Lecture #2
Introduction to Git

AMath 483/583 - Spring 2016

Announcements

- Online Office Hours posted
- Still working on finding a quiz time
- Additional / Secondary resources updated for this week

Primary References

- Official Git Documentation - Chapters 1,2,3
- (See Syllabus for additional / secondary refs.)

What is Git?

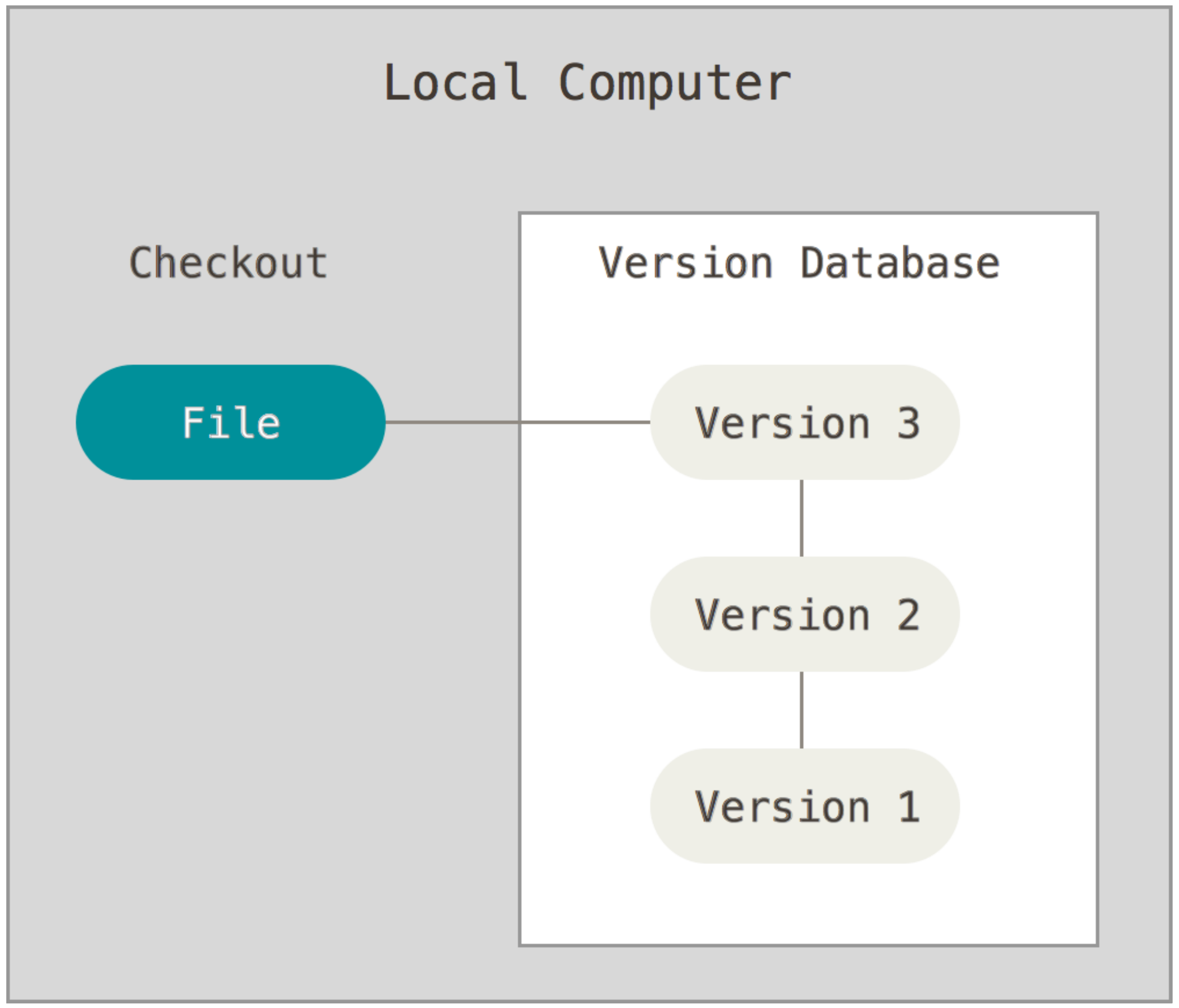
- A backup tool
- A change tracking tool
- A collaboration tool
- A history documentation tool
- “Version Control System”
  - git, mercurial, bazaar, subversion (SVN), concurrent versions system (CVS)

What is Git

- (In this class) Homework submission tool

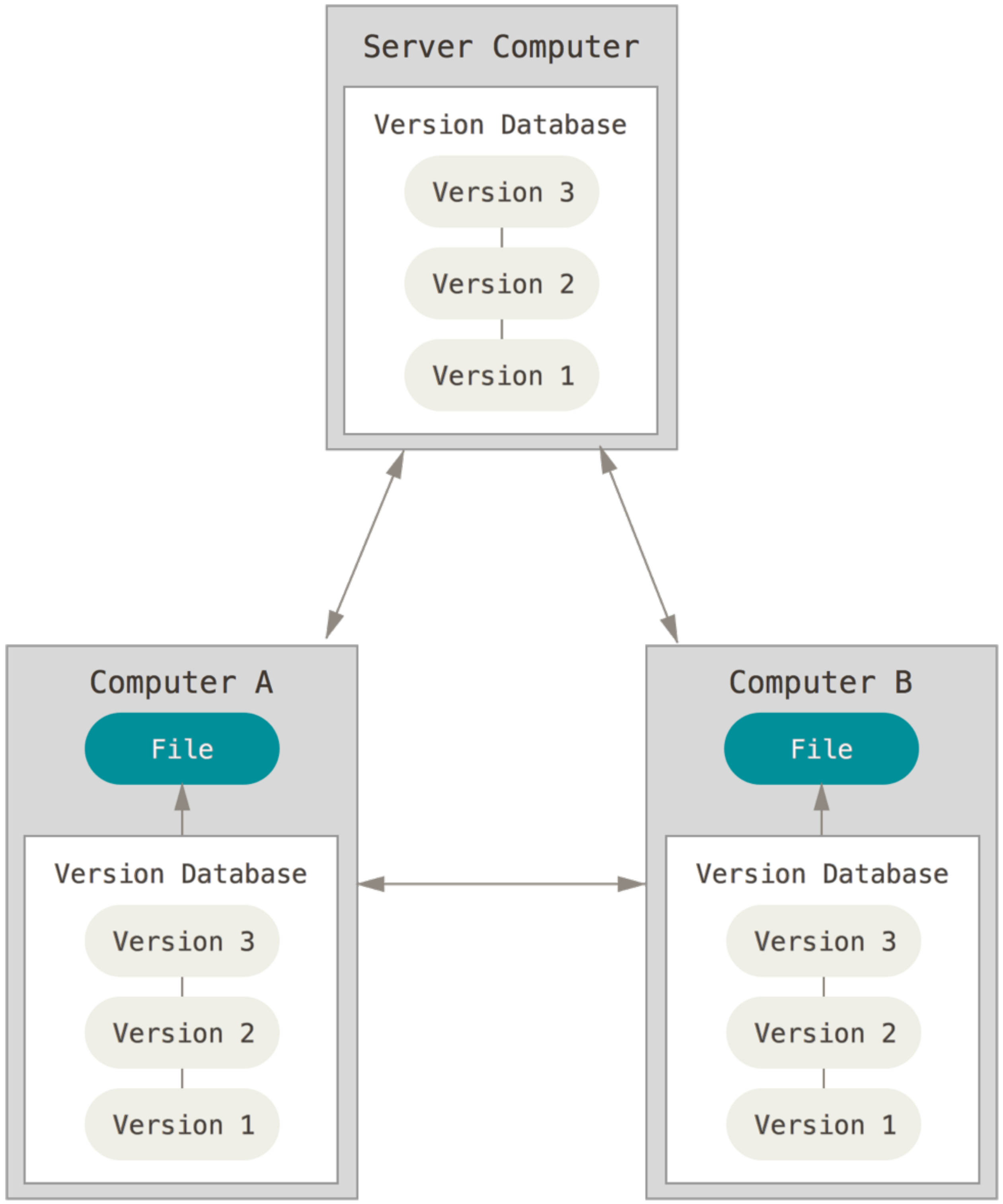
Local Backups

- Versions of `File` kept on Local Computer
- Can only share current version of `File`

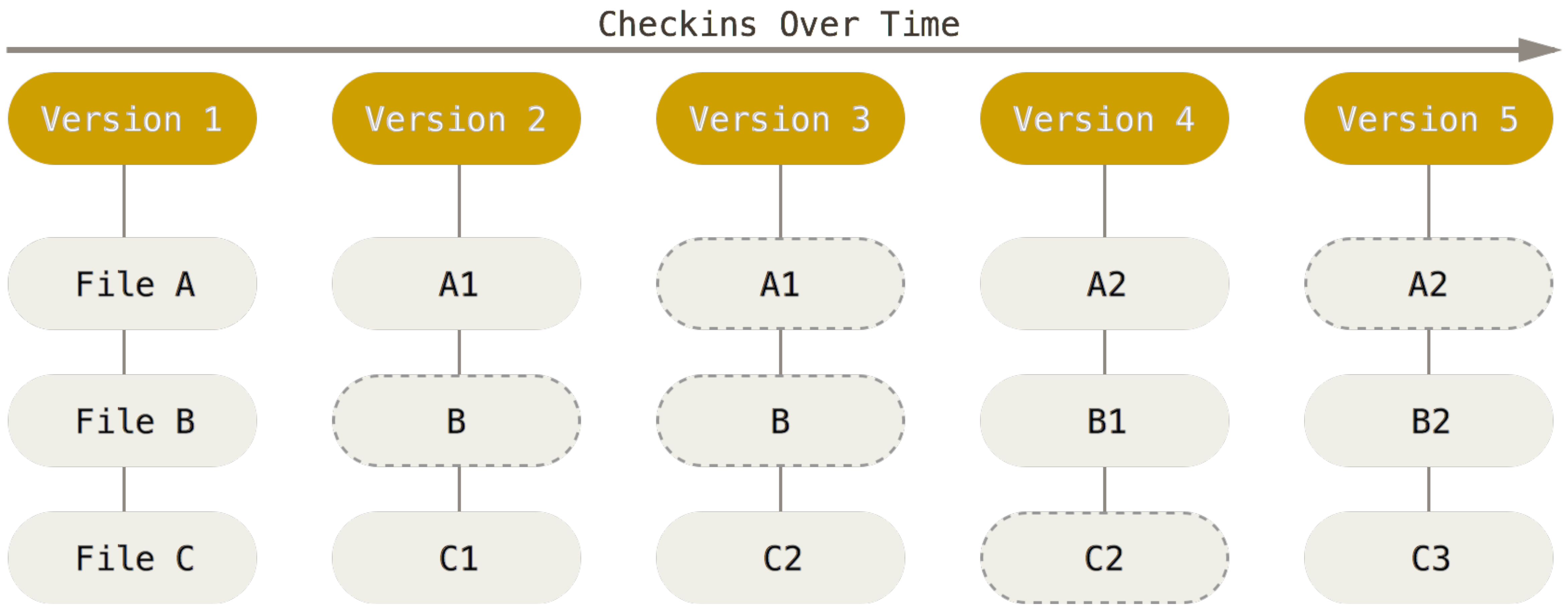


Distributed Version Control

- Everyone has complete version history of `File`

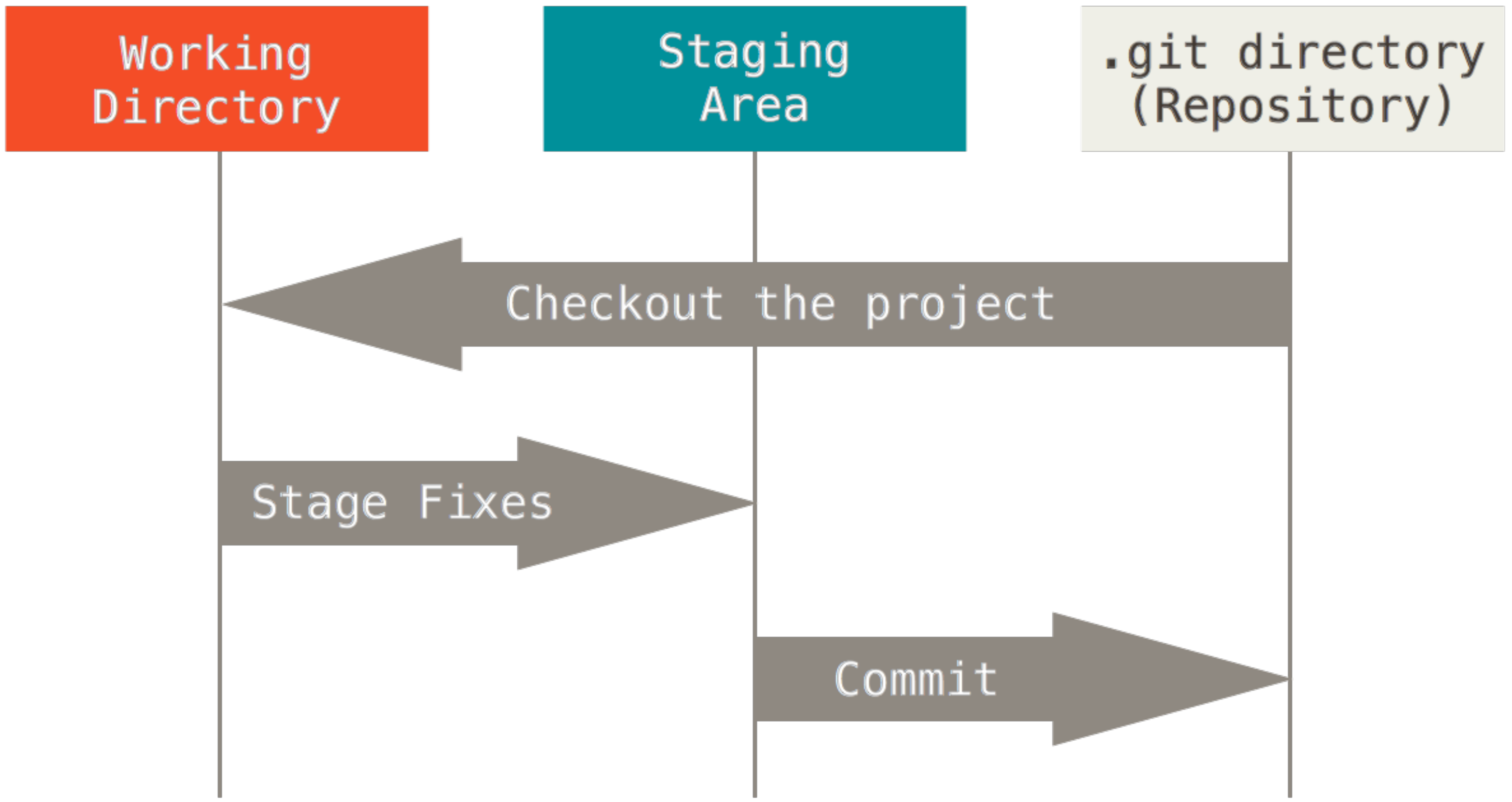


Git Basics



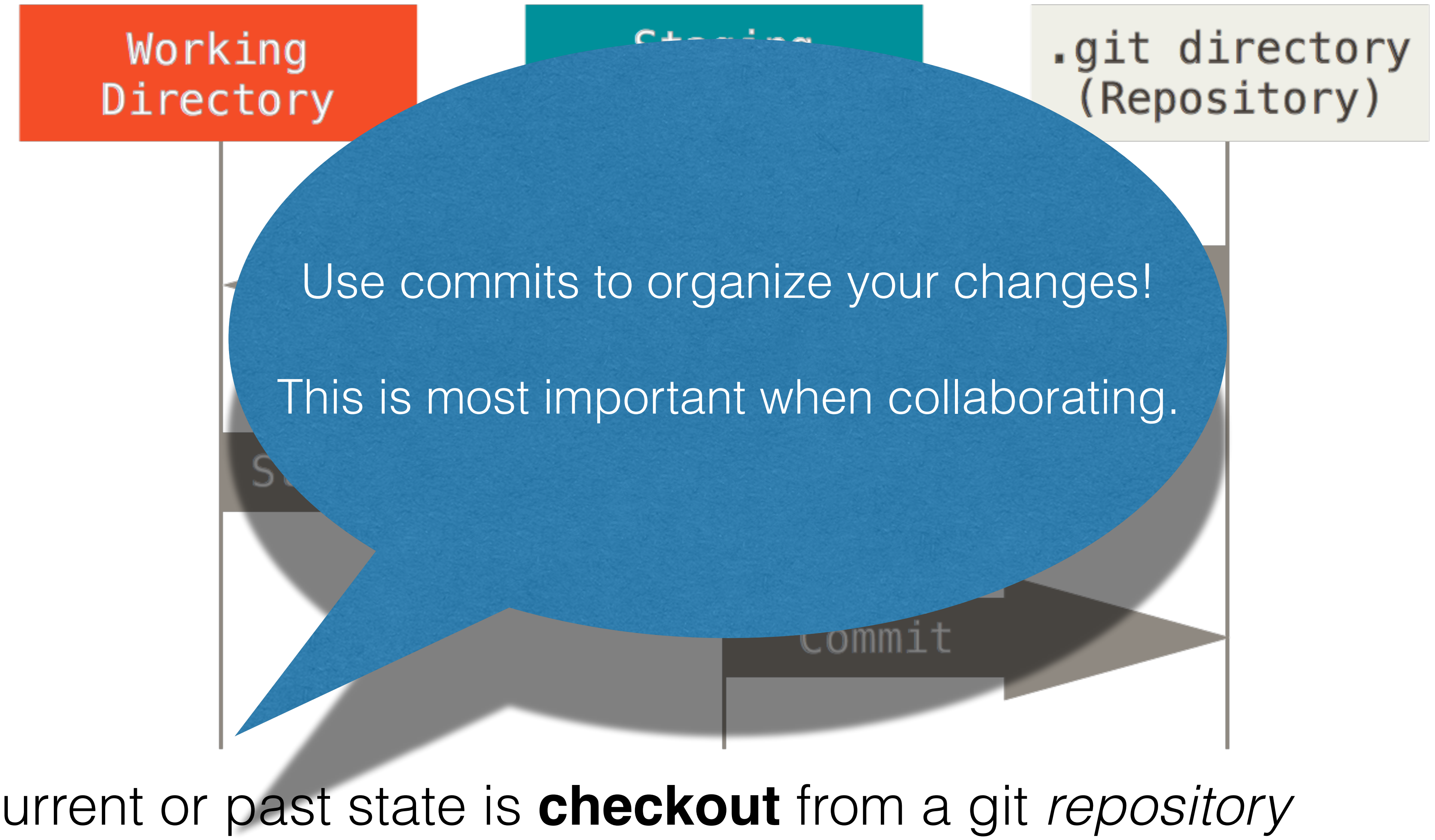
- Git keeps track of the state of files over time
  - save the state of your files every time you “commit”
  - simple use: backup system

Git Basics - Workflow



- A current or past state is **checkout** from a git *repository*
- Changes to this state are **add**ed to a staging area
- Staged files are **committed** to the local git database

Git Basics - Workflow


Working Directory
.git directory (Repository)
Use commits to organize your changes!
This is most important when collaborating.
Commit

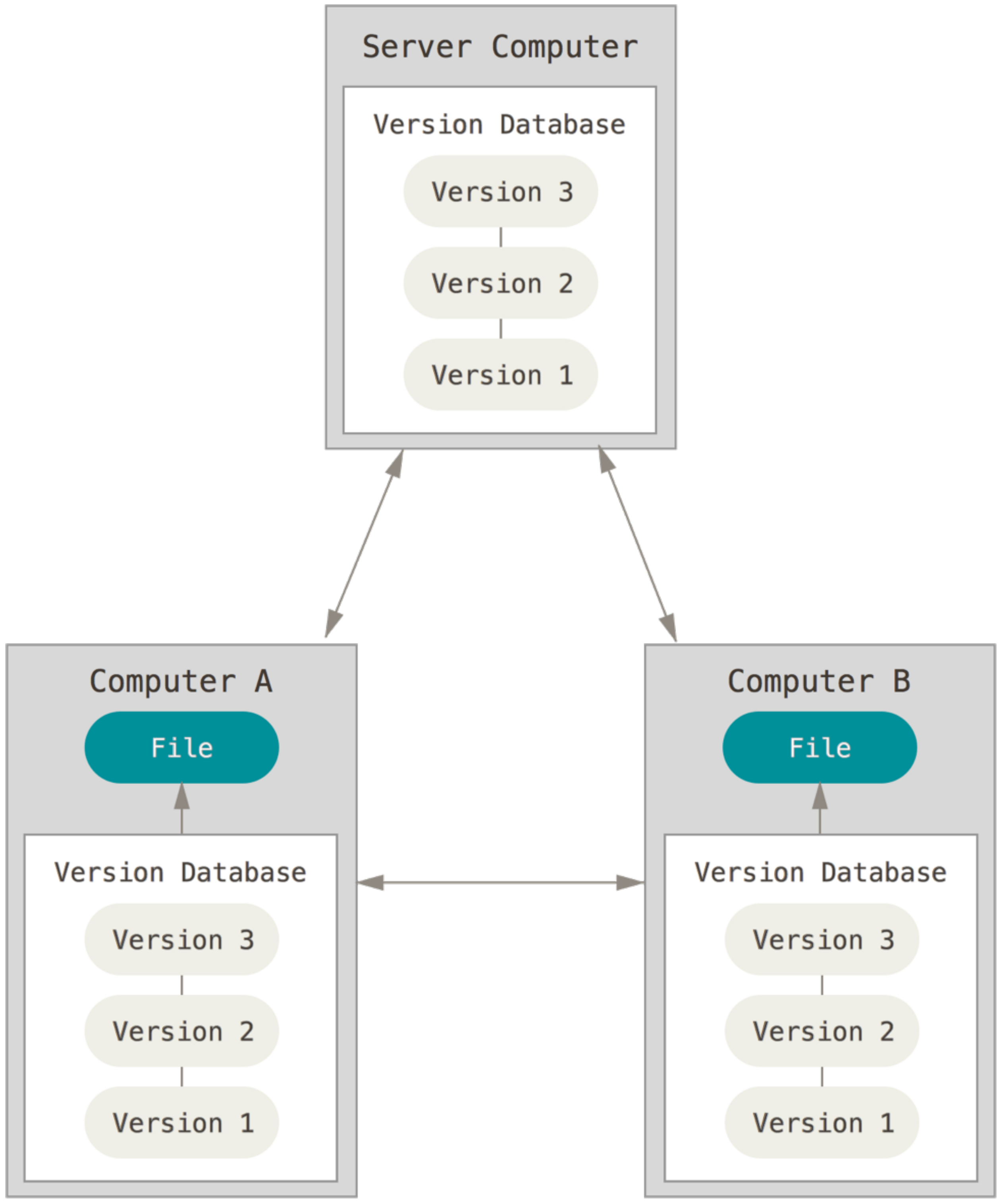
- A current or past state is **checkout** from a git *repository*
- Changes to this state are **add**ed to a staging area
- Staged files are **committed** to the local git database

Demonstration

`config`ure global settings, `init`ialize a Git repo, `add` files, `commit`, view history `log`

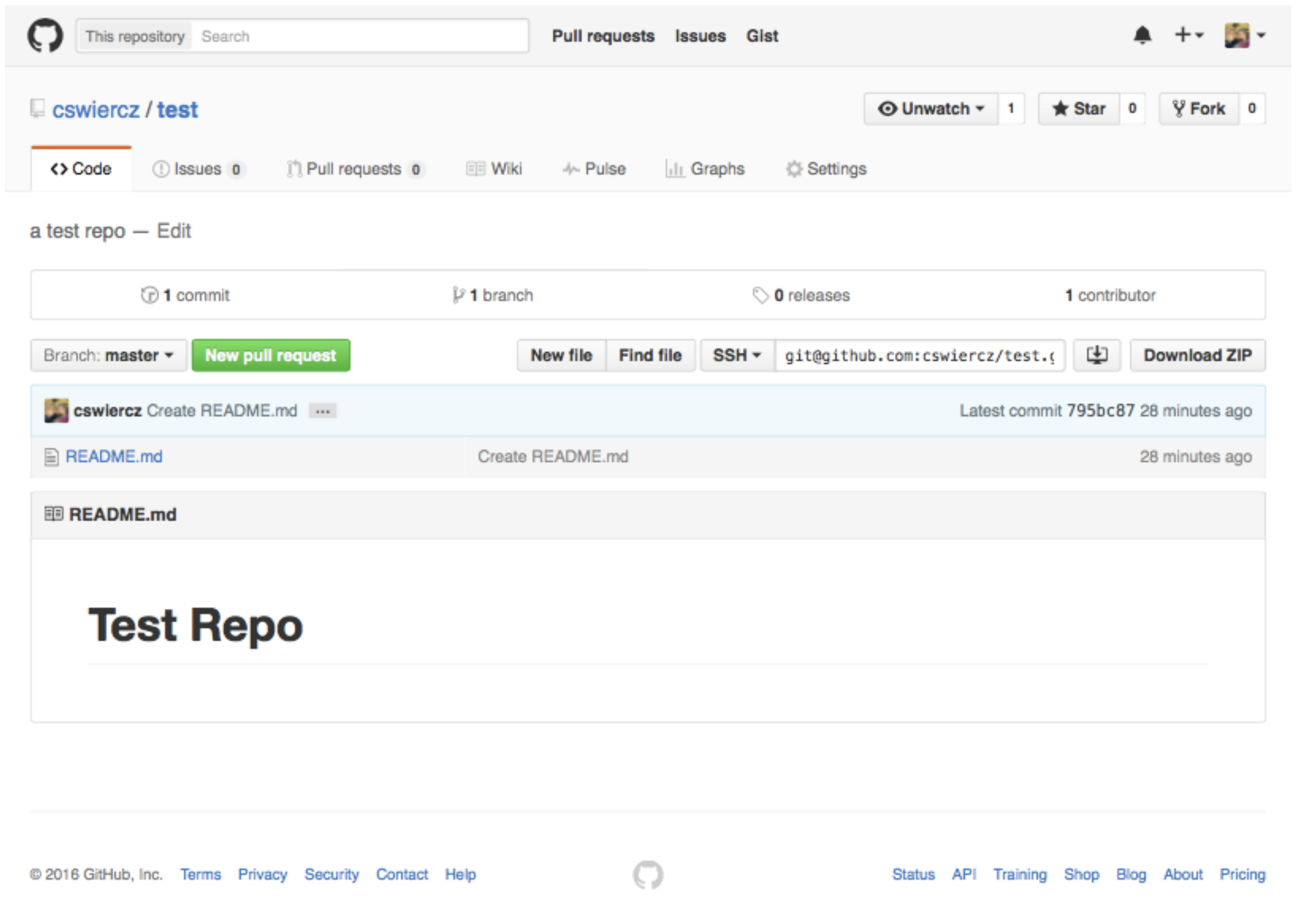
Working with Remotes

- Remote repositories make it possible to backup, share, and collaborate
- A `remote` is any repo containing the same .git database as your repo
  - sharing `commits` = synchronizing repo databases
  - sharing `commits` = `pull`ing changes from other repos
  - (by default, no `push` permissions)



GitHub

- Public remotes for everyone (as opposed to private remote on your computer)
- Plus additional tools for easy collaboration
  - Pull Requests
  - Issues Pages
  - “Social Networking”

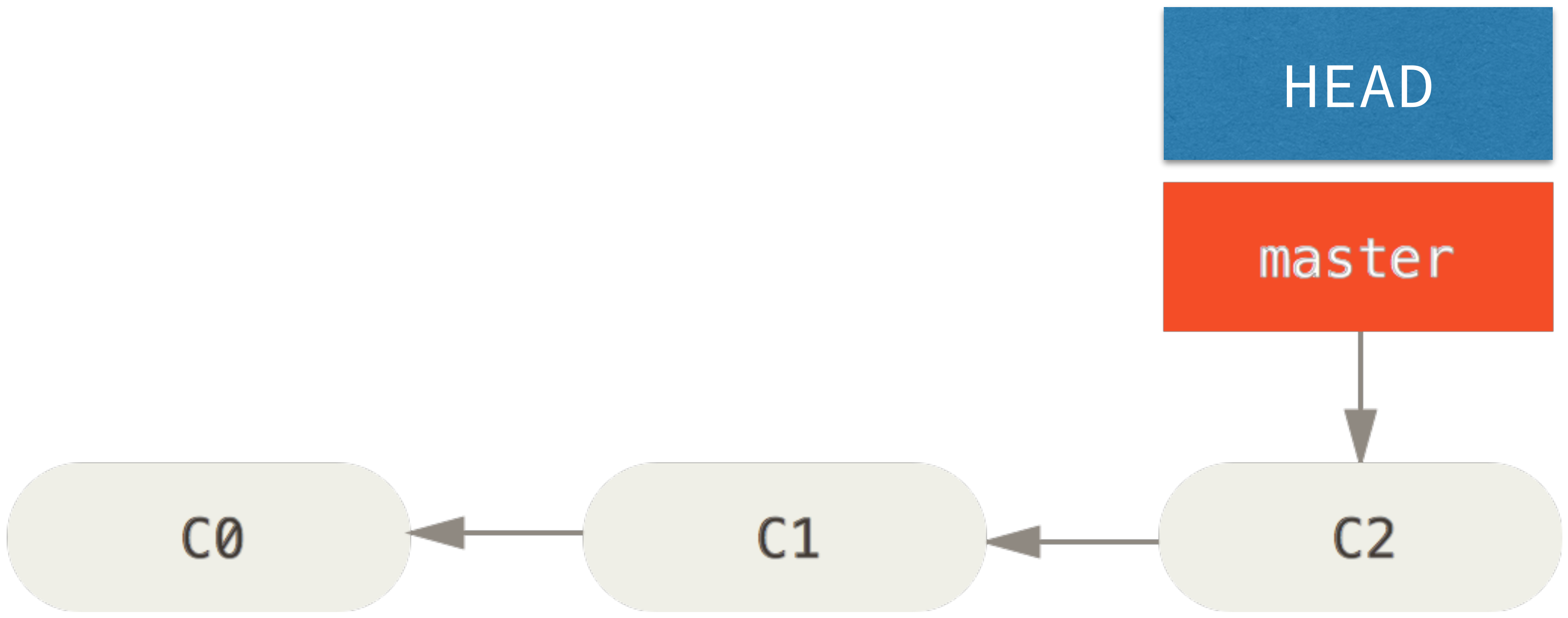


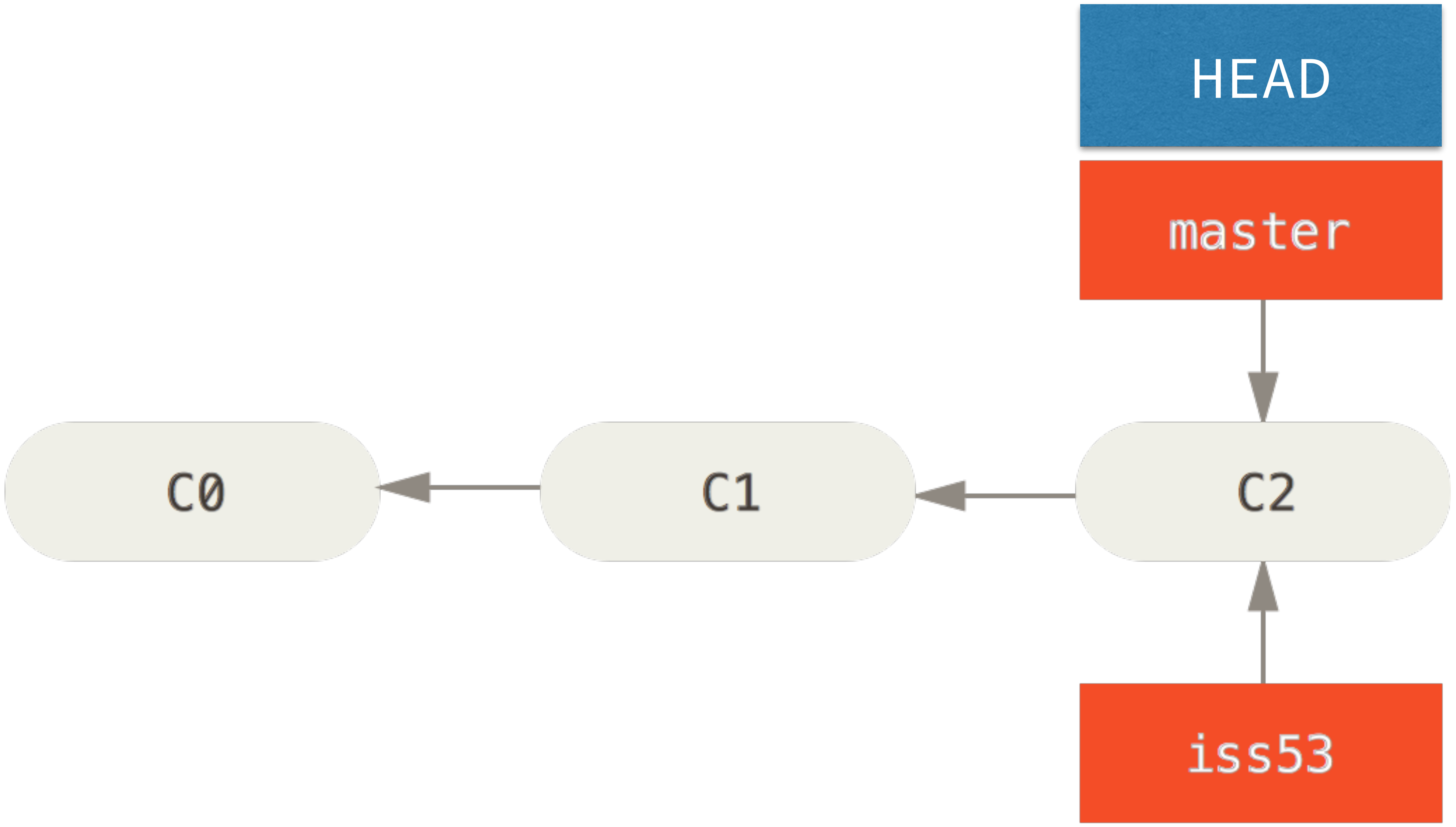
Demonstration

Set up public GitHub repo, `push` local changes.

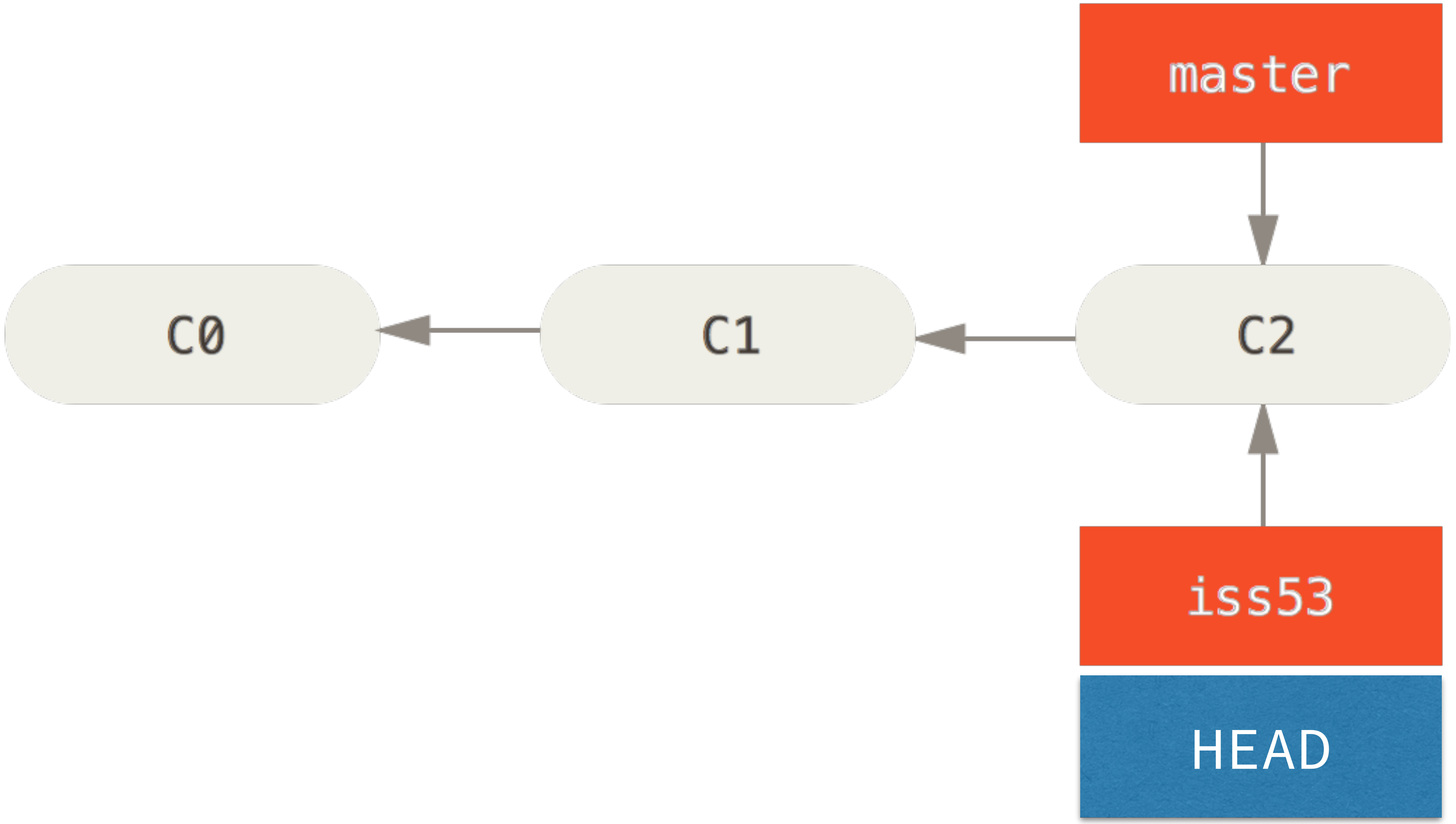
Branching

- One of the most powerful tools provided by git
- Easily test features / additions without affecting original code
- Organizes how multiple developers contribute to code
- `master` branch used as last working state

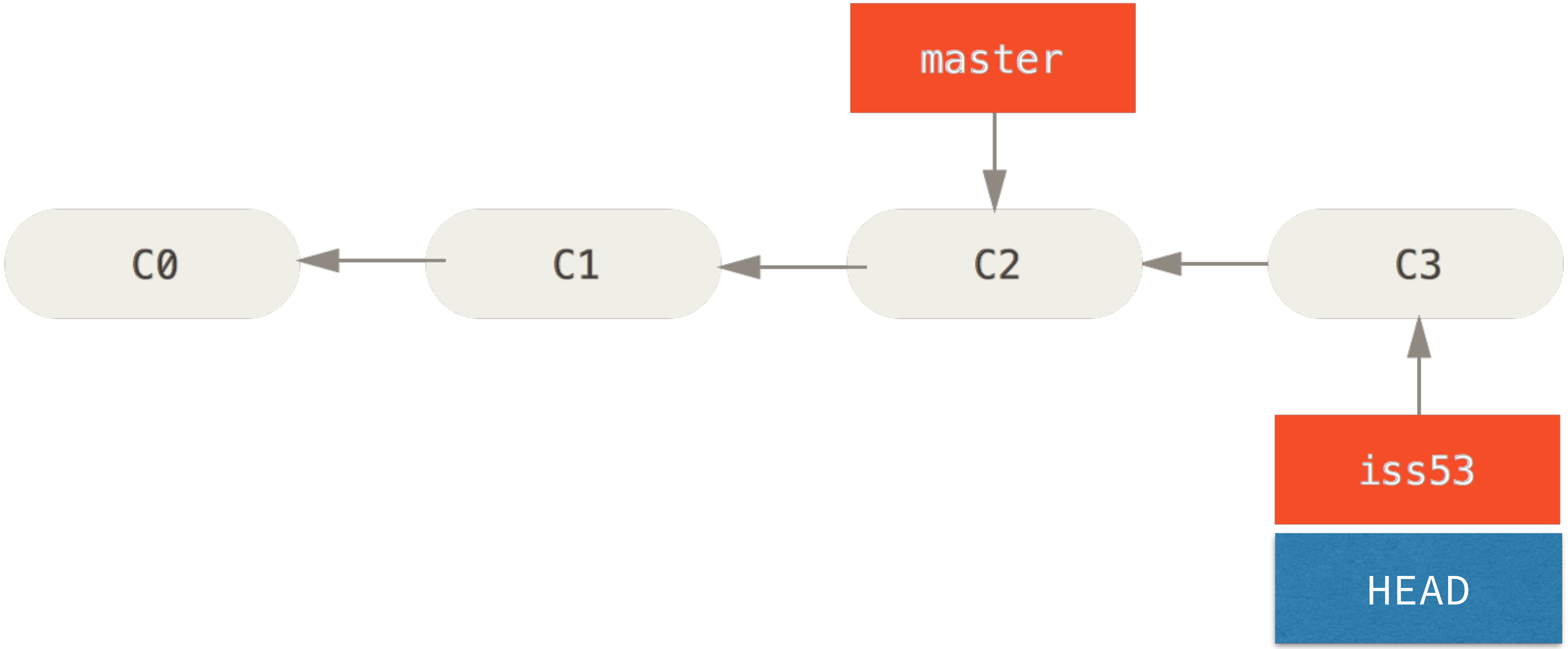

HEAD
master
C0
C1
C2



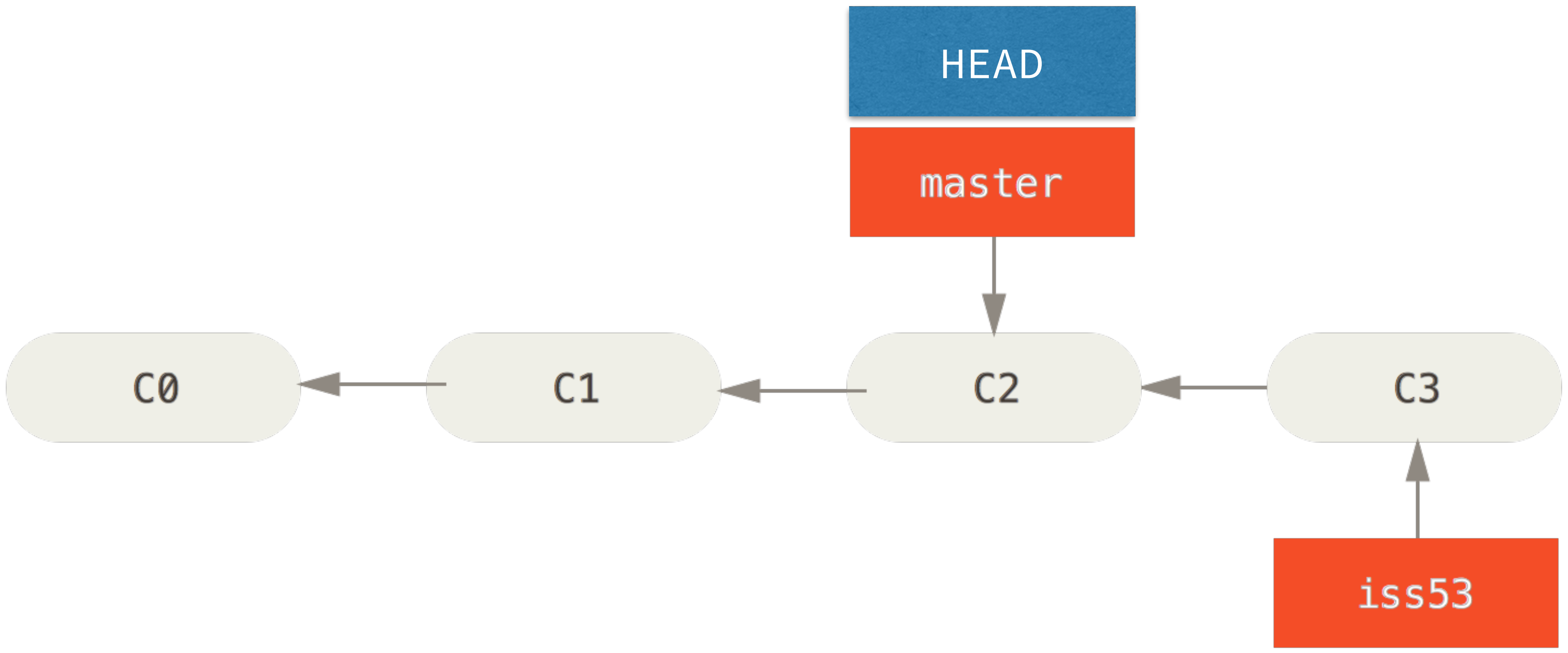
```
$ git branch iss53
```



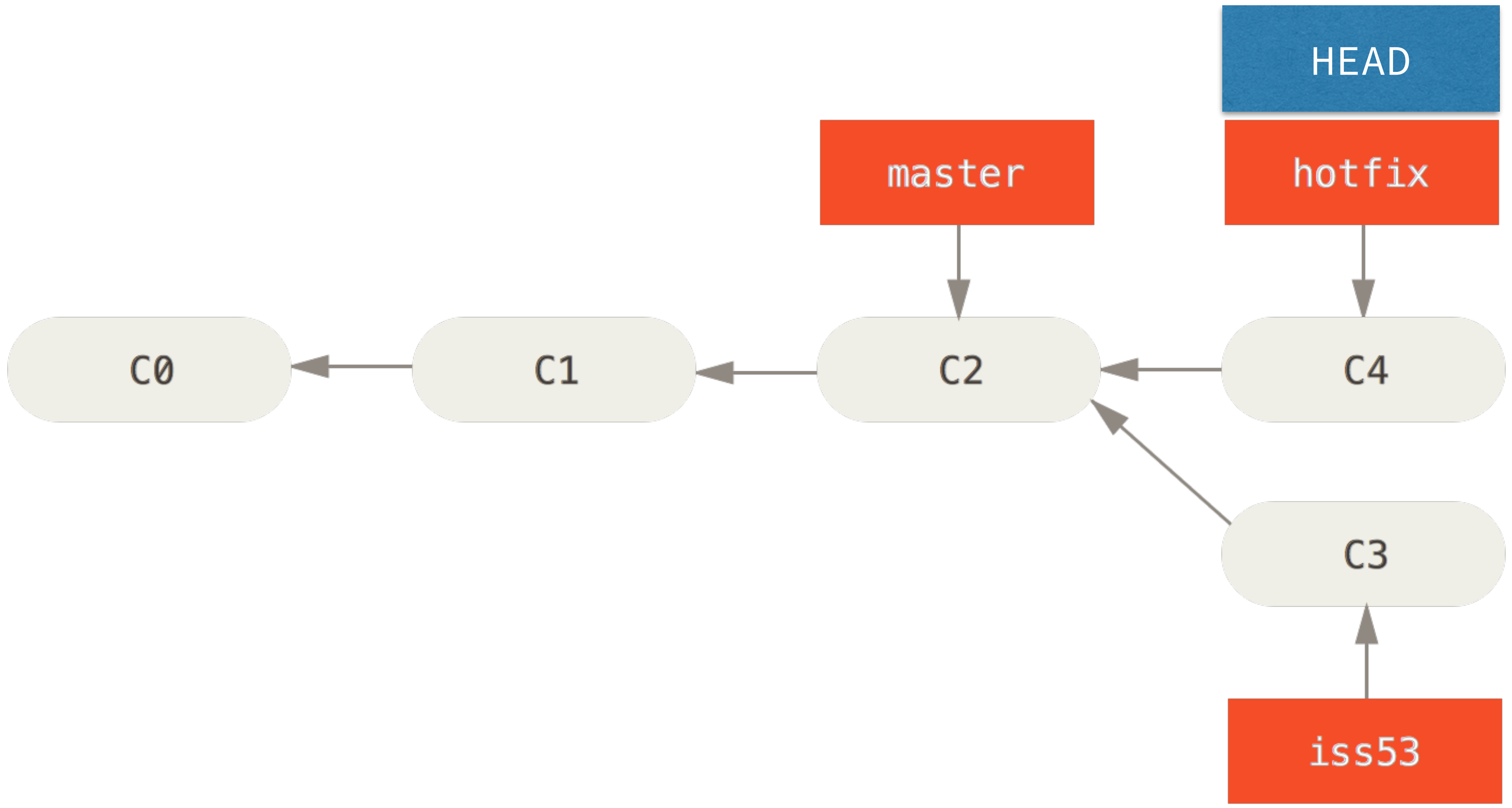
```
$ git checkout iss53
```



```
$ emacs README.md
$ git commit -a -m “working on issue”
```



```
$ git checkout master
```



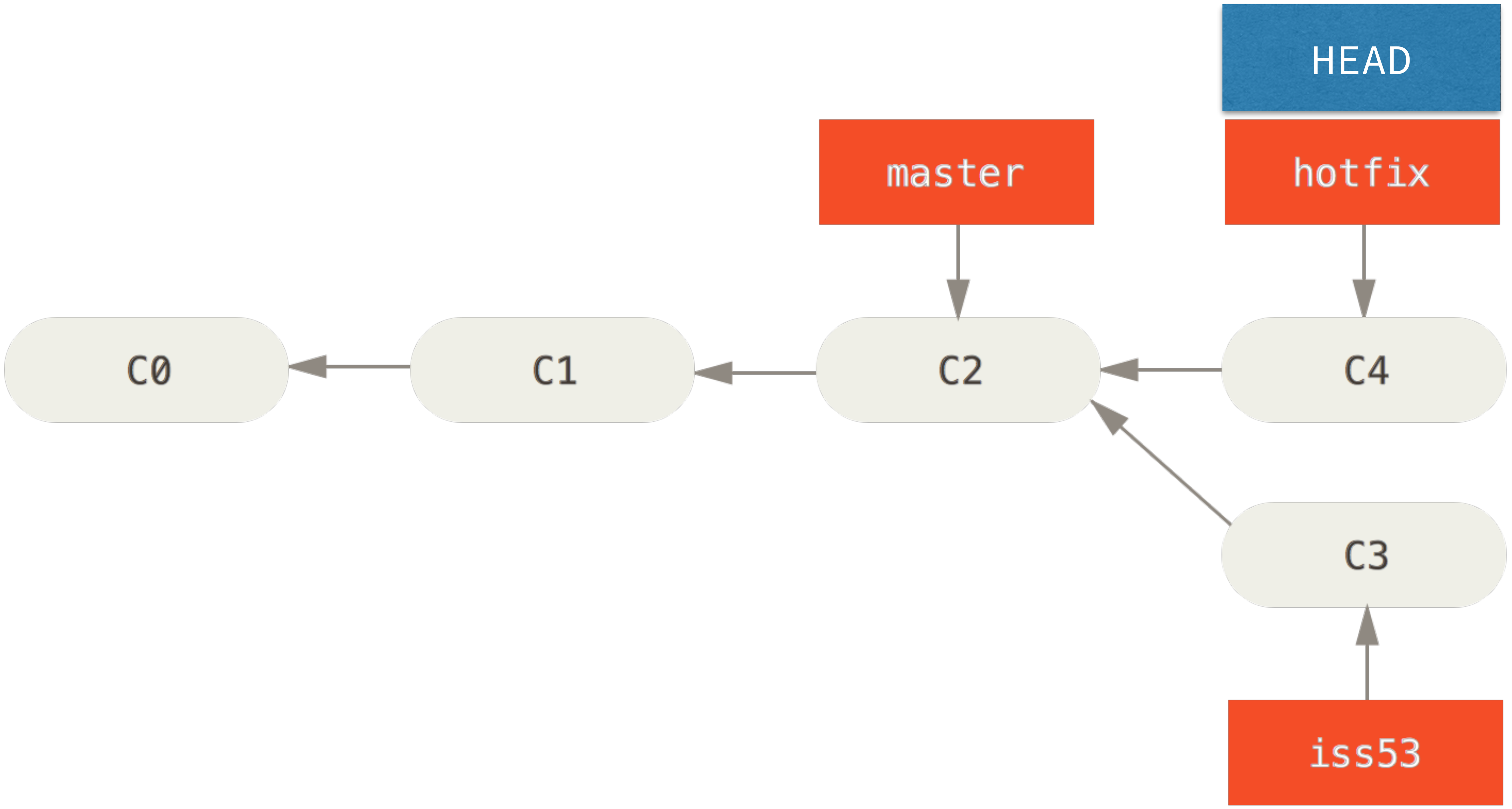
```
$ git checkout -b hotfix
$ emacs README.md
$ git commit -a -m “gotta fix it”
```

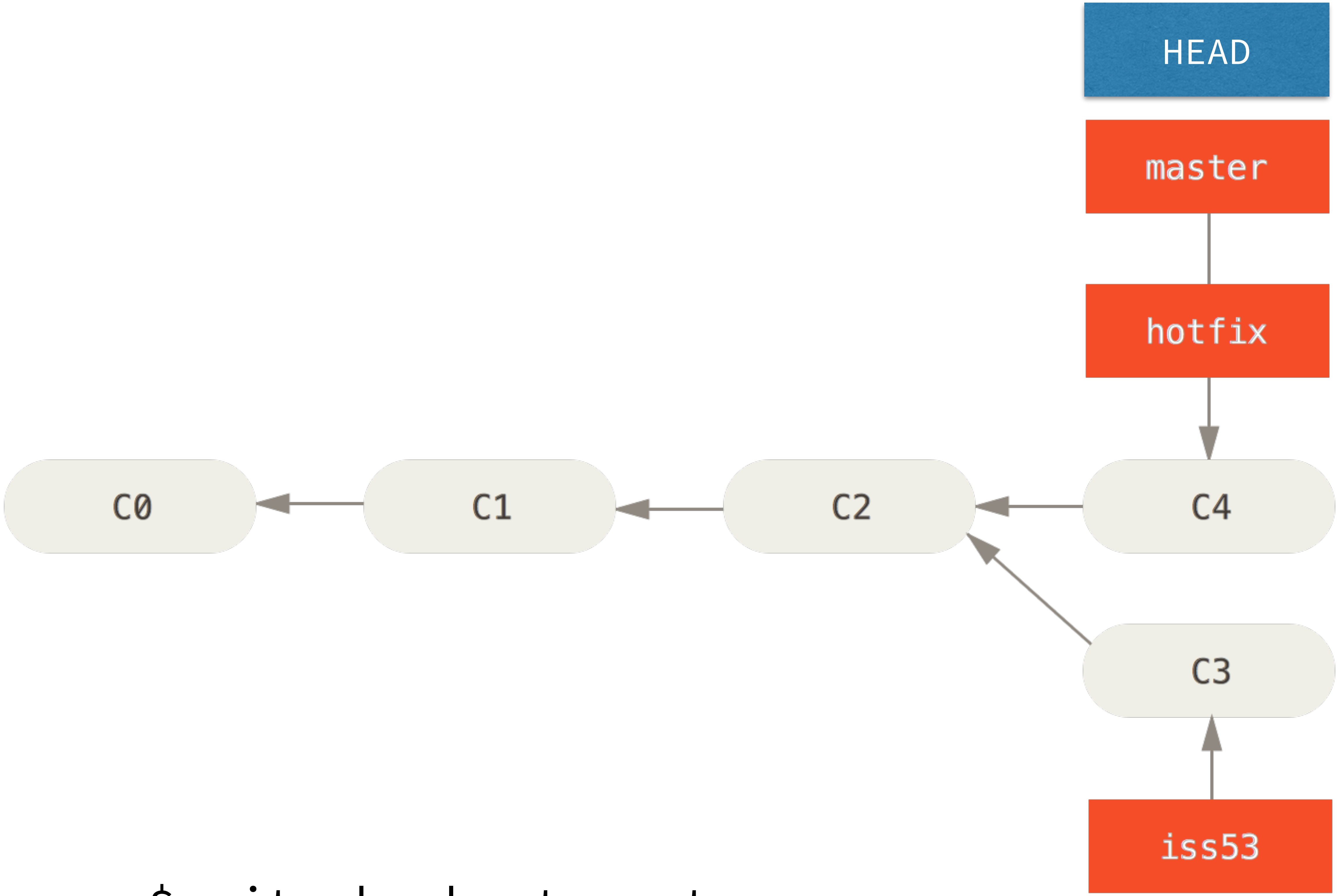
Demonstration

Create a test branch, make changes, switch to master branch, make more changes, merge test branch.

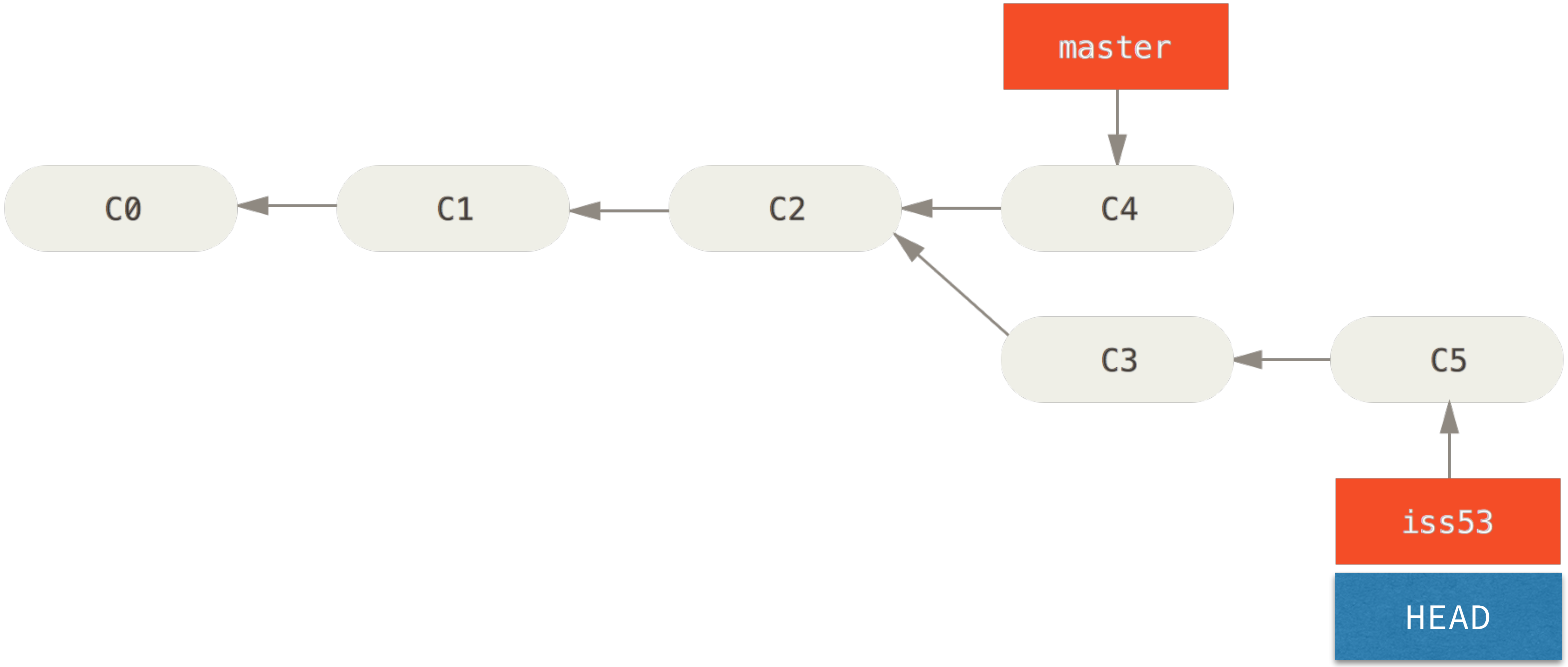
Merging

- Take changes in one branch and merge them into other
- Two types:
  - fast-forward (easy) — only requires moving the branch/commit pointer
  - non-fast forward (harder) — cannot simply move pointer

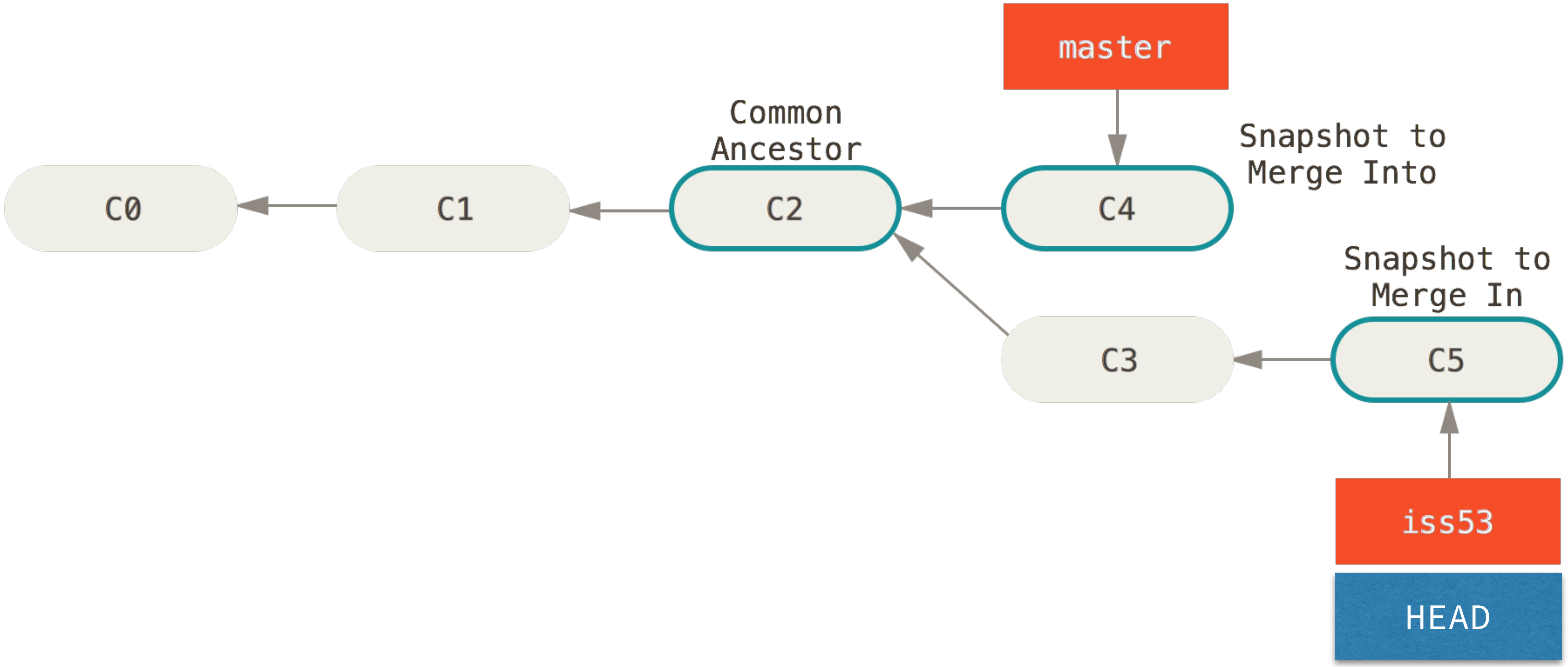

HEAD
master
hotfix
C0
C1
C2
C4
C3
iss53

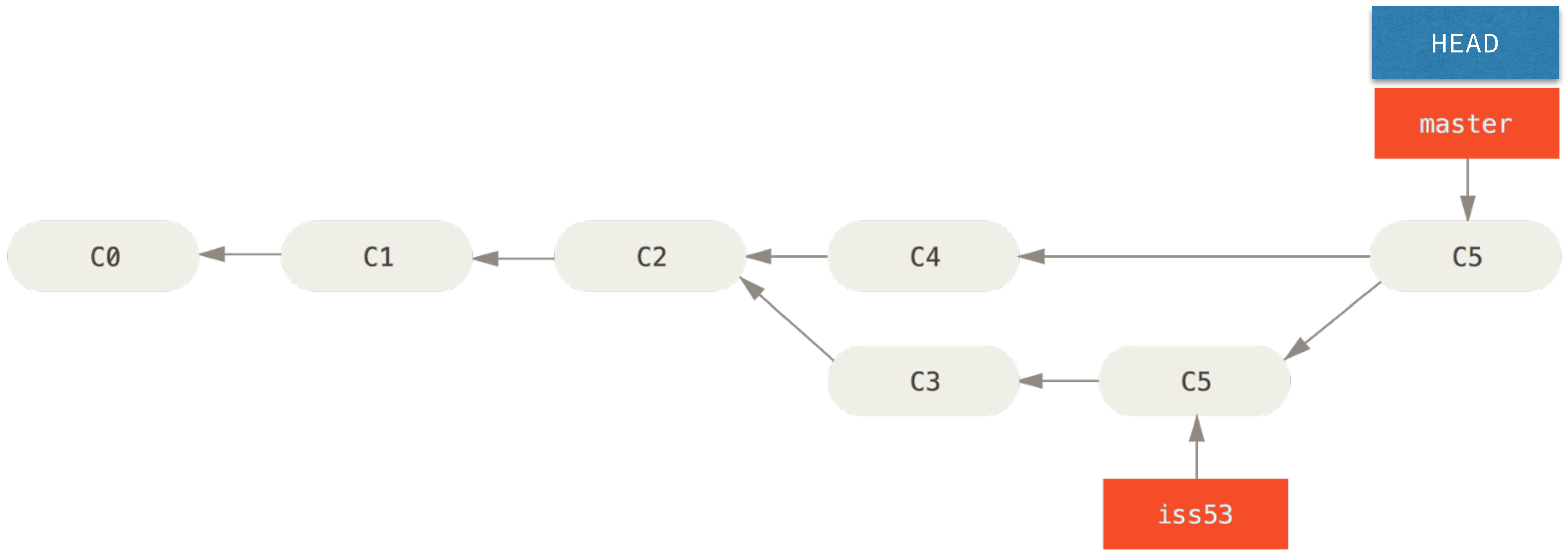


```
$ git checkout master
$ git merge hot fix
```



```
$ git branch -d hot fix
$ git checkout iss53
(make further changes and commit)
```


master
Common Ancestor
Snapshot to Merge Into
C0
C1
C2
C4
Snapshot to Merge In
C3
C5
iss53
HEAD



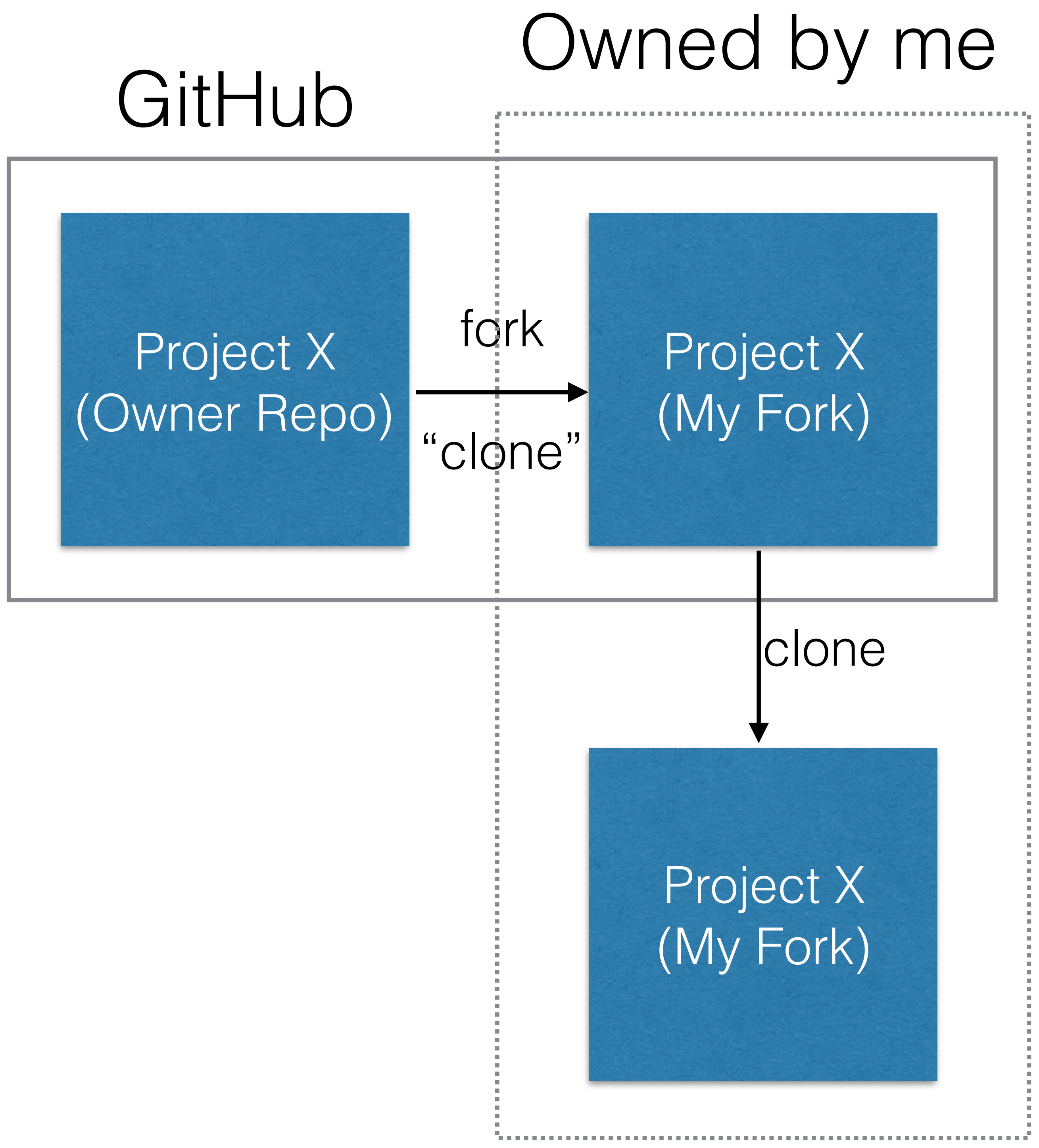
```
$ git checkout master
$ git merge iss53
(later) $ git branch -d iss53
```

Demonstration

Merging branches.

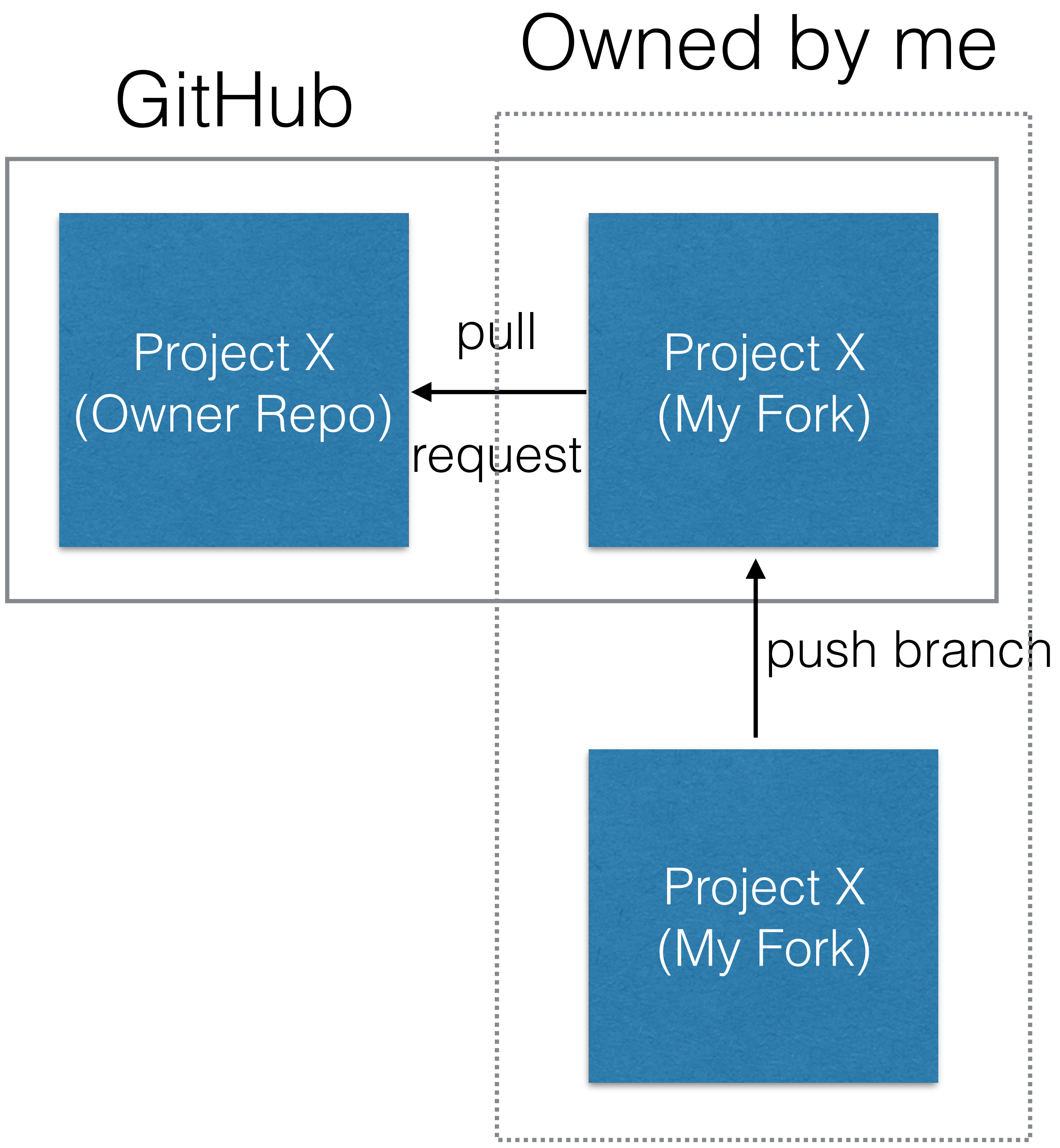
GitHub Workflow

- "I want to contribute to project X"
- "Fork" Project X (GitHub for "clone")
- Make local "clone" on your computer
- Create working branch
- Push branch to your fork
- Initiate pull request



GitHub Workflow

- "I want to contribute to project X"
- "Fork" Project X (GitHub for "clone")
- Make local "clone" on your computer
- Create working branch
- Push branch to your fork
- Initiate pull request



Demonstration

Forking a repo on GitHub, initiating a pull request.
Accepting a pull request.

Real World Practice

- Fork a repo you think is interesting (e.g. http://github.com/sympy/sympy)
- Read the online documentation and find a place to improve, add an example, etc.
- Find where the documentation is stored in the repo
- Make changes, push branch, initiate pull request