

Git Tutorial

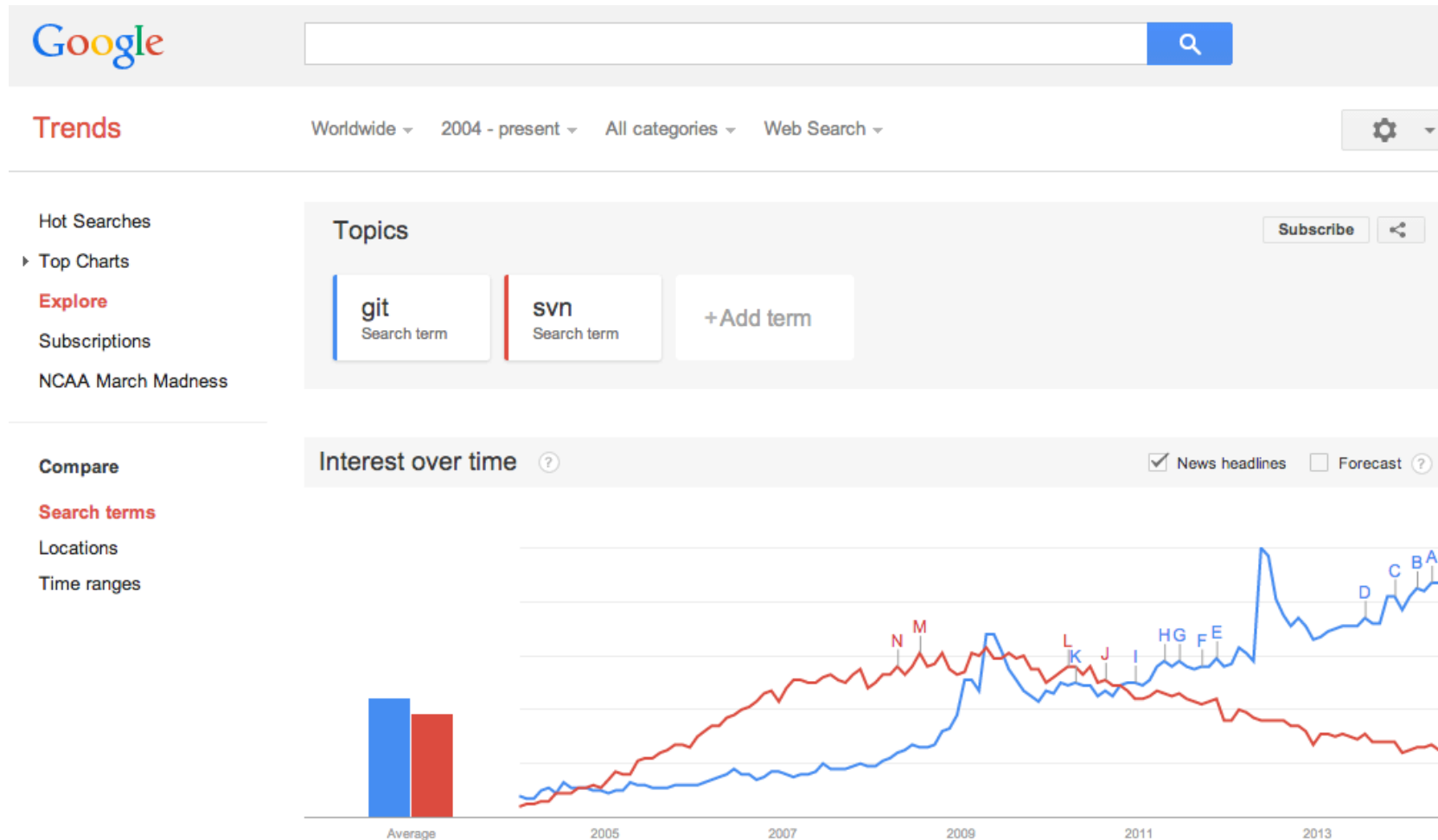
Soonho Kong

2014/05/19, CMU

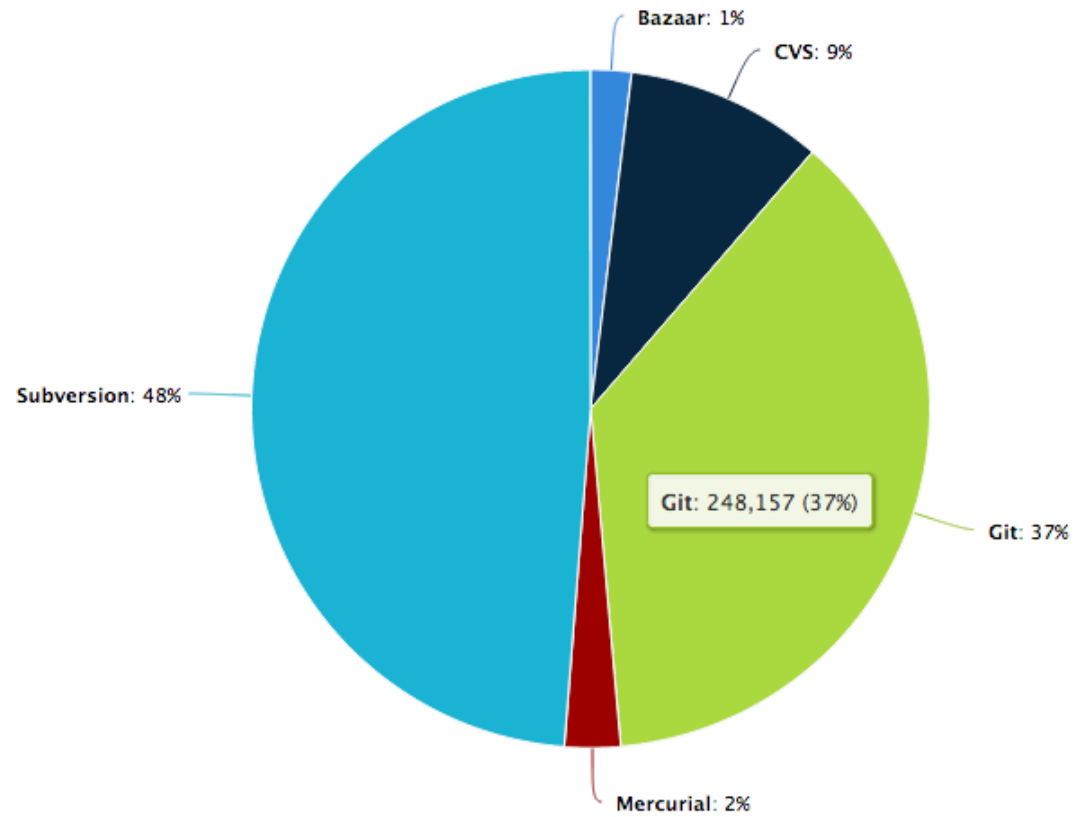
<http://www.cs.cmu.edu/~soonhok>

Introduction





Google Trend: git vs. svn



Ohloh: Compare Repositories

NAME

git - the stupid content tracker

SYNOPSIS

```
git [--version] [--help] [-C <path>] [-c <name>=<value>]  
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]  
      [-p|--paginate|--no-pager] [--no-replace-objects] [--bare]  
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]  
      <command> [<args>]
```

DESCRIPTION

Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high-level operations and full access to internals.

Git: "The stupid content tracker"

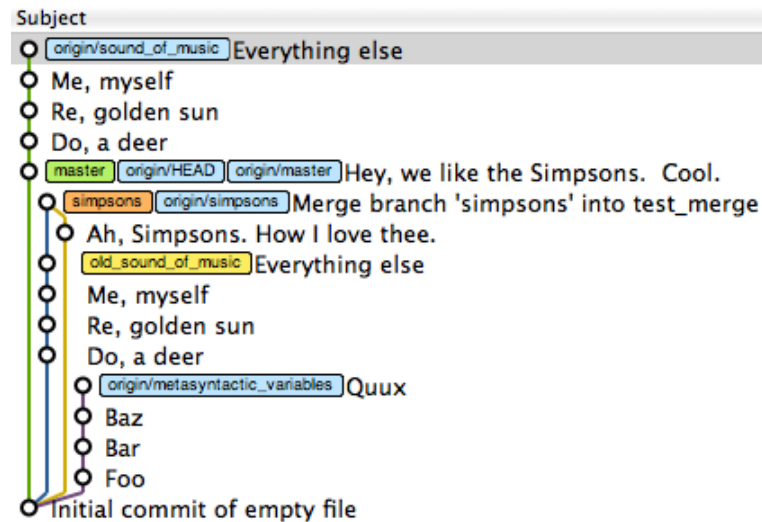


"I did **not** really expect **anyone to use it**
because it's so **hard** to use..."

- "Linus Torvalds goes off on Linux and Git"
(Fake Interview)

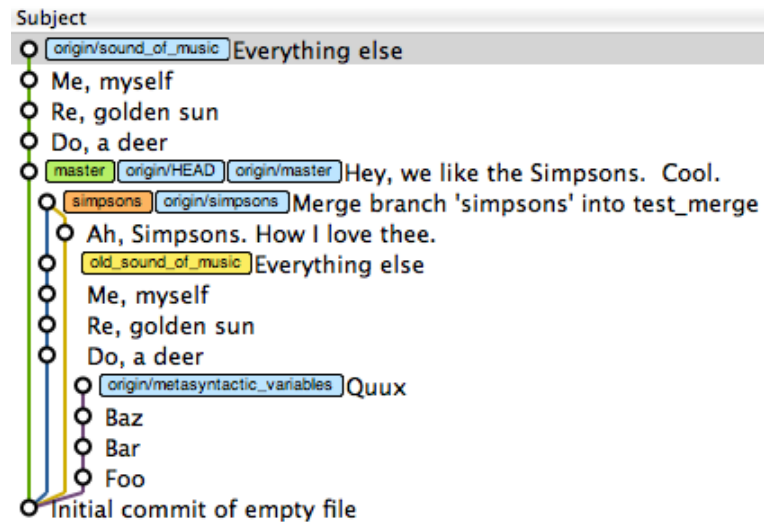
Concepts

Git \simeq DAG



"Once you realize that **git** is **just a DAG** with **commit objects** as **vertices**, and **pointers (refs)** into that graph, it becomes a lot **simpler to understand**"

Git \simeq DAG



Repository \simeq Graph

Node (Commit) \simeq Files

Label (refs) \simeq Branch / Tag / etc

Commit Three Stage Thinking



working



staging



history

Commit Three Stage Thinking



working

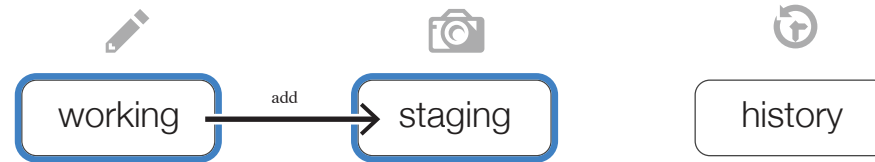


staging

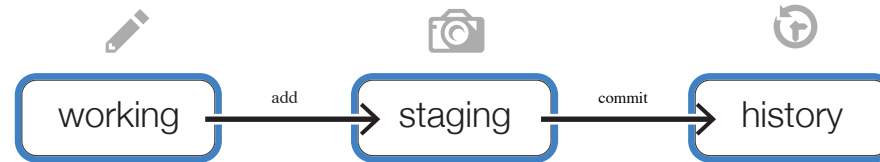


history

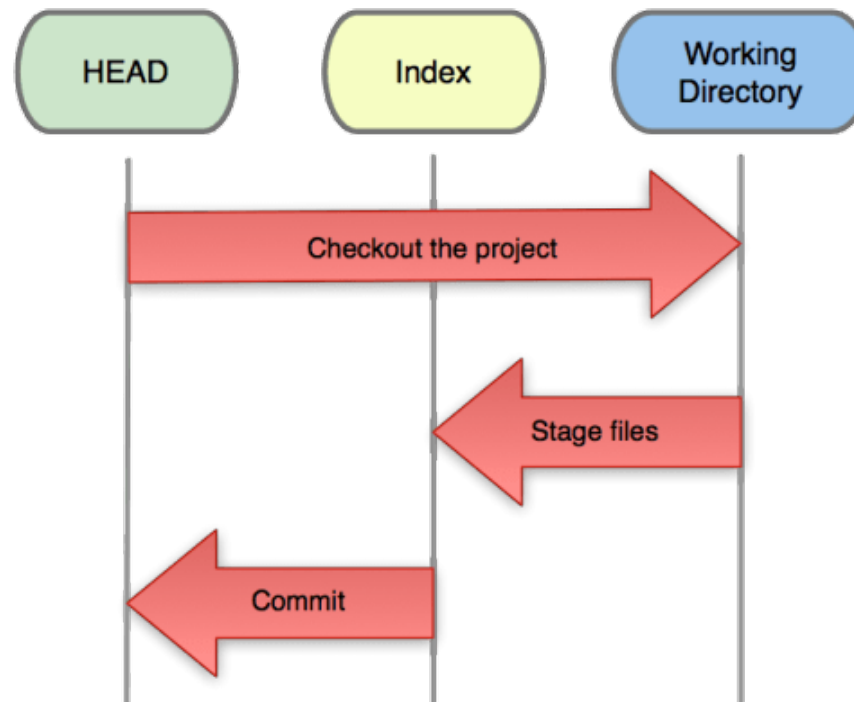
Commit Three Stage Thinking



Commit Three Stage Thinking



Workflow



Install and Config

Install

OSX

```
$ brew install git
```

Linux

```
$ sudo apt-get install git
```

Windows

```
Visit http://git-scm.com/
```

Config: Name & Email

```
$ git config --global user.name "your-name"  
$ git config --global user.email "your-email-address"
```

Config: Useful Settings

```
# Colorize console output
$ git config --global color.ui auto

# Force files to be LF on Mac/Linux
$ git config --global core.autocrlf input

# Force Windows to convert to CRLF
# on checkout and to LF on `add`
$ git config --global core.autocrlf true
```

Config: Like a Boss!

By default for "git log -all"

```
commit 3eefa541df947b46f4c1e9e98ec44afd100a91f7
Author: Soonho Kong <soonhok@cs.cmu.edu>
Date: Tue May 13 15:31:57 2014 -0400

    feat(cmake): use capd-4.0

    fix cmakefile to add autoreconf

    not use boost

    use libcapd.a

commit f23e070b1a610f4b7b2b5eb4a0863a6108f7308a
Author: Daniel Bryce <danbryce410@gmail.com>
Date: Thu May 15 07:52:28 2014 -0500

    feat(dsolvers): variable selection heuristic for bmc

commit 3d6f6994d76c586746c9f4cc6e512378261842bc
Author: Soonho Kong <soonhok@cs.cmu.edu>
Date: Thu May 15 00:09:18 2014 -0400

    use libcapd.a

commit e475ff37904b52e07e11301d397f04e58a6b4463
Author: Soonho Kong <soonhok@cs.cmu.edu>
```

Config: Like a Boss!

Now, "git la"

```
* 3eefa54 (capd4) feat(cmake): use capd-4.0 Soonho Kong, 6 days ago
| * f23e070 (dan/bmc_heuristic) feat(dsolvers): variable selection heuristic for bmc Daniel Bryce, 4 days ago
| * 76636e8 feat(heuristics): added variable selection heuristic based upon mode paths Daniel Bryce, 12 days ago
| * f6e1143 (dan/stable, dan/master) fix(cmake): Added lzma to linux build Daniel Bryce, 2 weeks ago
| * 5b0e9cd feat(smtsolver): added debugging and nra_solver suggestions for variable selection Daniel Bryce, 2 weeks ago
| | * 3d6f699 (blessed/capd) use libcapd.a Soonho Kong, 4 days ago
| | * e475ff3 not use boost Soonho Kong, 5 days ago
| | * 537a459 (origin/capd4, capd) fix cmake: add autoreconf Soonho Kong, 5 days ago
| | * 970fe14 feat(cmake): use capd-4.0 Soonho Kong, 6 days ago
| | /
| | /
| | /
| 46ef66c (blessed/master, master) feat(opensmt/smtsolvers): add --model option Soonho Kong, 6 days ago
| cc2cce5 fix(opensmt/smtsolvers): rename short_sat => nra_short_sat Soonho Kong, 6 days ago
| | * 2af0548 (refs/original/refs/heads/lib2) feat(opensmt/smtsolvers): add --model option Soonho Kong, 6 days ago
| | * 7ff98fe fix(dsolver/ode_solver): add more log messages Soonho Kong, 13 days ago
| | * c24d907 feat(test): added short_sat to testing Daniel Bryce, 3 weeks ago
| | * 8242c53 feat(solver): Added flags for short_sat and enabled ICP to stop if constraint is SAT even if width is not small enough Daniel Bryce, 3 weeks ago
| | * 3d59c71 feat(dsolvers): icp and nra solvers use util/interval Soonho Kong, 3 weeks ago
| | * 6b84ac4 feat(src): use namespace dreal Soonho Kong, 3 weeks ago
| | * ec983cf feat(dsolvers/icp_solver): add rp_problem display for debugging Soonho Kong, 3 weeks ago
| | * 5b60205 fix(ode_solver): polarity check for invariants was succeeding for negative invariants Daniel Bryce, 4 weeks ago
| | * d533494 fix(dsolvers/ode_solver): clean up LOG msgs Soonho Kong, 4 weeks ago
| | * 91c7df8 fix(dsolvers/icp_solver): clean up LOG msgs Soonho Kong, 4 weeks ago
| | * a438534 chore: update copyright info Soonho Kong, 4 weeks ago
| | * 23f3c9c fix(dsolvers/nra_solver): clean up LOG msgs Soonho Kong, 5 weeks ago
| | * 2650089 fix(dsolver/ode_solver): give better error msg for inv violation Soonho Kong, 5 weeks ago
| | * c1c70de (lib1) refactor: use "nra_" and "nra_ODE" prefix for config vars Soonho Kong, 5 weeks ago
| | * 09d30fd feat: use selective logging Soonho Kong, 5 weeks ago
| | * e915cb9 refactor: move src/dsolvers/util => src/util Soonho Kong, 5 weeks ago
| | * 30b3f96 fix(dsolver/icp_solver): small bug lingering from old rebase where not ignoring forall constraints. Also updated planning examples Daniel Bryce, 5 weeks ago
| | * 473dd28 fix(dsolver/icp_solver.cpp): use LOG(INFO) to call pprint_vars/lits Soonho Kong, 5 weeks ago
| | * ac630b6 fix(dsolver/util/scoped_env.cpp): operator<< Soonho Kong, 5 weeks ago
| | * 079b28c feat: remove dsolver/util/logger Soonho Kong, 5 weeks ago
| | * 4d8a284 feat: use glog instead of util/logger.h Soonho Kong, 5 weeks ago
| | * 0caa50c fix(dsolvers/icp_solver): move LOG into the conditional Soonho Kong, 5 weeks ago
| | * cb10fe3 feat(dsolver/icp_solver): use glog Soonho Kong, 5 weeks ago
```

<http://www.jayway.com/2012/11/27/configure-git-like-a-boss/>

Basic(init/add/commit)

<http://try.github.io>

Init

From Scratch

```
# New project  
$ git init newproject  
$ cd newproject  
# ...start coding
```

Init

From Existing Dir

```
# Legacy project tree
$ cd existingproject
$ git init

# Add all the code
$ git add .
$ git commit -m "Initial import"
```


Commit Your First Commit

```
git add  
git status  
git commit -m "Helpful message"
```

Network

Network



Network




local



origin

Network

PUBLIC  soonhokong / dreal
forked from dreal/dreal

 Unwatch 1

 Star 0

 Fork 3

Working copy of the dReal repo. I use this repo for performing experiments and storing work-in-progress
<http://dreal.cs.cmu.edu> — Edit

 1,379 commits

 4 branches

 0 releases


 5 contributors



branch: master


dreal / +


This branch is 0 commits ahead and 0 commits behind master









 Pull Request

 Compare

checkpoint

 soonhokong authored on May 12

latest commit bee5818766 

 benchmarks	benchmark(hybrid_systems/bouncing_ball): fix bouncing_ball.drh to be ...	on Apr 30
 bin	fix(bin/dReach): have better output messages	on Apr 23
 doc	doc(ode-visualization): add documentation for visualization	on Apr 15
 src	checkpoint	on May 12
 tools	fix(code_bmc.ml): fix mode index in variable name	on May 12
 .gitignore	feat(non_uniform_delta): Barn Door experiments for memo	on Mar 28
 .mailmap	chore(.mailmap): update & clean up	on Apr 23
 .travis.osx.yml	chore(travis): only run testcases labeled with "normal" on travis	on May 5

 Code

 Pull Requests 0

 Wiki


 Pulse


 Graphs


 Network


 Settings

SSH clone URL

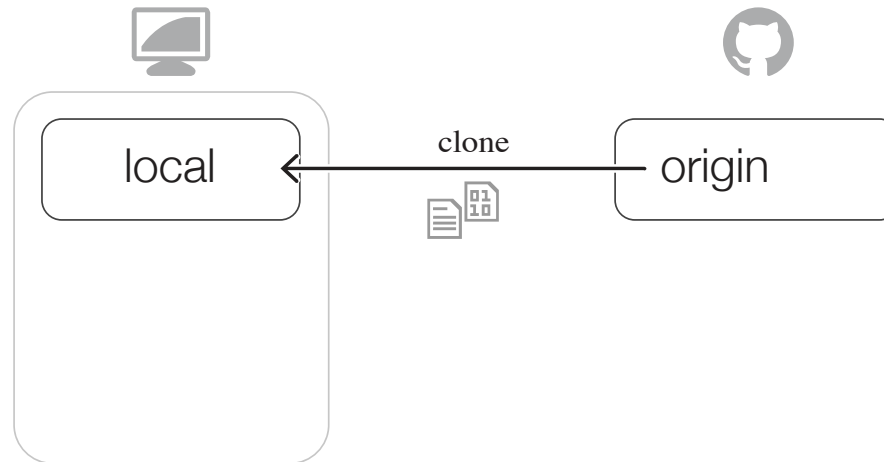


You can clone with [HTTPS](#), [SSH](#), or [Subversion](#). 

 Clone in Desktop

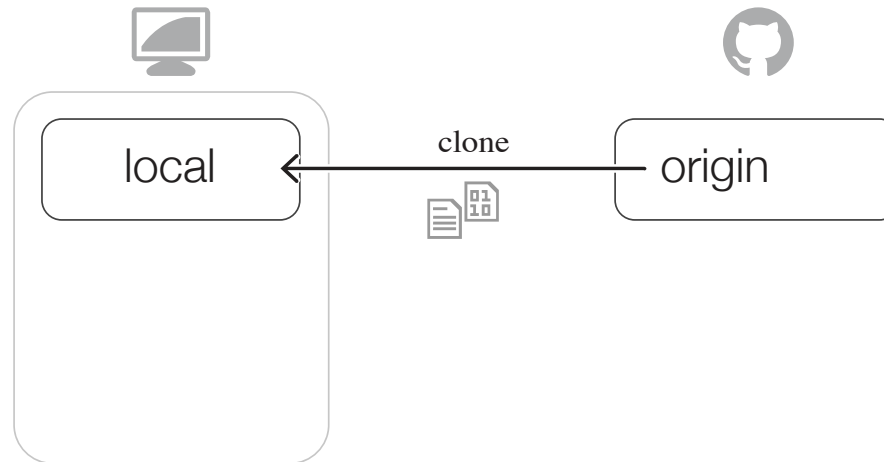
 Download ZIP

Network



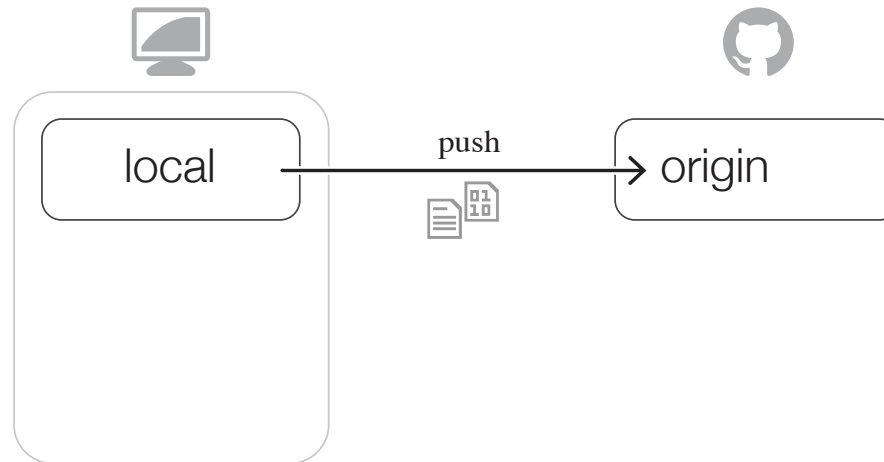
```
$ git clone git@github.com:soonhokong/dreal
Cloning into 'dreal'...
remote: Reusing existing pack: 79707, done.
remote: Counting objects: 127, done.
remote: Compressing objects: 100% (98/98), done.
remote: Total 79834 (delta 37), reused 74 (delta 29)
Receiving objects: 100% (79834/79834), 49.25 MiB | 2.34 MiB/s, done.
Resolving deltas: 100% (39555/39555), done.
Checking connectivity... done.
```

Network



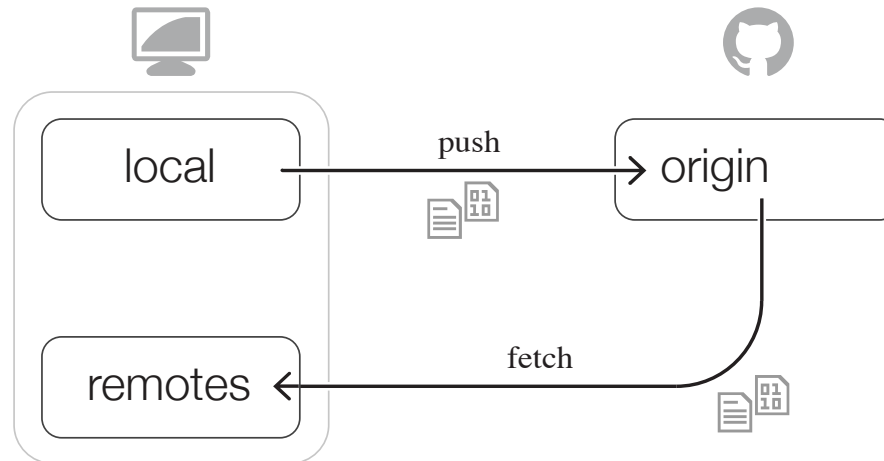
```
$ git remote -v  
origin  git@github.com:soonhokong/dreal (fetch)  
origin  git@github.com:soonhokong/dreal (push)
```

Network



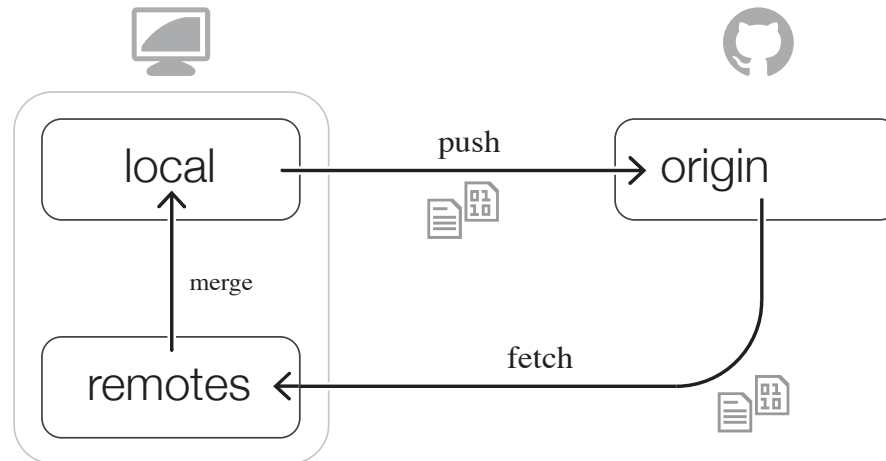
```
$ git push
Counting objects: 9, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 286 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
To git@github.com:soonhokong/dreal
    bee5818..40155b4  master -> master
```


Network



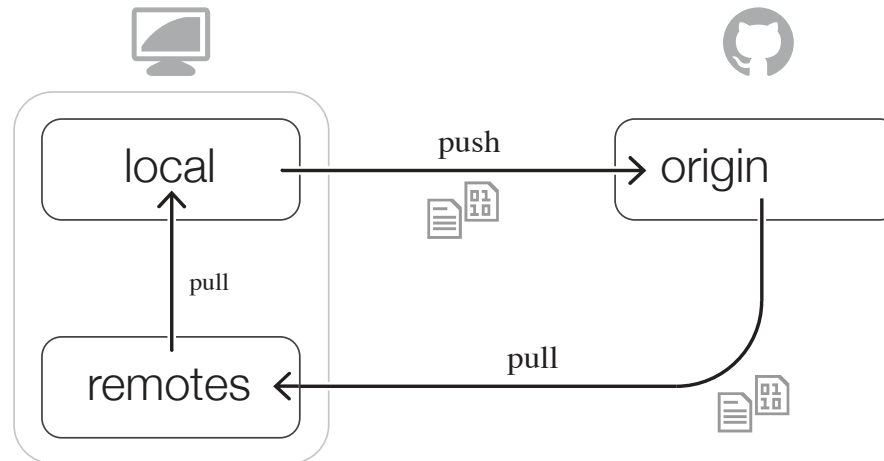
```
$ git fetch origin
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From github.com:soonhokong/dreal
    bee5818..40155b4  master    -> origin/master
```

Network



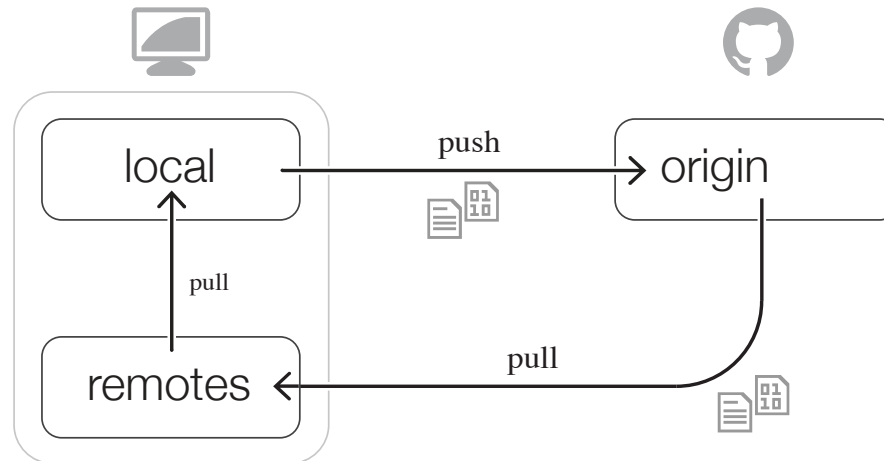
```
$ git merge origin/master
Updating bee5818..40155b4
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
```

Network



```
$ git pull origin master
From github.com:soonhokong/dreal
 * branch          master      -> FETCH_HEAD
Updating bee5818..40155b4
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
```

Network



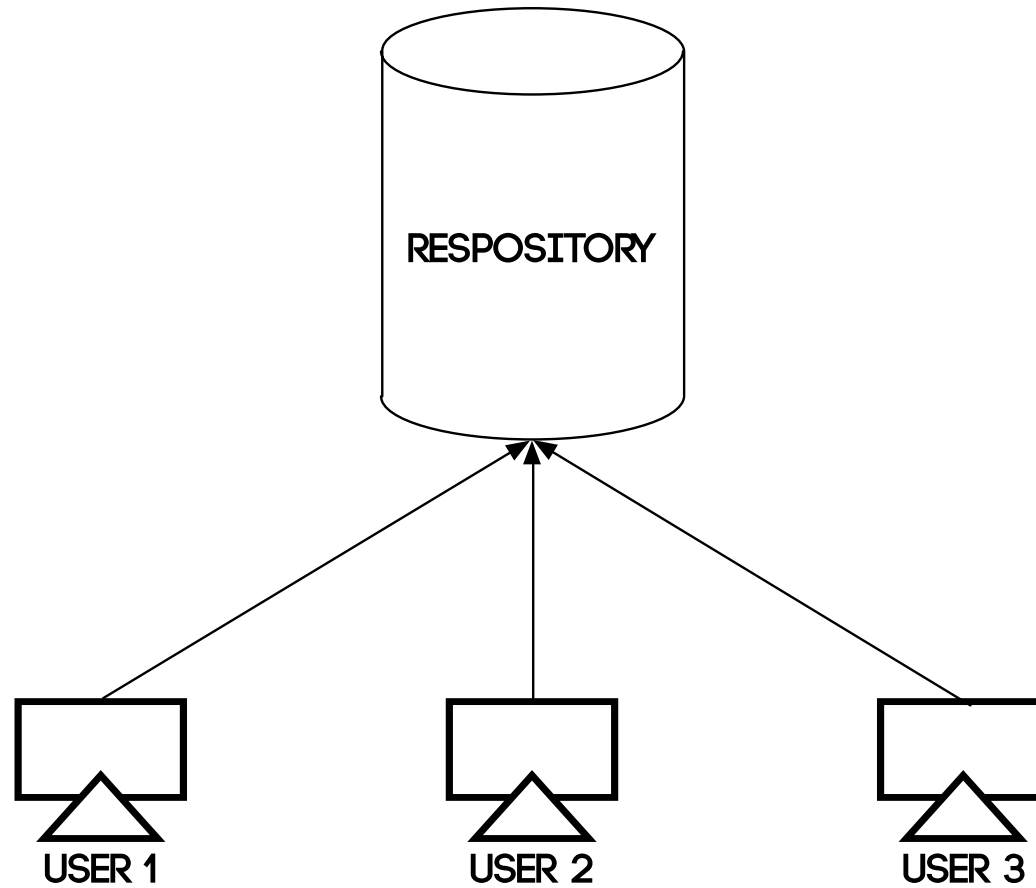
Don't use git **pull**!

Use git **fetch** and then **git rebase** (or git **merge**)

<http://longair.net/blog/2009/04/16/git-fetch-and-merge>

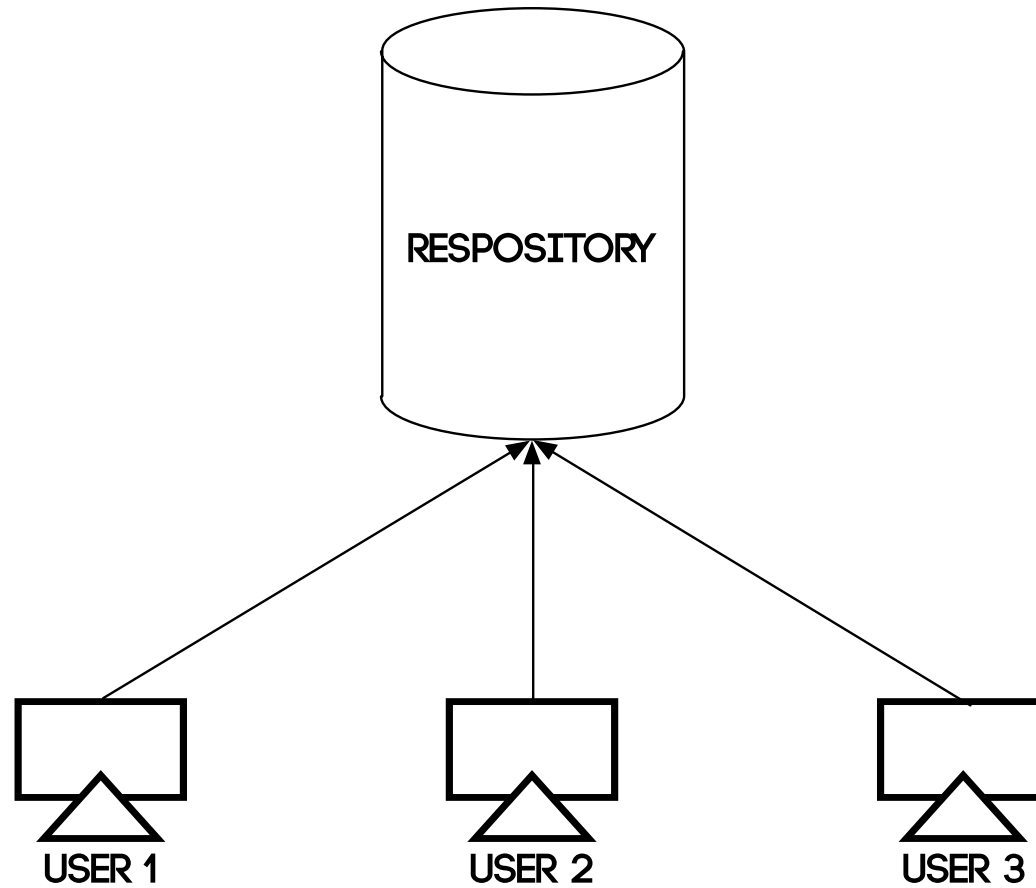
Centralized Workflow

Centralized Workflow



The way people are using **SVN**

Centralized Workflow

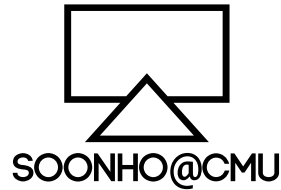
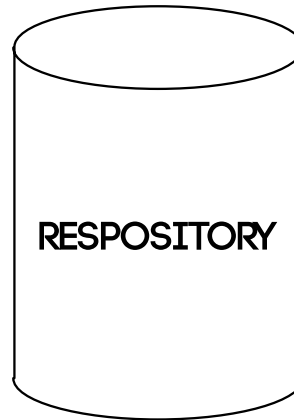


You can use **git** in this way, too

Centralized Workflow

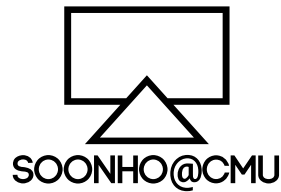
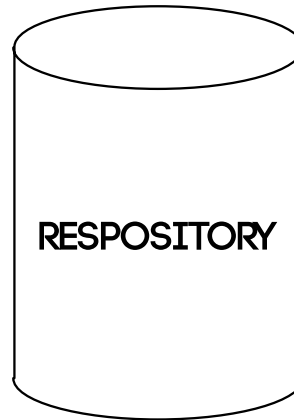
What's a **problem**?

Problem of Centralized Workflow



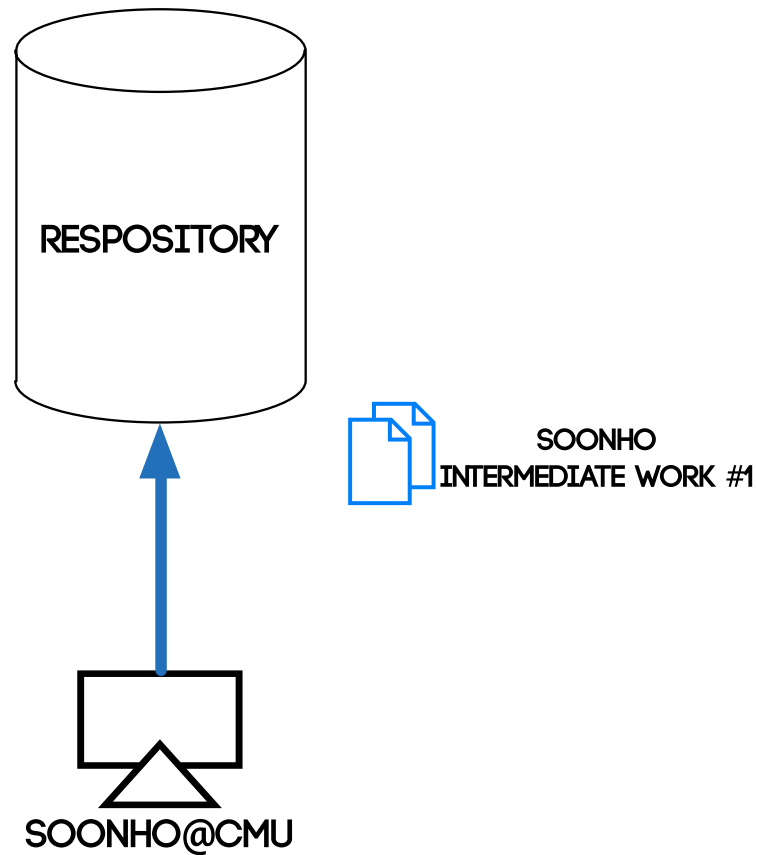
I'm working at CMU...

Problem of Centralized Workflow



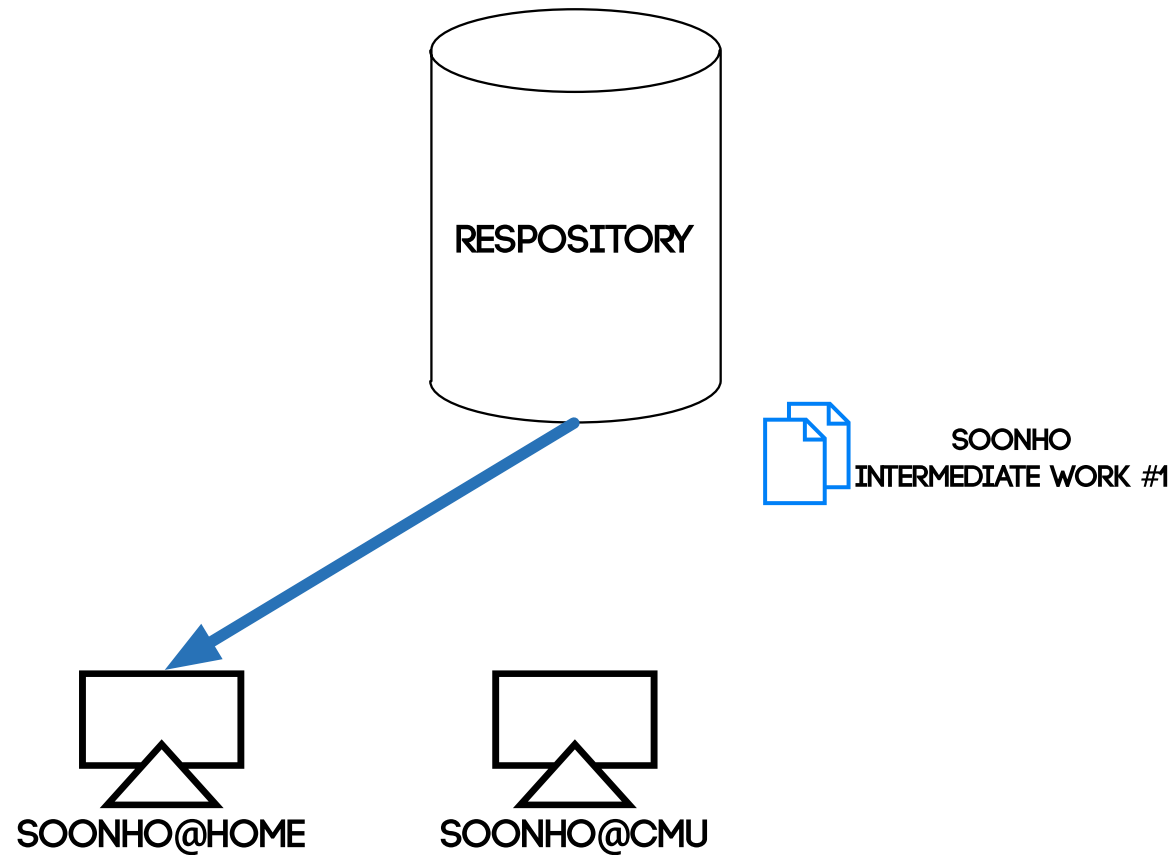
I want to go home and **resume** the work.

Problem of Centralized Workflow



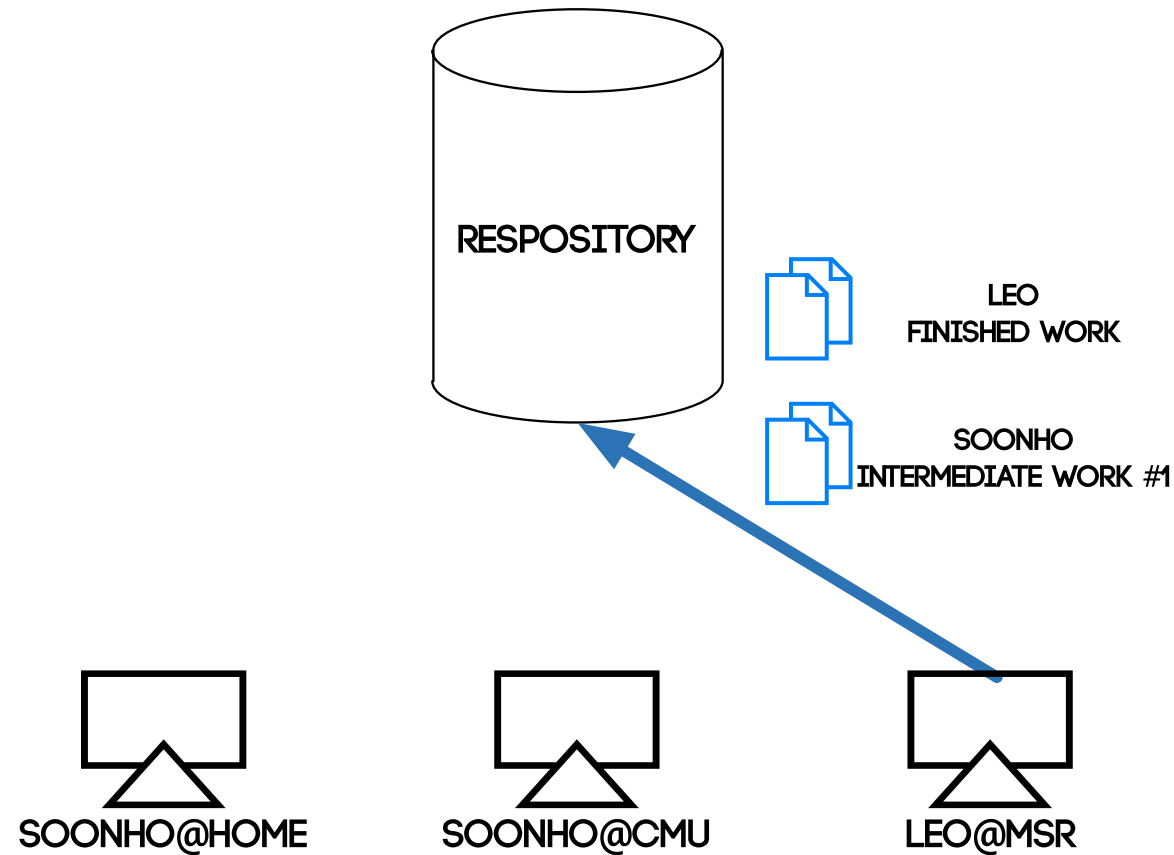
So I **push** what I've done so far to the repo

Problem of Centralized Workflow



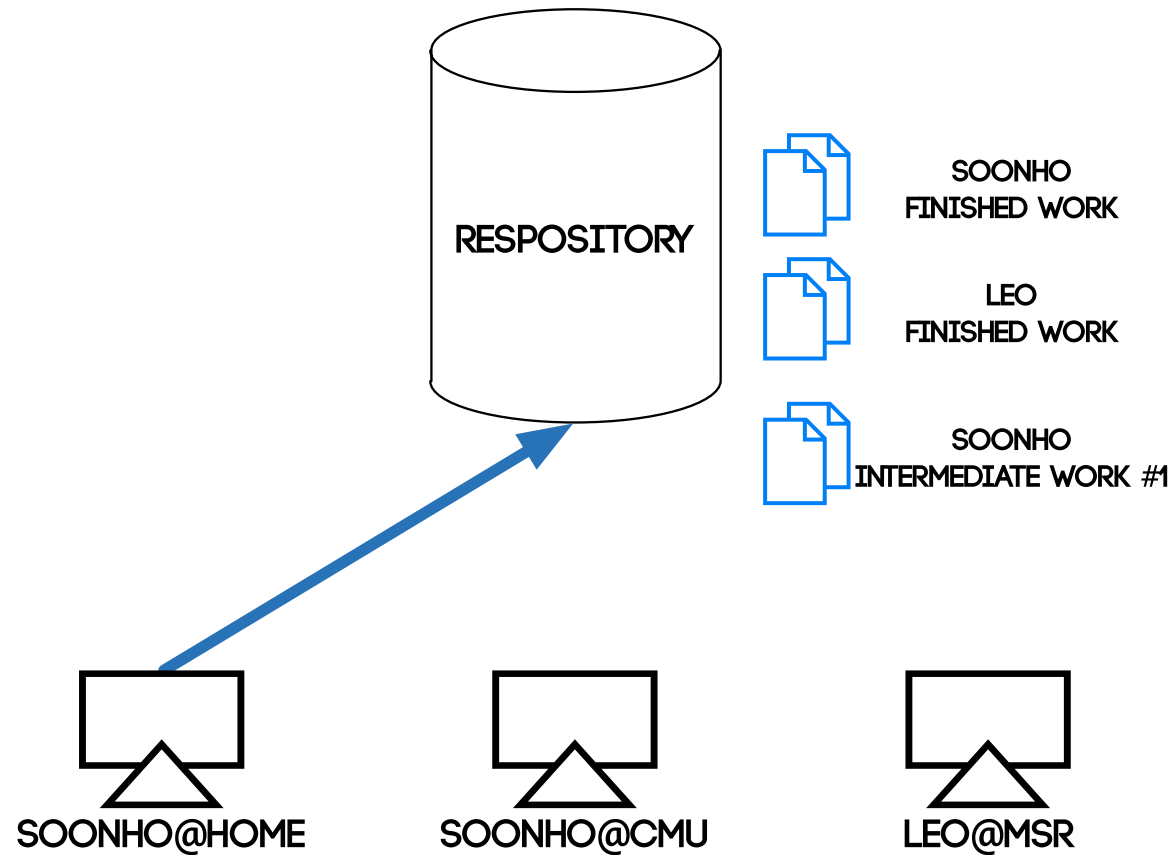
@Home, I **pull** what I pushed and **resume** the work.

Problem of Centralized Workflow



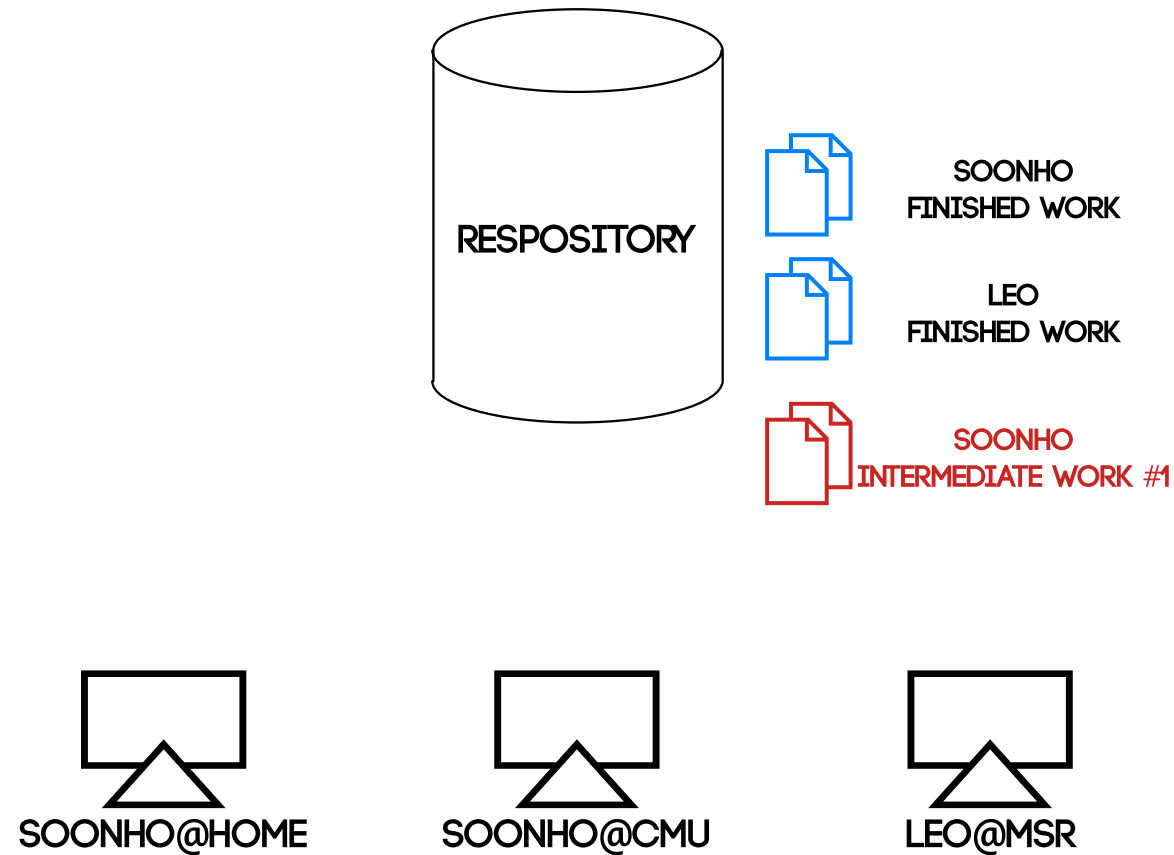
In the meantime, Leo **pushed** a commit to the repo.

Problem of Centralized Workflow



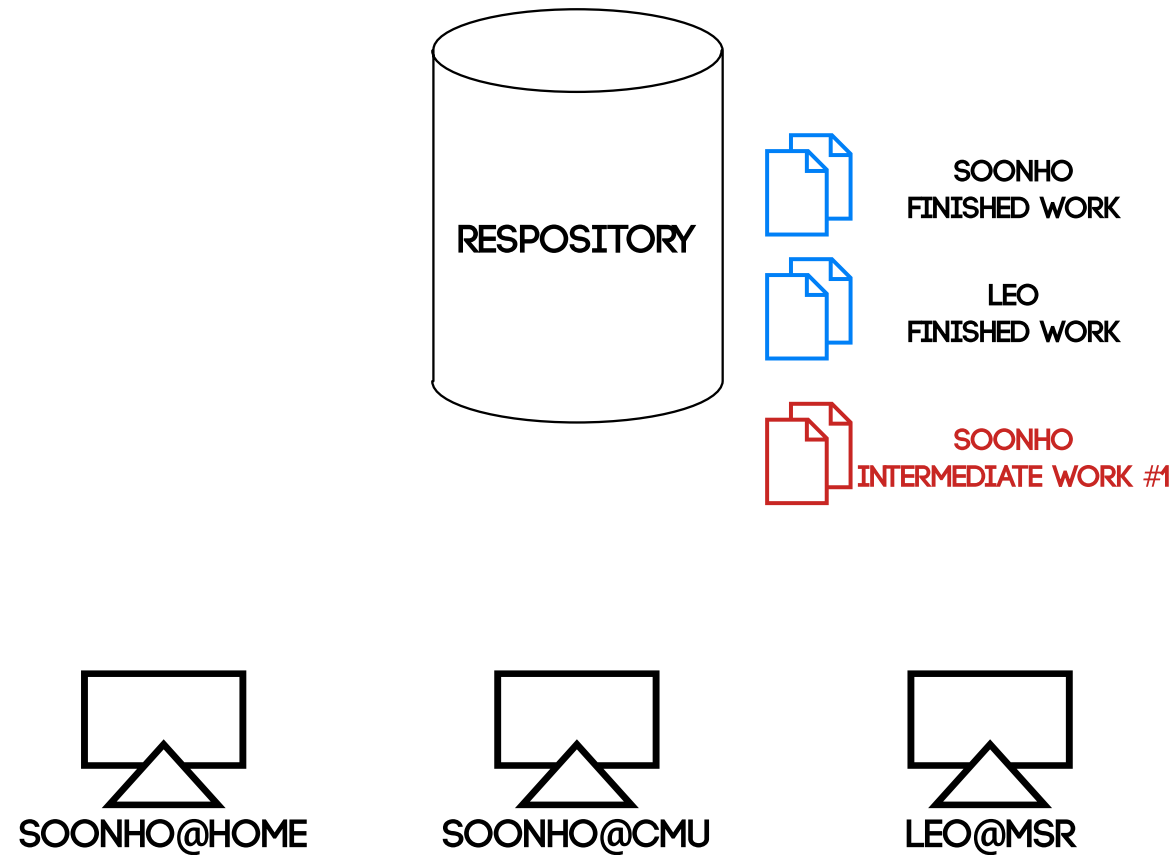
I finished my part at home, **push** to the repo.

Problem of Centralized Workflow



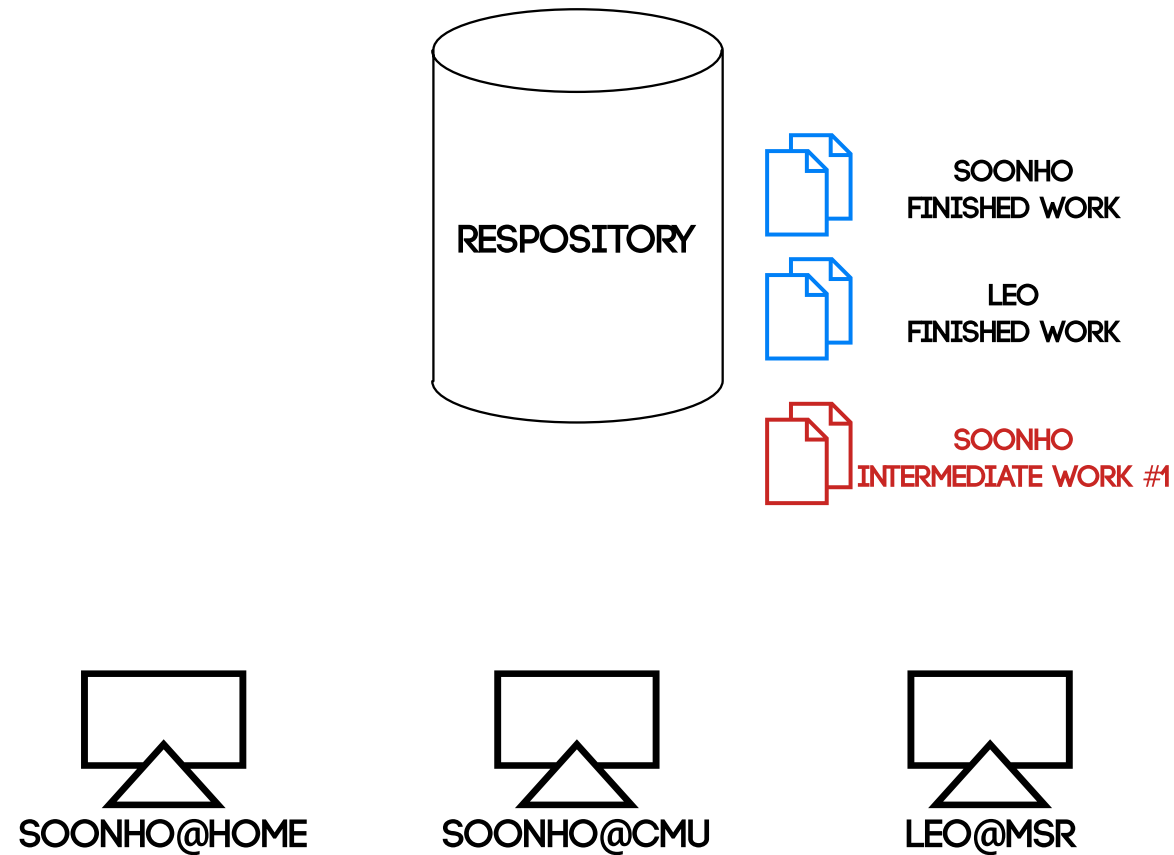
Problem: Repo will be filled with **intermediate** commits

Problem of Centralized Workflow



Problem: Single Sync Point = ↑ **Merge Conflicts**

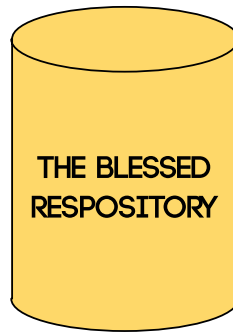
Problem of Centralized Workflow



Not Scalable (≤ 5 members)



Distributed workflow

Distributed workflow









There is the **Blessed** repository.


Distributed workflow



PUBLIC  **leanprover / lean**  6  20  8












Lean Theorem Prover — Edit


 **2,461** commits  **1** branch  **0** releases  **2** contributors

 branch: **master** **lean** / 

feat(kernel/inductive): generate computational rules RHS for inductiv... 

 **leodemoura** authored 31 minutes ago latest commit 2aacb769dd 

 doc	doc(demo): remove Lean 0.1 demo files	Saturday at 1:43pm
 images	chore(*): add logo	on Apr 29
 script	fix(script/demangle_cpptype): process line-by-line, instead of waitin...	on Sep 29, 2013
 src	feat(kernel/inductive): generate computational rules RHS for inductiv...	31 minutes ago
 tests	feat(kernel/inductive): generate computational rules RHS for inductiv...	31 minutes ago
 .gitignore	chore(.gitignore): ignore .md.lean files	on Dec 27, 2013
 .travis.osx.yml	chore(travis): use lean-build@googlegroups	on May 2
 .travis.windows.yml	chore(travis): use lean-build@googlegroups	on May 2
 .travis.yml	chore(travis): fix bitbucket push	on May 4
 LICENSE	Add LICENSE file	on Jul 15, 2013
 README.md	doc(README.md): use self-hosted icon for cdas	on May 5

 **Code**

 **Issues** 1

 **Pull Requests** 0

 **Wiki**


 **Pulse**


 **Graphs**


 **Network**

 **Settings**

SSH clone URL

git@github.com:lec 

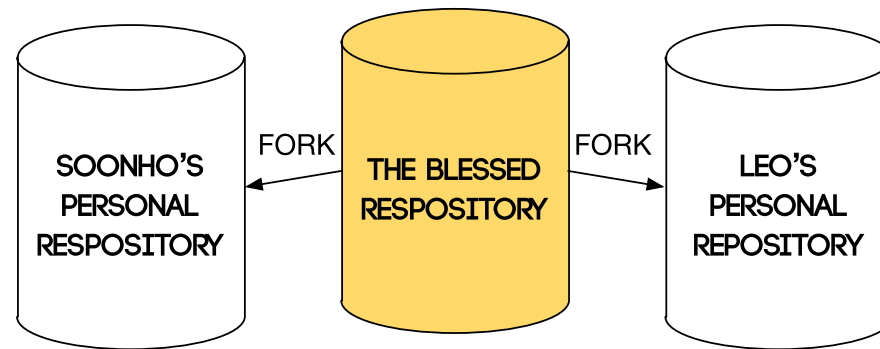
You can clone with **HTTPS**, **SSH**,
or **Subversion**. 

 **Clone in Desktop**

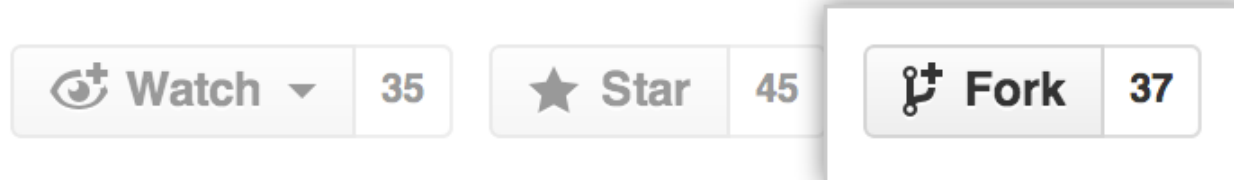
 **Download ZIP**

Distributed workflow

Distributed workflow



Distributed workflow



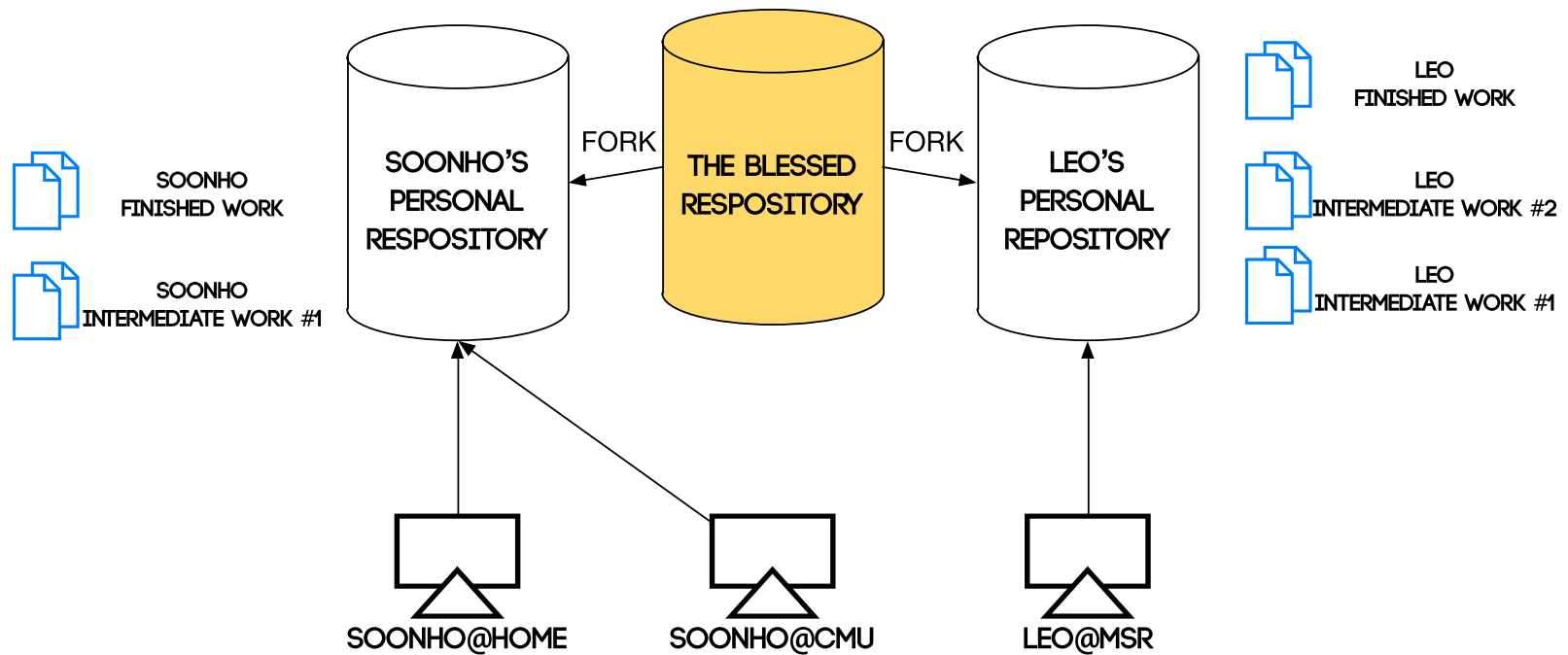
Clone **personal** repo (origin):

```
$ git clone git@github.com:soonhokong/dreal.git
$ git remote -v
origin  git@github.com:soonhokong/dreal.git (fetch)
origin  git@github.com:soonhokong/dreal.git (push)
```

Add **blessed** repo:

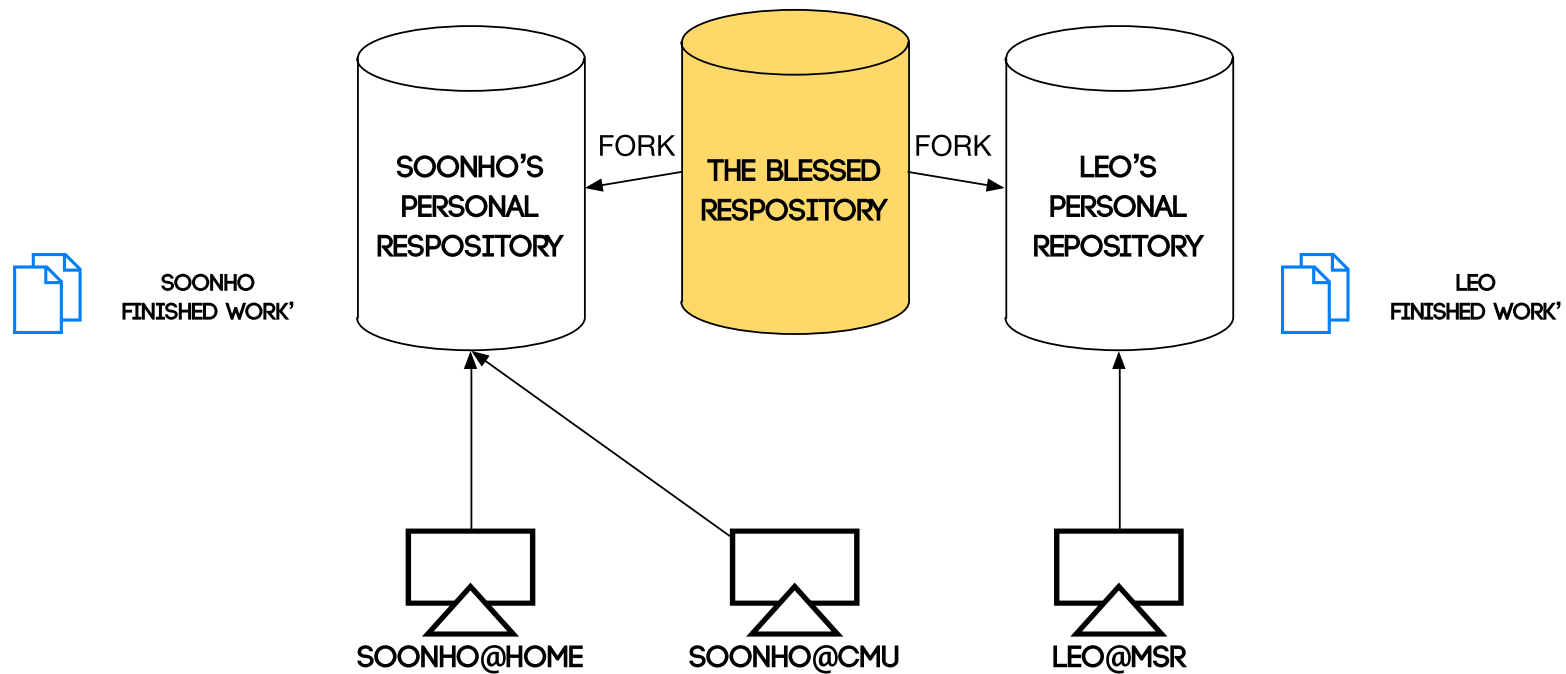
```
$ git remote add blessed git@github.com:dreal/dreal
$ git remote -v
blessed git@github.com:dreal/dreal.git (fetch)
blessed git@github.com:dreal/dreal.git (push)
origin  git@github.com:soonhokong/dreal.git (fetch)
origin  git@github.com:soonhokong/dreal.git (push)
```

Distributed workflow



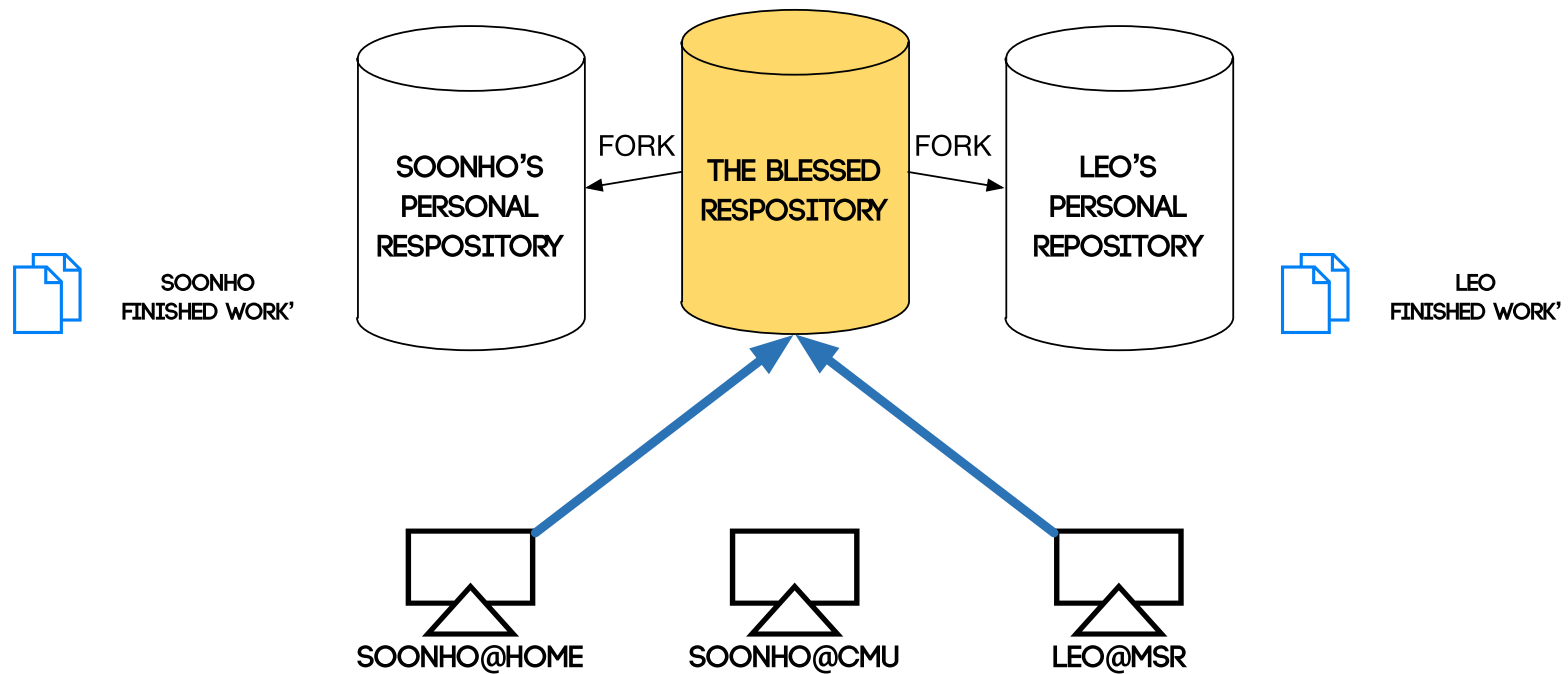
Most of time, we push to **personal** repos.

Distributed workflow



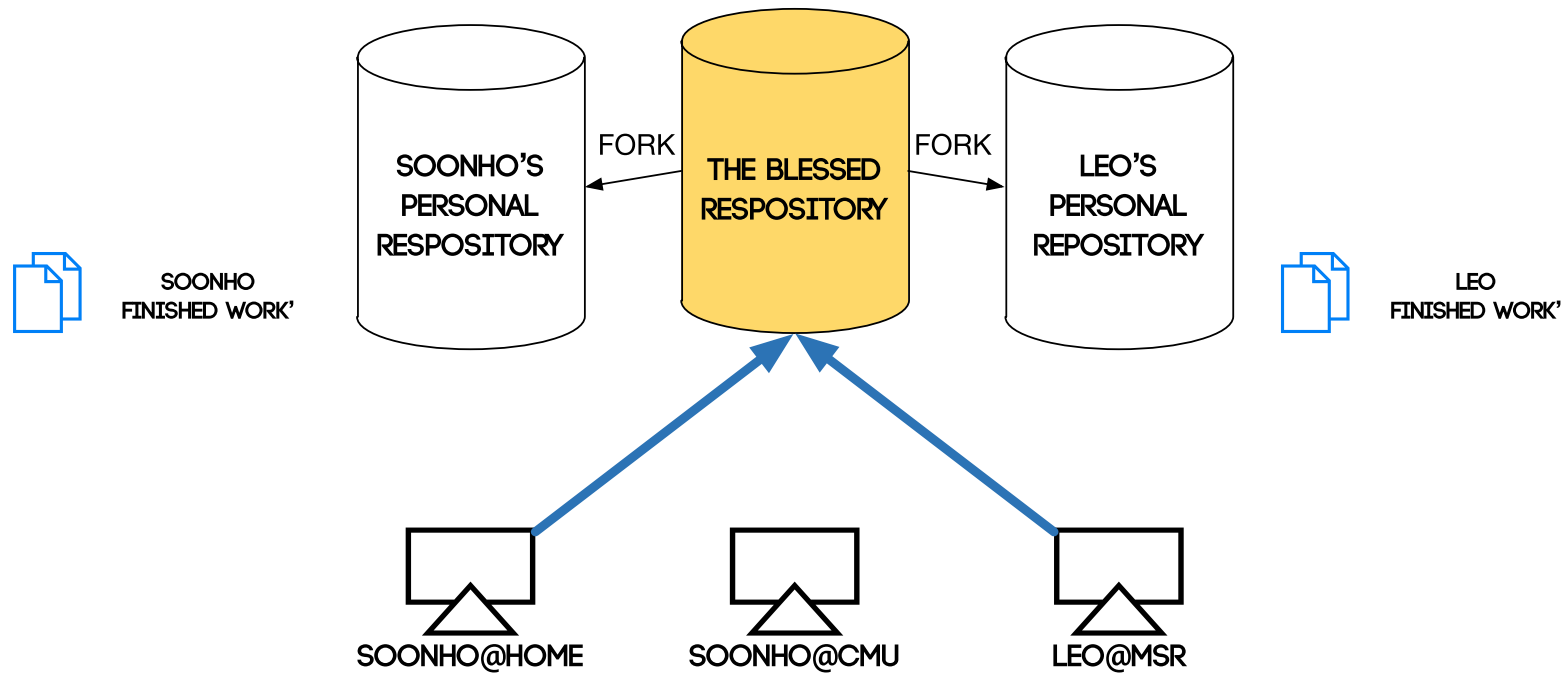
You can do **destructive** update (**forced** push) on your repo.

Distributed workflow



When the work is solid, we push to the **blessed**.

Distributed workflow



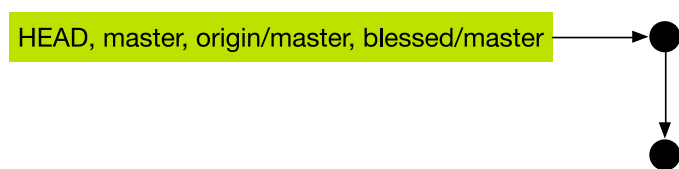
or make a **pull-request**.

<https://help.github.com/articles/using-pull-requests>

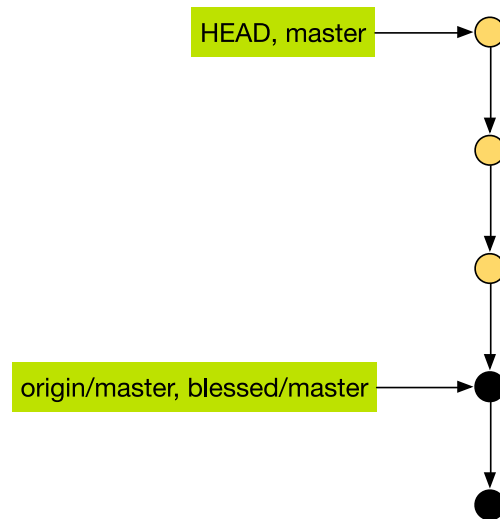
Rebase

Many **meaningful** git operations
can be expressed
in terms of the **rebase** command.

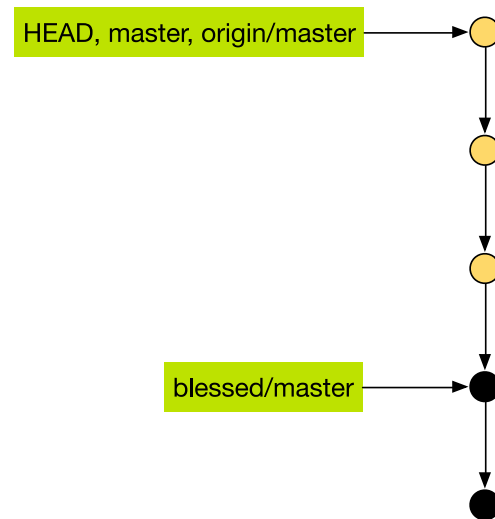
Merge vs. Rebase



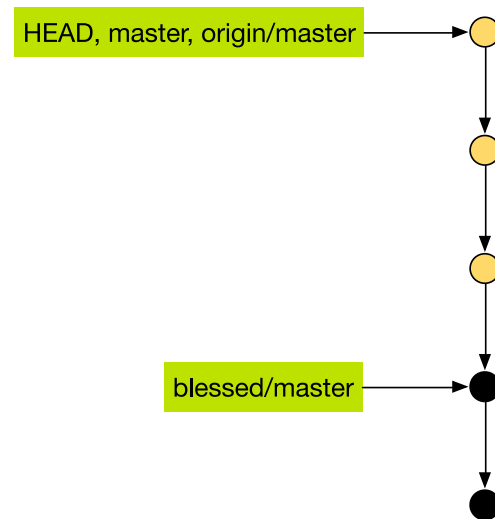
Merge vs. Rebase



Merge vs. Rebase



Merge vs. Rebase

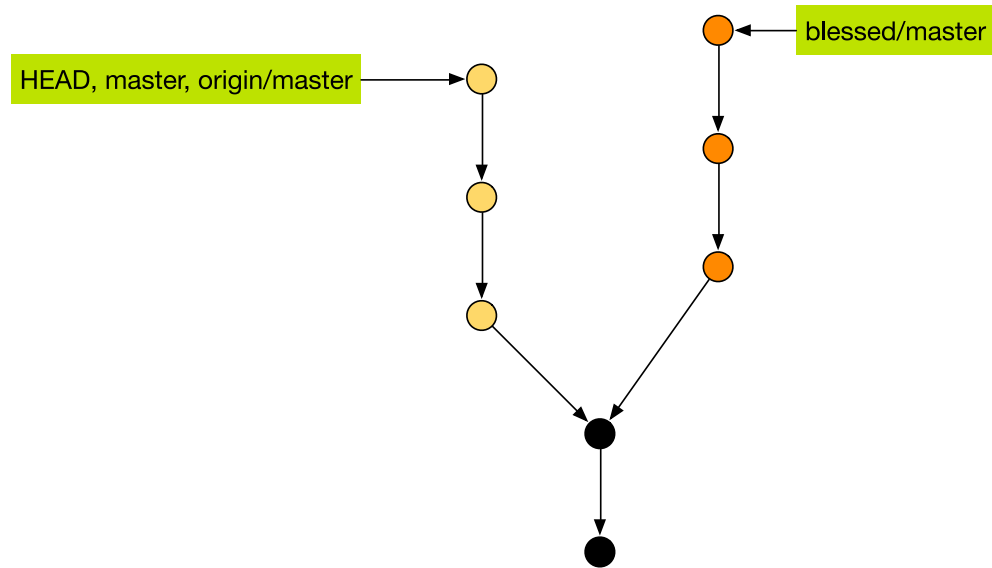


Merge vs. Rebase

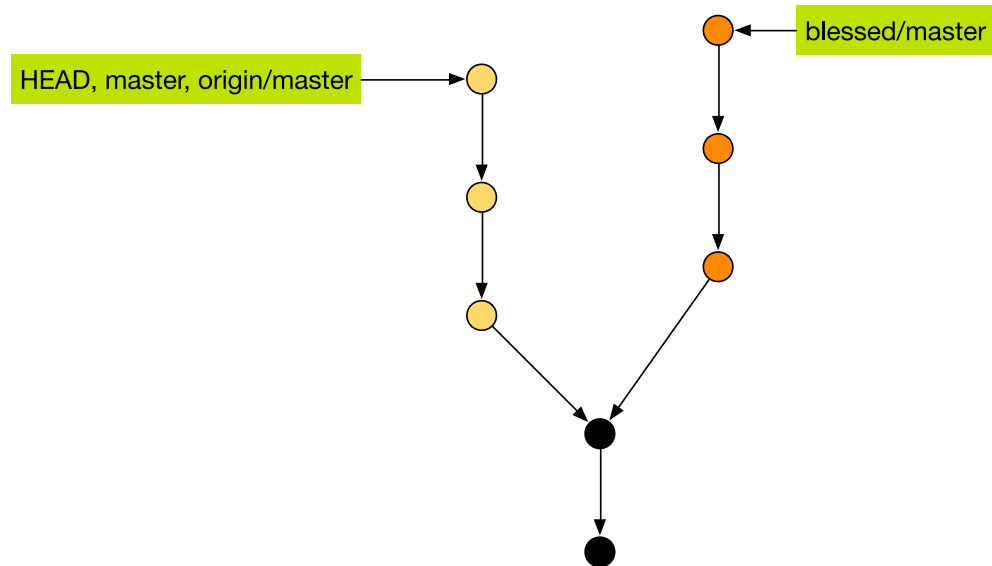
```
$ git push blessed master
Counting objects: 28, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 955 bytes | 0 bytes/s, done.
Total 9 (delta 6), reused 0 (delta 0)
To git@github.com:dreal/dreal.git
 ! [rejected]          master -> master (non-fast-forward)
error: failed to push some refs to 'git@github.com:dreal/dreal.git'
hint: Updates were rejected because a pushed branch tip is behind its re
hint: counterpart. Check out this branch and integrate the remote change
hint: (e.g. 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details
```

non-fast-forward??

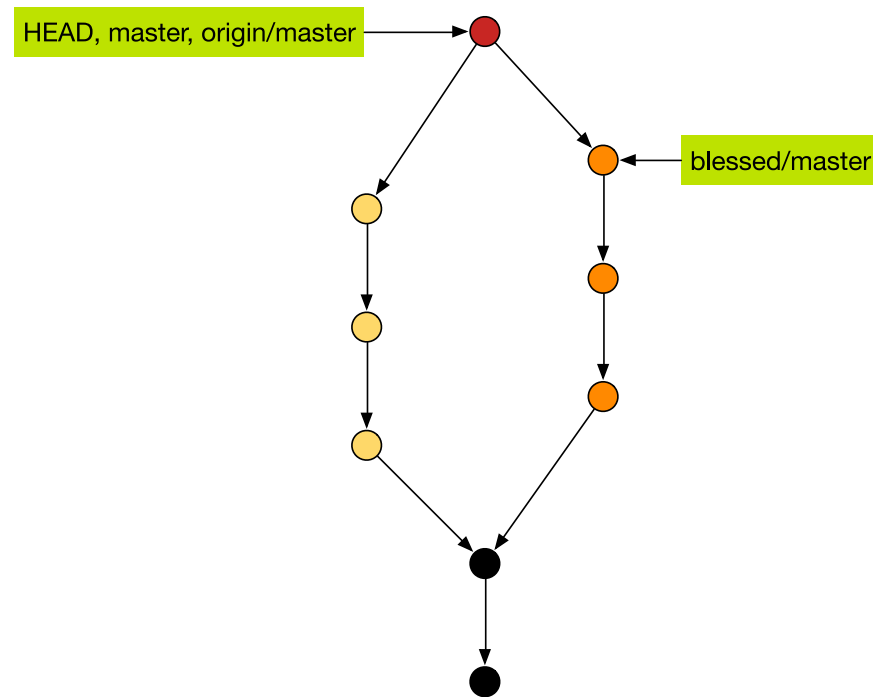
Merge vs. Rebase



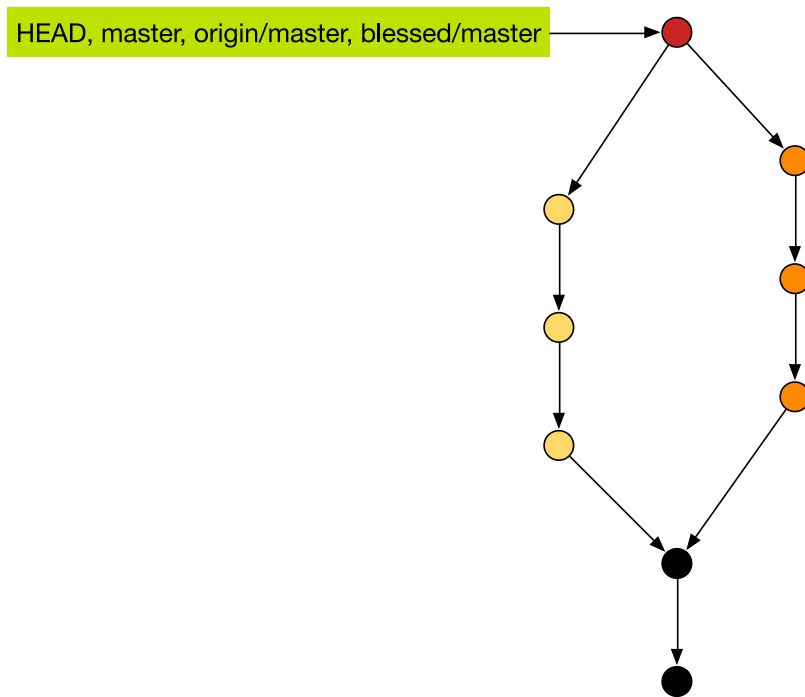
Merge vs. Rebase



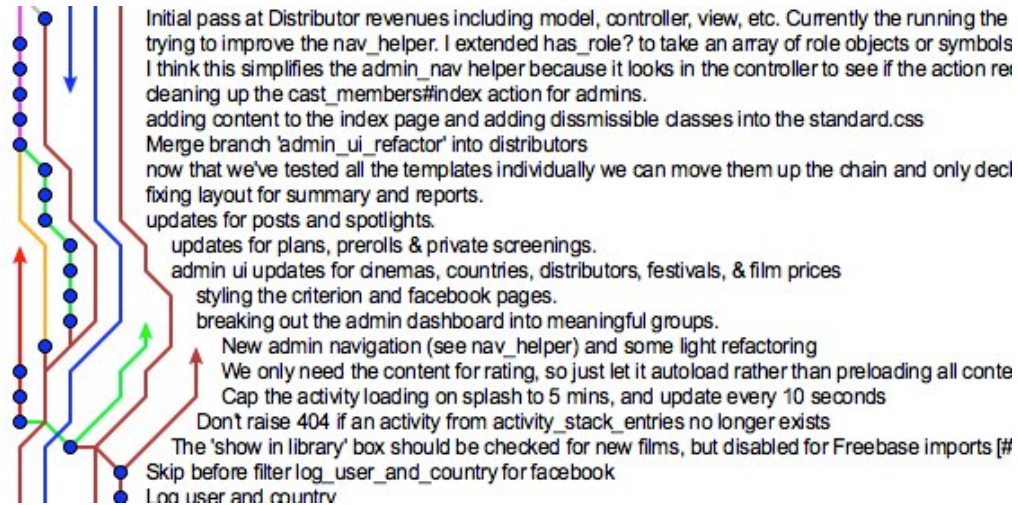
Merge vs. Rebase



Merge vs. Rebase

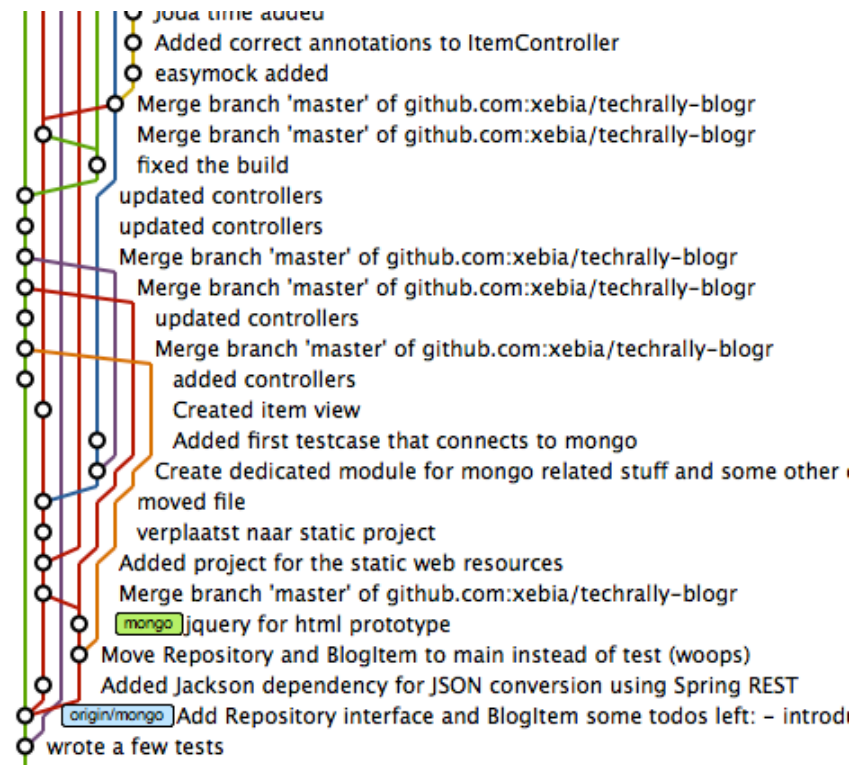


Merge vs. Rebase



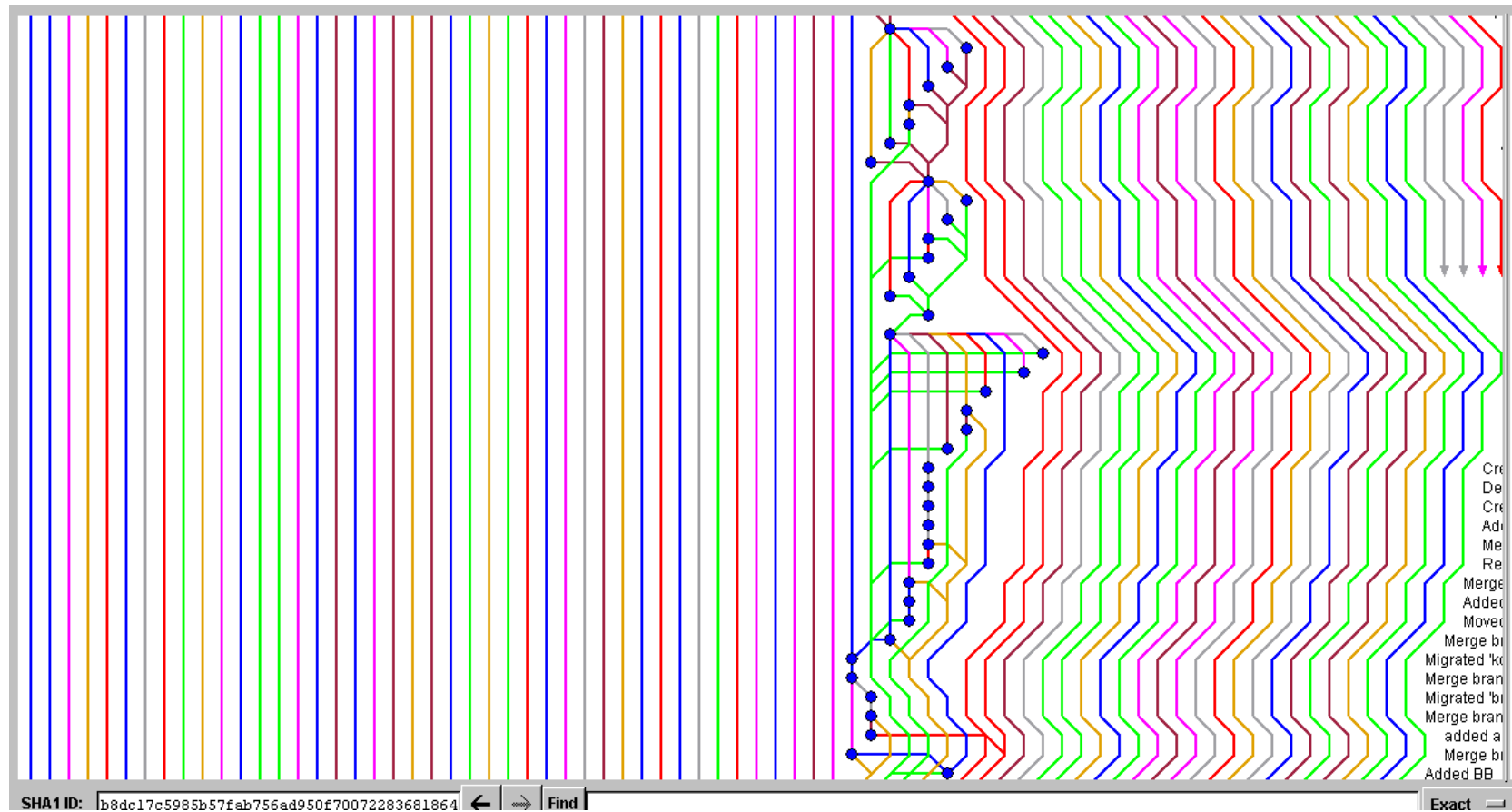
OK?

Merge vs. Rebase



OK?

Merge vs. Rebase



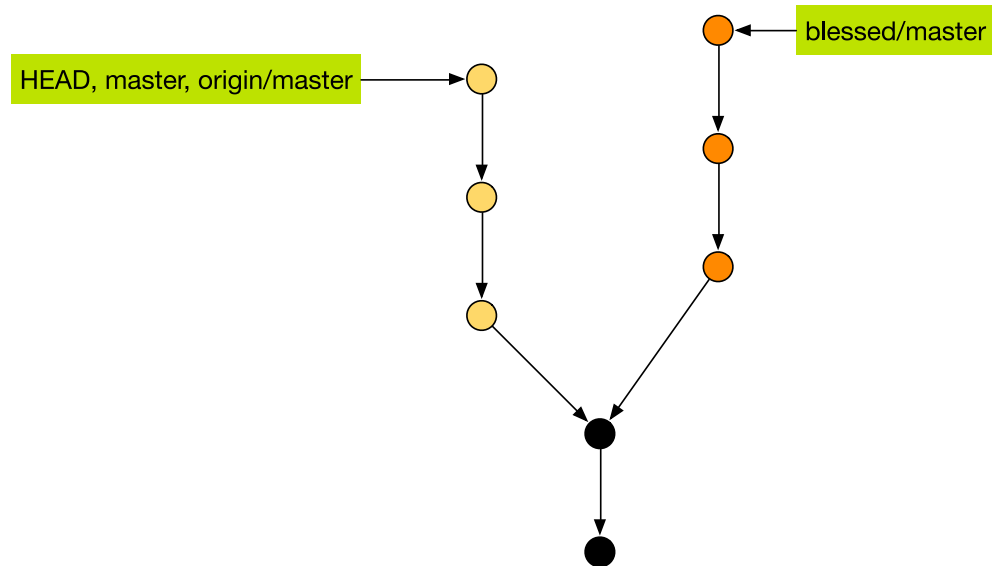
Merge Hell

Version Control Pitfalls

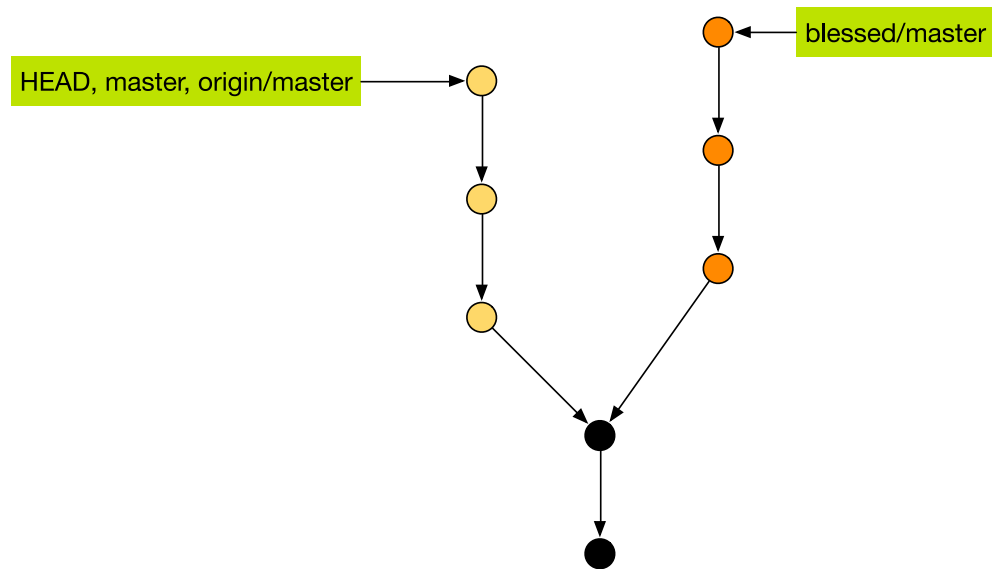
1. History should be written in **chronological** order.
2. Commits are **immutable** and **immovable** objects.

git rebase will free you!!

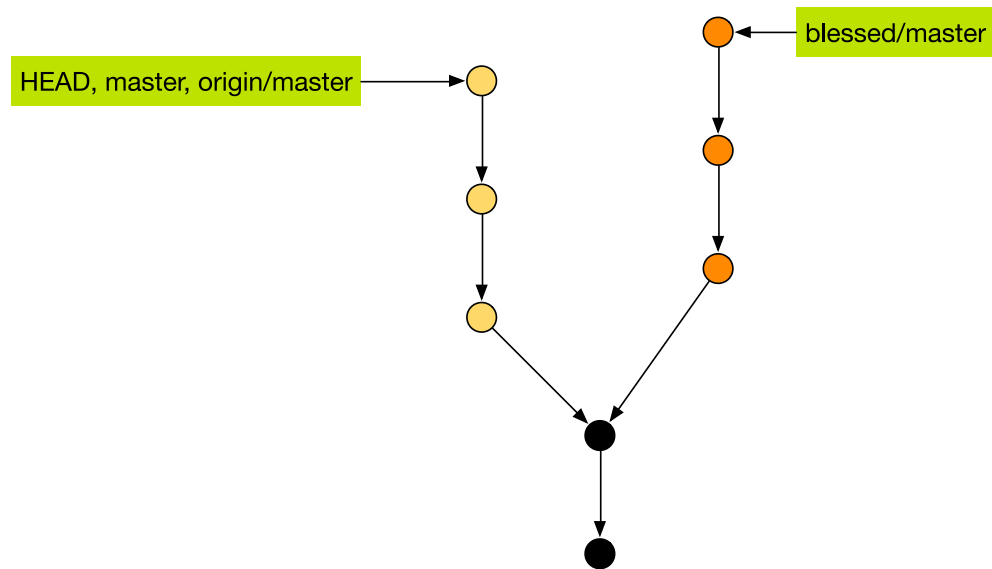
Merge vs. Rebase



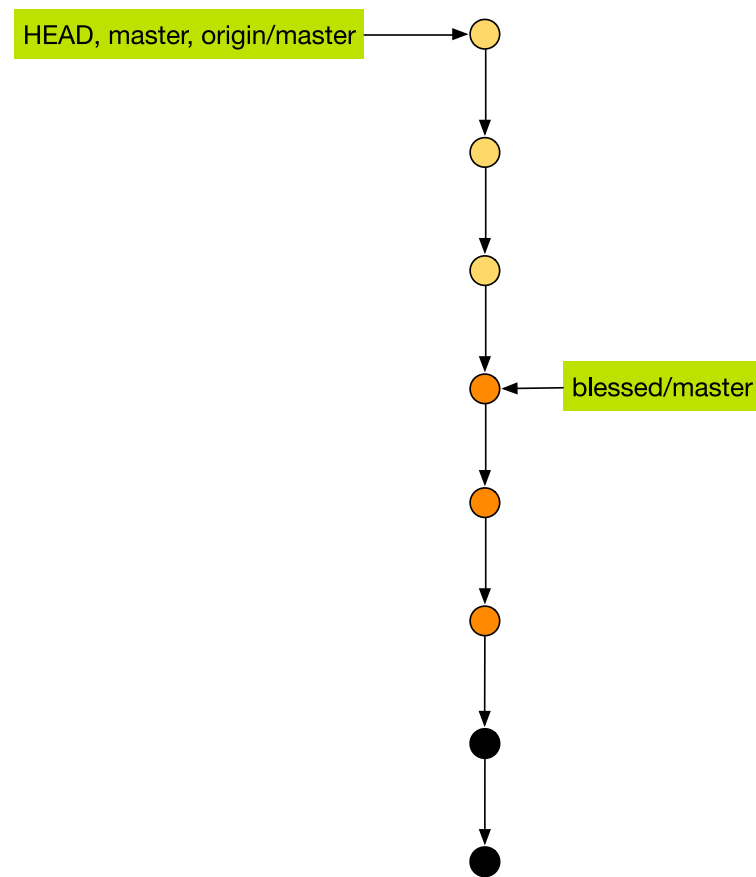
Merge vs. **Rebase**



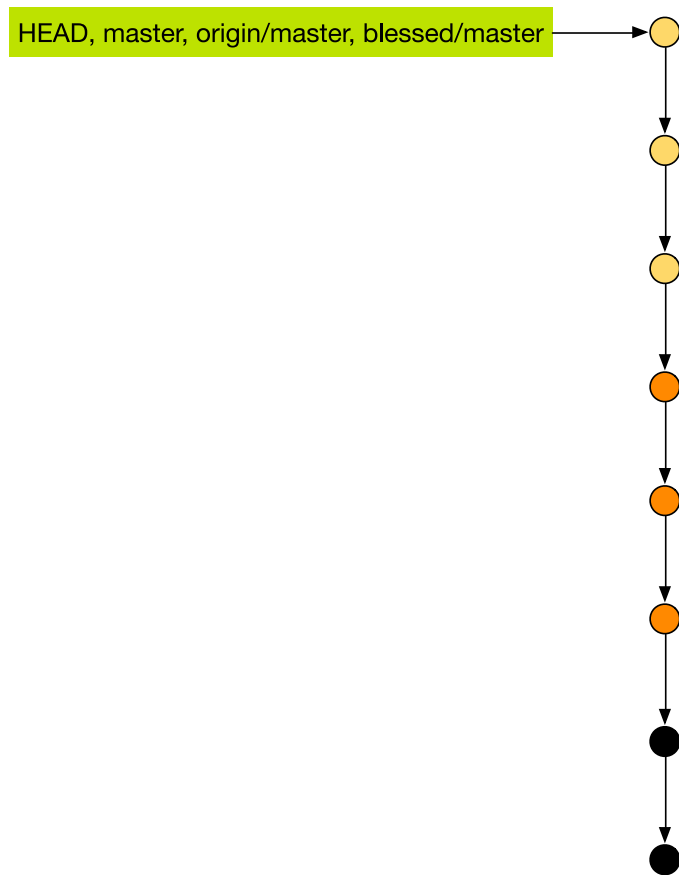
Merge vs. **Rebase**



Merge vs. **Rebase**



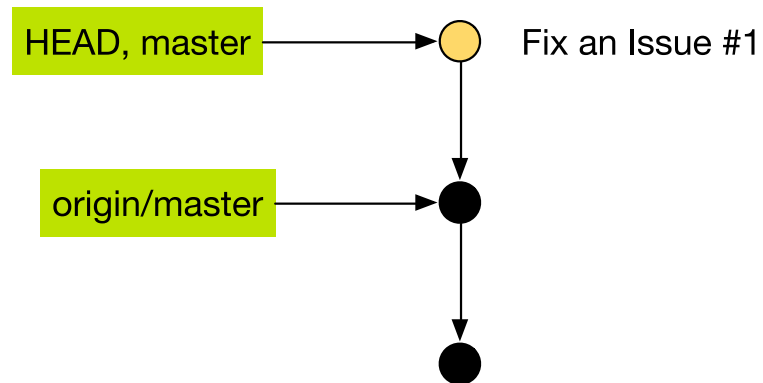
Merge vs. **Rebase**



Interactive Rebase

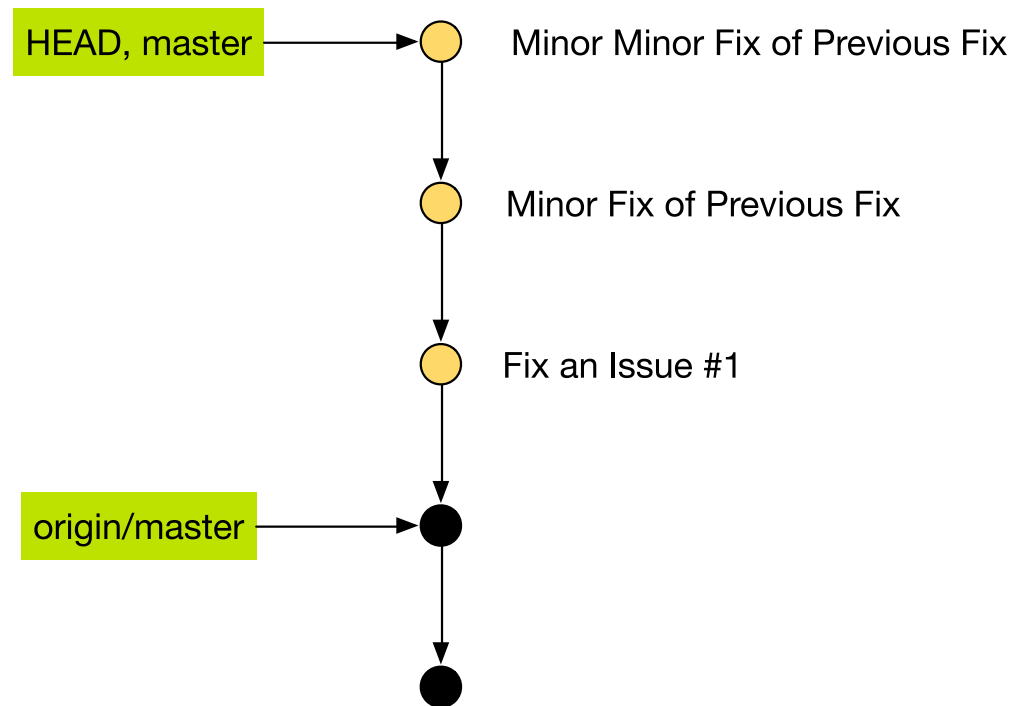


Interactive Rebase

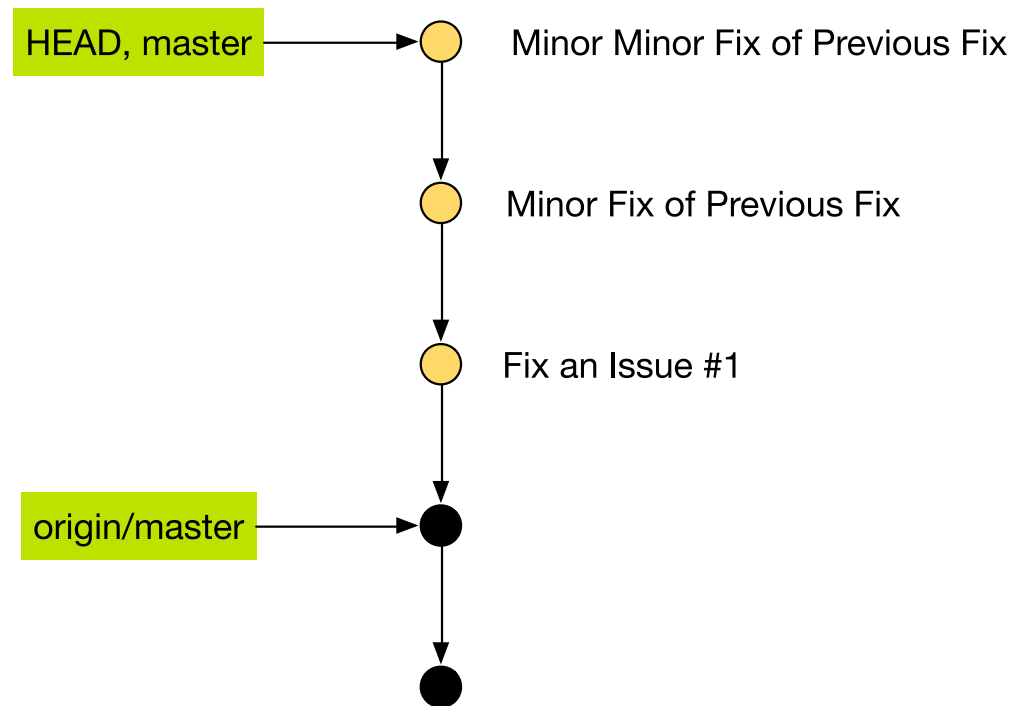


Interactive Rebase

Interactive Rebase

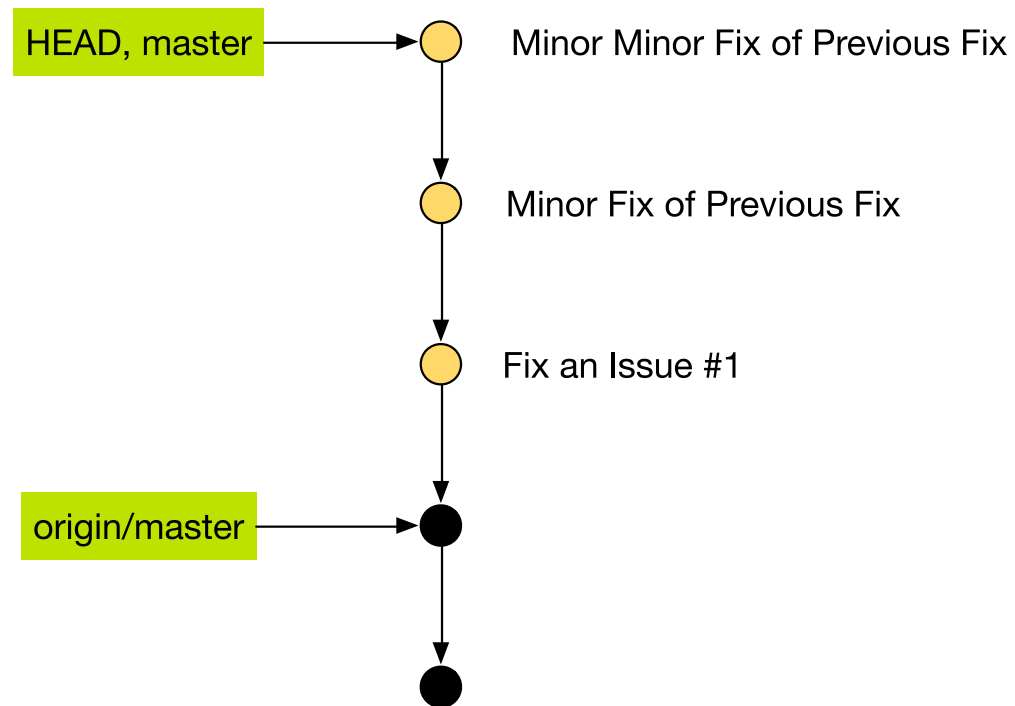


Interactive Rebase



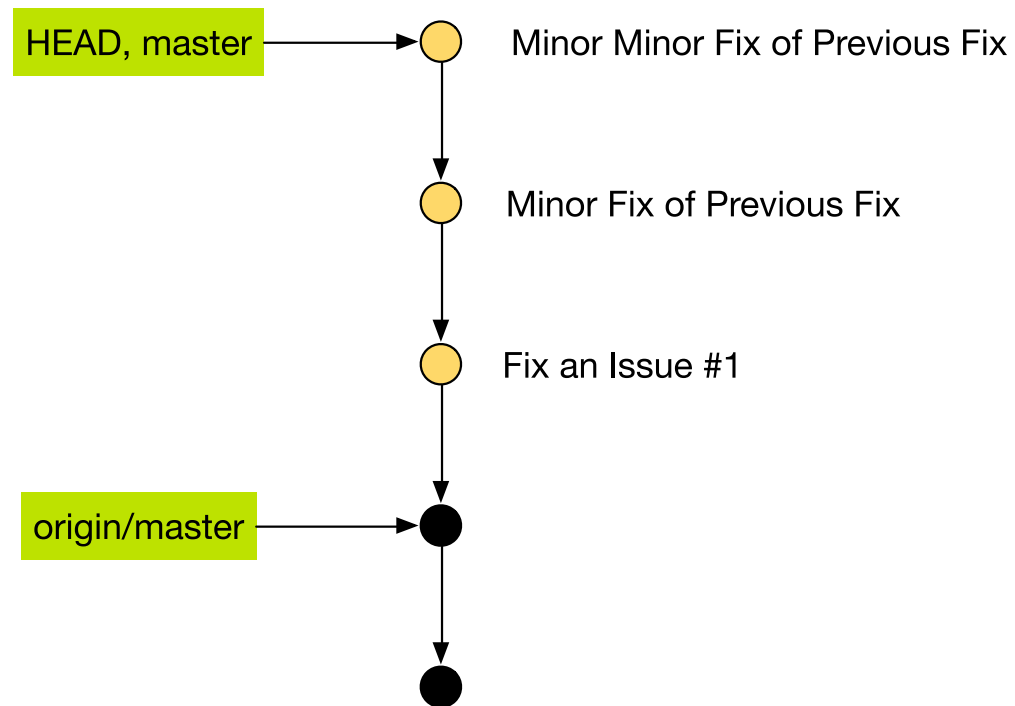
I want to **combine** the three commits into one.

Interactive Rebase



git rebase origin/master doesn't do anything.

Interactive Rebase



Run **git rebase -i origin/master**

Interactive Rebase

Run **git rebase -i origin/master**

```
pick 40155b4 Fix an Issue #1
pick 42e5db9 Minor Fix of Previous Fix
pick a31c7cd Minor Minor Fix or Previous Fix

# Rebase ec2ae2b..cf0977c onto ec2ae2b
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
#
# These lines can be re-ordered; they are executed from top to bottom
```

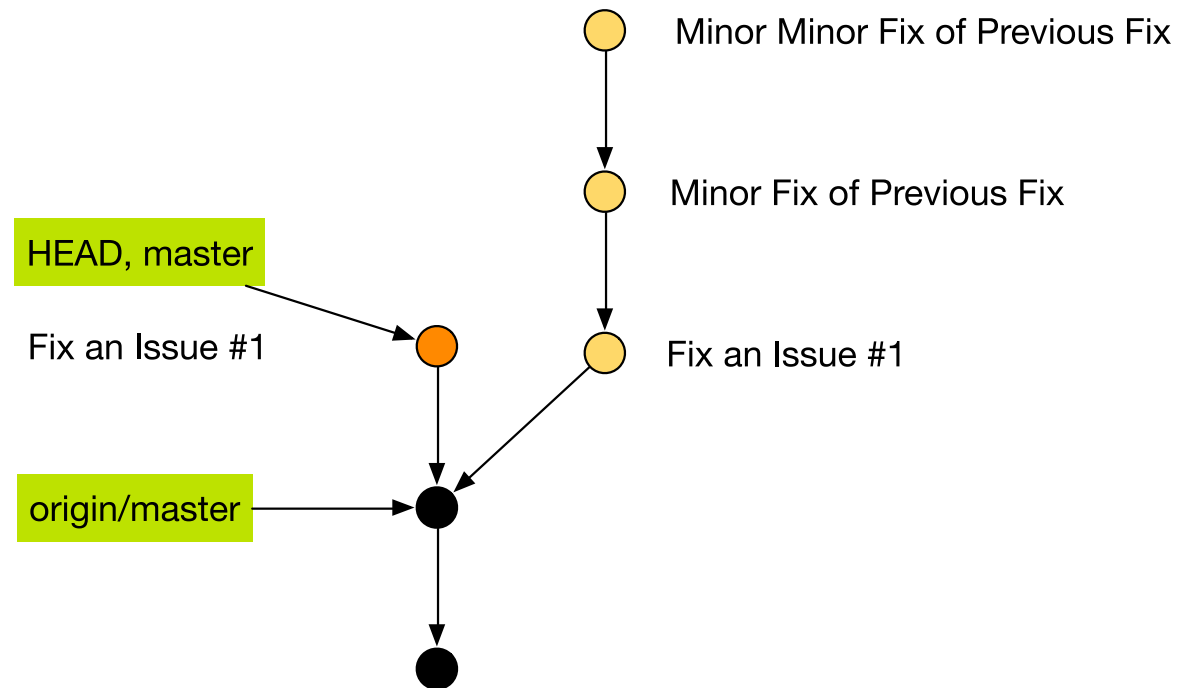
Interactive Rebase

Run **git rebase -i origin/master**

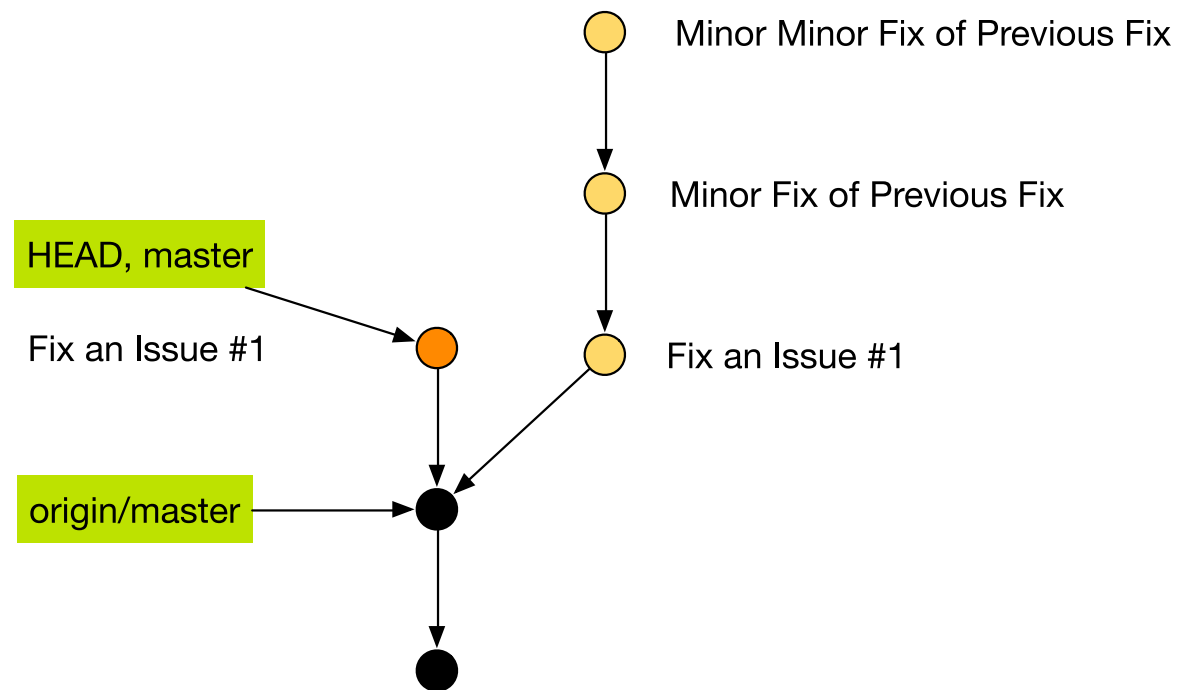
```
pick 40155b4 Fix an Issue #1
f 42e5db9 Minor Fix of Previous Fix
f a31c7cd Minor Minor Fix or Previous Fix

# Rebase ec2ae2b..cf0977c onto ec2ae2b
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
#
# These lines can be re-ordered; they are executed from top to bottom
```

Interactive Rebase

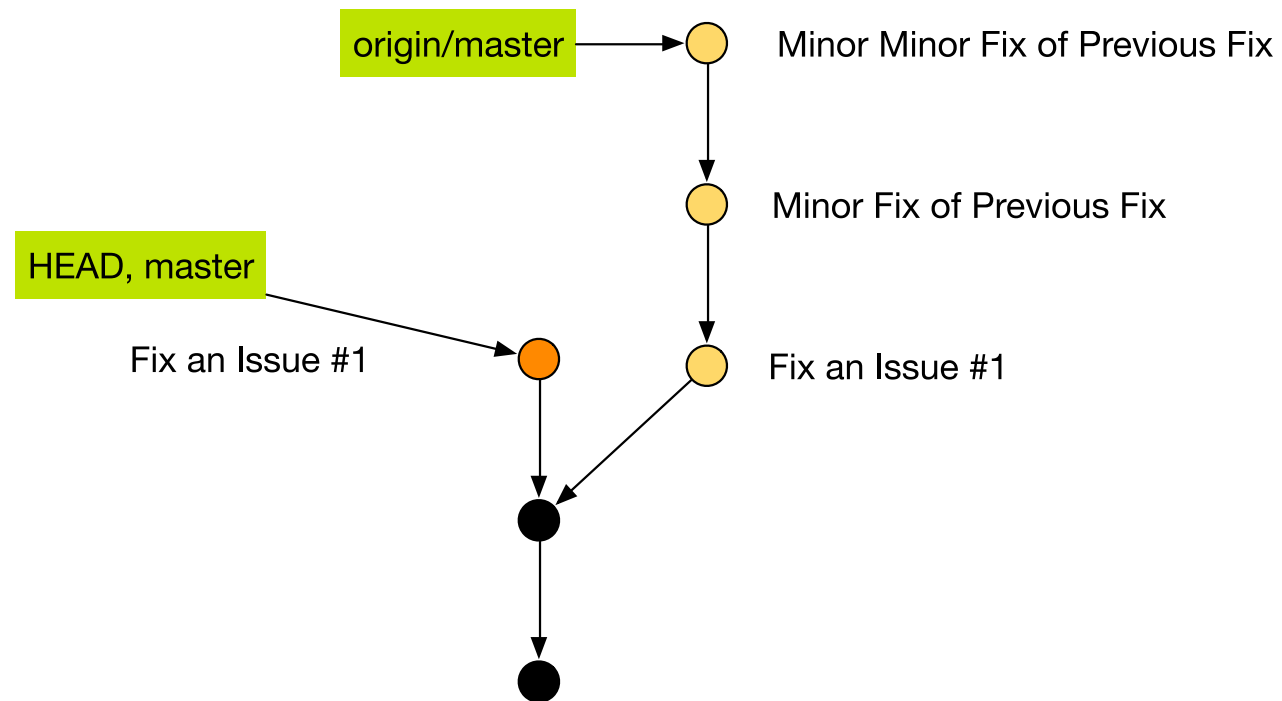


Warning on Rebase



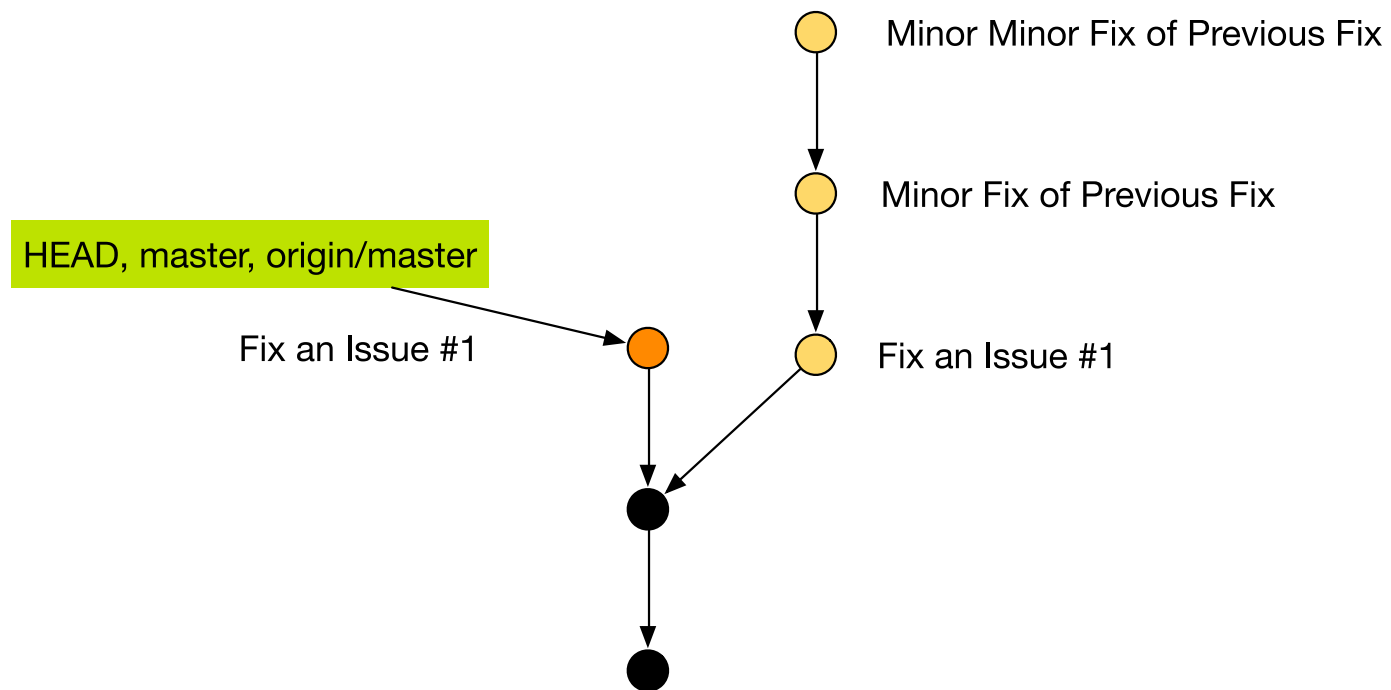
Rebase will create a **new** commit.

Warning on Rebase



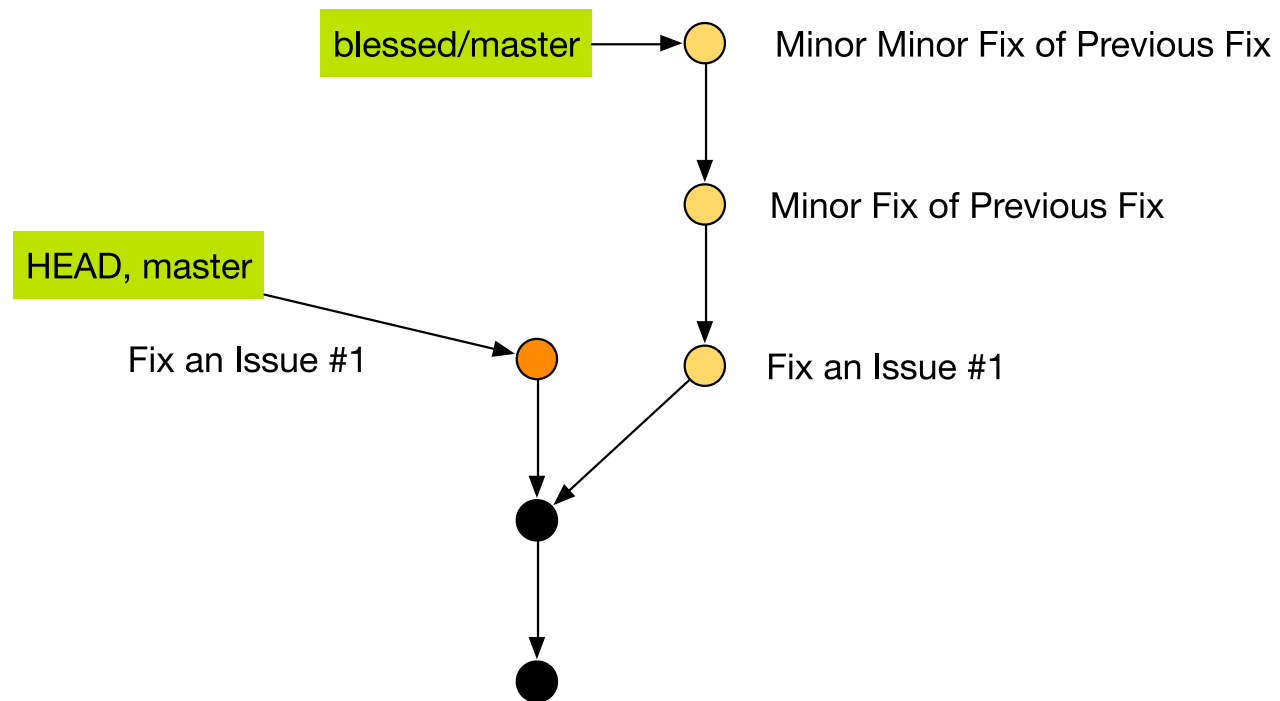
It's OK to rebase and **destructively** update **personal repo**.

Warning on Rebase



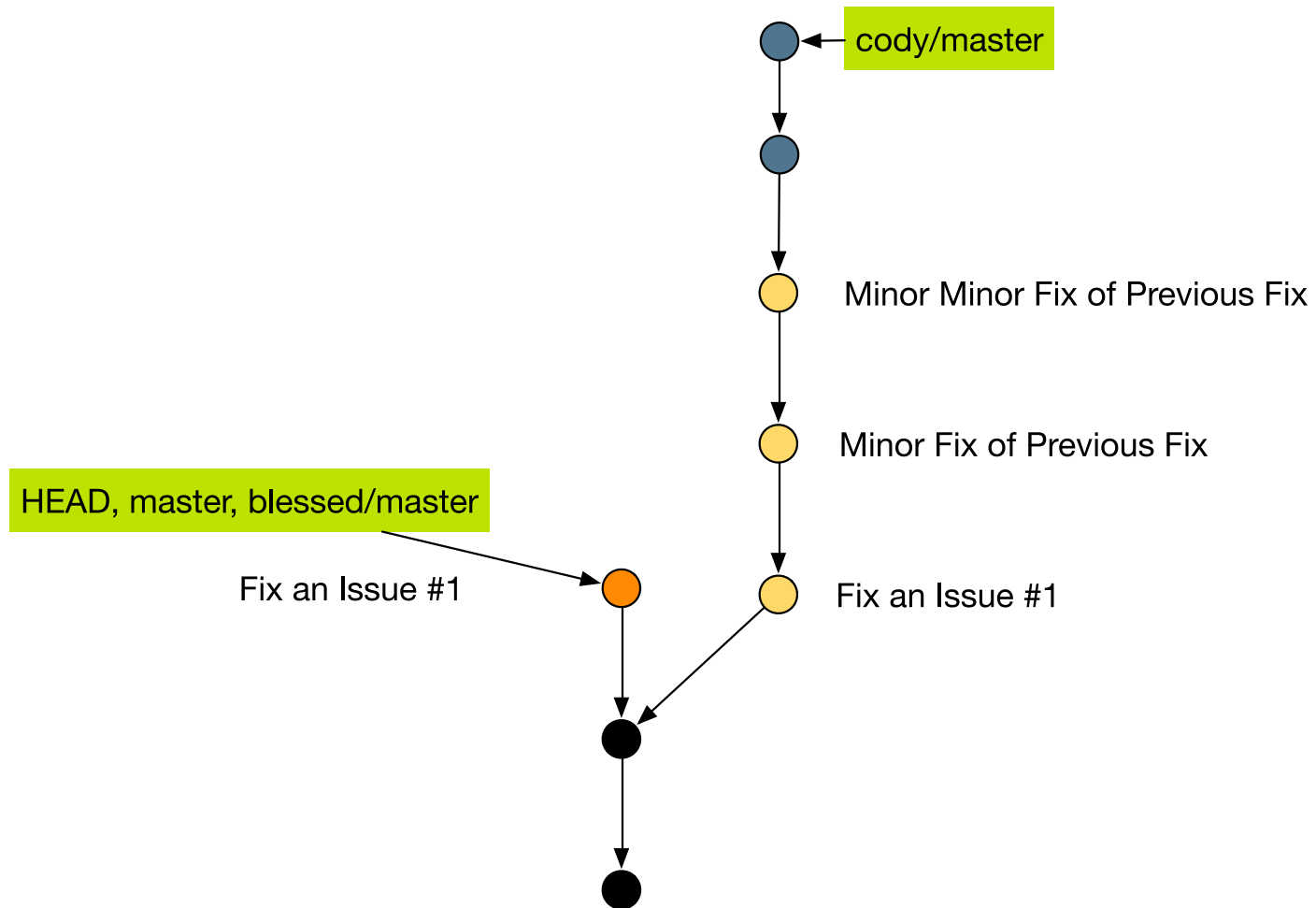
It's OK to rebase and **destructively** update **personal repo**.

Warning on Rebase



But you should **not destructively** update the **blessed repo**.

Warning on Rebase



Because other users base on **blessed/master**.

Reset

"Git Reset Demystified"

<http://git-scm.com/blog/2011/07/11/reset.html>

Magit (emacs's git-mode)

(Available from [package-list](#))

- <https://github.com/mtdavidson/magit-cheatsheet/>
- <http://vickychijwani.github.io/2011/09/09/Magit-Part-I/>
- <http://vickychijwani.github.io/2011/11/26/An-Introduction-to-Magit-Part-II/>

Github tips

[**https://github.com/tiimgreen/github-cheat-sheet**](https://github.com/tiimgreen/github-cheat-sheet)

Other Useful Resources

- Quick Guide : <http://rogerdudler.github.io/git-guide/>
- Complete Reference : <http://git-scm.com/book>
- Visual Git Reference :
<http://marklodato.github.io/visual-git-guide/index-en.html>
- Nice Interactive Tutorial on Git Branches:
<http://pcottle.github.io/learnGitBranching/>

Thank you