch as the review progresses, rather than having to open a new one

Sparse checkout and reference repositories to minimize disk / workarea size

Good ideas

Using git

Parallel developments are mostly no-brainers now

git is already showing a big edge for large changes, distributed across the whole code base

Topic branch development model

Simply the correct thing to do

Github pull requests make it a breeze, even for newcomers

Leveraging Github labels to keep track of approvals

Having a comment bot to handle the various +1 / -1 as incredible as it might sound, turned out to be an extremely good idea

Random quotes

*The slogan for the Subversion was ‘**CVS done right**’. And if you start with that kind of slogan, there is nowhere you can go. **There is no way to do CVS right.***

Linus Torvalds

I was about to cry due to the sanity of an such an idea (moving CMSSW to Github).

Anonymous master student

Please do not use “vim” for the tutorial, it locks our users’ shell.

Anonymous

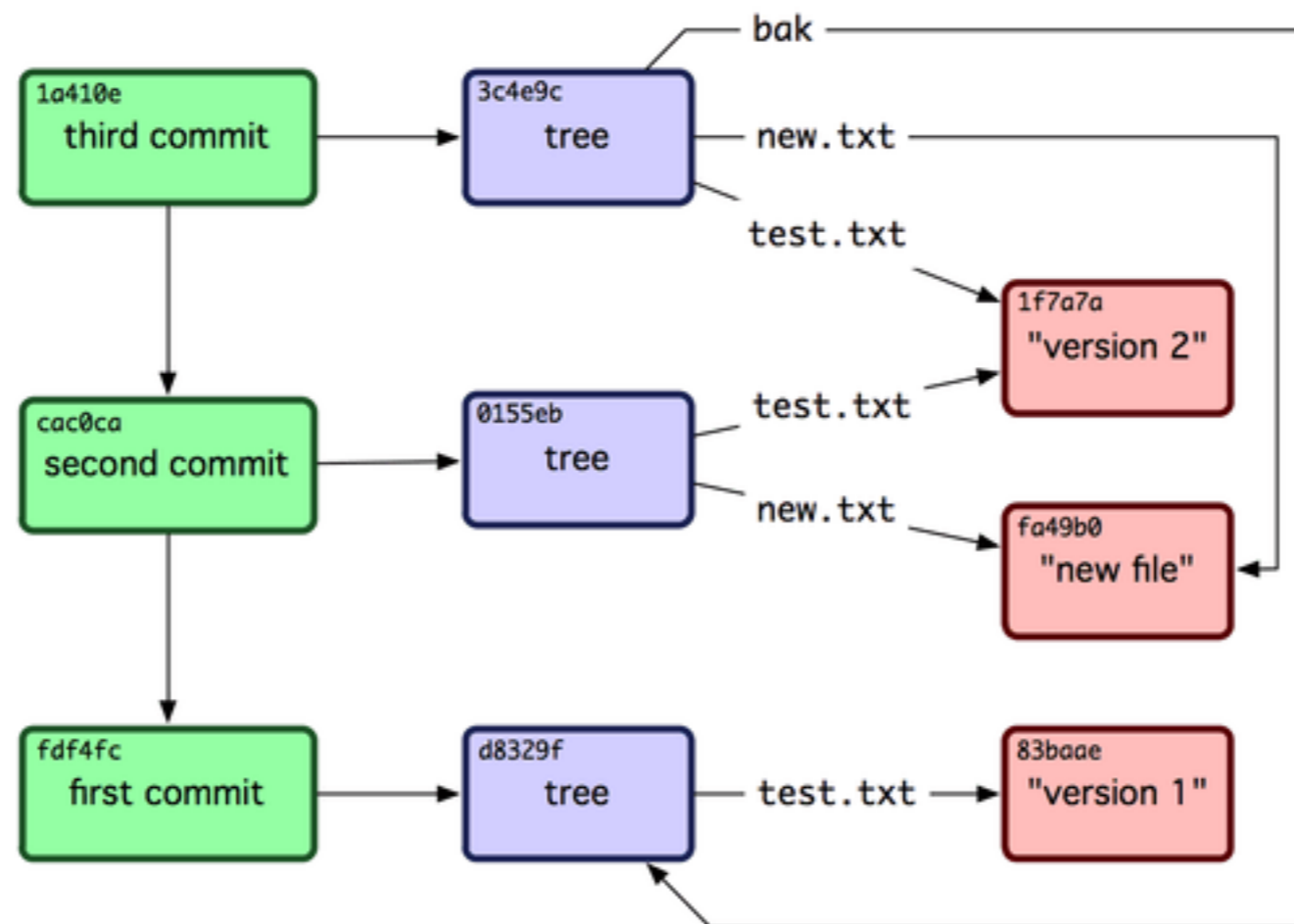
Non-technical questions sometimes don't have an answer at all.

Linus Torvalds

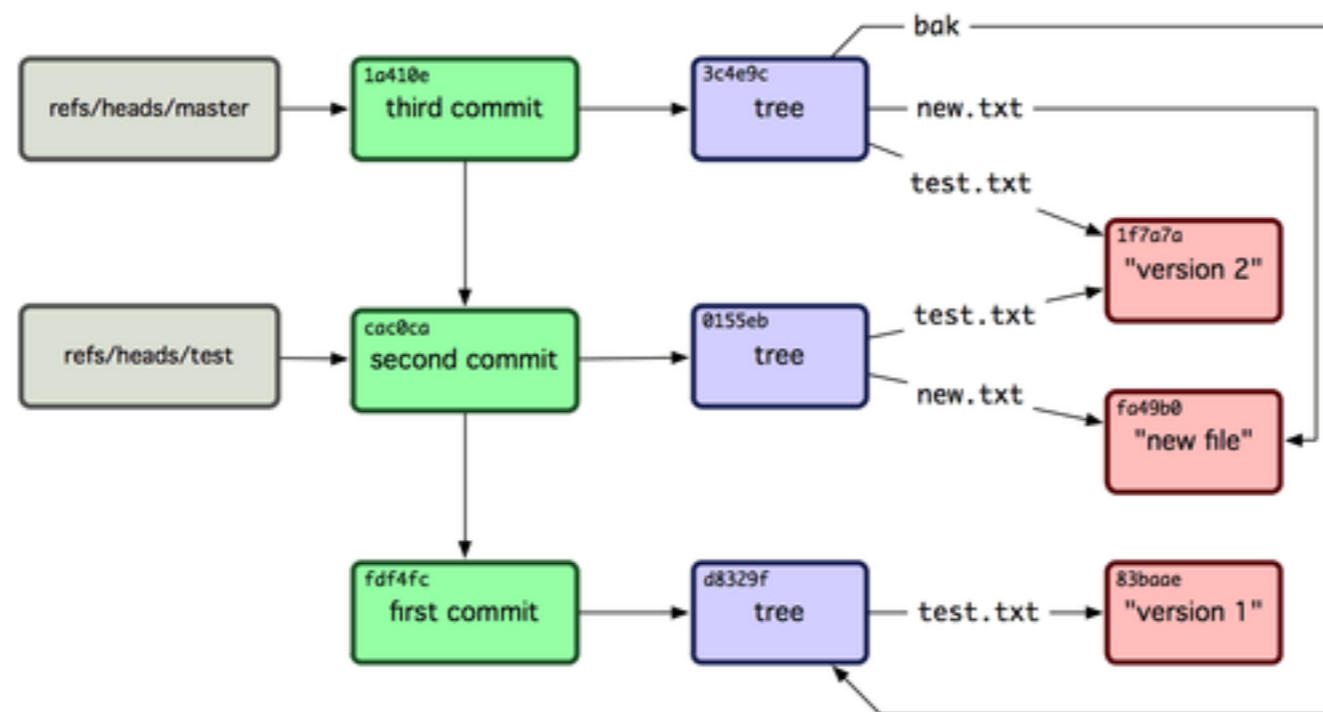
<http://cms-sw.github.io/cmssw>

Backup slides

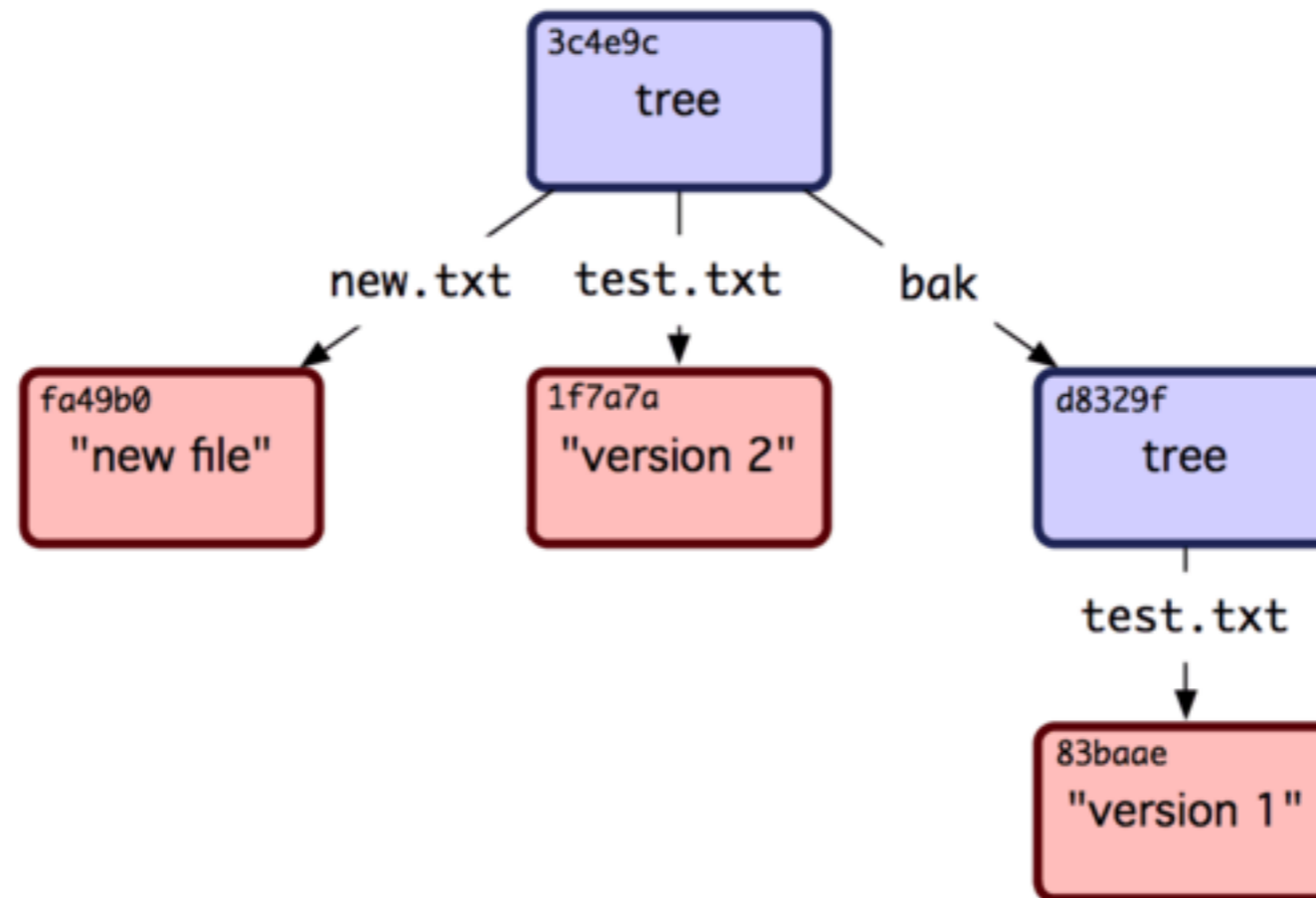
git DB structure



git DB structure



git DB structure



git workflow

