

How to Use Git to Solve World Hunger

a humble talk by Scott Chacon

Me

Scott Chacon



github
SOCIAL CODE HOSTING

**schacon**

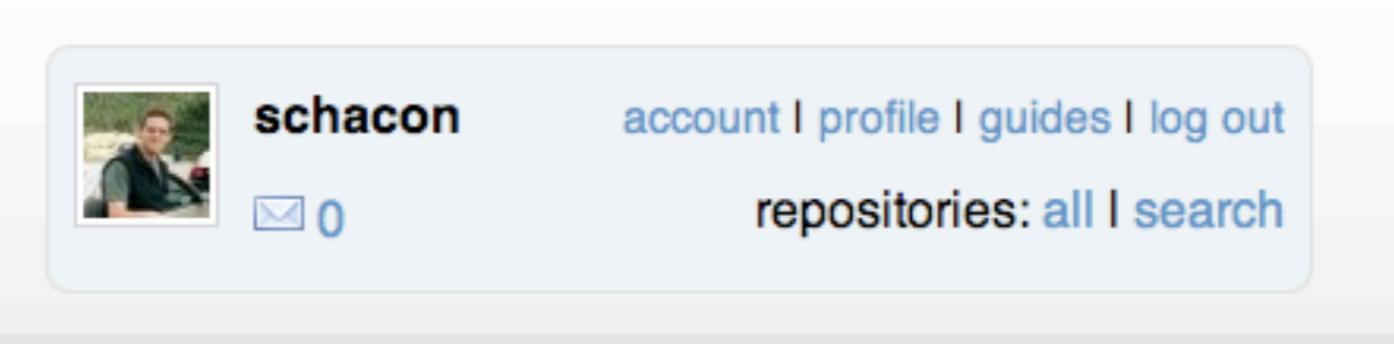
✉ 0

[account](#) | [profile](#) | [guides](#) | [log out](#)[repositories: all](#) | [search](#)

Your Repositories (create a new one)

[all](#) | [public](#) | [private](#) | [sources](#) | [forks](#) [asgit](#) [fireeaglet](#) [git-lighthouse](#) [git-ruby](#) [git-source](#) [munger](#) [ruby-git](#) [simplegit](#) [ticgit](#)

github.com/schacon



 **schacon** account | profile | guides | log out
✉ 0 repositories: all | search

Your Repositories (create a new one)

[all](#) | [public](#) | [private](#) | [sources](#) | [forks](#)

 [asgit](#)

 [fireeaglet](#)

 [git-lighthouse](#)

 [git-ruby](#)

 [git-source](#)

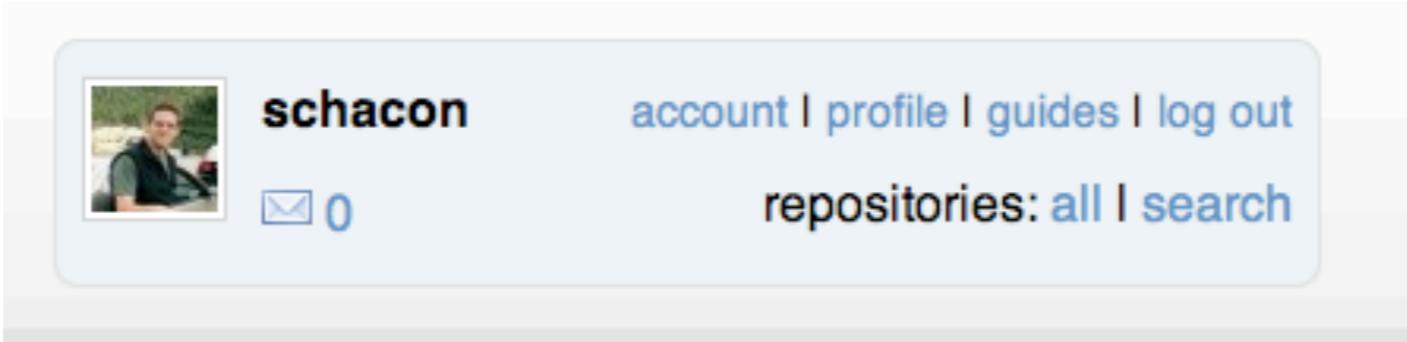
 [munger](#)

 [ruby-git](#)

 [simplegit](#)

 [ticgit](#)

github.com/schacon



A screenshot of a GitHub user profile. At the top left is a small square thumbnail of a man's face. To its right is the username "schacon". Further to the right are links for "account", "profile", "guides", and "log out". Below the username is a blue envelope icon followed by the number "0". To the right of the envelope are links for "repositories: all" and "search".

Your Repositories (create a new one)

[all](#) | [public](#) | [private](#) | [sources](#) | [forks](#)

 [asgit](#)

 [fireeaglet](#)

 [git-lighthouse](#)

 [git-ruby](#)

 [git-source](#)

 [munger](#)

 [ruby-git](#)

 [simplegit](#)

 [ticgit](#)



\$9

Git Internals

Source code control and beyond

by Scott Chacon

THE EXPERT'S VOICE®

Pro Git

Patrick Aljord and Scott Chacon

Apress®

git-scm.com



+++ git

git the fast version control system

A cartoon orange character with a single sprout on its head is standing in a field of small green trees.

Home About Git Documentation Download Related Tools Wiki

The [Git User's Survey 2008](#) is up! Please devote a few minutes of your time to fill it out, so we can improve Git!

Git is...

Git is an **open source, distributed version control system** designed to handle everything from small to very large projects with speed and efficiency.

Every Git clone is a full-fledged repository with complete history and full revision tracking capabilities, not dependent on network access or a central server.
Branching and merging are fast and easy to do.

Git is used for version control of files, much like tools such as [Mercurial](#), [Subversion](#), [CVS](#), [Perforce](#), [Bitkeeper](#), and [Visual SourceSafe](#).

Projects using Git

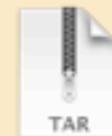
- [Git](#)
- [Linux Kernel](#)
- [Ruby on Rails](#)
- [WINE](#)
- [Fedora](#)
- [X.org](#)
- [Rubinius](#)
- [VLC](#)
- [Prototype](#)

Download Git

The latest stable Git release is

v1.6.0.2

[release notes](#) (2008-09-12)



[tar.bz2 \(sign\)](#)



[tar.gz \(sign\)](#)

[Other Download Options](#)

[Source and History](#)

Git Quick Start

Cloning and Creating a Patch

```
$ git clone git://github.com/git/hello-world.git  
$ cd hello-world
```

Creating and Committing

```
$ cd (project-directory)  
$ git init
```

Got a Question?

If you are curious about something, feel free to ask on the [IRC](#) channel or the [mailing list](#).

book.git-scm.com



The Git Community Book

Home About Git Documentation Download Related Tools Wiki

The Git Community Book

Welcome to the Git Community Book. This book has been built by dozens of people in the Git community, and is meant to help you learn how to use Git as quickly and easily as possible.

If you see anything out of date, have a suggestion on how to improve it, or would like to help add to the book, please see the [How to Contribute](#) page, or just send [our maintainer](#) a note.

Download

Click [here](#) to download the current PDF version of this book.



New to Git?

If you want to really understand Git, You may want to start at [the beginning](#).

If you just want to jump right in, you can skip right to [setting it up](#).

1. Introduction

[Welcome to Git](#)

[The Git Object Model](#)

[Git Directory and Working Directory](#)

[The Git Index](#)

2. First Time

[Installing Git](#)

5. Advanced Git

[Creating New Empty Branches](#)

[Modifying your History](#)

[Advanced Branching And Merging](#)

[Finding Issues - Git Bisect](#)

[Finding Issues - Git Blame](#)

[Git and Email](#)

[Customizing Git](#)

schacon@gmail.com

</me>

Git Crash Course

What is Git?

**Git is an open source,
distributed version control
system designed for speed
and efficiency**

**Git is an open source,
distributed version control
system designed for speed and
efficiency**

version control

mercurial (hg)

bazaar

subversion (svn)

version control

concurrent version system (cvs)

perforce

visual source safe

SoS

mercurial (hg)

bazaar

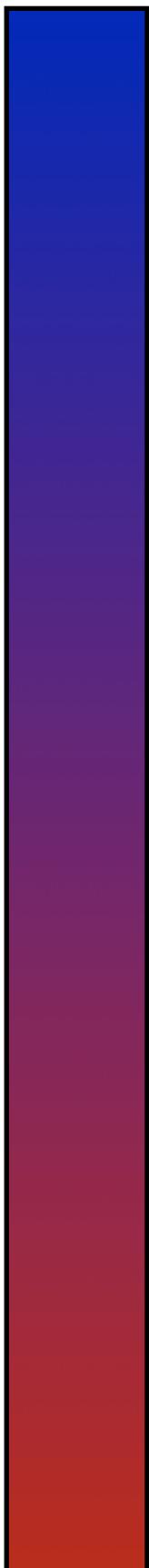
subversion (svn)

version control

concurrent version system (cvs)

perforce

visual source safe





SoS

mercurial (hg)

“not bad”

bazaar

subversion (svn)

version control

concurrent version system (cvs)

perforce

visual source safe



SoS

mercurial (hg)

bazaar

subversion (svn)

version control

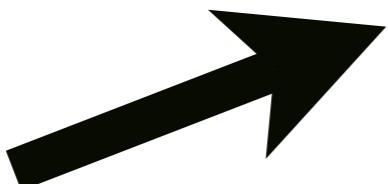
concurrent version system (cvs)

perforce

visual source safe “kill self”



git



mercurial (hg)

bazaar

subversion (svn)

version control

concurrent version system (cvs)

perforce

visual source safe

**Git is an open source,
distributed version control
system designed for speed
and efficiency**

git-scm.com

The screenshot shows the official website for Git (git-scm.com). The header features the word "git" in large letters with a small icon above it, followed by the tagline "the fast version control system". Below the header is a navigation bar with links for Home, About Git, Documentation, Download, Tools & Hosting, and Wiki. A cartoon character of a brown bear holding a branch is visible on the right side of the header.

Git is...

Git is an **open source, distributed version control system** designed to handle everything from small to very large projects with speed and efficiency.

Every Git clone is a full-fledged repository with complete history and full revision tracking capabilities, not dependent on network access or a central server.

Branching and merging are fast and easy to do.

Git is used for version control of files, much like tools such as [Mercurial](#), [Bazaar](#), [Subversion](#), [CVS](#), [Perforce](#), and [Visual SourceSafe](#).

Projects using Git

- [Git](#)
- [Linux Kernel](#)
- [Perl](#)
- [Ruby on Rails](#)
- [Android](#)
- [WINE](#)
- [Fedora](#)
- [X.org](#)
- [VLC](#)
- [Prototype](#)

Download Git

The latest stable Git release is **v1.6.1.1**

[Release notes \(2009-01-25\)](#)

 [tar.bz2 \(sign\)](#)

 [tar.gz \(sign\)](#)

[Other Download Options](#)

[Source and History](#)

Git Quick Start

Git is an open source,
distributed version control
system designed for speed
and efficiency

Fully Distributed

(almost) everything is local

which means

which means

everything is fast

which means

everything is fast

every clone is a backup

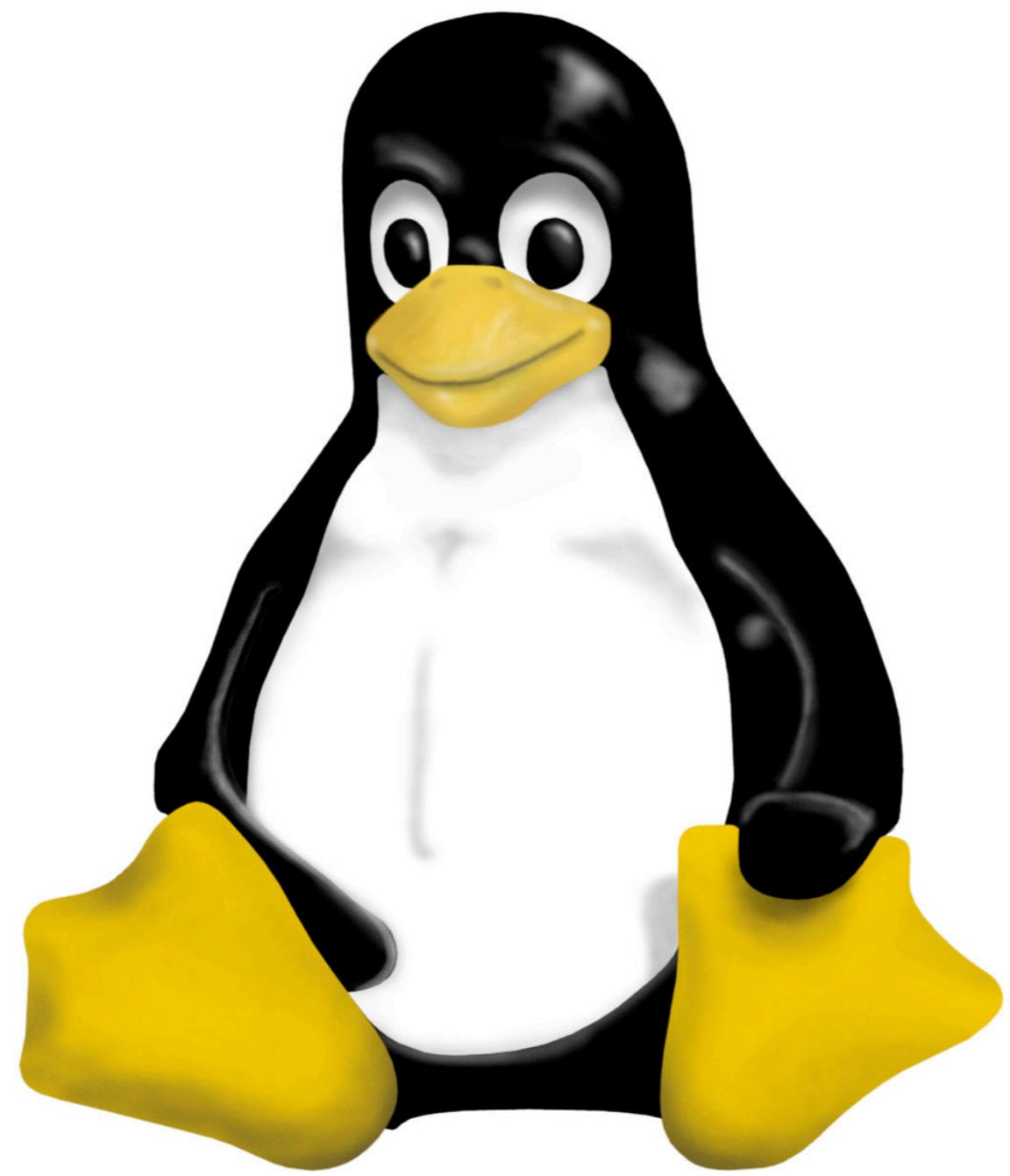
which means

everything is fast

every clone is a backup

work offline

**Git is an open source,
distributed version control
system designed for
speed and efficiency**

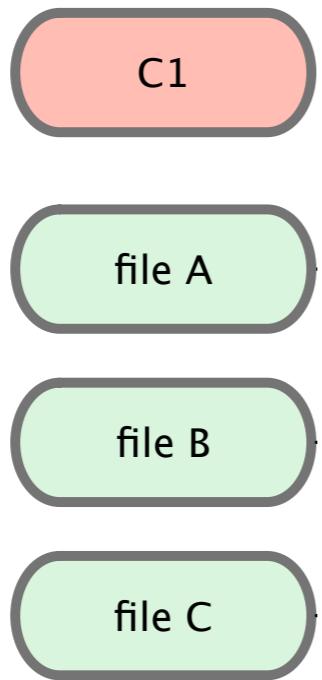
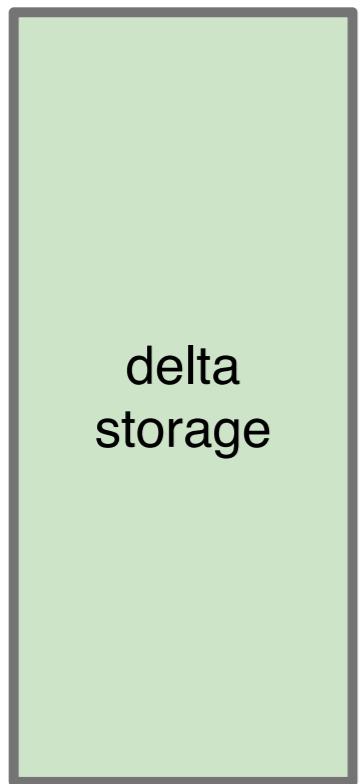


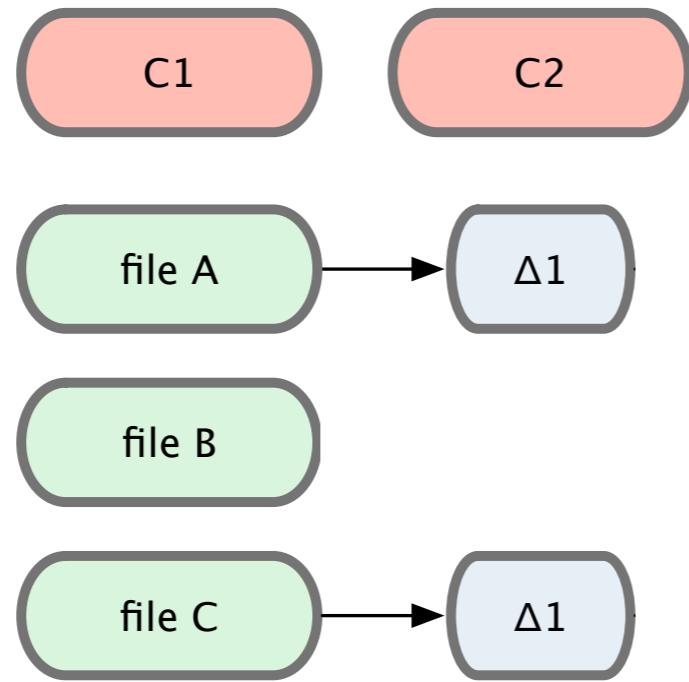
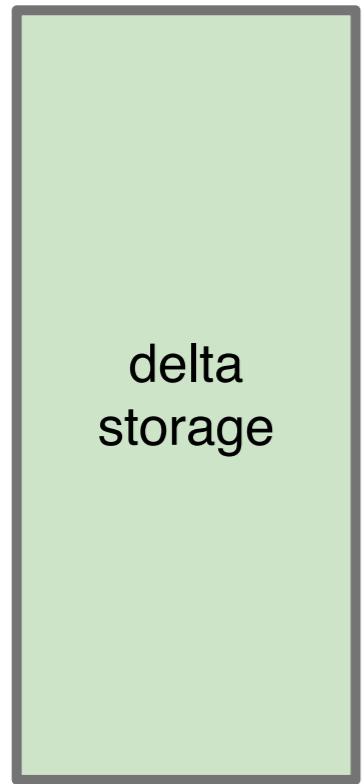
Immutable

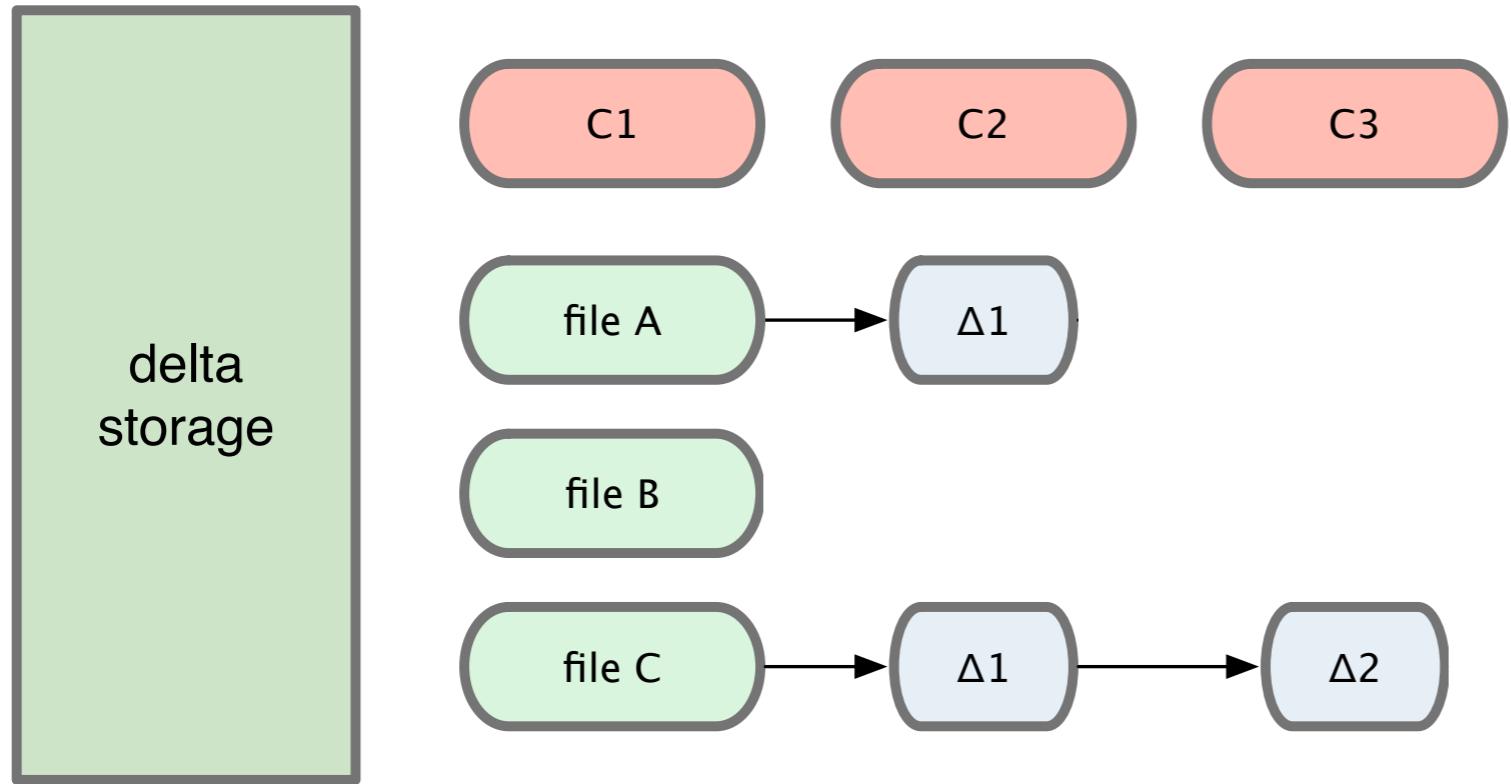
(almost) never removes data

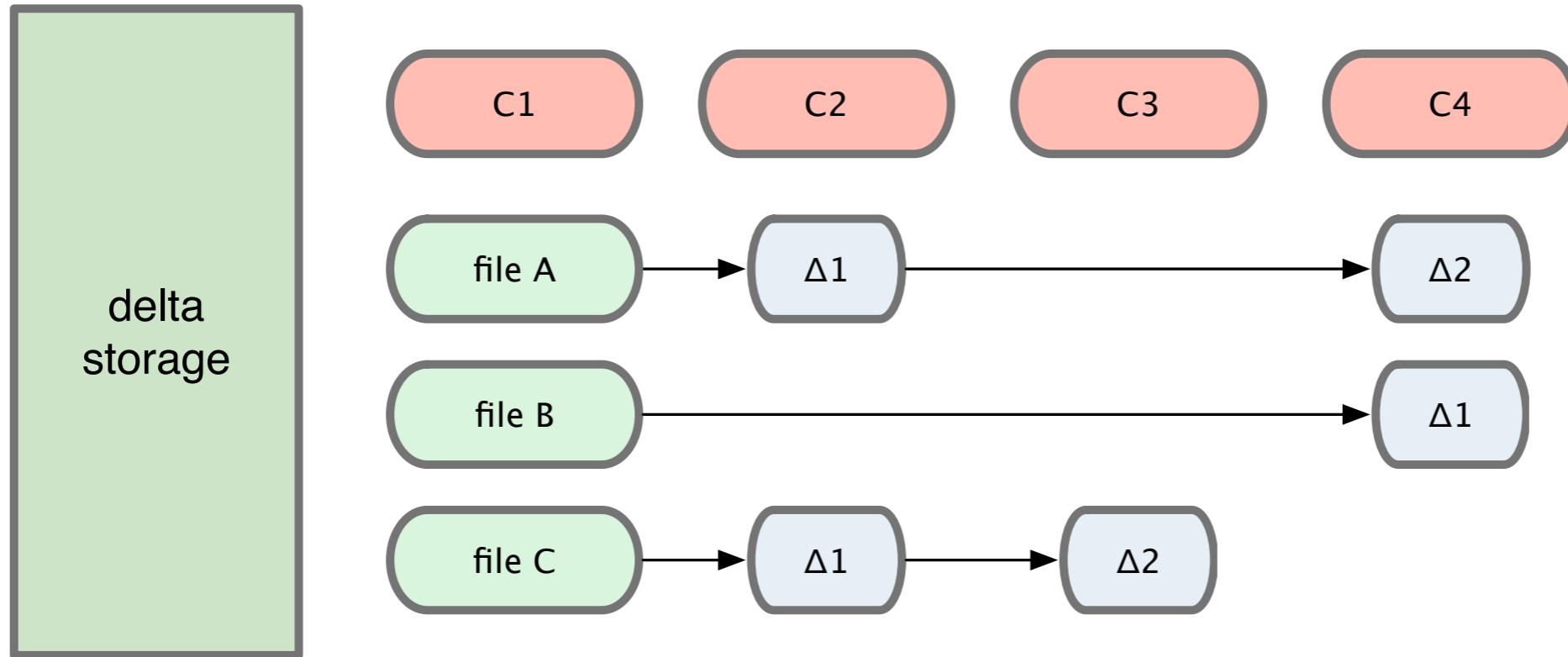
Snapshots, not Patches

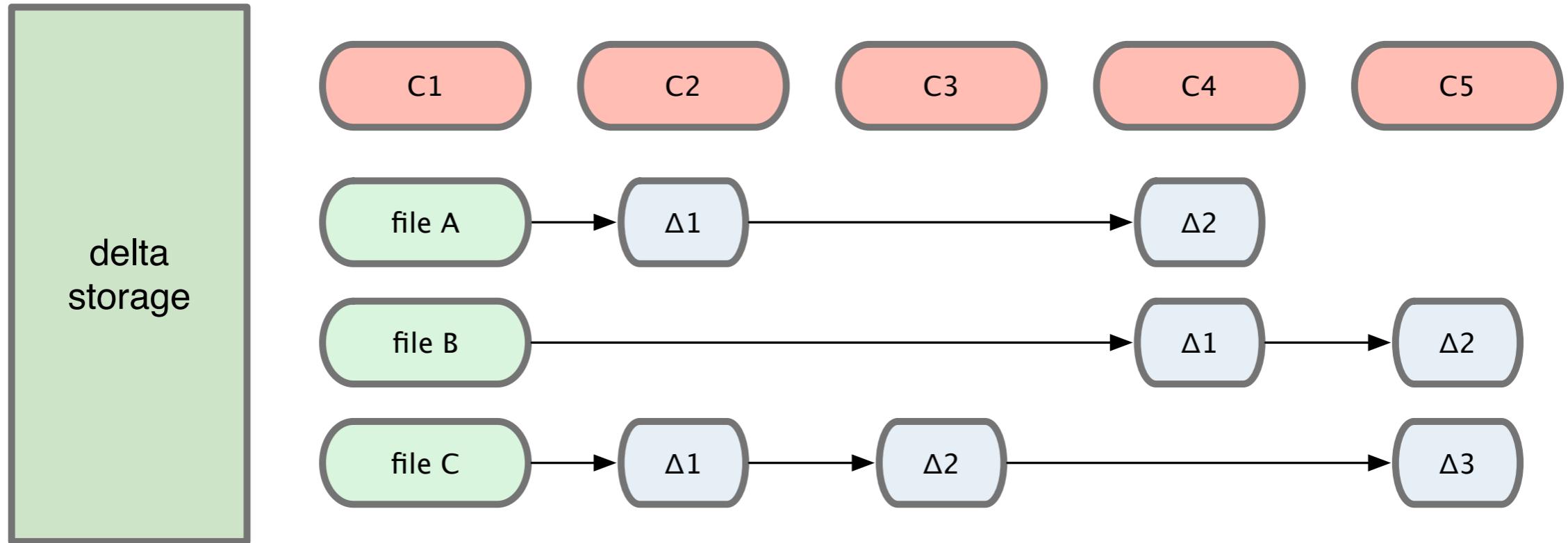
delta
storage

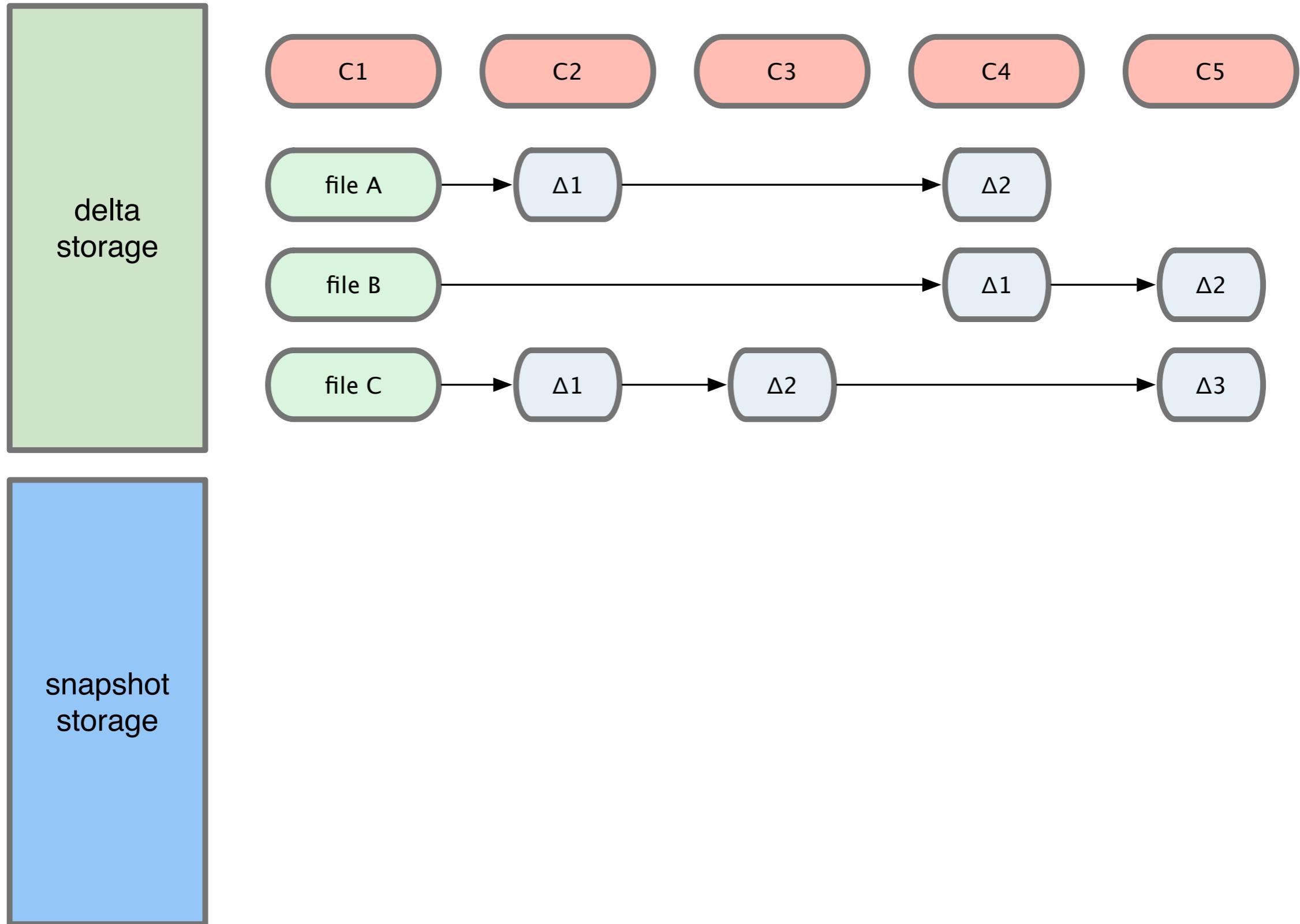


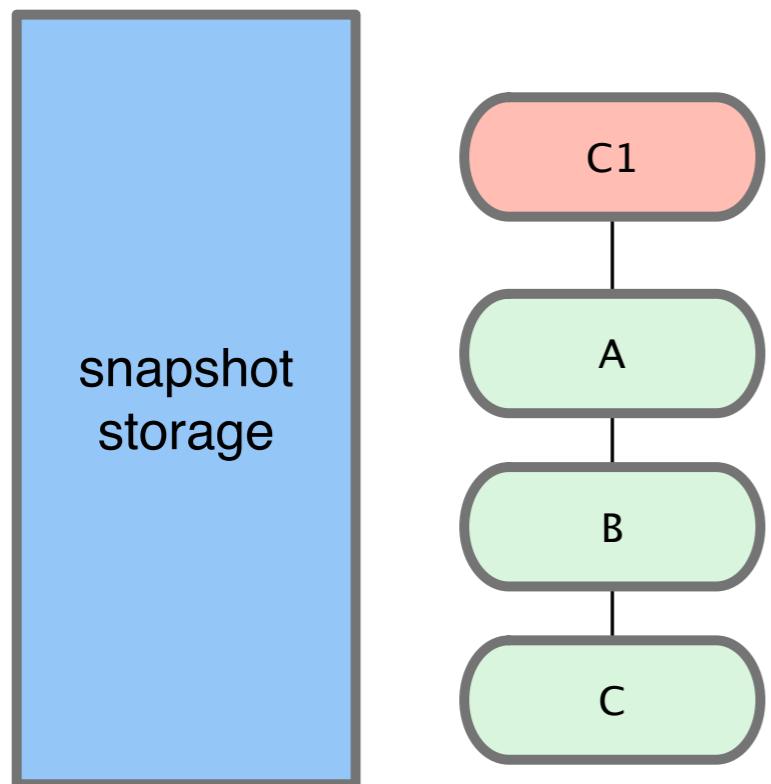
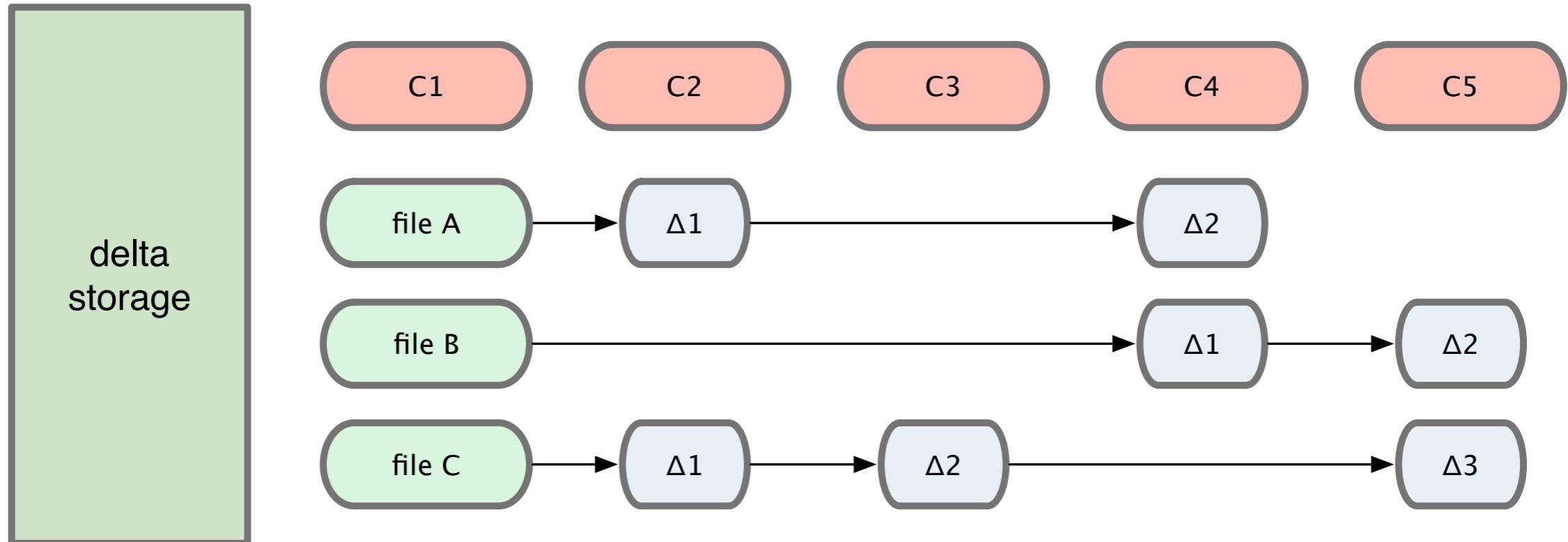


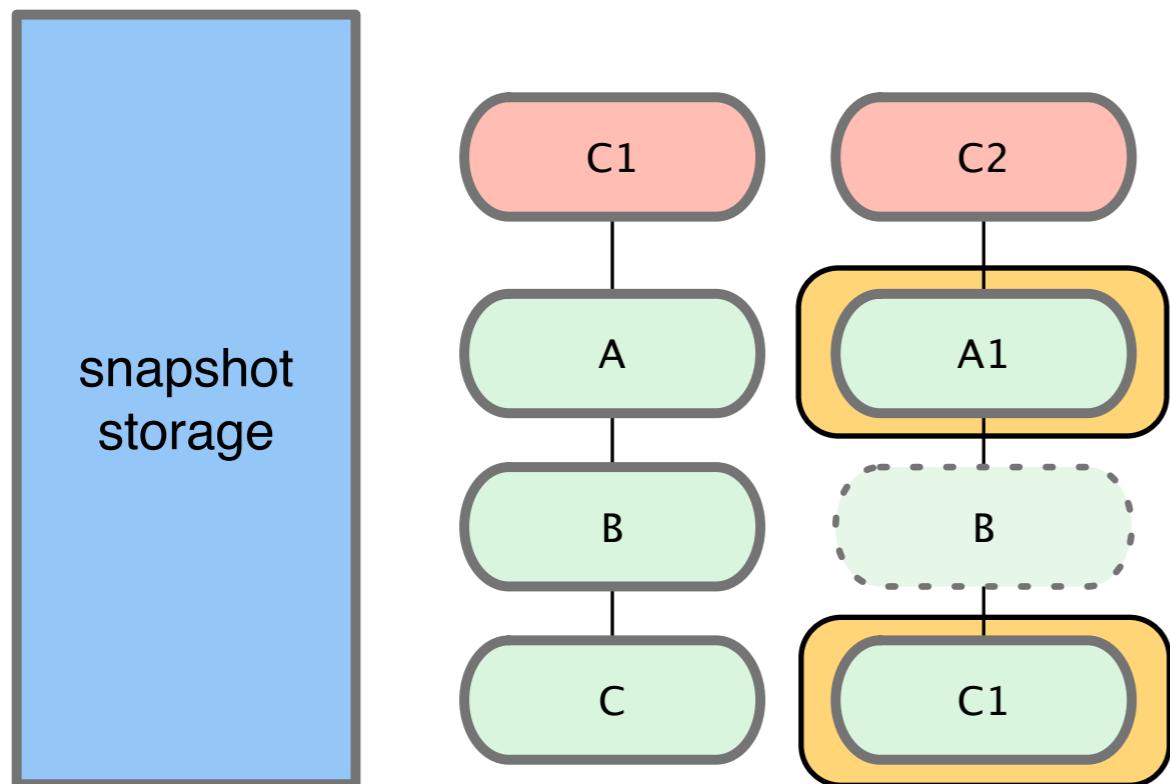
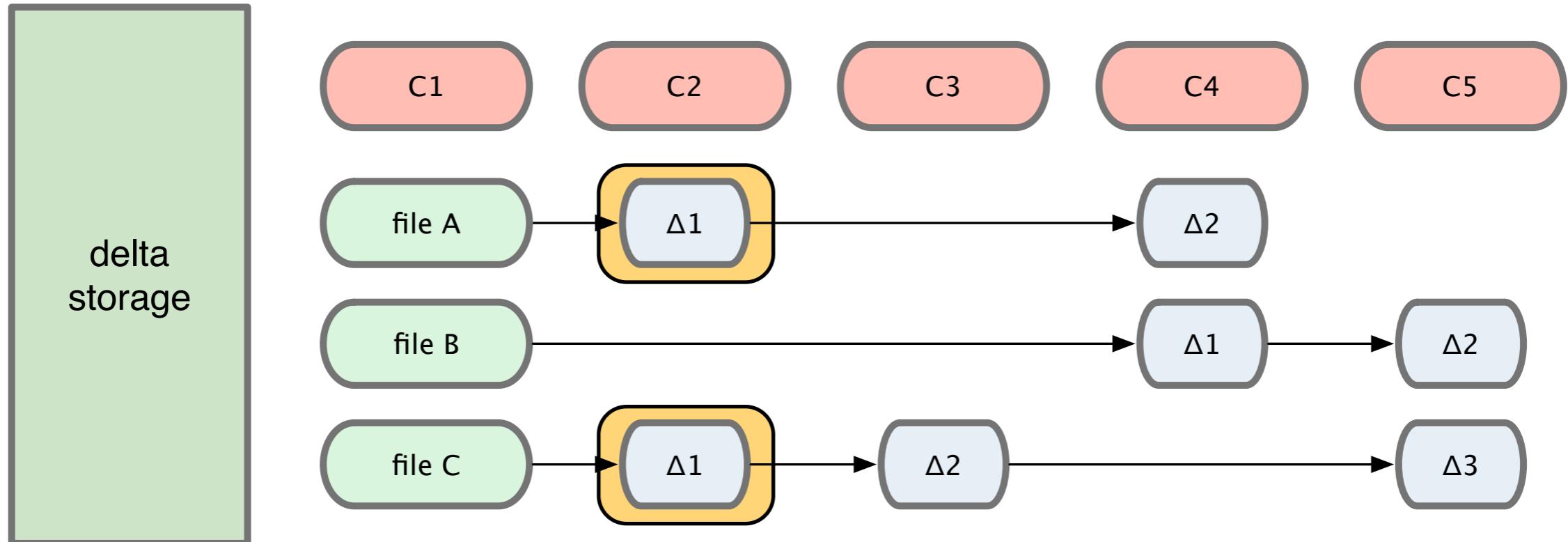


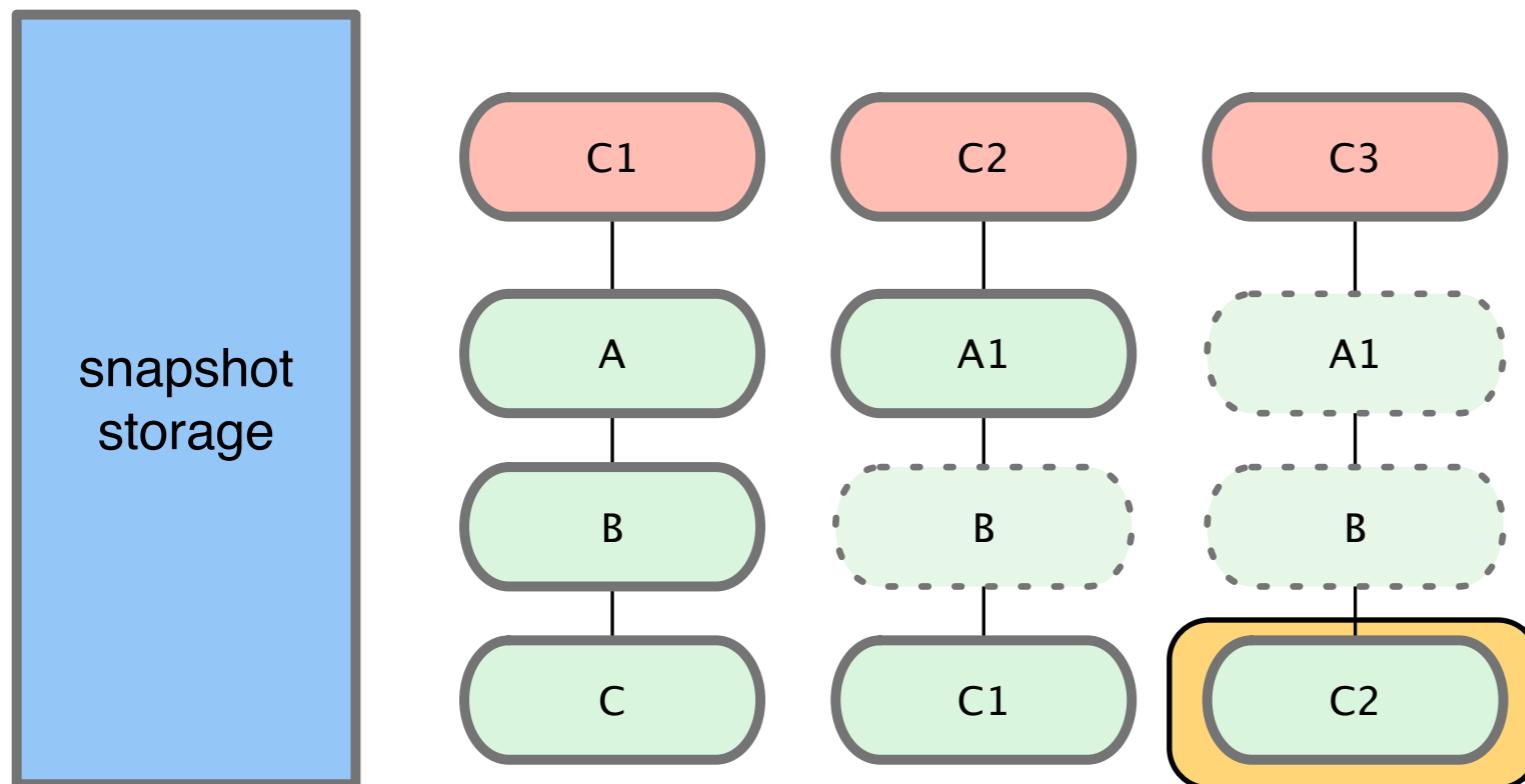
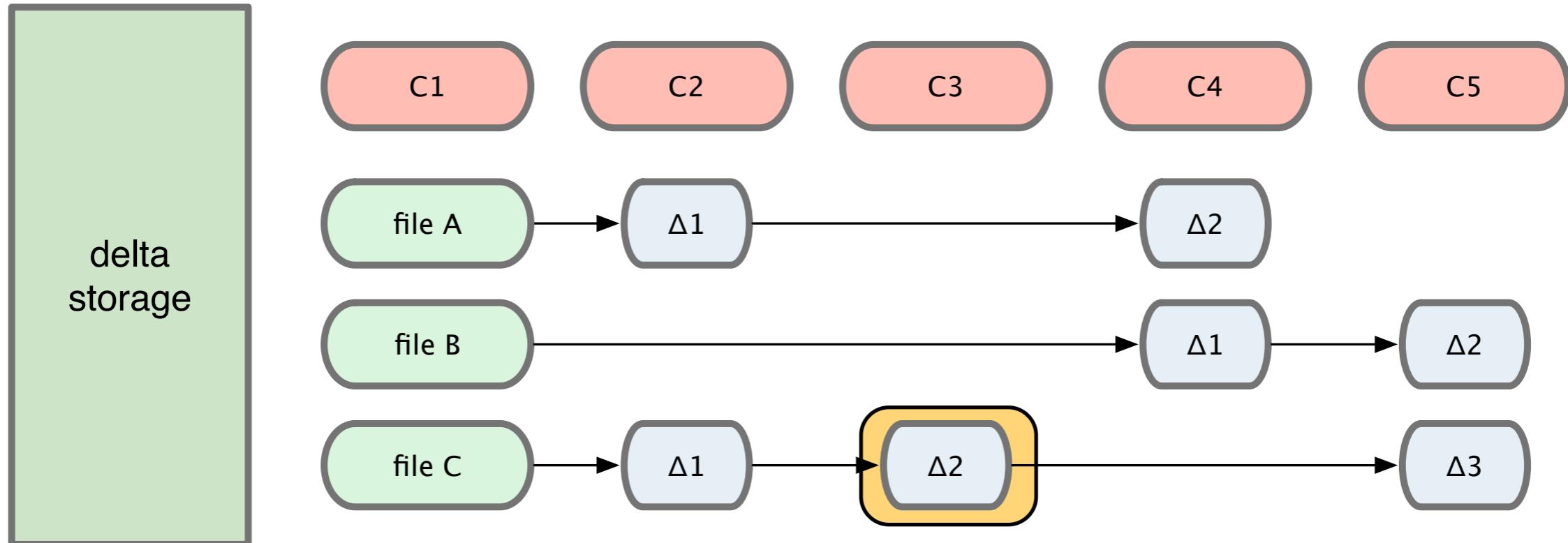


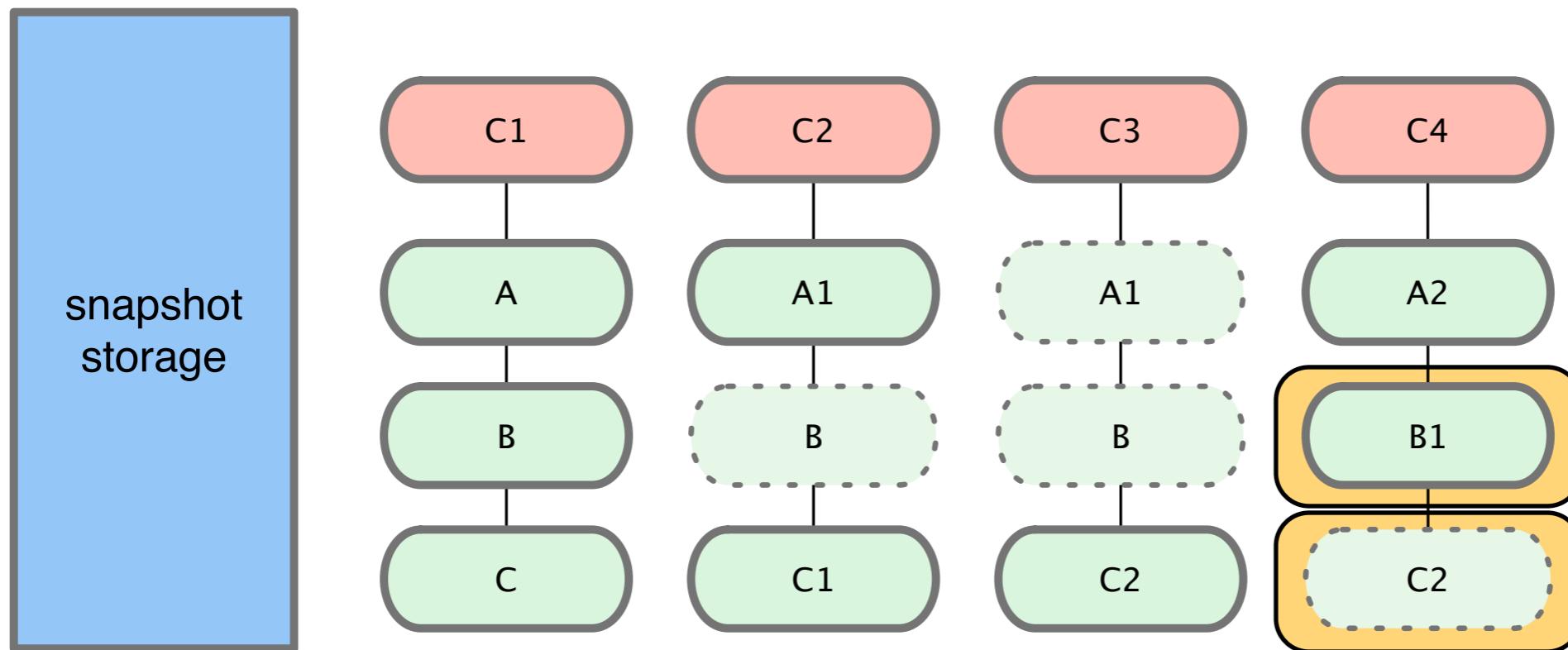
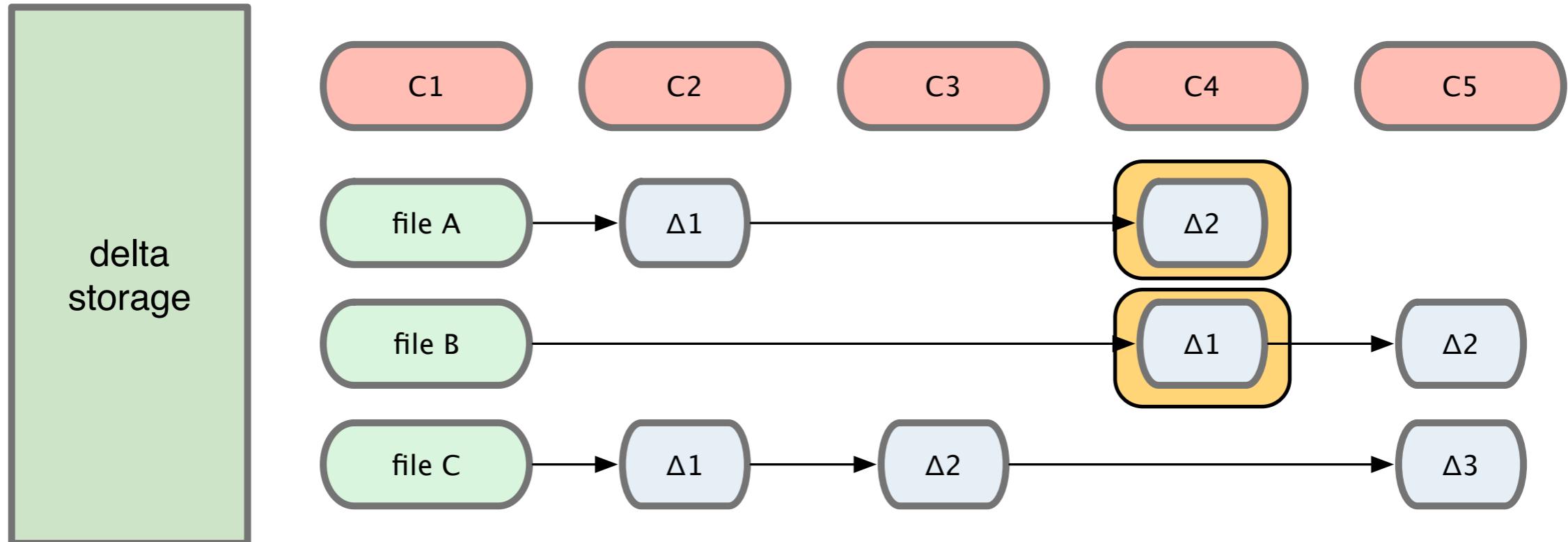


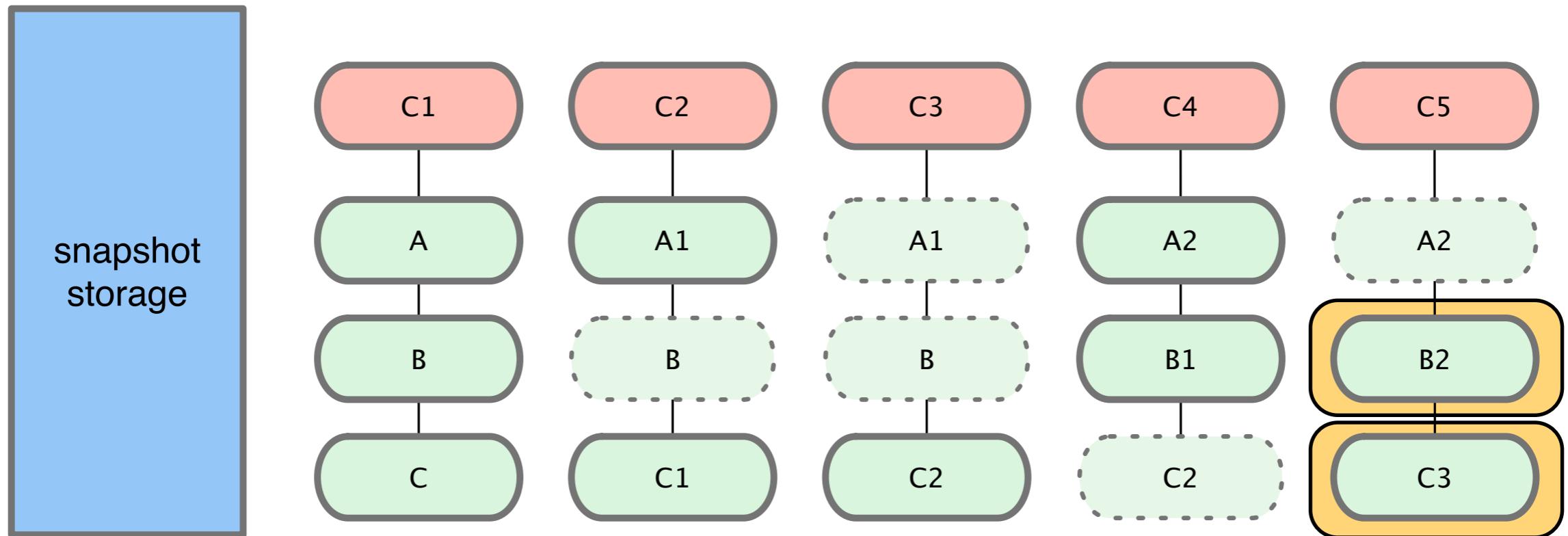
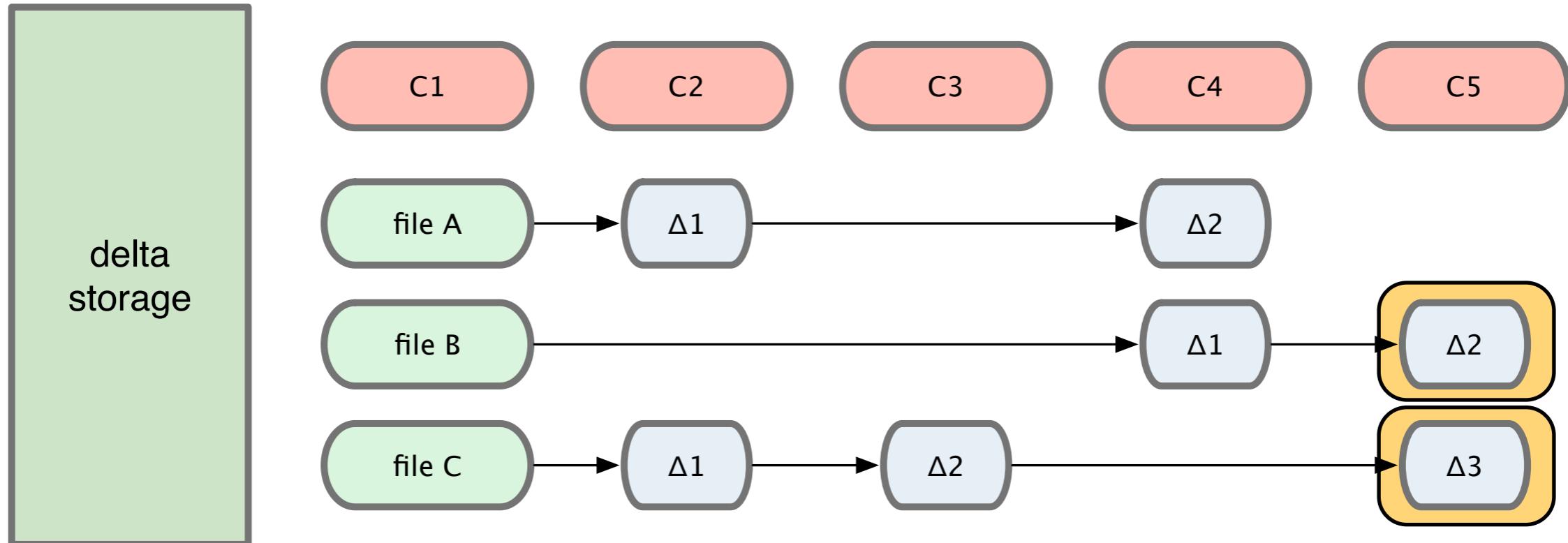


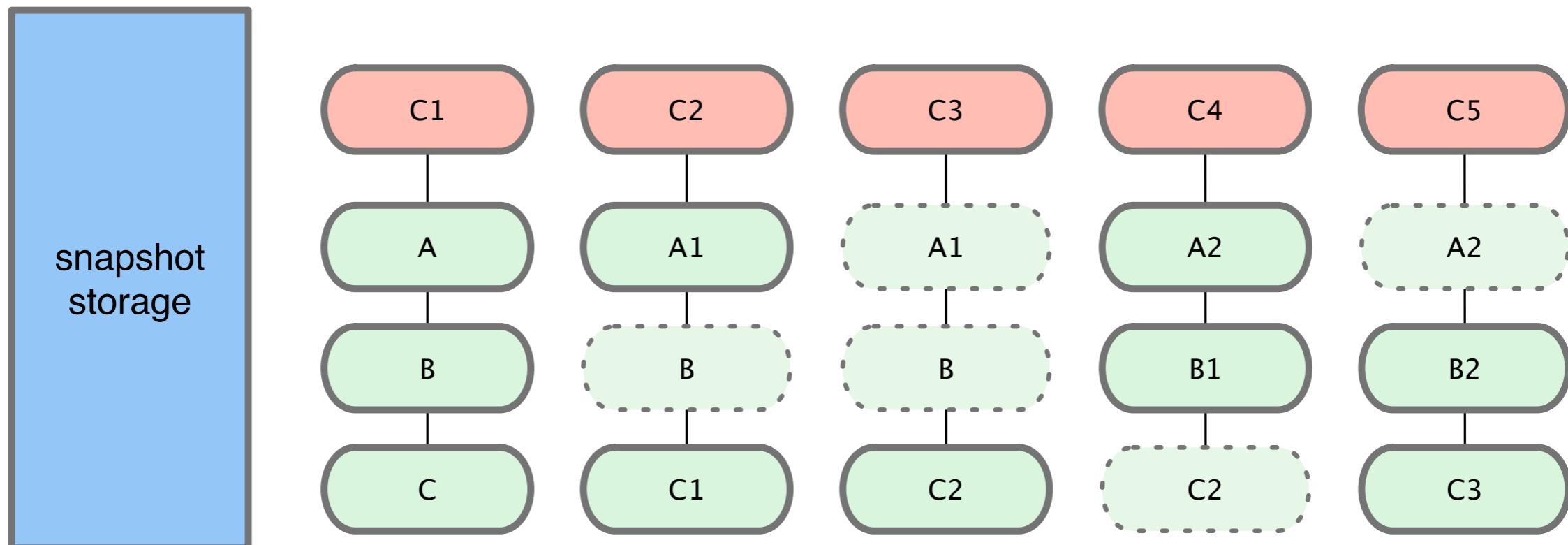
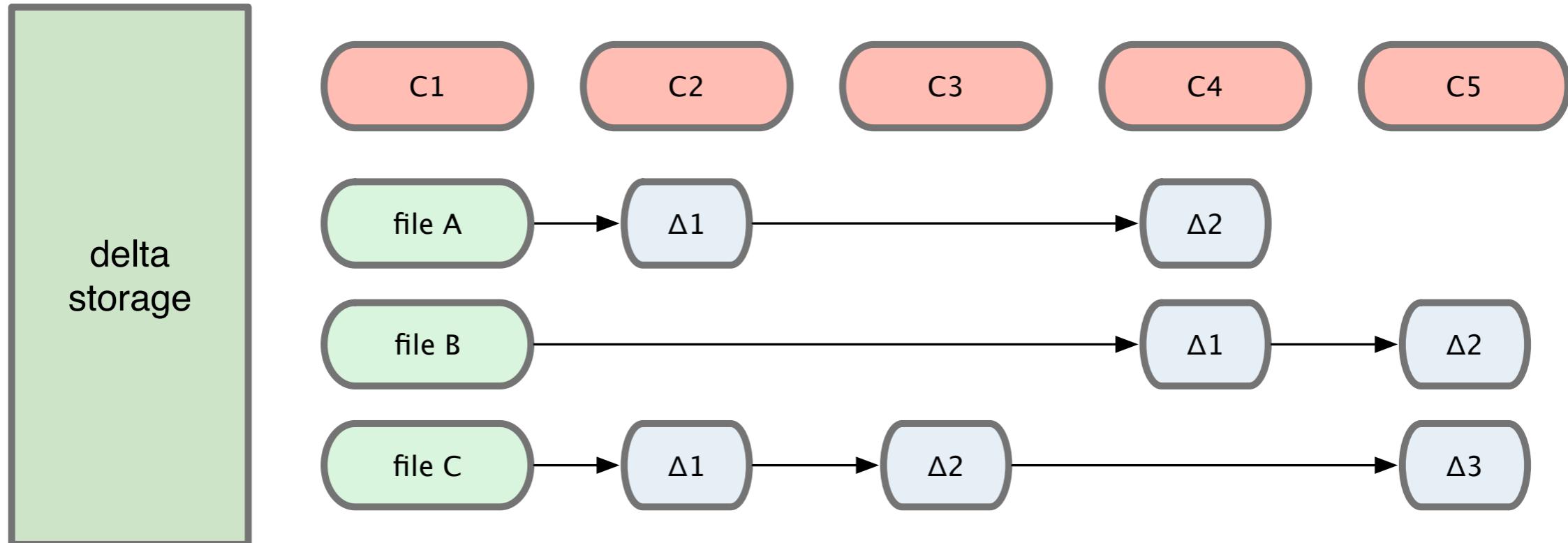












Why Git?

git ist schnell!

No Network Needed

No Network Needed

Performing a diff

No Network Needed

Performing a diff

Viewing file history

No Network Needed

Performing a diff

Viewing file history

Committing changes

No Network Needed

Performing a diff

Viewing file history

Committing changes

Merging branches

No Network Needed

Performing a diff

Viewing file history

Committing changes

Merging branches

Obtaining any other revision of a file

No Network Needed

Performing a diff

Viewing file history

Committing changes

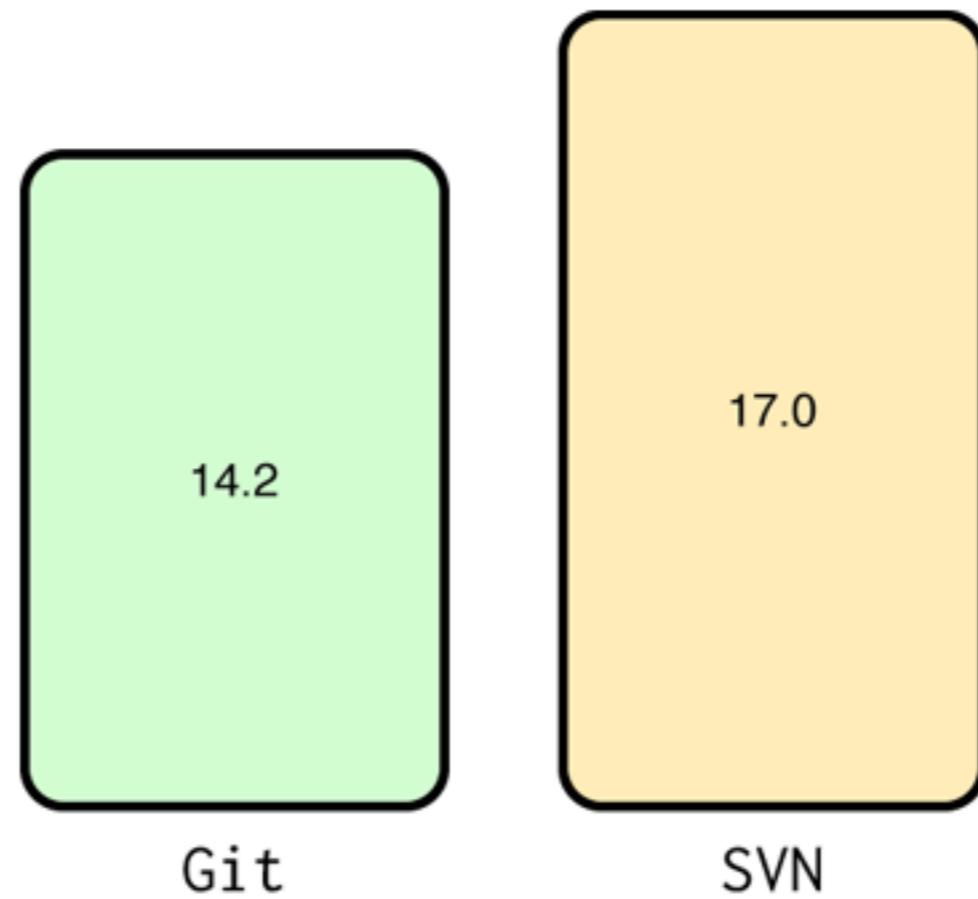
Merging branches

Obtaining any other revision of a file

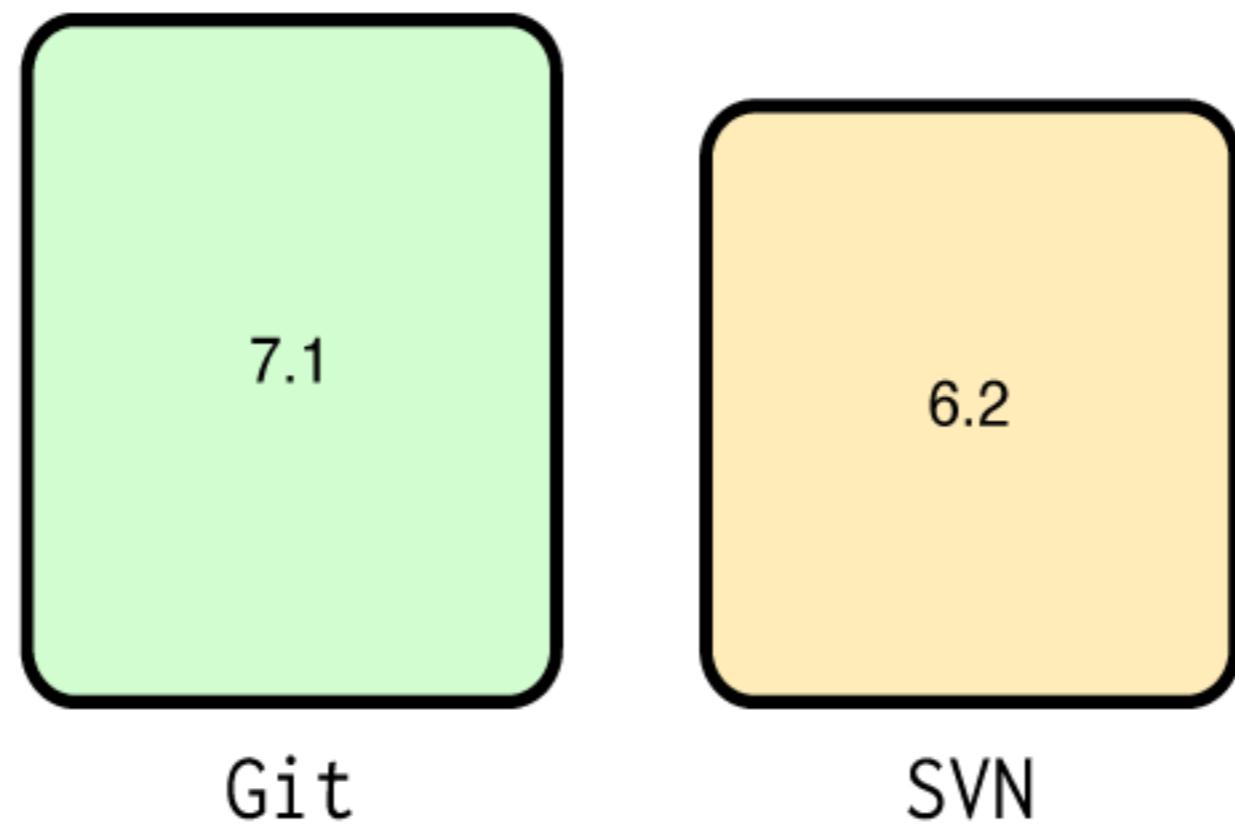
Switching branches

jQuery tests

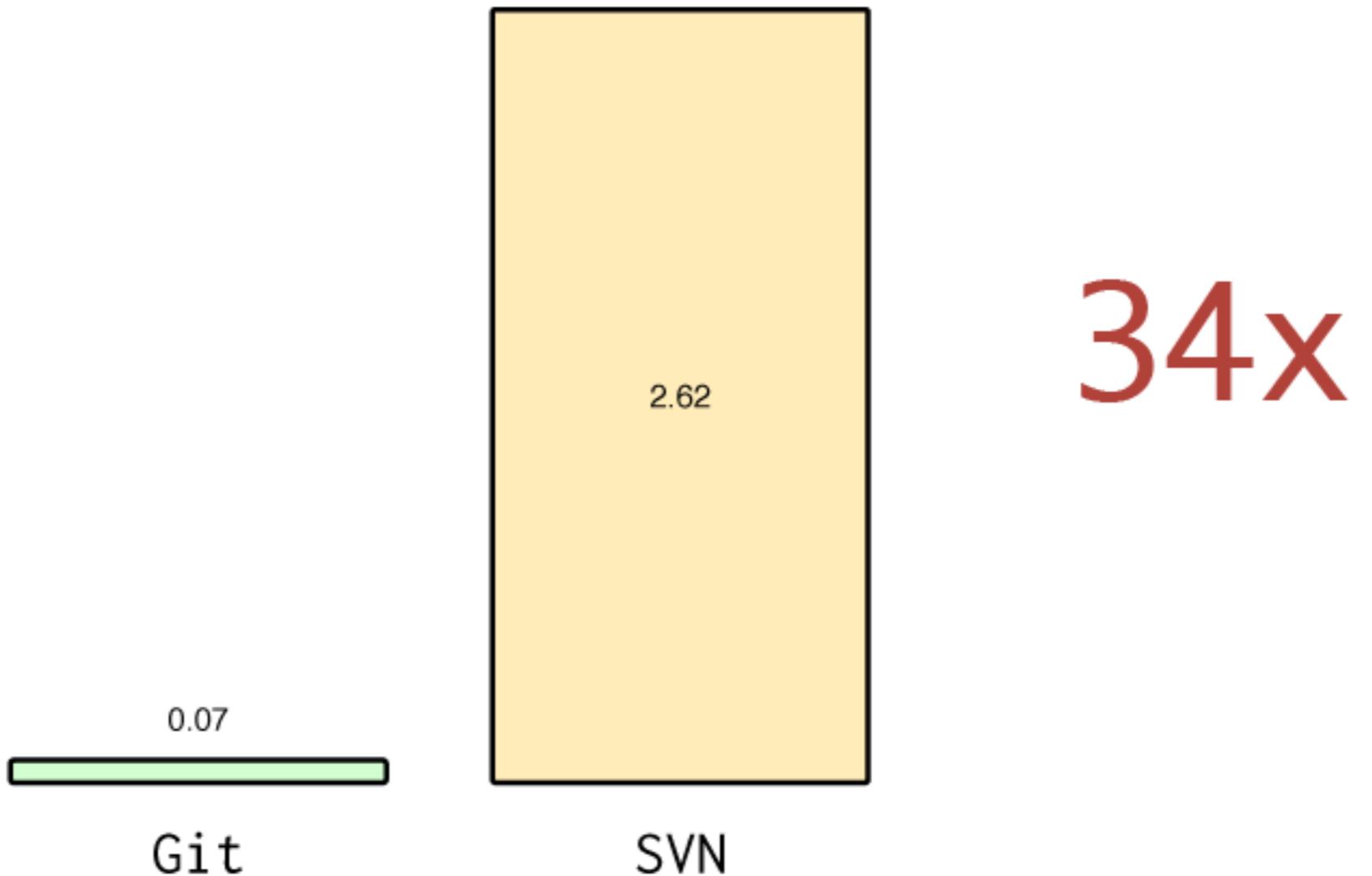
Clone/Checkout



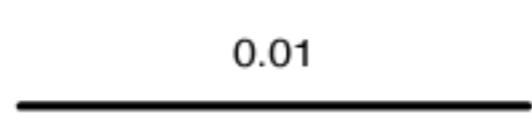
Size



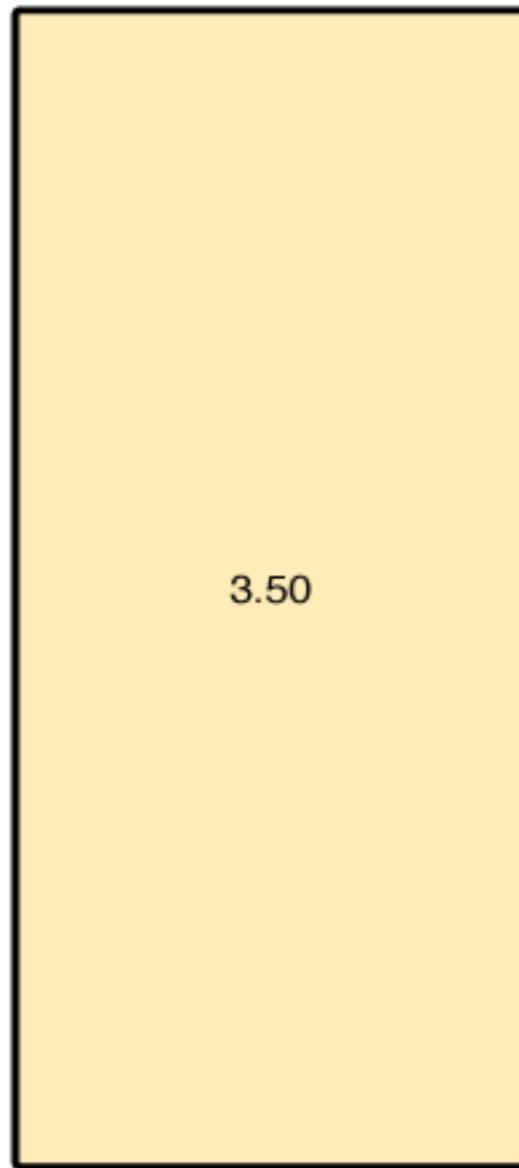
Log



Diff



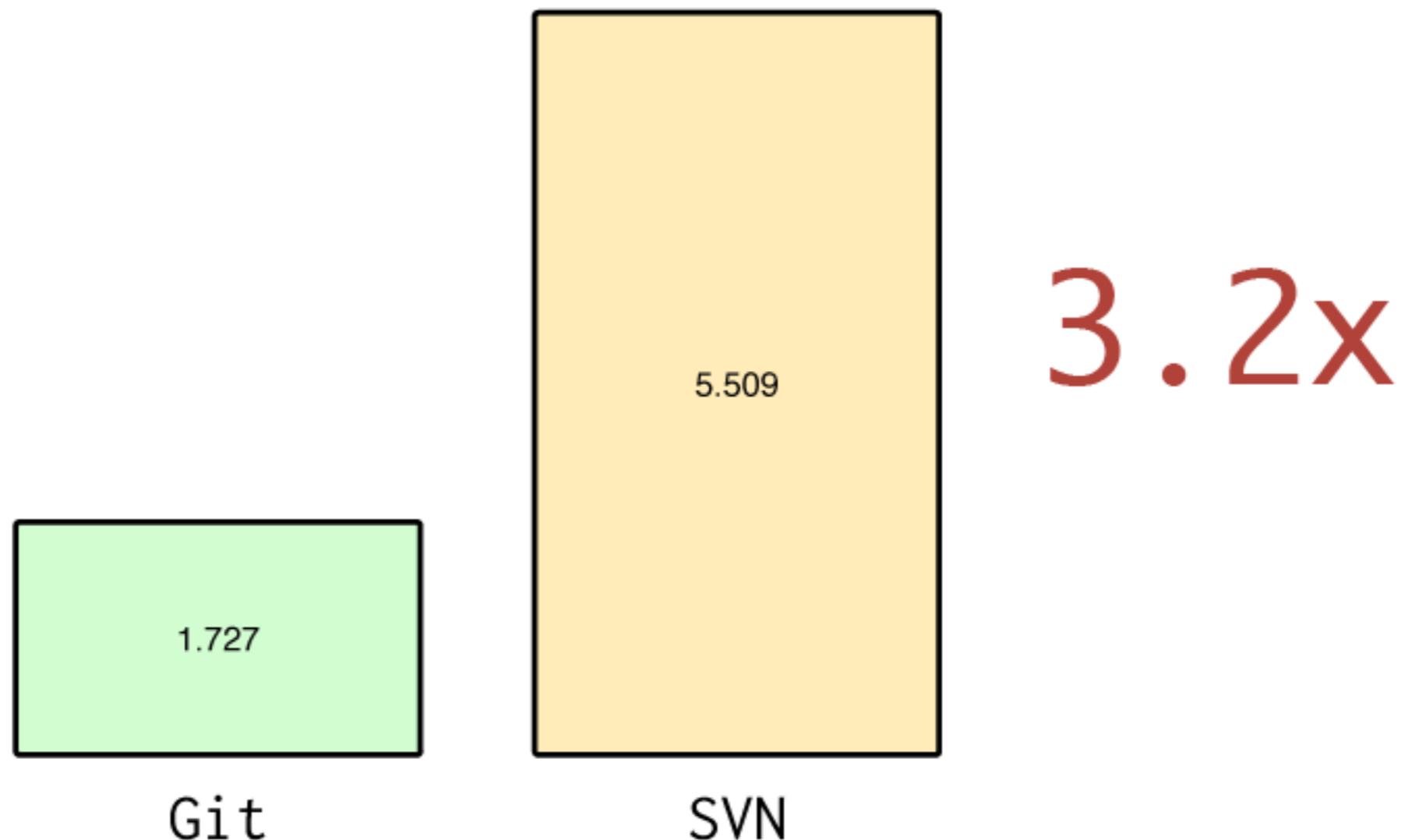
Git



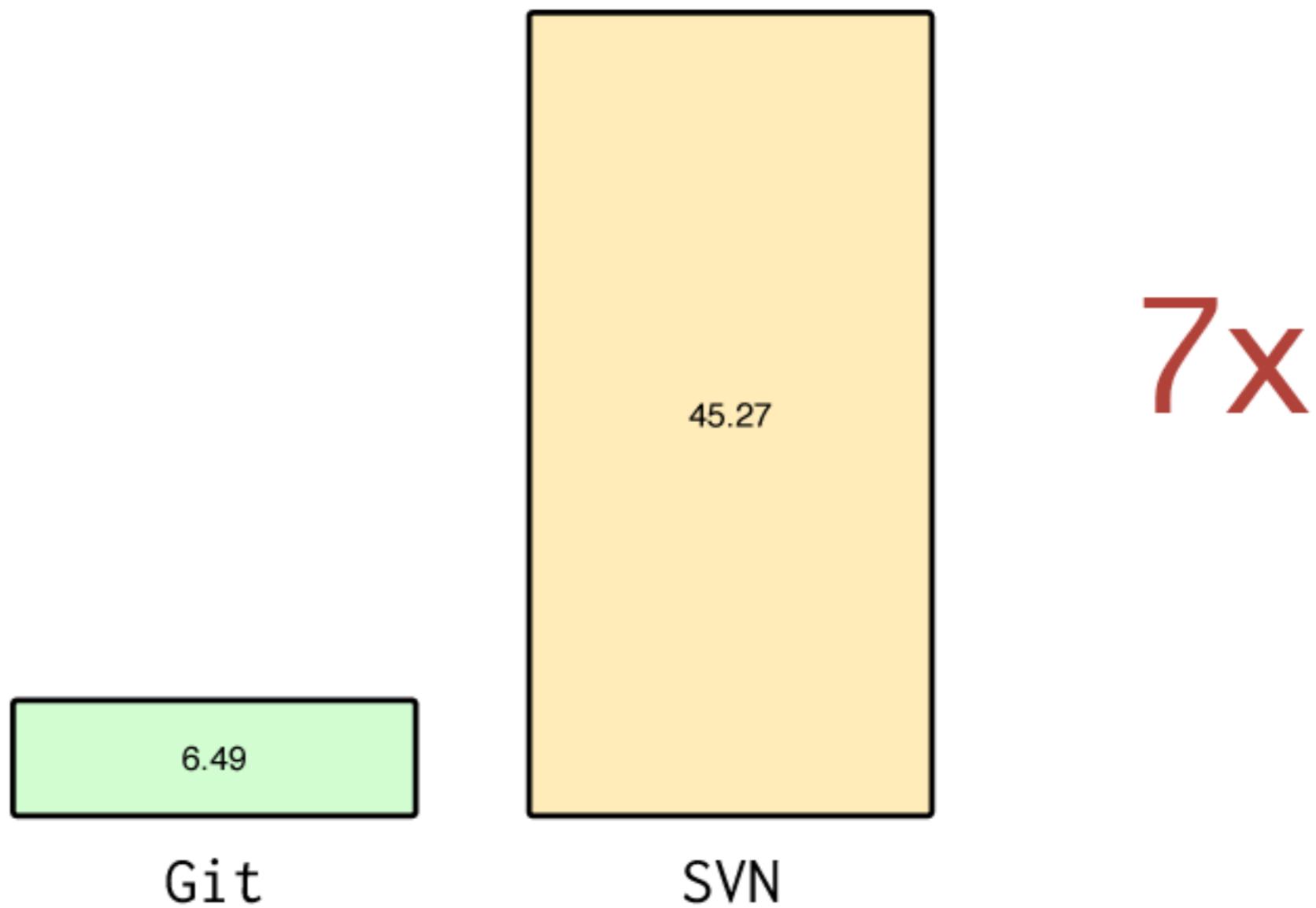
SVN

318x

Commit

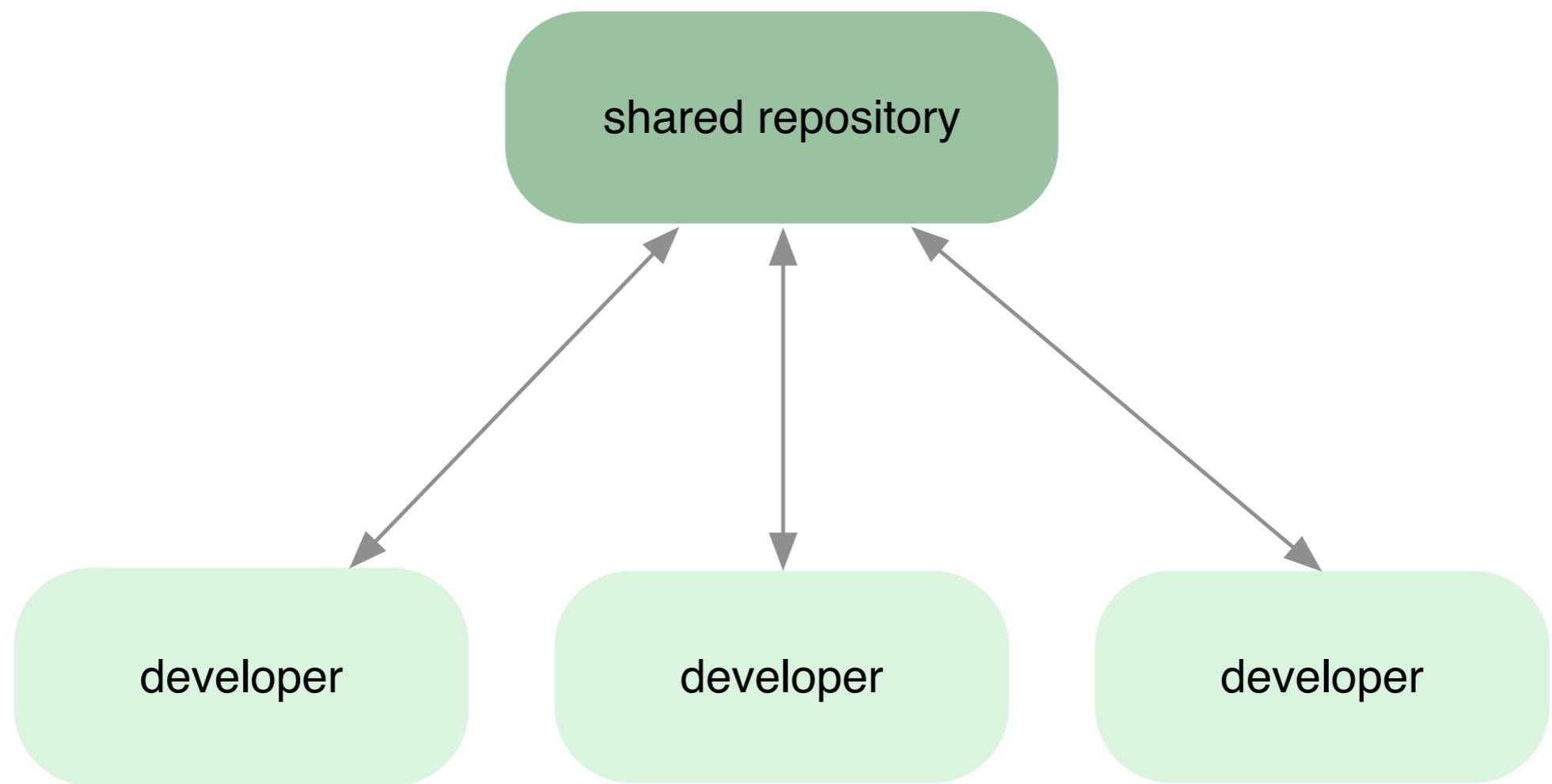


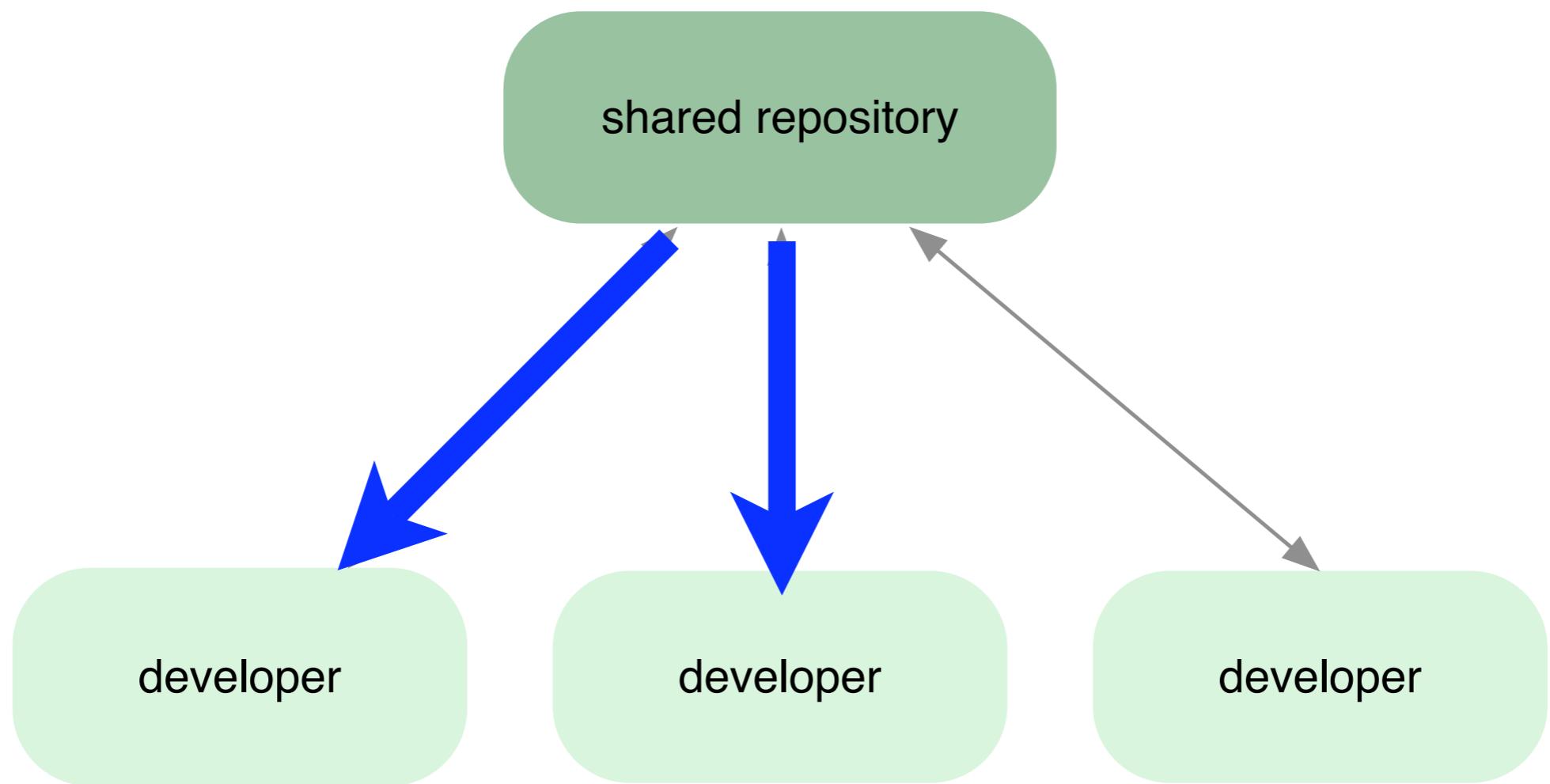
Substantial Commit

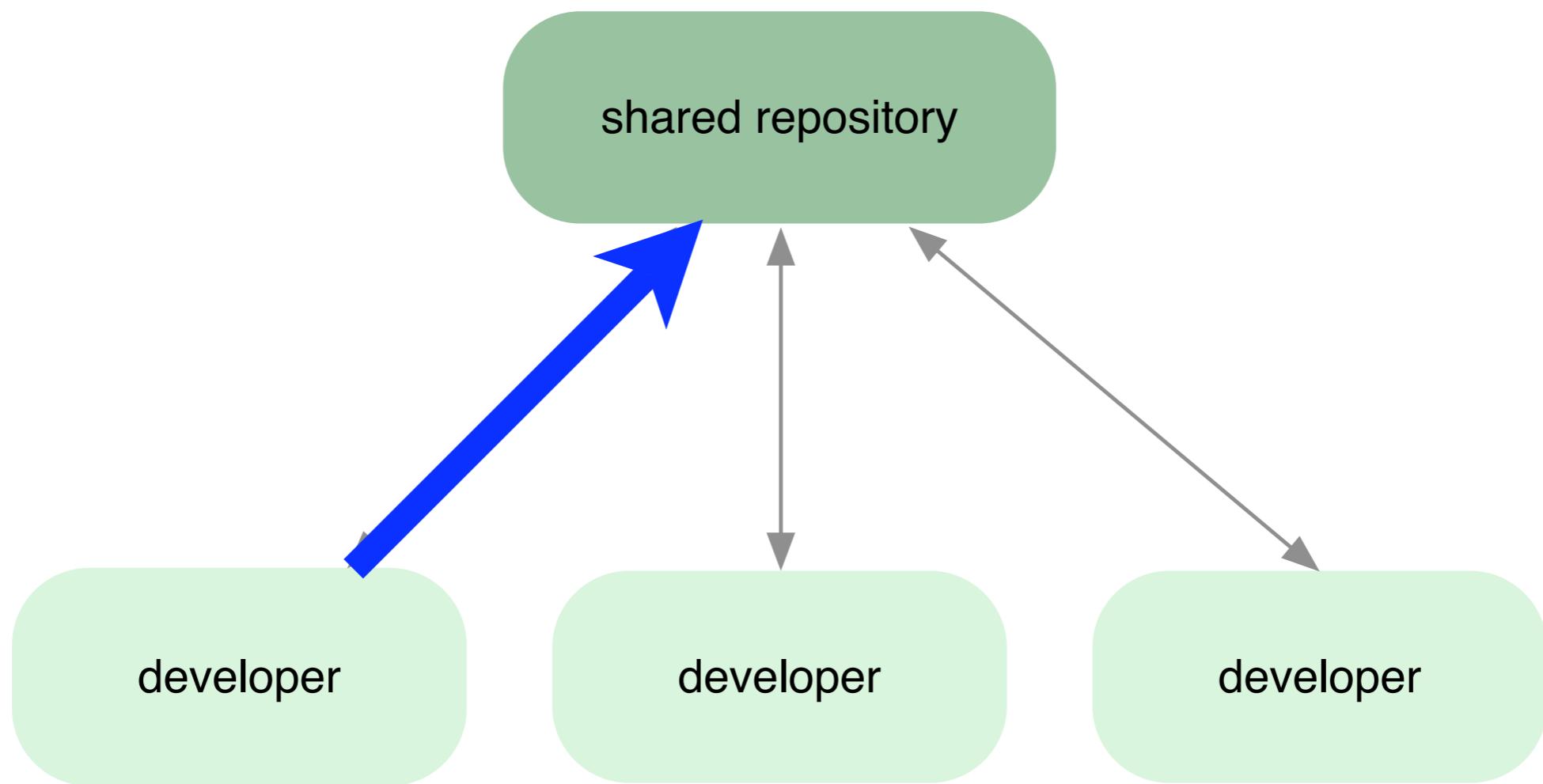


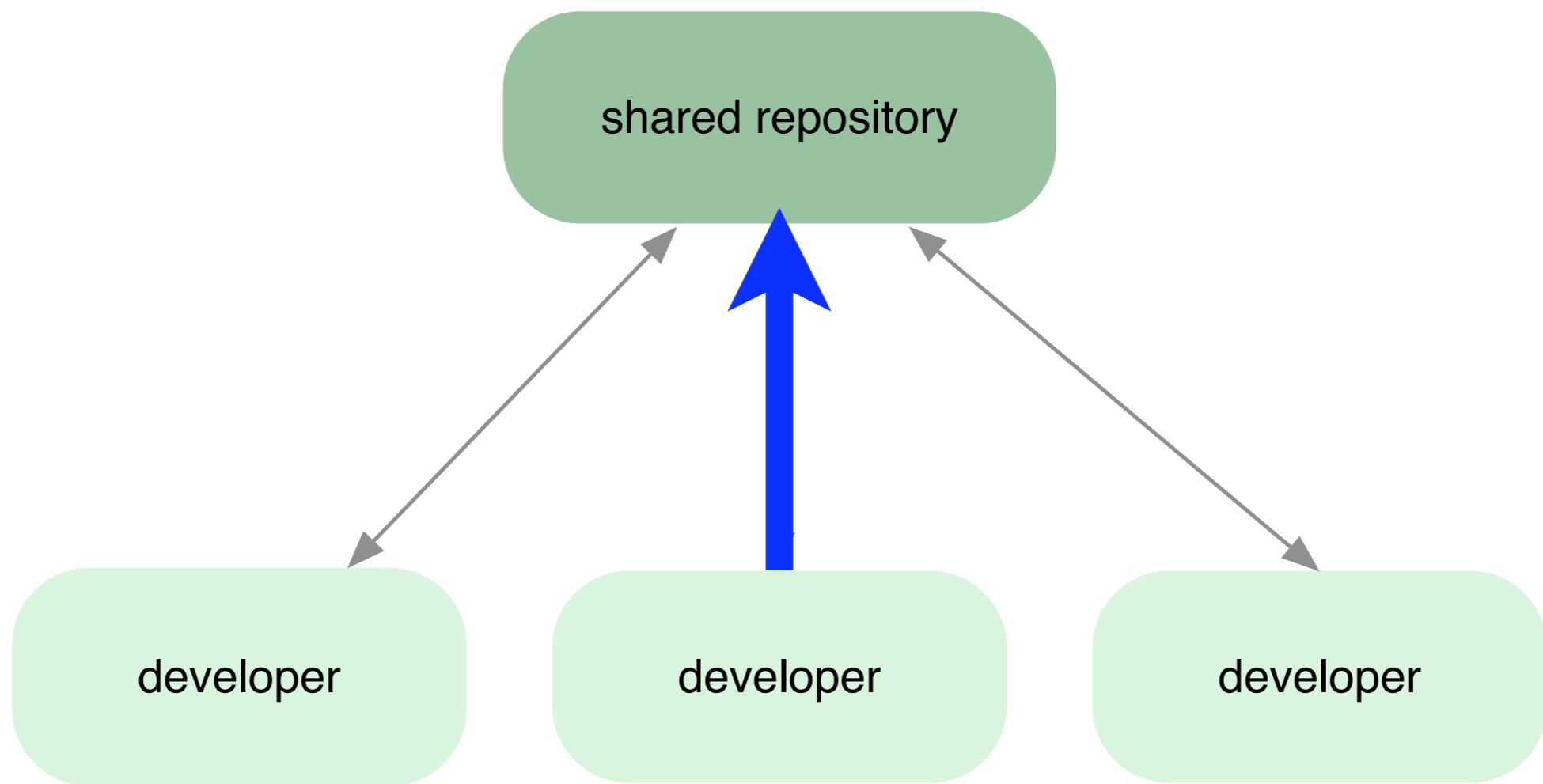
more workflows

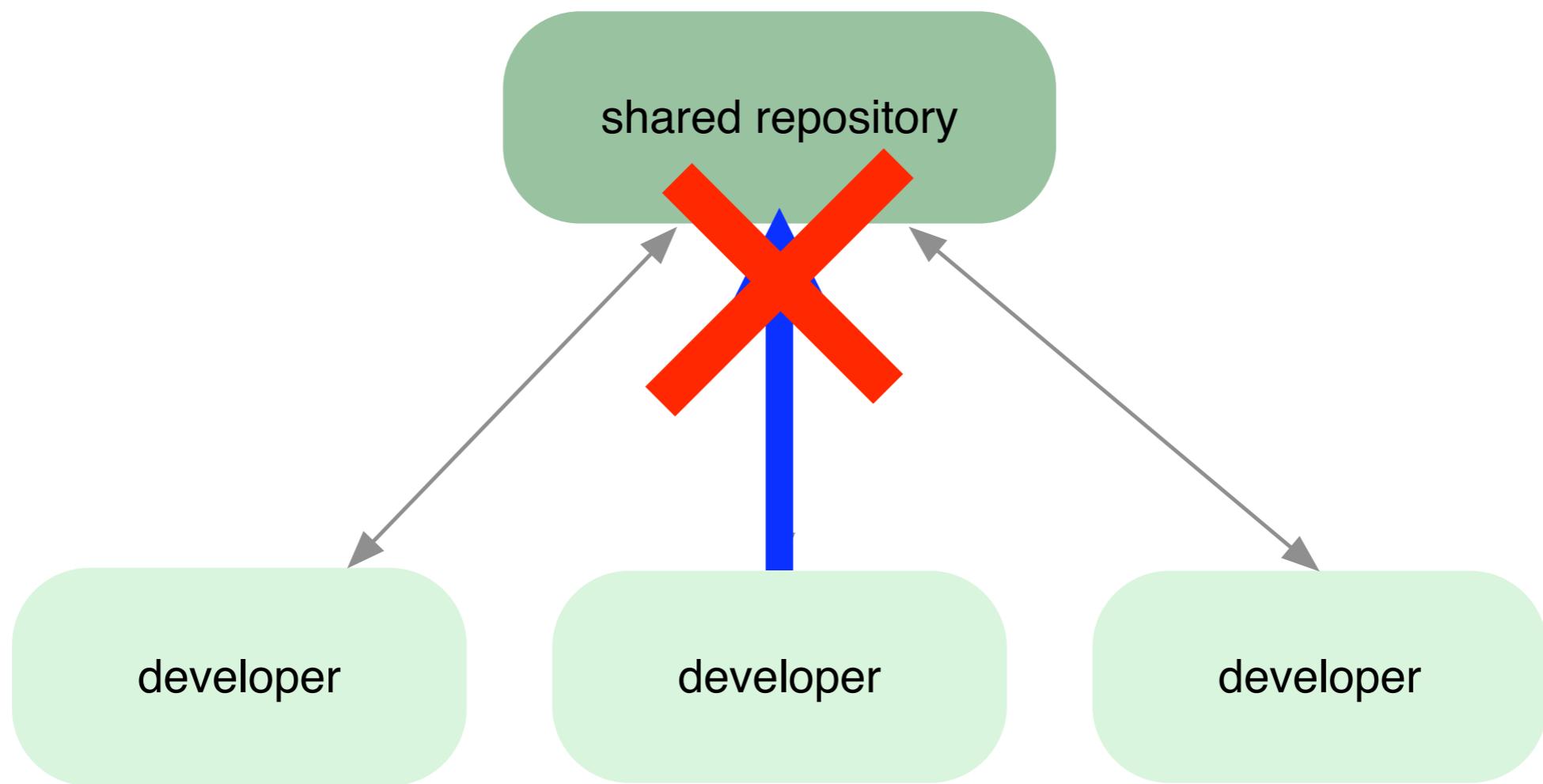
centralized

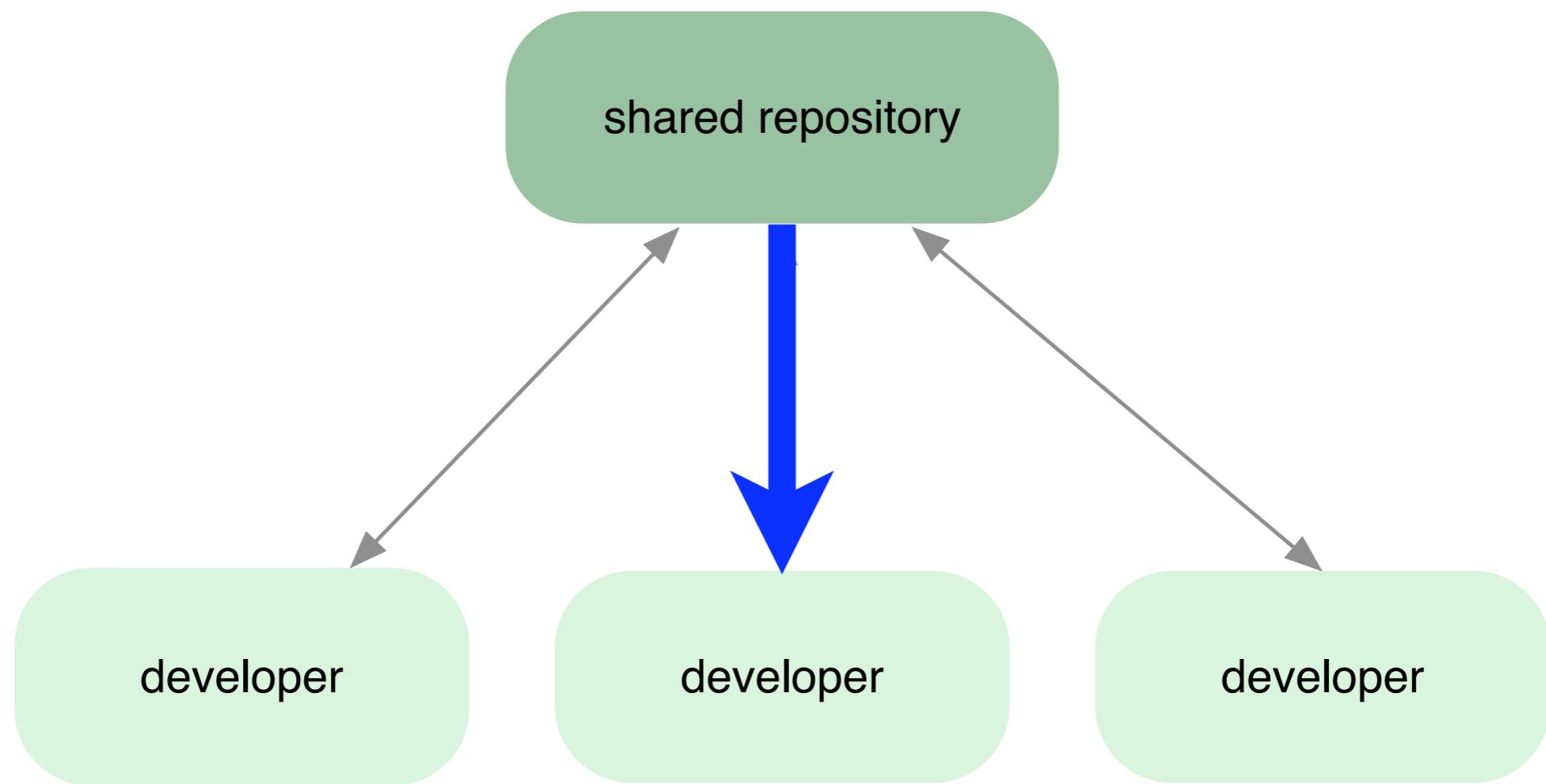


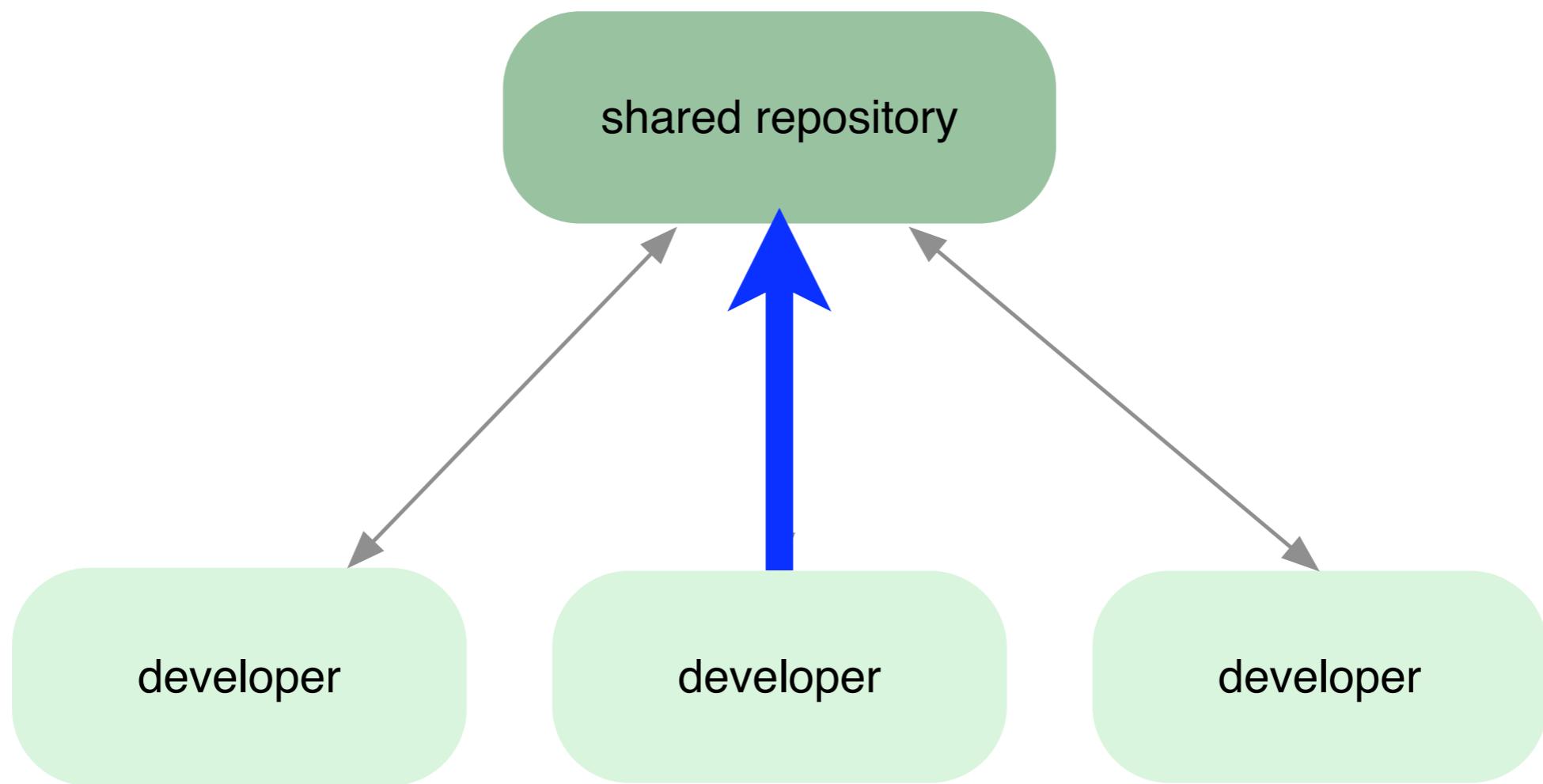




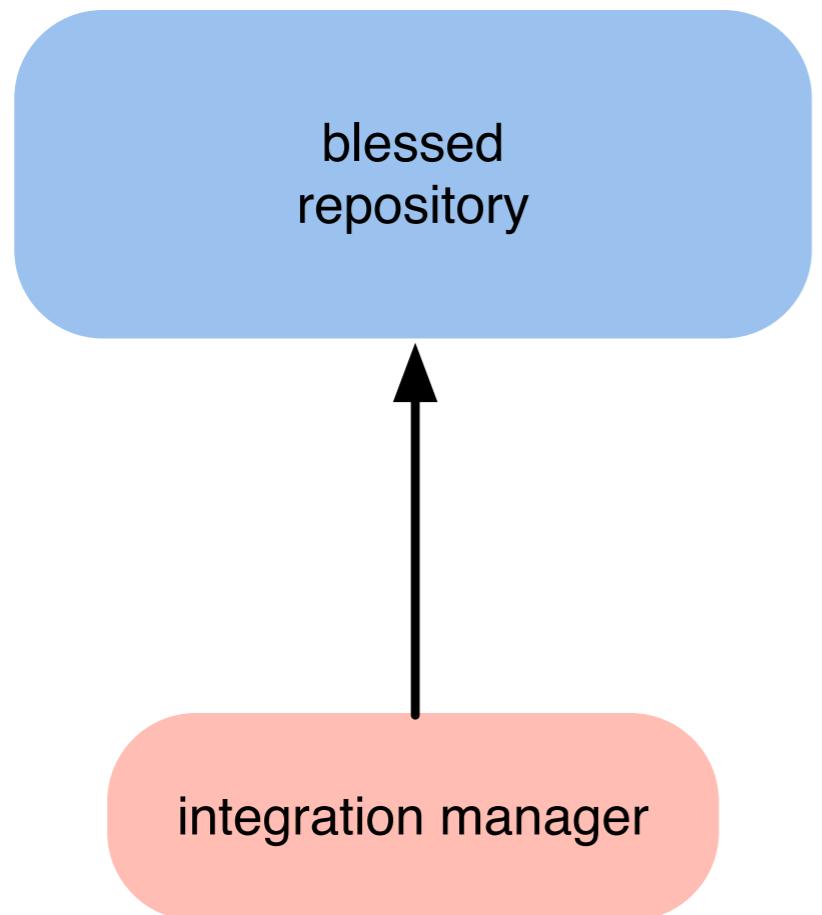


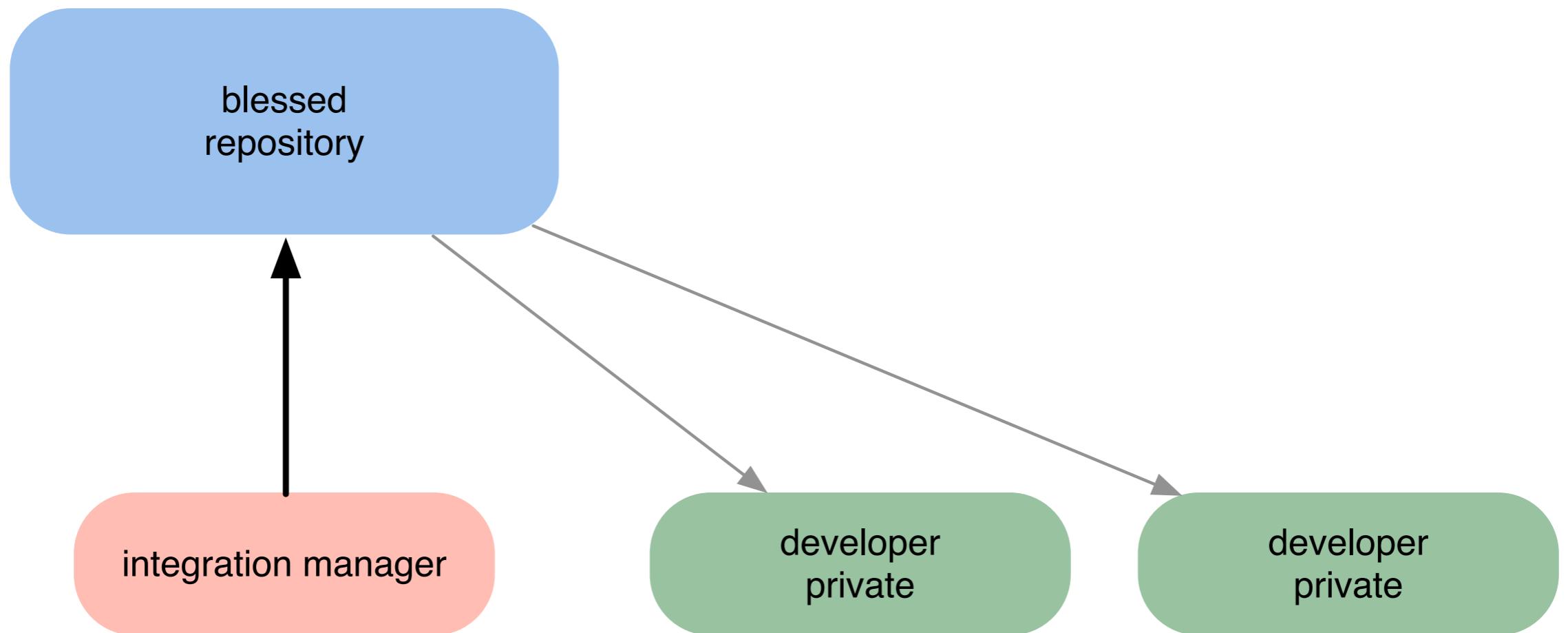


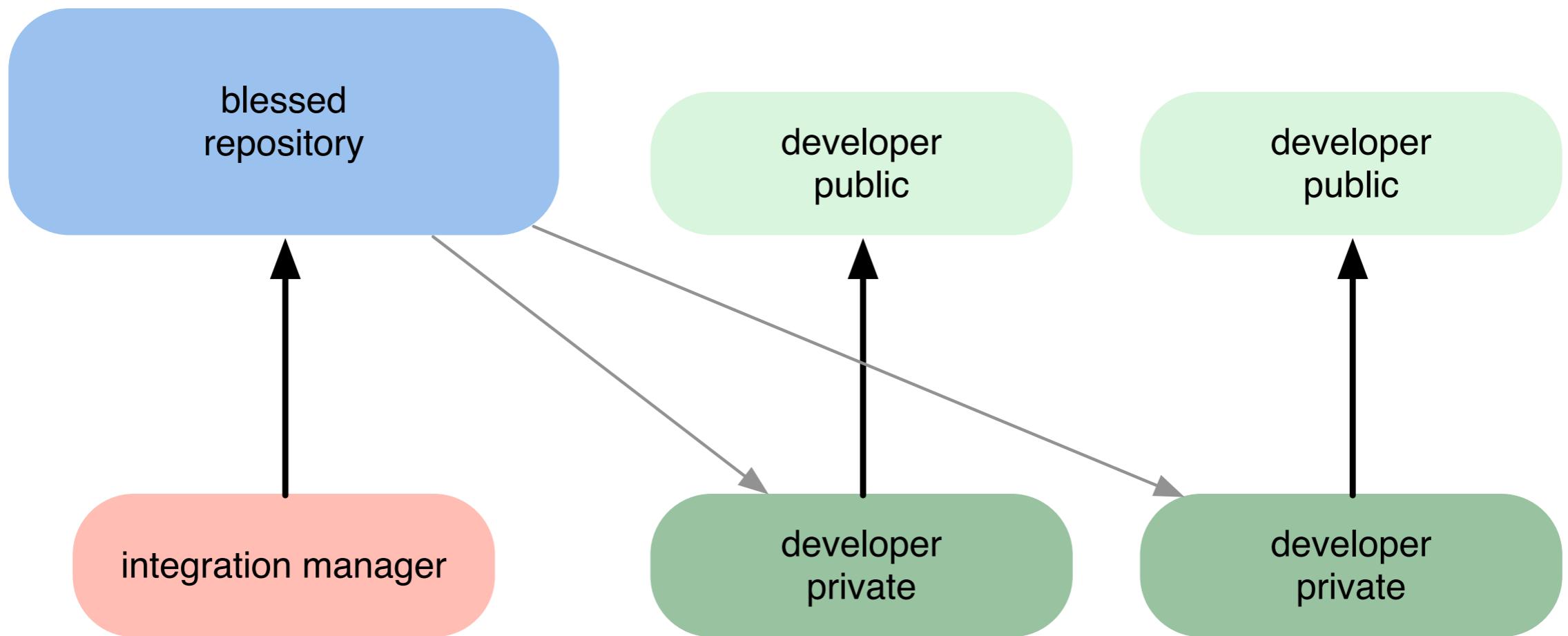


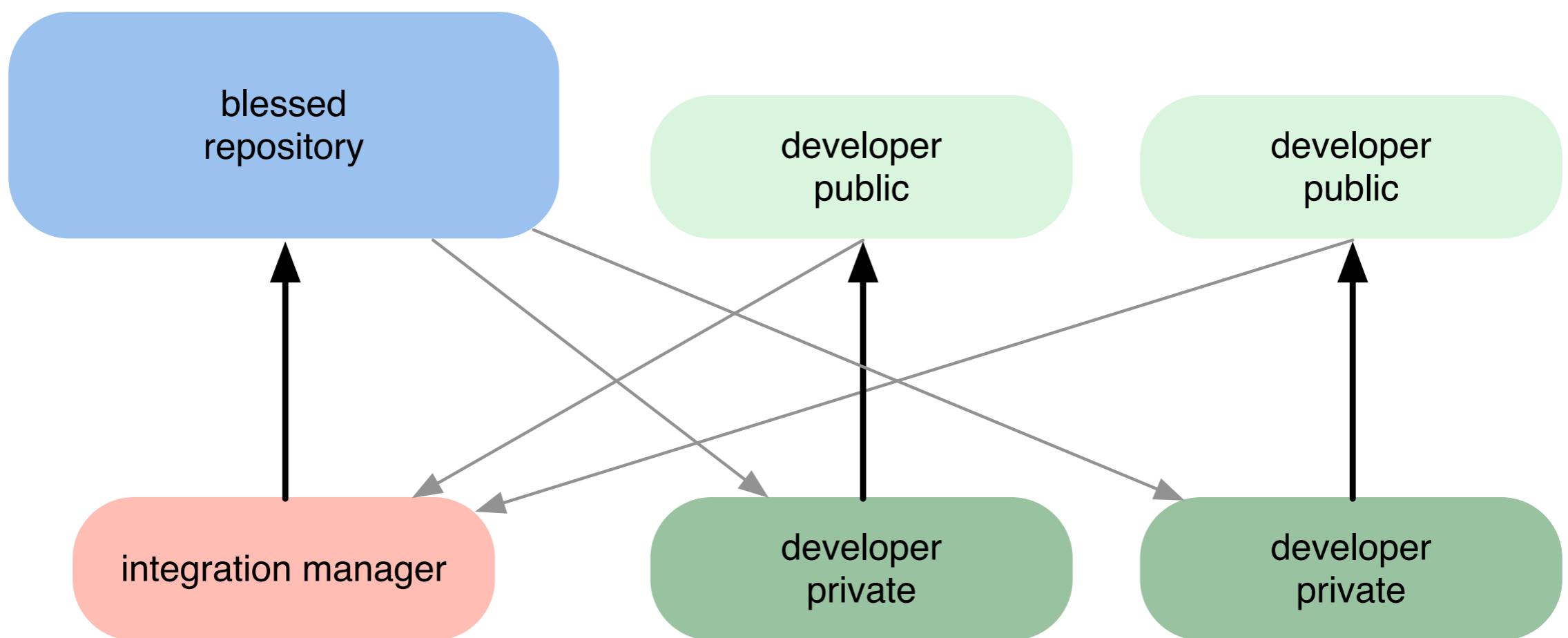


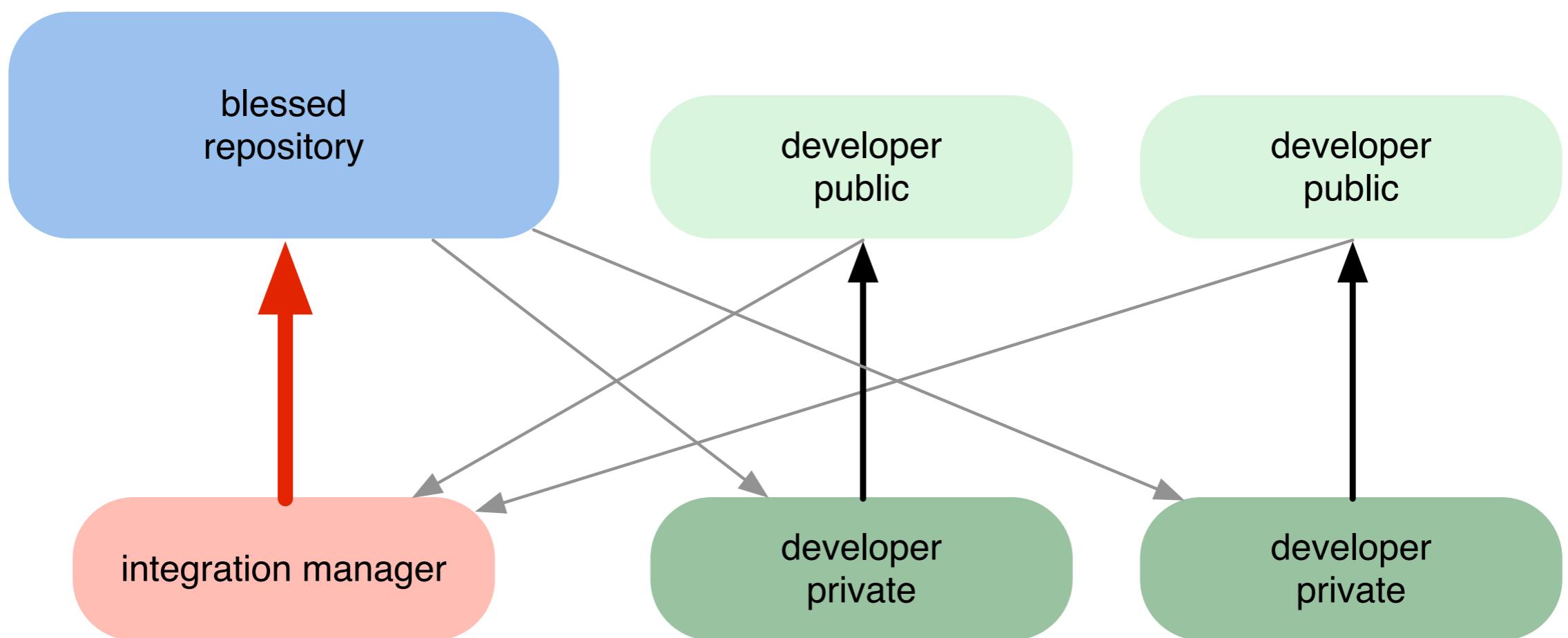
integration manager



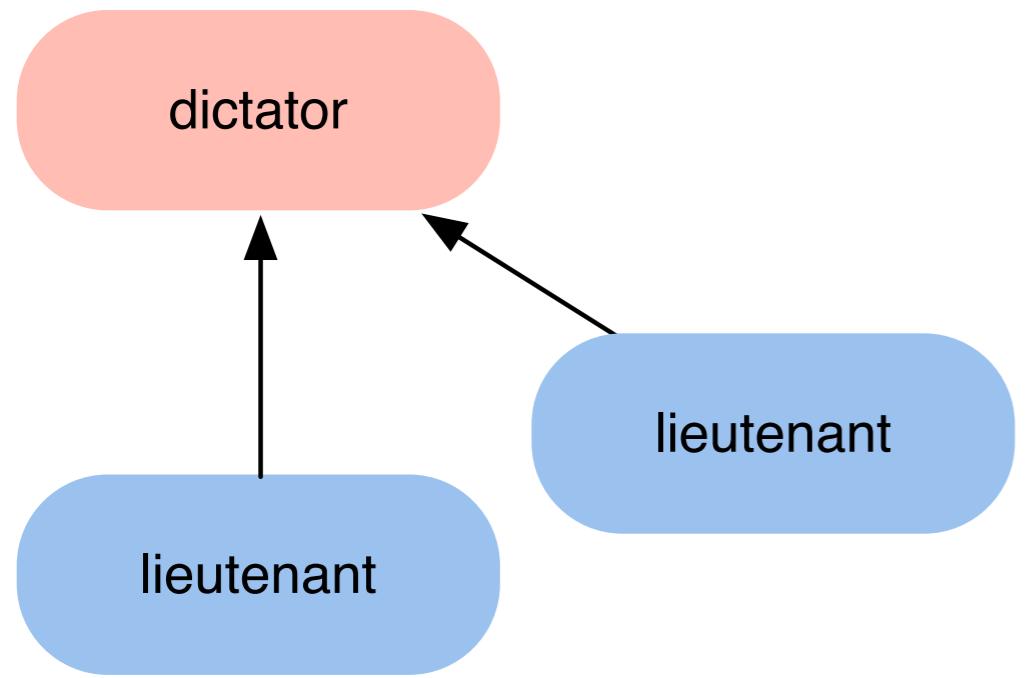


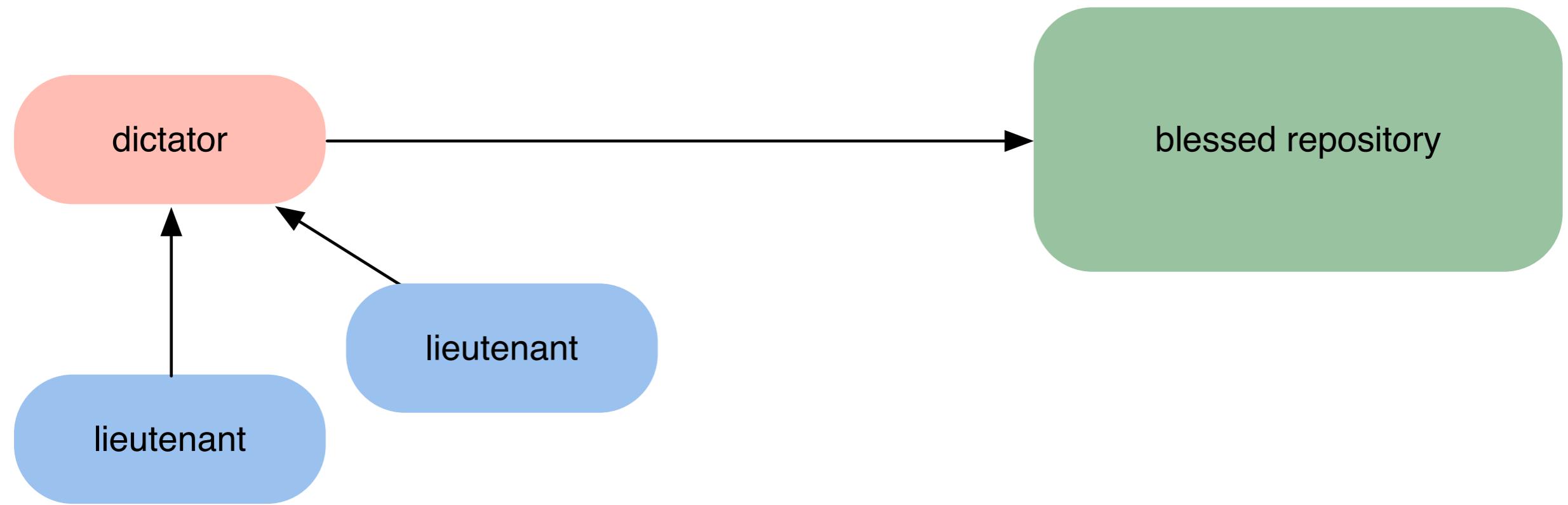


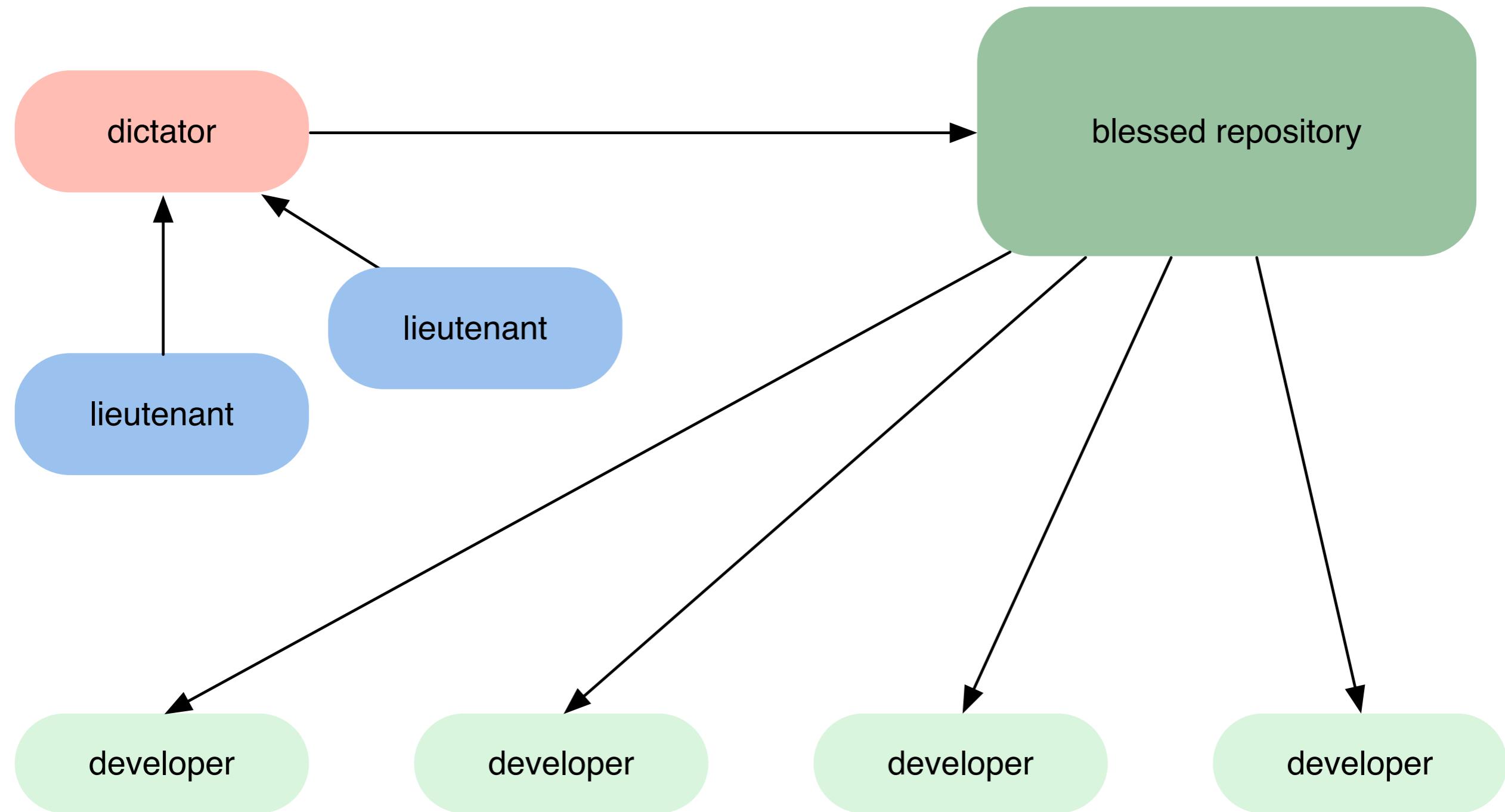


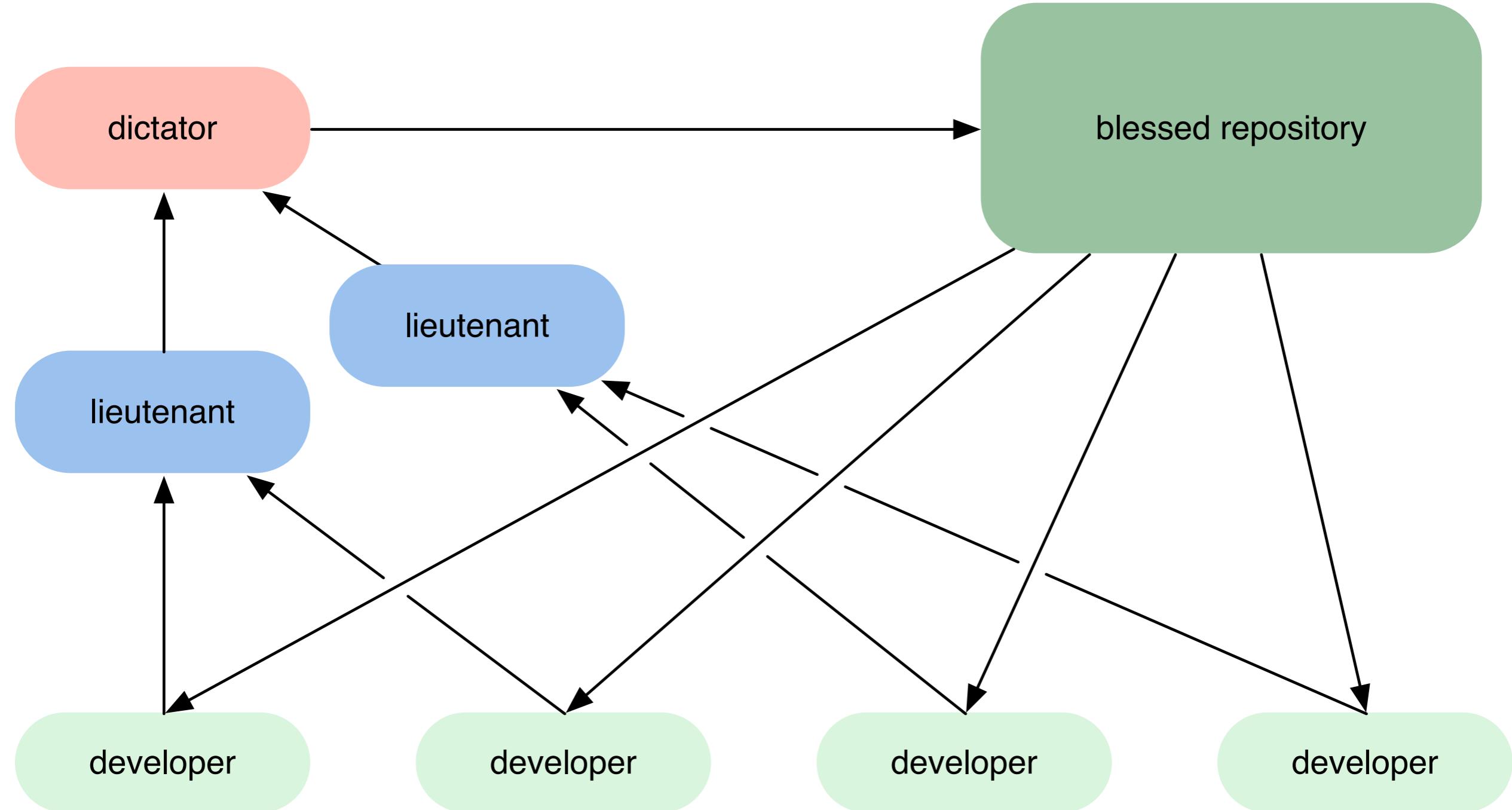


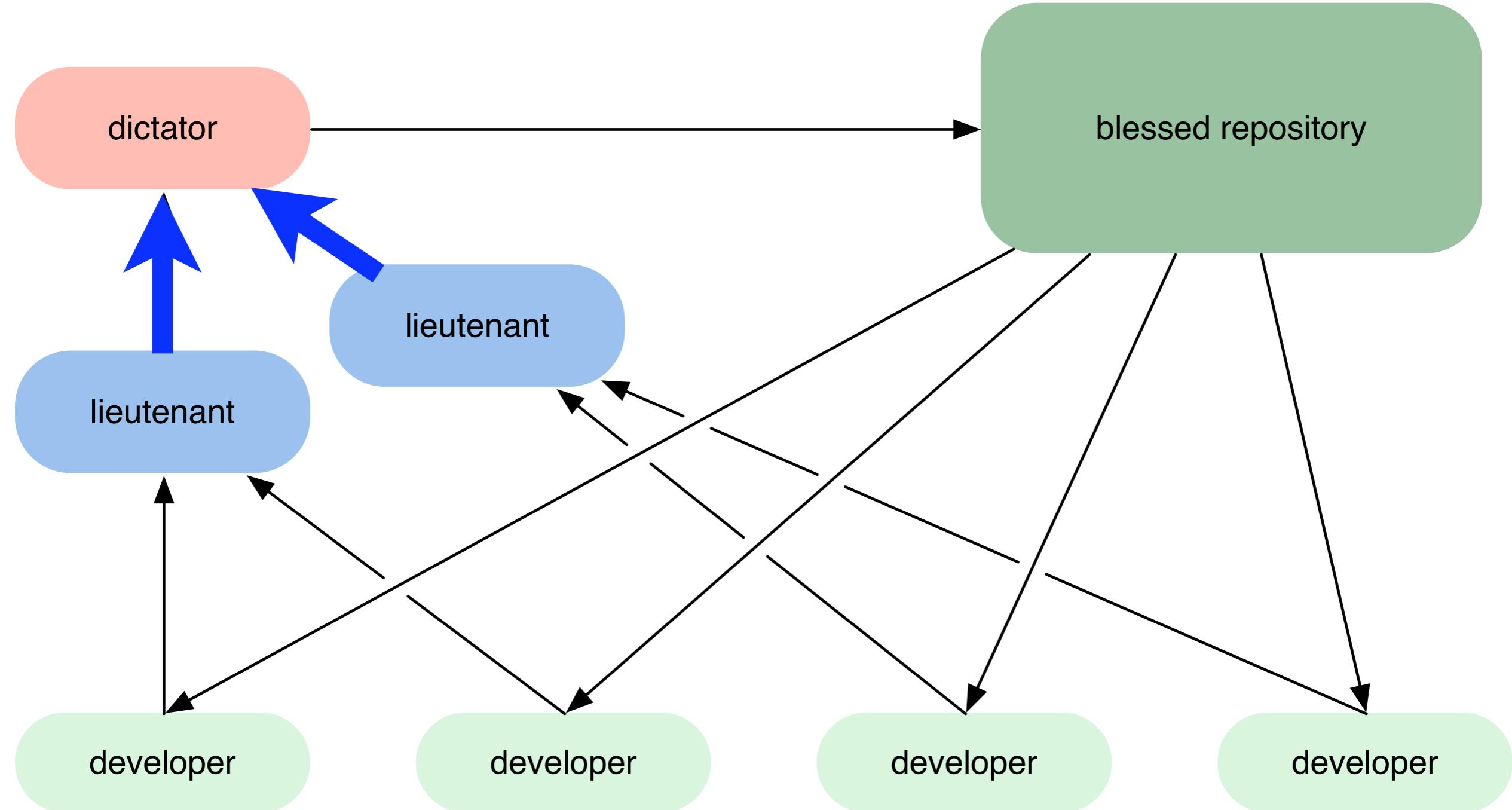
dictator / lieutenant

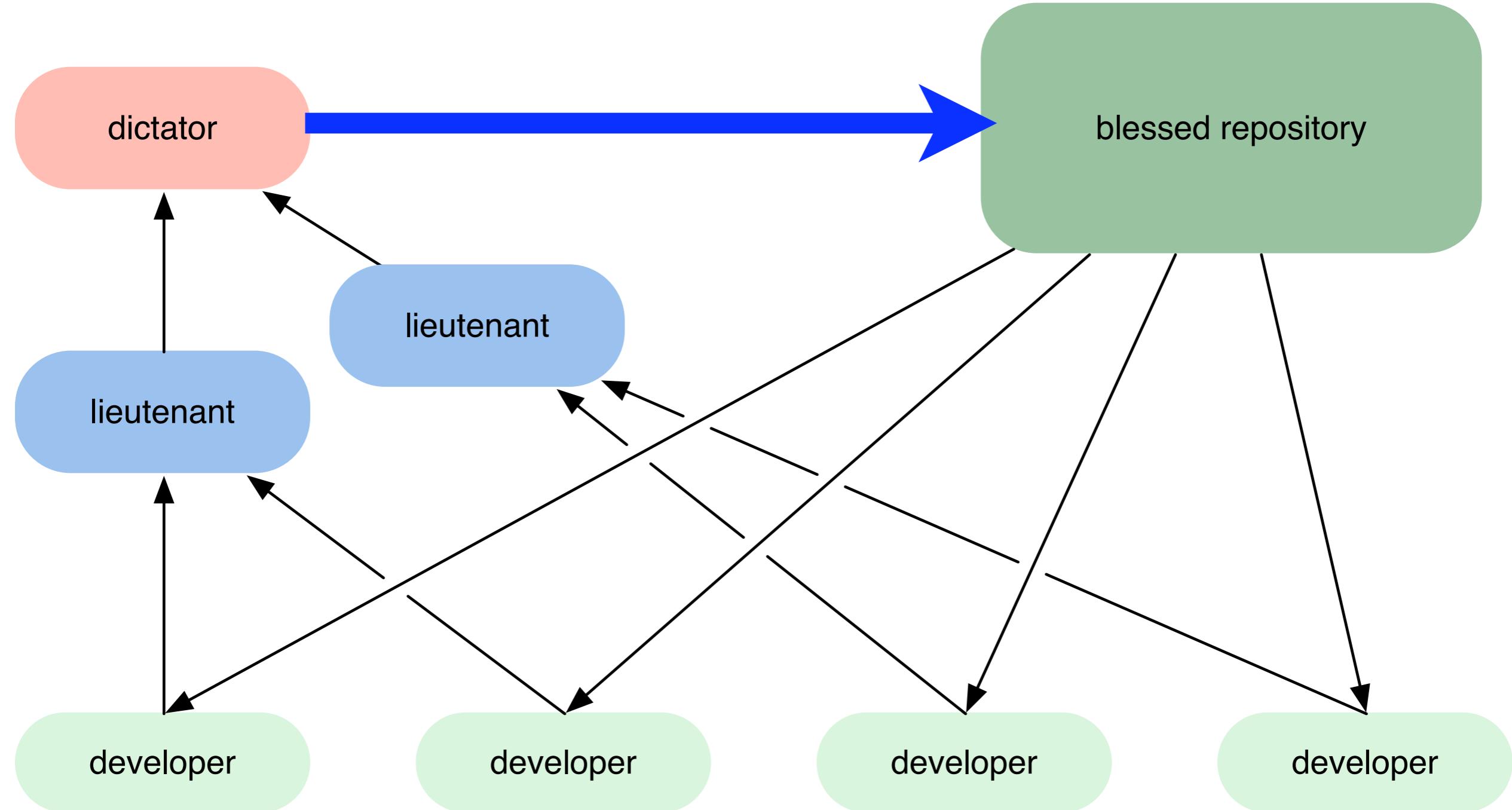


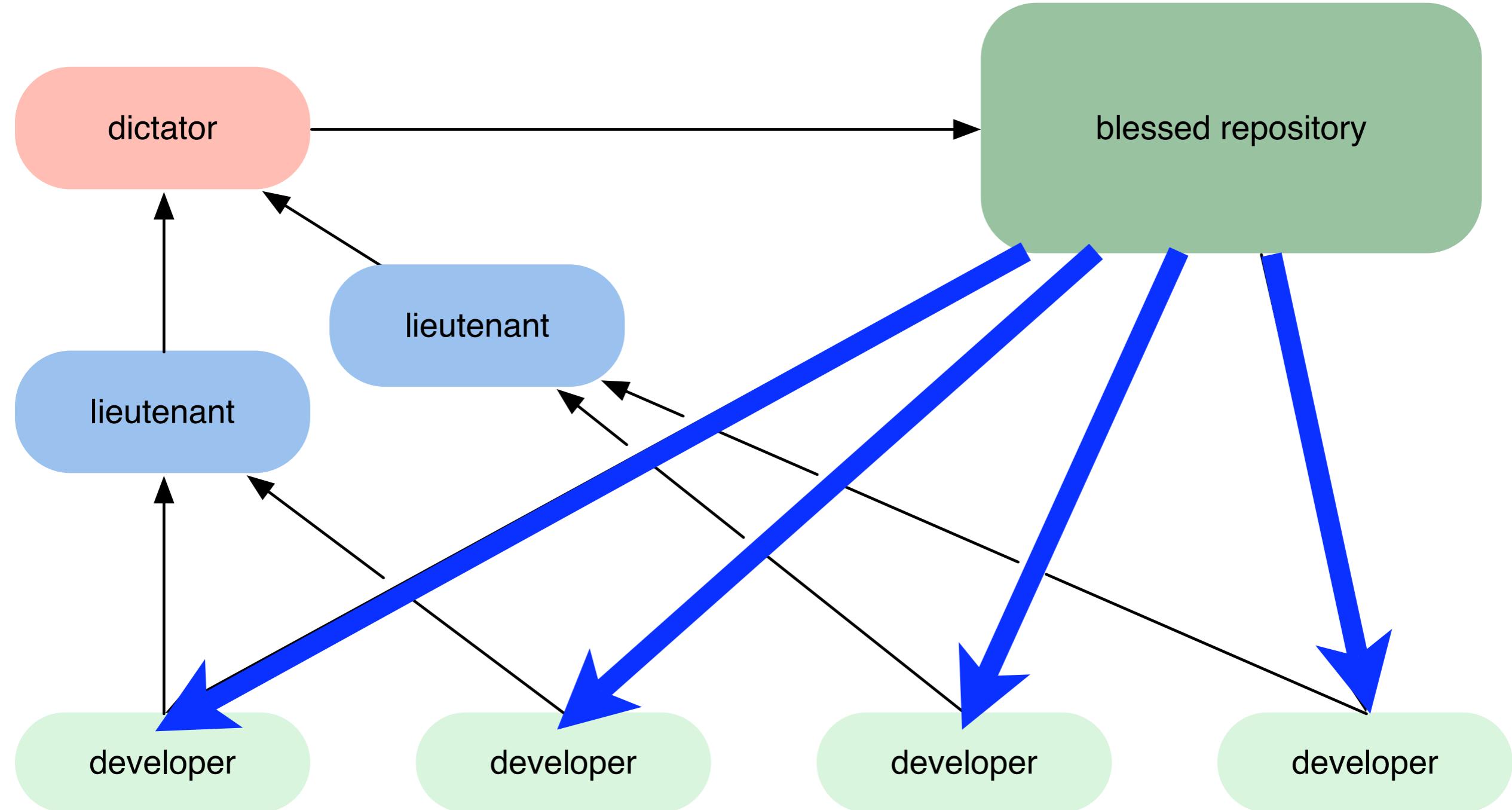


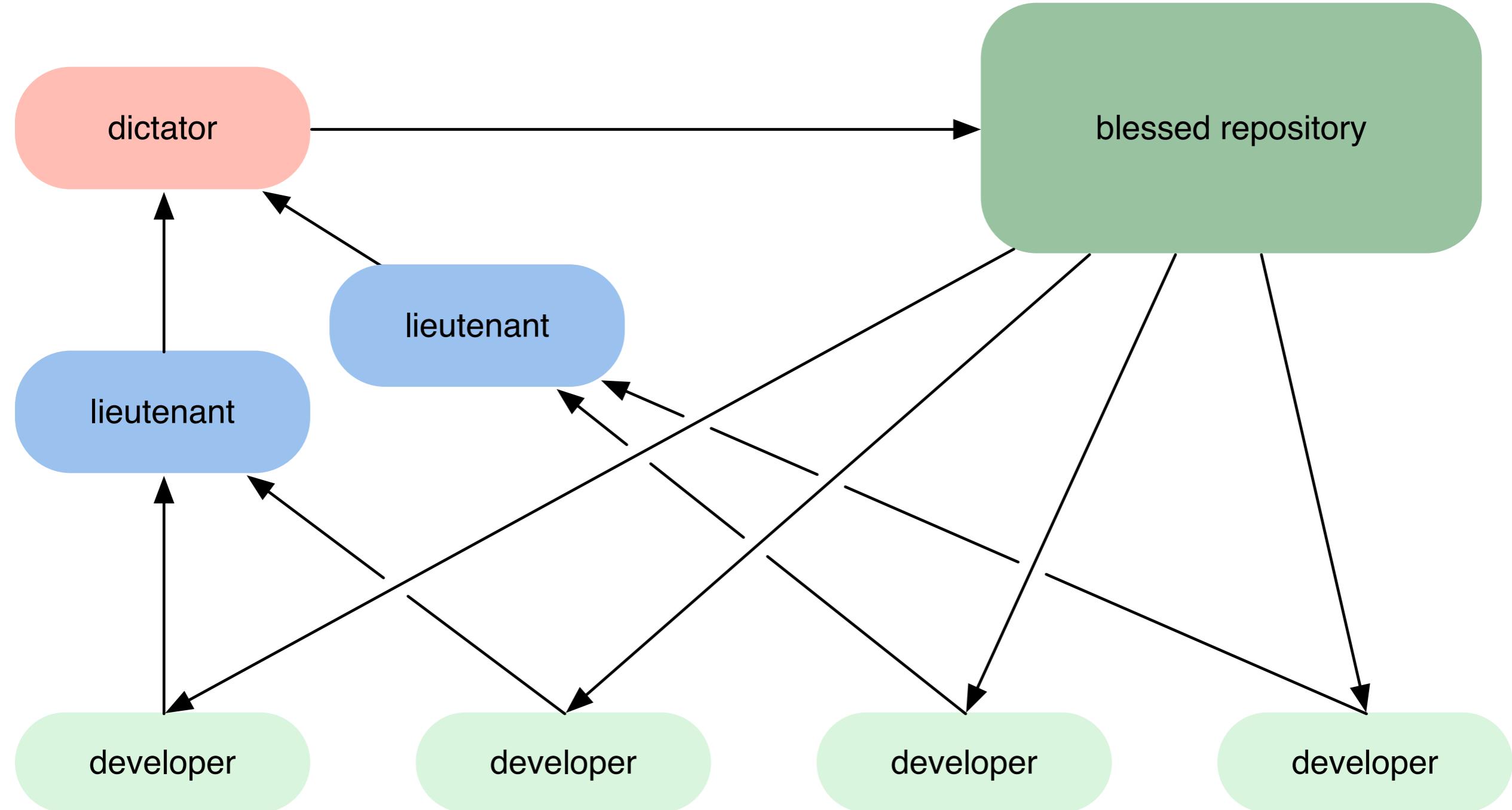












cheap local branching

cheap local branching
and easy merging

```
$ du -h testrepo/  
78M testrepo/
```

```
$ du -h testrepo/  
78M testrepo/  
$ cd testrepo/
```

```
$ du -h testrepo/  
78M testrepo/  
$ cd testrepo/  
$ time git checkout -b newbranch
```

```
$ du -h testrepo/  
78M testrepo/  
$ cd testrepo/  
$ time git checkout -b newbranch  
Switched to a new branch "newbranch"  
  
real 0m0.091s  
user 0m0.016s  
sys 0m0.023s
```

```
$ du -h testrepo/  
78M testrepo/  
$ cd testrepo/  
$ time git checkout -b newbranch  
Switched to a new branch "newbranch"  
  
real 0m0.091s  
user 0m0.016s  
sys 0m0.023s
```

```
$ time cp -Rf testrepo r2
```

```
real 0m13.684s
user 0m0.085s
sys  0m1.215s
```

| 3.68 vs 0.09

so what?

**frictionless context
switching**



JELLY/GELÉE

Personal Lubricant
Lubrifiant personnel

*#1 Doctor
Recommended Brand
Gentle and Safe
Water-Based*

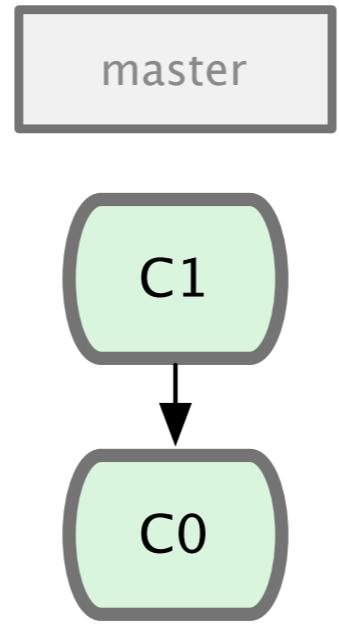
• • •

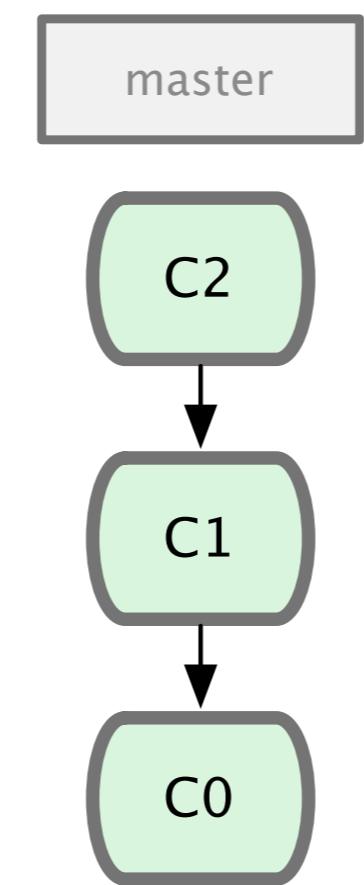
*La marque n° 1
des médecins
Doux et sûr
Fabriqué à base d'eau*

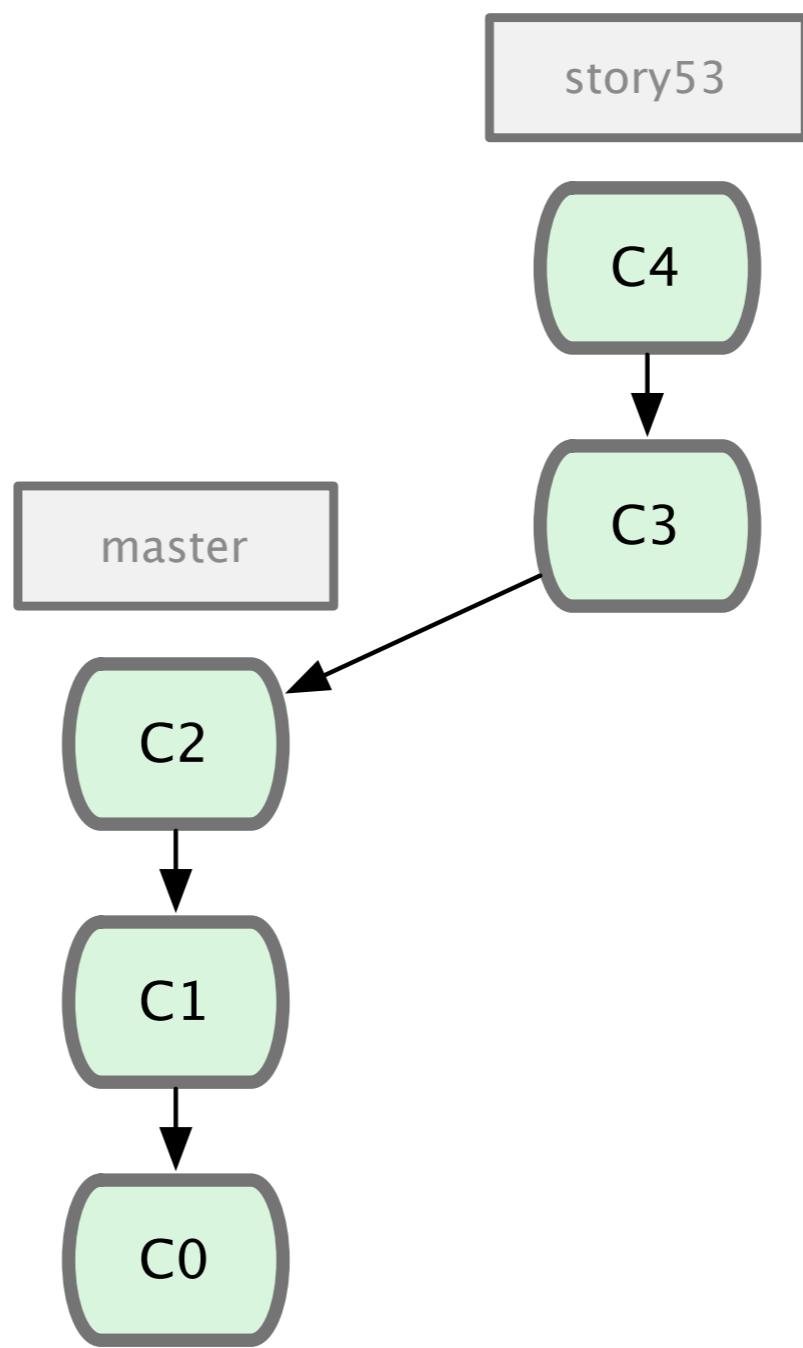
Net Wt. 4 oz (113 g)

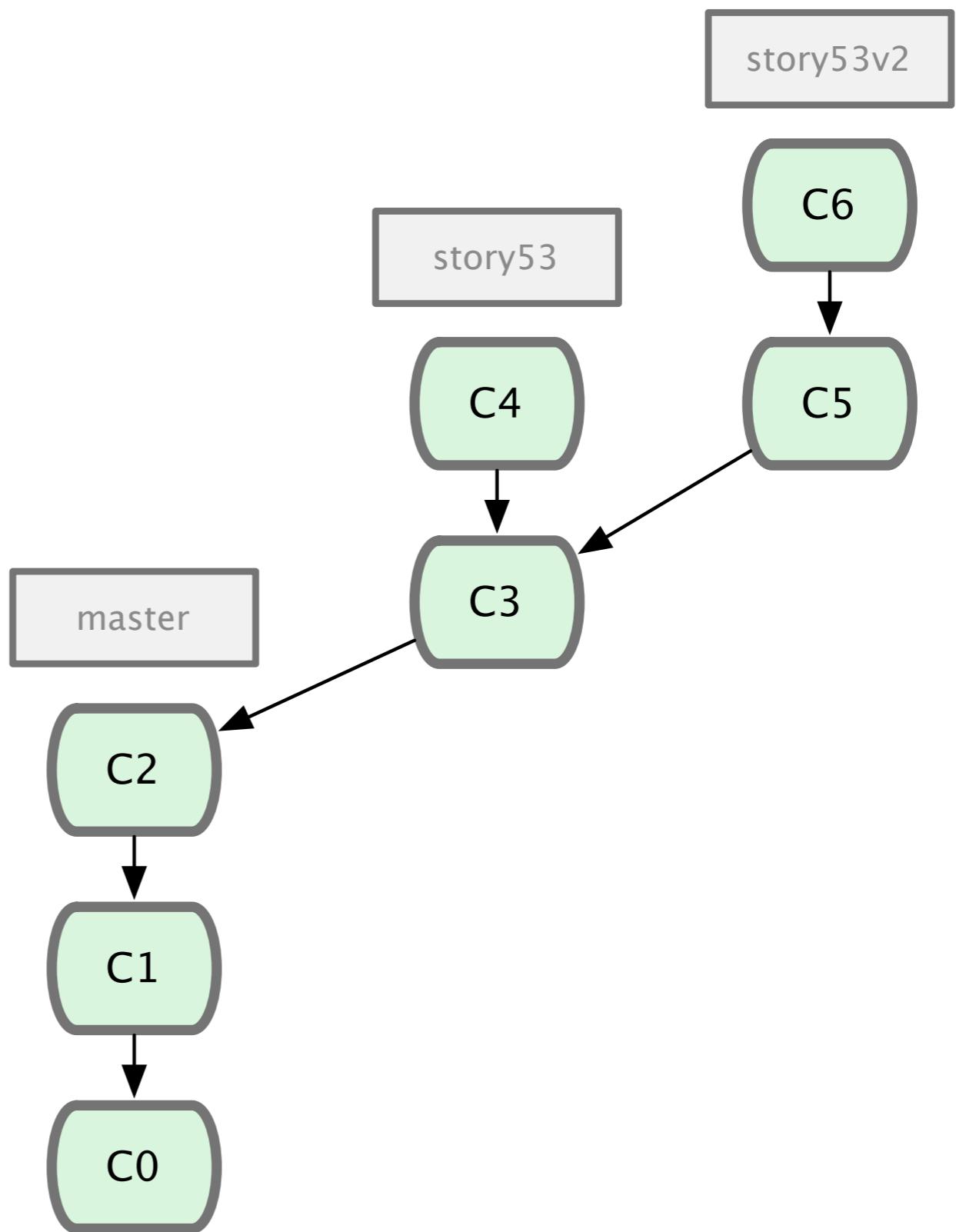
try out new ideas

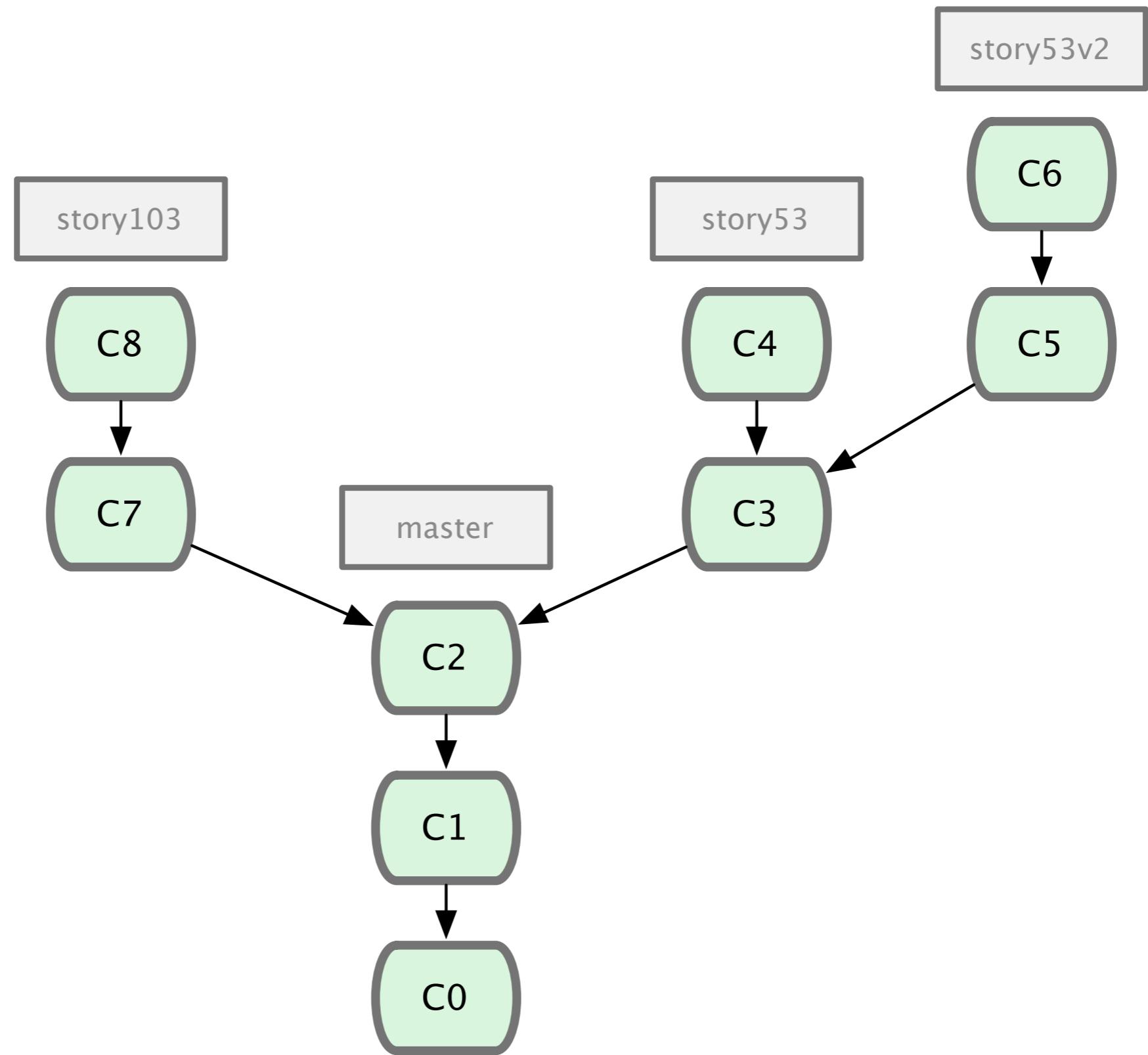
non-linear development

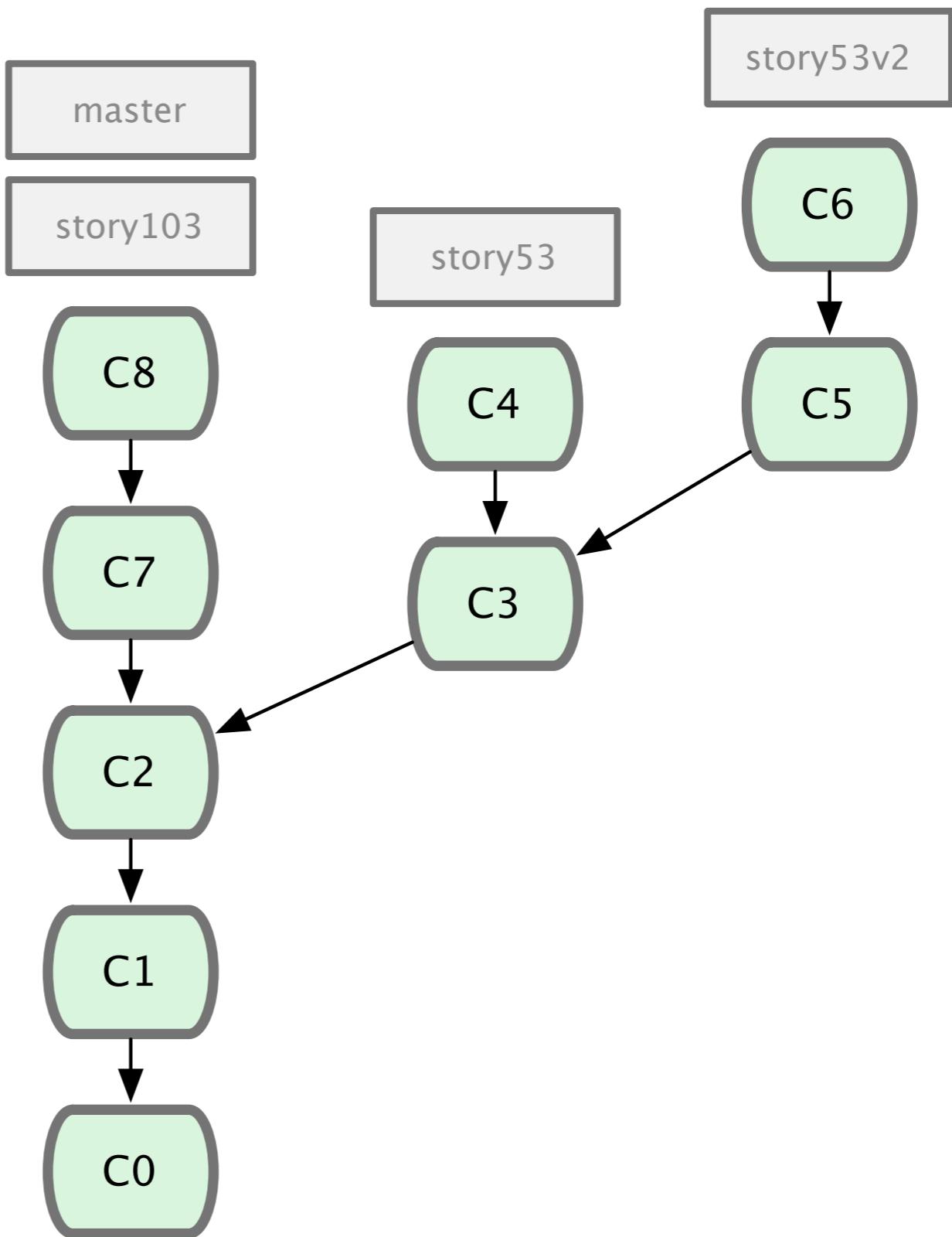


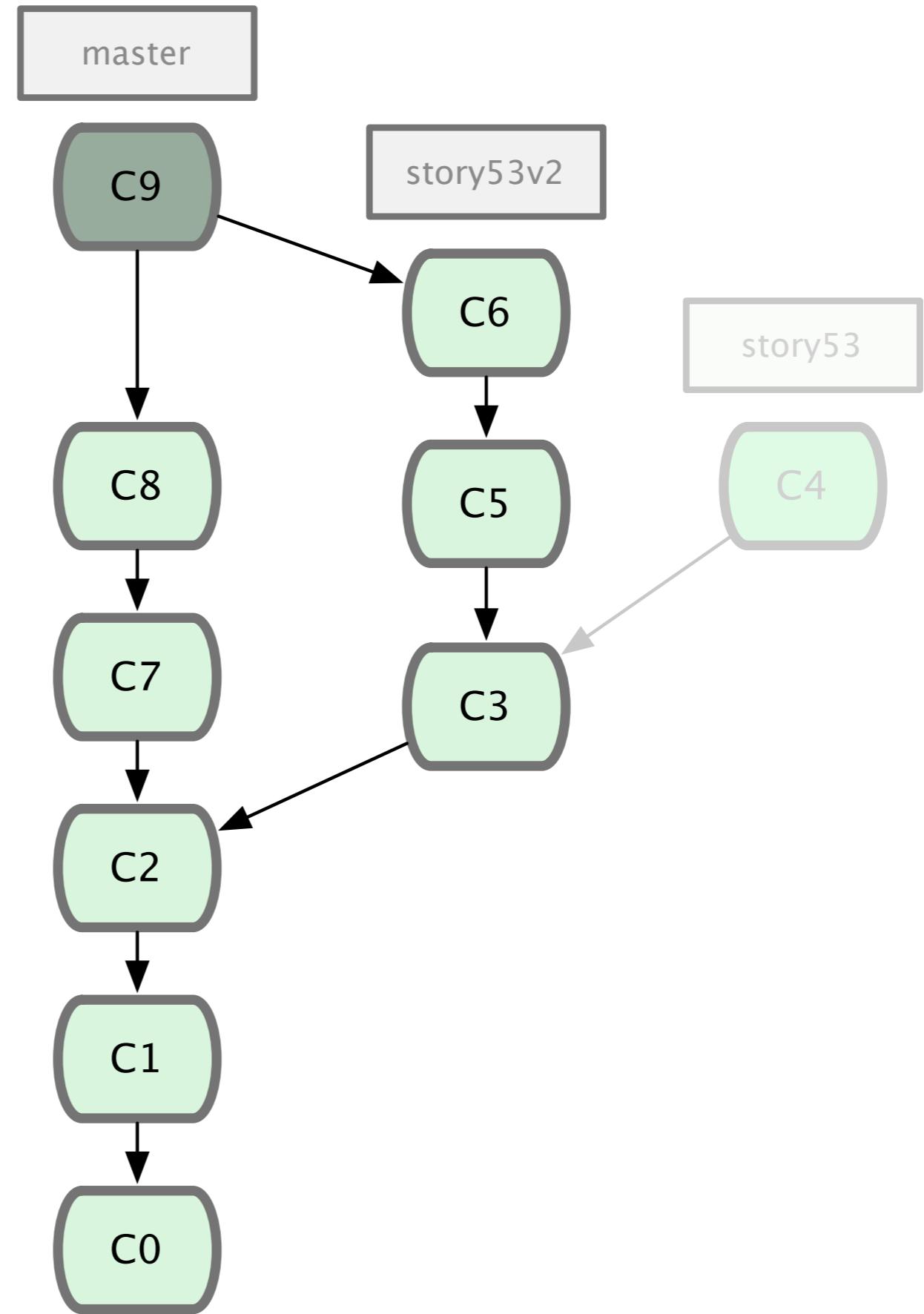




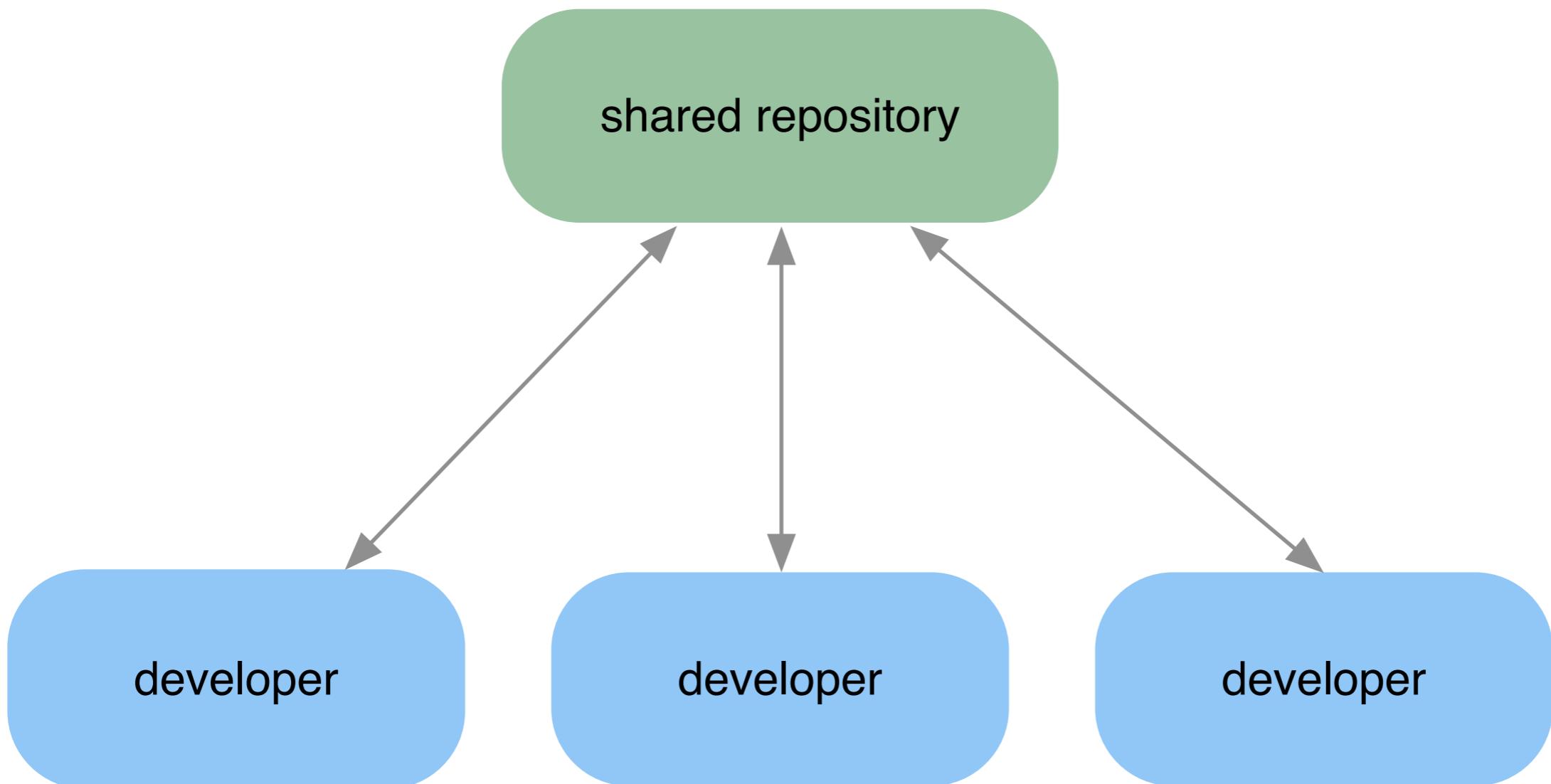


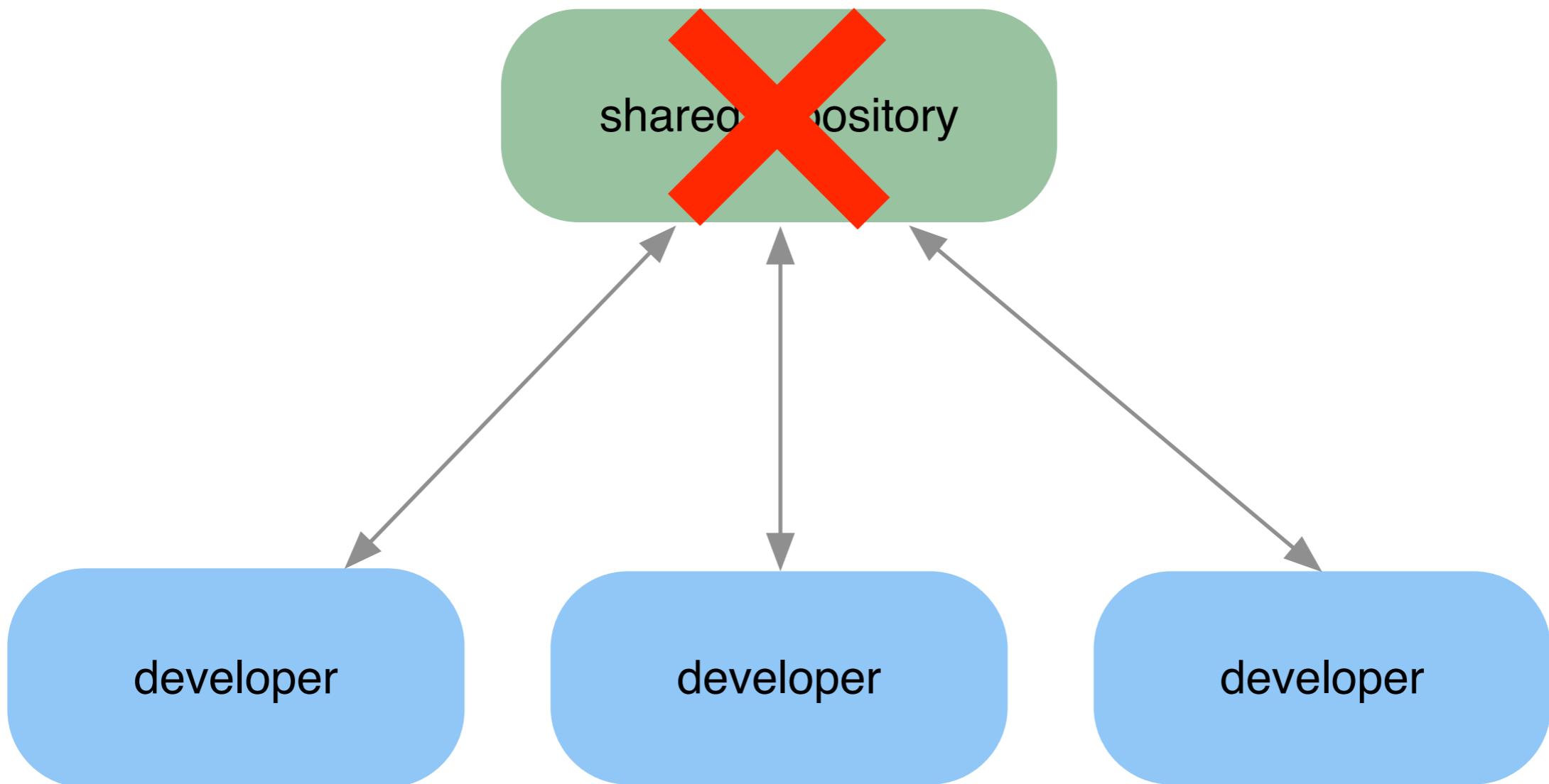


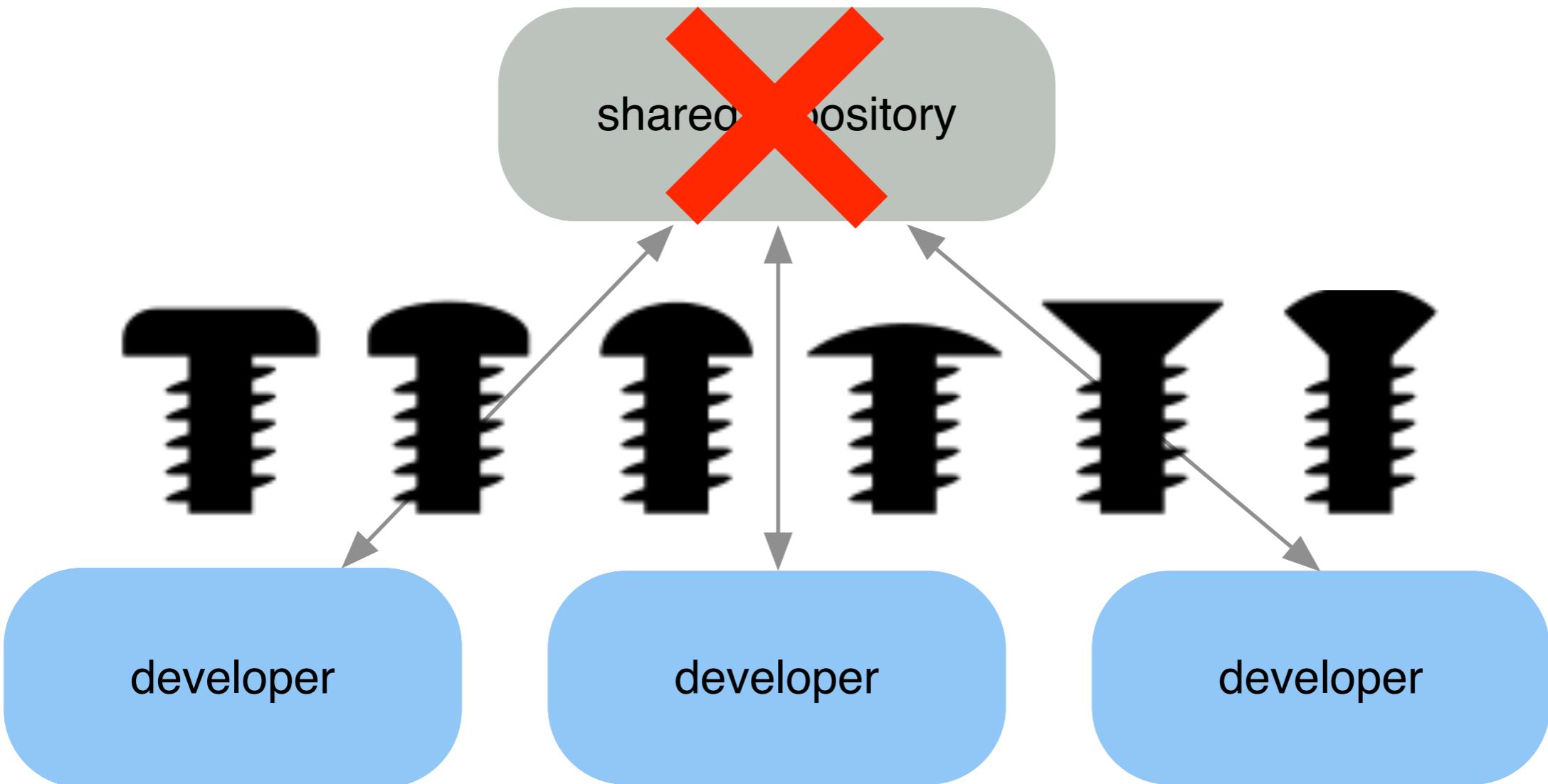




no necessary SPOF







shared repository

A central gray rounded rectangle contains the text "shared repository" in black. A large black X is drawn across the text. Three blue rounded rectangles, each containing the word "developer", are positioned below the central box, connected to it by thin lines.

developer

developer

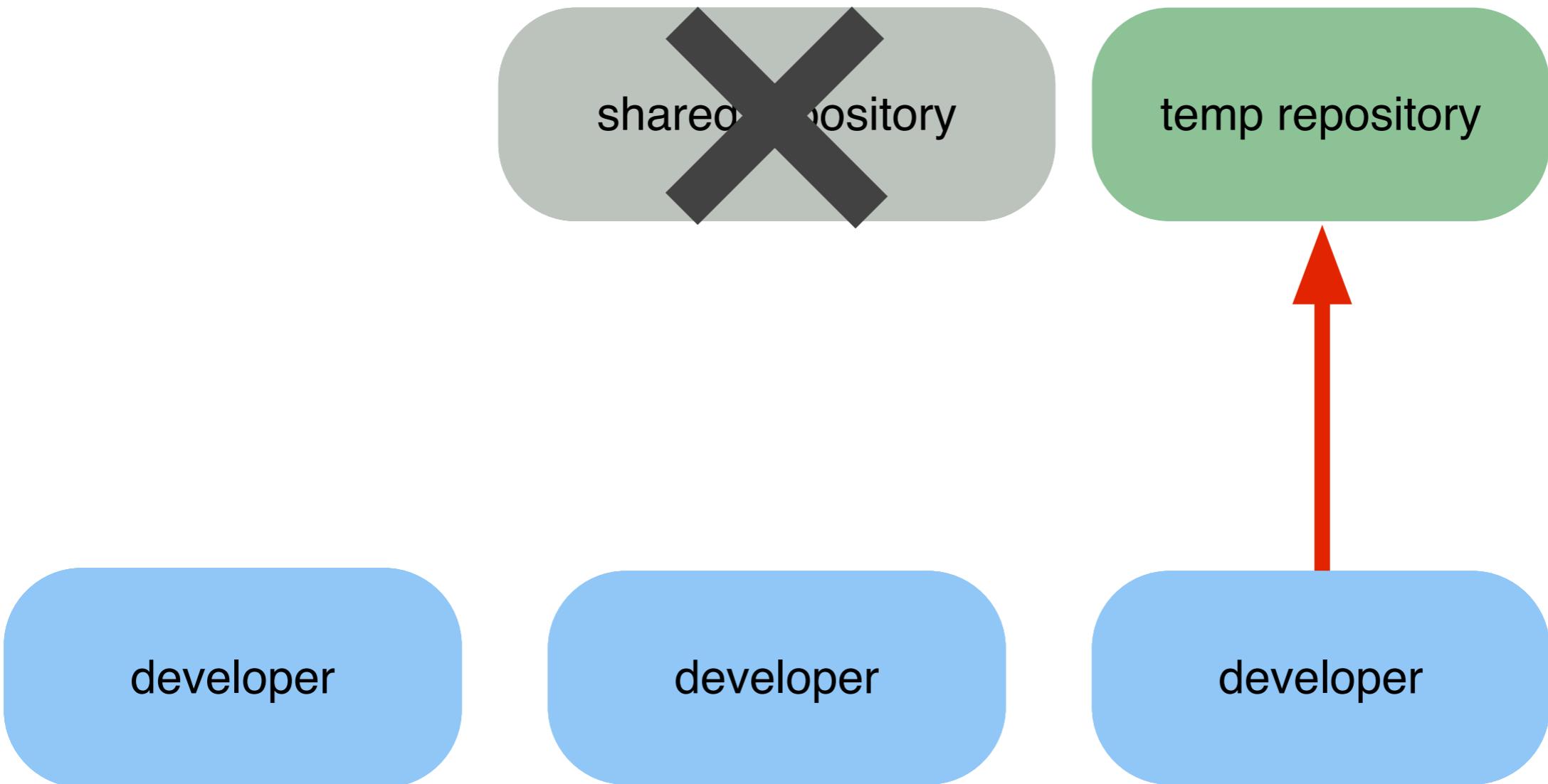
developer

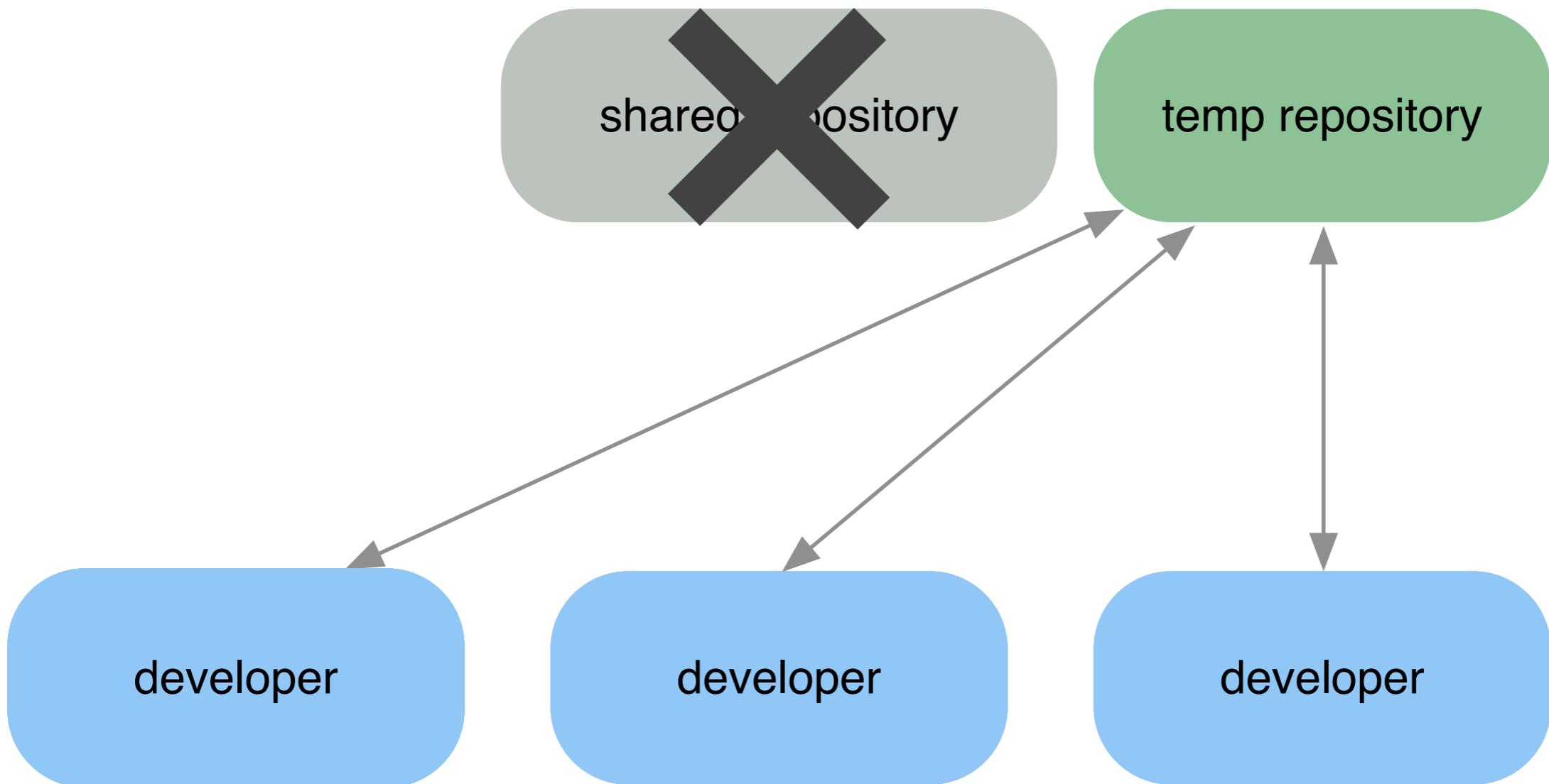
developer

developer

shared repository







How Do I Use Git?

First Steps

```
$ git config --global user.name "Scott Chacon"  
$ git config --global user.email "schacon@gmail.com"
```

Getting A Repo

git init

```
$ touch hello_world.rb
```

```
$ touch hello_world.rb  
$ git init
```

```
$ tree -a  
.  
|-- .git  
|   |-- HEAD  
|   |-- branches  
|   |-- config  
|   |-- description  
|   |-- hooks  
|   |   |-- post-commit.sample  
|   |   |-- post-receive.sample  
|   |   |-- ...  
|   |   |-- pre-rebase.sample  
|   |   `-- update.sample  
|   |-- info  
|   |   `-- exclude  
|   |-- objects  
|   |   |-- info  
|   |   `-- pack  
|   |-- refs  
|   |   |-- heads  
|   |   `-- tags  
|   `-- remotes  
`-- hello_world.rb
```

11 directories, 25 files

```
$ tree -a  
.  
|-- .git  
|   |-- HEAD  
|   |-- branches  
|   |-- config  
|   |-- description  
|   |-- hooks  
|   |   |-- post-commit.sample  
|   |   |-- post-receive.sample  
|   |   |-- ...  
|   |   |-- pre-rebase.sample  
|   |   `-- update.sample  
|   |-- info  
|   |   `-- exclude  
|   |-- objects  
|   |   |-- info  
|   |   `-- pack  
|   |-- refs  
|   |   |-- heads  
|   |   `-- tags  
|   `-- remotes  
`-- hello_world.rb
```

11 directories, 25 files

```
$ tree -a
```

```
.
|--- .git
|   |--- HEAD
|   |--- branches
|   |--- config
|   |--- description
|   |--- hooks
|   |   |--- post-commit.sample
|   |   |--- post-receive.sample
|   |   |--- ...
|   |   |--- pre-rebase.sample
|   |   `--- update.sample
|   |--- info
|   |   `--- exclude
|   |--- objects
|   |   |--- info
|   |   |--- pack
|   |--- refs
|   |   |--- heads
|   |   |--- tags
|   |--- remotes
`--- hello_world.rb
```

11 directories, 25 files

git clone

```
$ git clone
```

```
$ git clone git://github.com/schacon/grit.git
```

```
$ git clone git://github.com/schacon/grit.git mygrit
```

```
$ git clone git://github.com/schacon/grit.git mygrit
Initialized empty Git repository in /home/schacon/mygrit/.git/
remote: Counting objects: 3220, done.
remote: Compressing objects: 100% (2093/2093), done.
remote: Total 3220 (delta 1134), reused 3149 (delta 1081)
Receiving objects: 100% (3220/3220), 1.79 MiB | 357 KiB/s, done.
Resolving deltas: 100% (1134/1134), done.
```

```
$ git clone git://github.com/schacon/grit.git mygrit
Initialized empty Git repository in /home/schacon/mygrit/.git/
remote: Counting objects: 3220, done.
remote: Compressing objects: 100% (2093/2093), done.
remote: Total 3220 (delta 1134), reused 3149 (delta 1081)
Receiving objects: 100% (3220/3220), 1.79 MiB | 357 KiB/s, done.
Resolving deltas: 100% (1134/1134), done.
```

```
$ cd mygrit
$
```

```
$ git clone git://github.com/schacon/grit.git mygrit
Initialized empty Git repository in /home/schacon/mygrit/.git/
remote: Counting objects: 3220, done.
remote: Compressing objects: 100% (2093/2093), done.
remote: Total 3220 (delta 1134), reused 3149 (delta 1081)
Receiving objects: 100% (3220/3220), 1.79 MiB | 357 KiB/s, done.
Resolving deltas: 100% (1134/1134), done.
```

```
$ cd mygrit
$ ls
API.txt      Manifest.txt README.txt benchmarks.rb examples lib
History.txt   PURE_TODO Rakefile benchmarks.txt grit.gemspec test
$
```

A Basic Workflow

A Basic Workflow

Edit files

Stage the changes

Review your changes

Commit the changes

working directory

index

repository

working directory

a working copy
of your project

index

repository

working directory

index

object database

repository

working directory

index

“staging”

repository

A Basic Workflow

Edit files

Stage the changes

Review your changes

Commit the changes

\$

\$ find .

```
$ find .  
./app.yaml  
./index.yaml  
./main.py
```

\$

```
$ vim main.py
```

```
$ vim main.py
```

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hello world!')

    def main():
        application = webapp.WSGIApplication([(('/', MainHandler)],
                                              debug=True)
        wsgiref.handlers.CGIHandler().run(application)

if __name__ == '__main__':
    main()
~
~
"main.py" 16L, 402C
```

```
$ vim main.py
```

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hello world!')

    def main():
        application = webapp.WSGIApplication([(('/', MainHandler)],
                                              debug=True)
        wsgiref.handlers.CGIHandler().run(application)

    if __name__ == '__main__':
        main()
~  
~  
"main.py" 16L, 402C
```

```
$ git status
```

```
$ git status
```

```
# On branch master
```

```
# Changed but not updated:
```

```
#   (use "git add <file>..." to update what
```

```
#
```

```
# modified:    main.py
```

```
#
```

```
no changes added to commit (use "git add" and
```

```
$ git status
```

```
# On branch master
```

```
# Changed but not updated:
```

```
#   (use "git add <file>..." to update what
```

```
#
```

```
# modified:    main.py
```

```
#
```

```
no changes added to commit (use "git add" and
```

```
$ git status
```

```
# On branch master
```

```
# Changed but not updated:
```

```
#   (use "git add <file>..." to update what
```

```
#
```

```
# modified:    main.py
```

```
#
```

```
no changes added to commit (use "git add" and
```

```
$ git status
```

```
# On branch master
```

```
# Changed but not STAGED:
```

```
#   (use "git add <file>..." to update what
```

```
#
```

```
# modified:    main.py
```

```
#
```

```
no changes added to commit (use "git add" and
```

A Basic Workflow

Edit files

Stage the changes

Review your changes

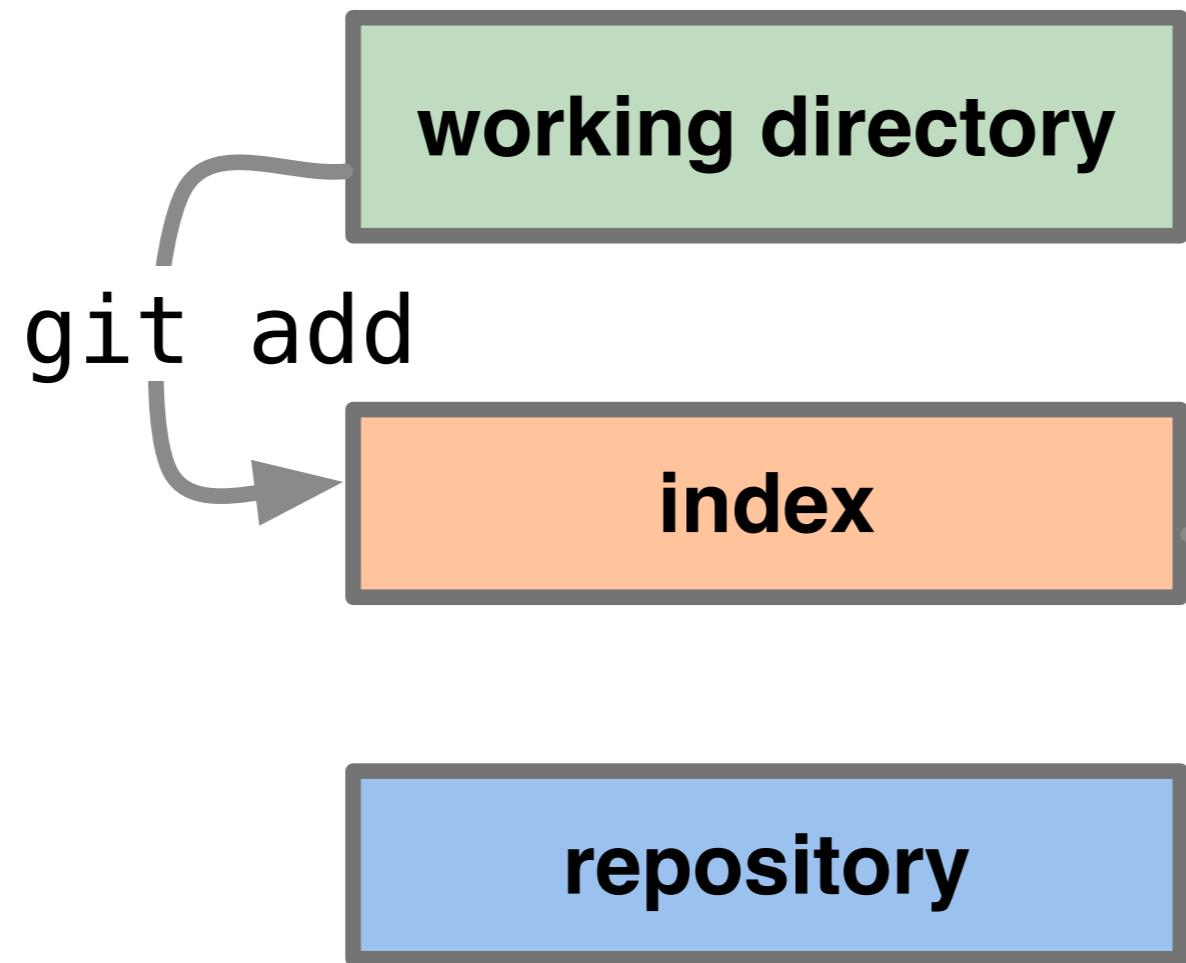
Commit the changes

git add

working directory

index

repository



\$

```
$ git add main.py
```

```
$ git add main.py
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>
#
# modified:    main.py
#
```

```
$ git add main.py
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>
#
# modified:    main.py
#
```

```
$ git add main.py
$ git status
# On branch master
# Changes THAT ARE STAGED :
#   (use "git reset HEAD <file>
#
# modified:    main.py
#
```

\$

```
$ vim app.yaml
```

```
$ vim app.yaml
```

```
application: chacon
version: 1
runtime: python
api_version: 1
```

```
handlers:
- url: .*
  script: main.py
```

```
~
```

```
~
```

```
~
```

```
"app.yaml" 8L, 101C
```

```
$ vim app.yaml
```

```
application: chacon
version: █
runtime: python
api_version: 1
```

```
handlers:
- url: .*
  script: main.py
```

```
~
```

```
~
```

```
~
```

```
"app.yaml" 8L, 101C
```

```
$ vim app.yaml
```

```
application: chacon
version: 2
runtime: python
api_version: 1
```

```
handlers:
- url: .*
  script: main.py
```

```
~
```

```
~
```

```
~
```

```
"app.yaml" 8L, 101C
```

\$

```
$ git status
```

\$ git status

```
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:   main.py
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be com
#
# modified:   app.yaml
#
```

\$

```
$ vim main.py
```

```
$ vim main.py
```

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hello world!')

def main():
    application = webapp.WSGIApplication(['/'], MainHandler),
                                         debug=True)
    wsgiref.handlers.CGIHandler().run(application)

if __name__ == '__main__':
    main()
~
```

```
$ vim main.py
```

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hola █ orld!')

def main():
    application = webapp.WSGIApplication([('/', MainHandler)],
                                          debug=True)
    wsgiref.handlers.CGIHandler().run(application)

if __name__ == '__main__':
    main()
~
~
"main.py" 16L, 402C
```

```
$ vim main.py
```

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hola Mundo')

def main():
    application = webapp.WSGIApplication(['/'], MainHandler),
                                         debug=True)
    wsgiref.handlers.CGIHandler().run(application)

if __name__ == '__main__':
    main()
~  
~  
"main.py" 16L, 402C
```

\$

```
$ git status
```

```
$ git status
```

```
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:   main.py
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be com
#
# modified:   app.yaml
# modified:   main.py
#
```

\$ git status

```
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:   main.py
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be com
#
# modified:   app.yaml
# modified:   main.py
#
```

Staged



\$ git status

```
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:   main.py
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be com
#
# modified:   app.yaml
# modified:   main.py
#
```

Unstaged



\$ git status

```
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   main.py
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be com
#
#       modified:   app.yaml
#       modified:   main.py
#
```



Staged

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hello World!')

def main():
    application =
webapp.WSGIApplication([(' '/',
debug=True)

wsgiref.handlers.CGIHandler().run(applicat

if __name__ == '__main__':
    main()
```

In Working Directory

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hola Mundo!')

def main():
    application =
webapp.WSGIApplication([(' '/',
debug=True)

wsgiref.handlers.CGIHandler().run(applicat

if __name__ == '__main__':
    main()
```

Staged

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hello World!')

    def main():
        application =
webapp.WSGIApplication([('/' debug=True)

wsgiref.handlers.CGIHandler().run(application)

if __name__ == '__main__':
    main()
```

In Working Directory

```
#!/usr/bin/env python
import wsgiref.handlers
from google.appengine.ext import webapp

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

    def get(self):
        self.response.out.write('Hola Mundo!')

    def main():
        application =
webapp.WSGIApplication([('/' debug=True)

wsgiref.handlers.CGIHandler().run(application)

if __name__ == '__main__':
    main()
```

You have to stage a file
after you edit it

You have to stage a file
after you edit it

You have to stage a file
after you edit it

\$

```
$ git add app.yaml main.py
```

```
$ git add app.yaml main.py
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:   app.yaml
# modified:   main.py
#
```

interesting tidbit

git add -p

A Basic Workflow

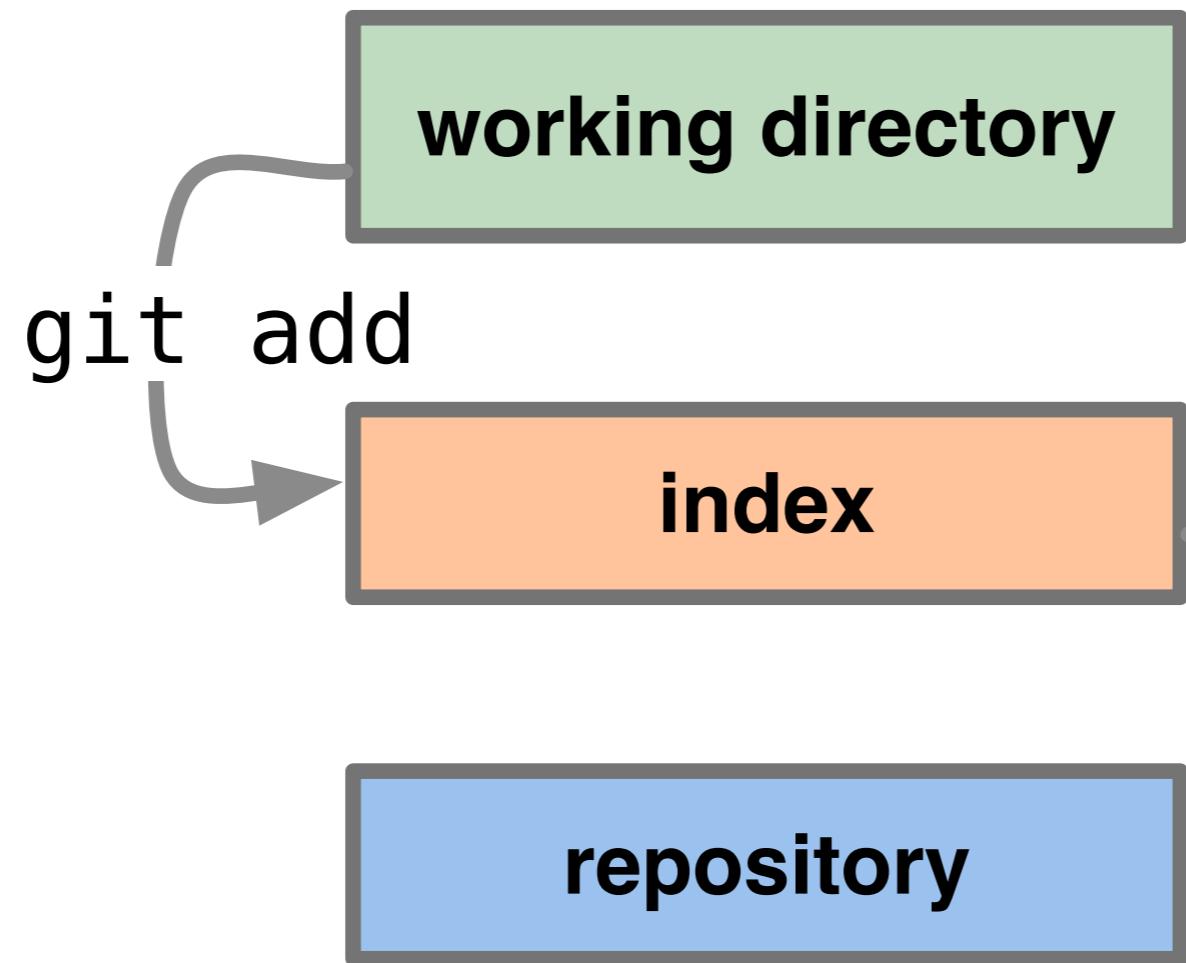
Edit files

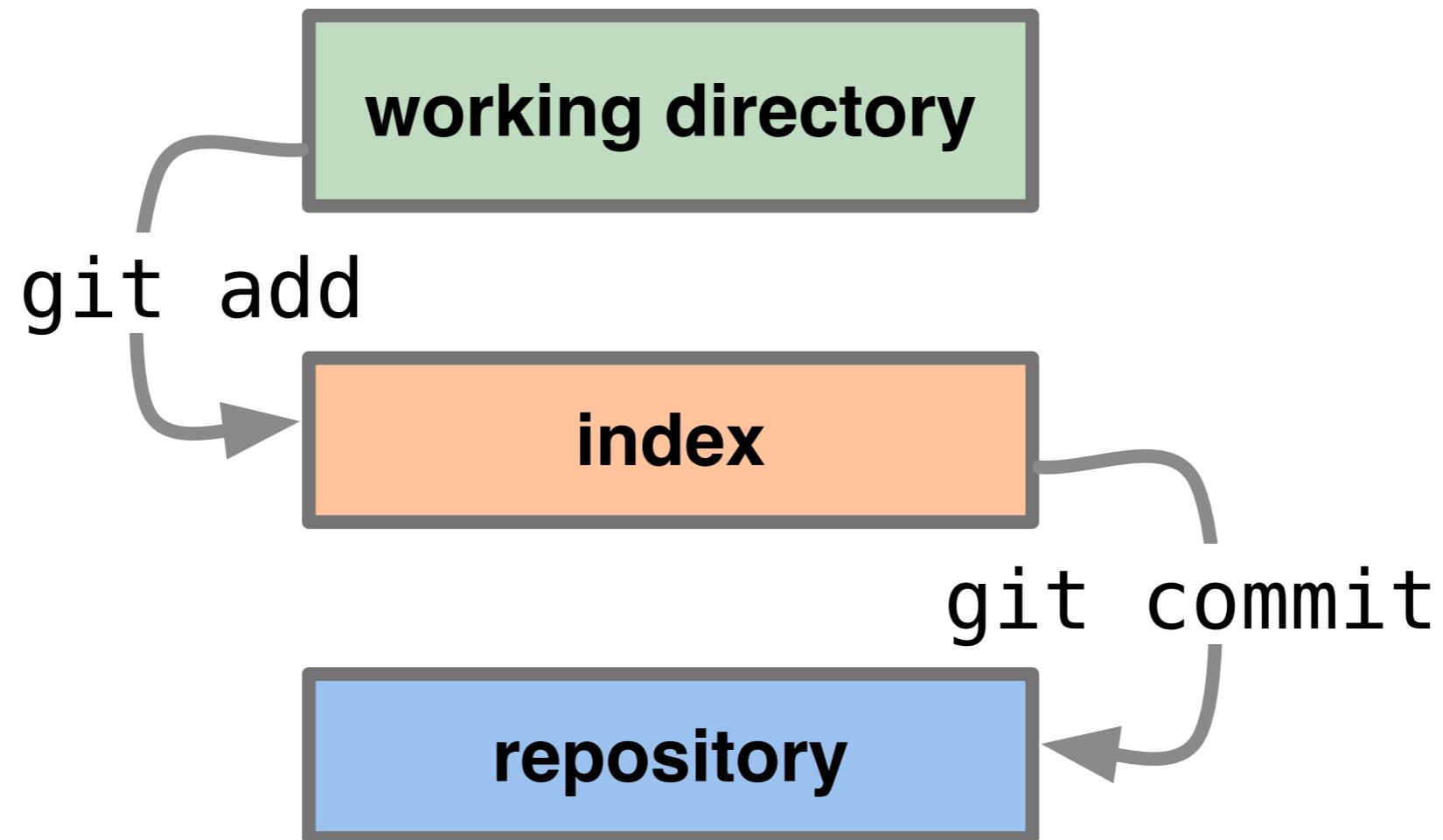
Stage the changes

Review your changes

Commit the changes

git commit





\$

```
$ git commit
```

```
$ git commit
```

\$ git commit

```
descriptive commit message █  
# Please enter the commit message for your changes. Lines starting  
# with '#' will be ignored, and an empty message aborts the commit.  
# On branch master  
# Changes to be committed:  
#   (use "git reset HEAD <file>..." to unstage)  
#  
#       modified:   app.yaml  
#       modified:   main.py  
#  
~  
~  
~  
~  
~  
".git/COMMIT_EDITMSG" 10L, 279C
```

```
$ git commit
```

```
Created commit 77d3001: descriptive commit message  
2 files changed, 4 insertions(+), 2 deletions(-)
```

A Basic Workflow

A Basic Workflow

Edit files

vim / emacs / etc

A Basic Workflow

Edit files

`vim / emacs / etc`

Stage the changes

`git add (file)`

A Basic Workflow

Edit files

`vim / emacs / etc`

Stage the changes

`git add (file)`

Review your changes

`git status`

A Basic Workflow

Edit files

`vim / emacs / etc`

Stage the changes

`git add (file)`

Review your changes

`git status`

Commit the changes

`git commit`

cheating...

A Basic Workflow

A Basic Editor Workflow

Edit files

`vim / emacs / etc`

A Basic Git Workflow

Edit files

`vim / emacs / etc`

Stage & commit the changes

`git commit -a`

What's going on here?

```
$ git commit
```

```
Created commit 77d3001: descriptive commit message  
2 files changed, 4 insertions(+), 2 deletions(-)
```

```
$ git commit
```

```
Created commit 77d3001: descriptive commit message  
2 files changed, 4 insertions(+), 2 deletions(-)
```

77d3001

77d3001

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

commit	size
tree	c4ec5
parent	a149e
author	Scott
committer	Scott
my commit message goes here and it is really, really cool	

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

```
tree c4ec543b0322744e55c5efc9b6c4e449d398dbff
parent a149e2160b3f7573768cdc2fce24d0881f3577e1
author Scott Chacon <schacon@gmail.com> 1223402504 -0700
committer Scott Chacon <schacon@gmail.com> 1223402504 -0700
```

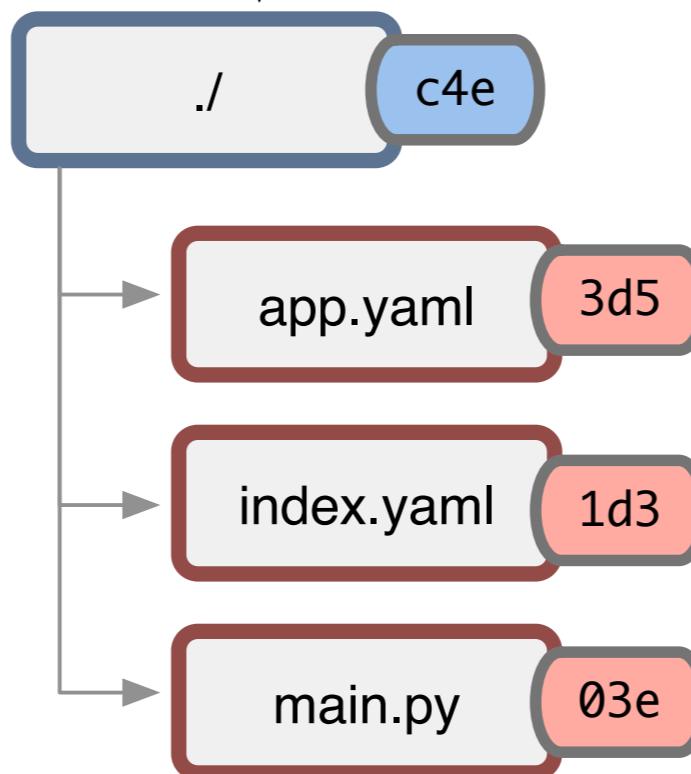
descriptive commit message

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

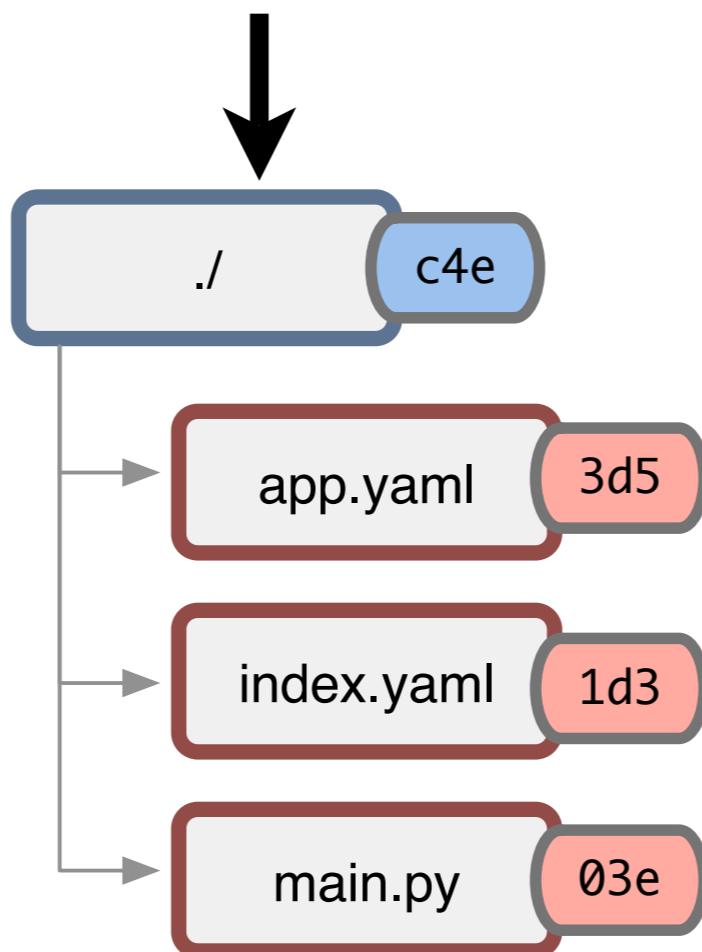
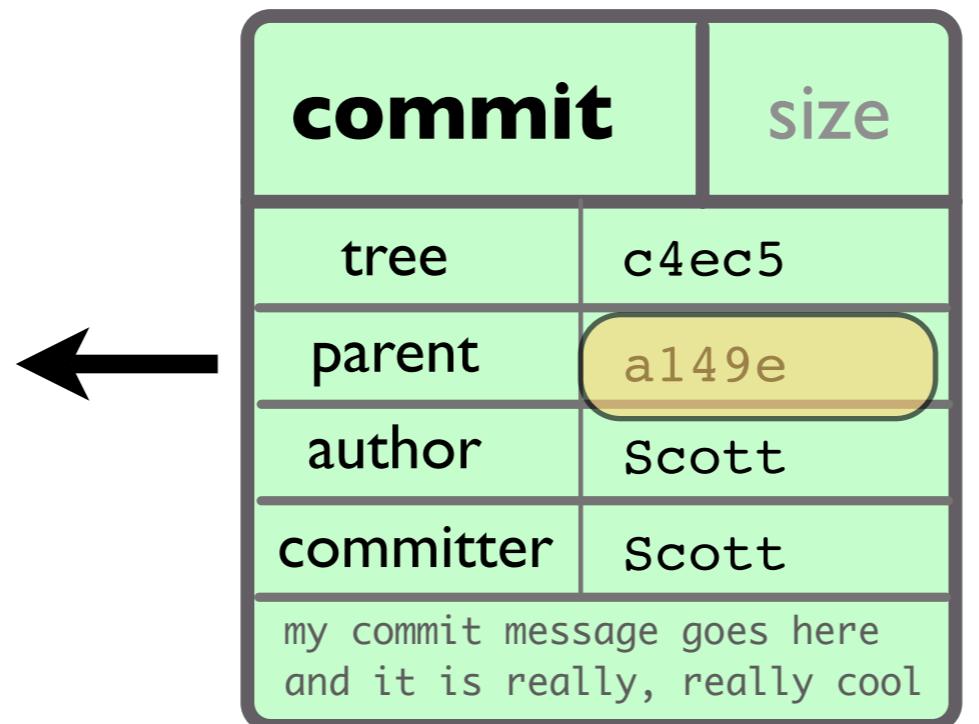
commit	size
tree	c4ec5
parent	a149e
author	Scott
committer	Scott
my commit message goes here and it is really, really cool	

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

commit	size
tree	c4ec5
parent	a149e
author	Scott
committer	Scott
my commit message goes here and it is really, really cool	

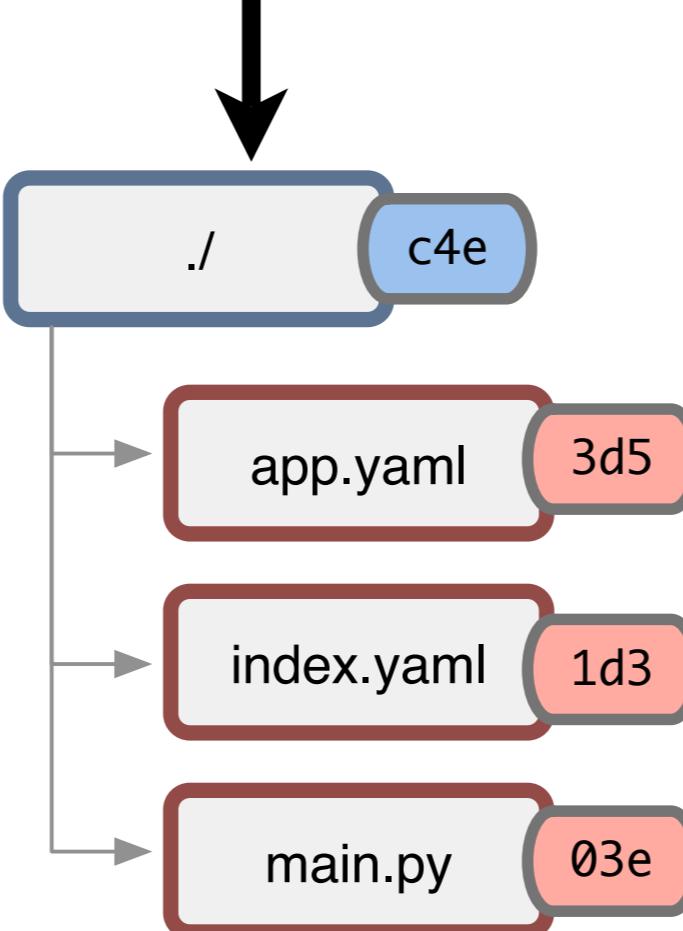
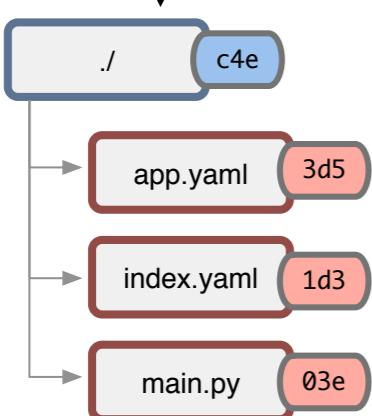


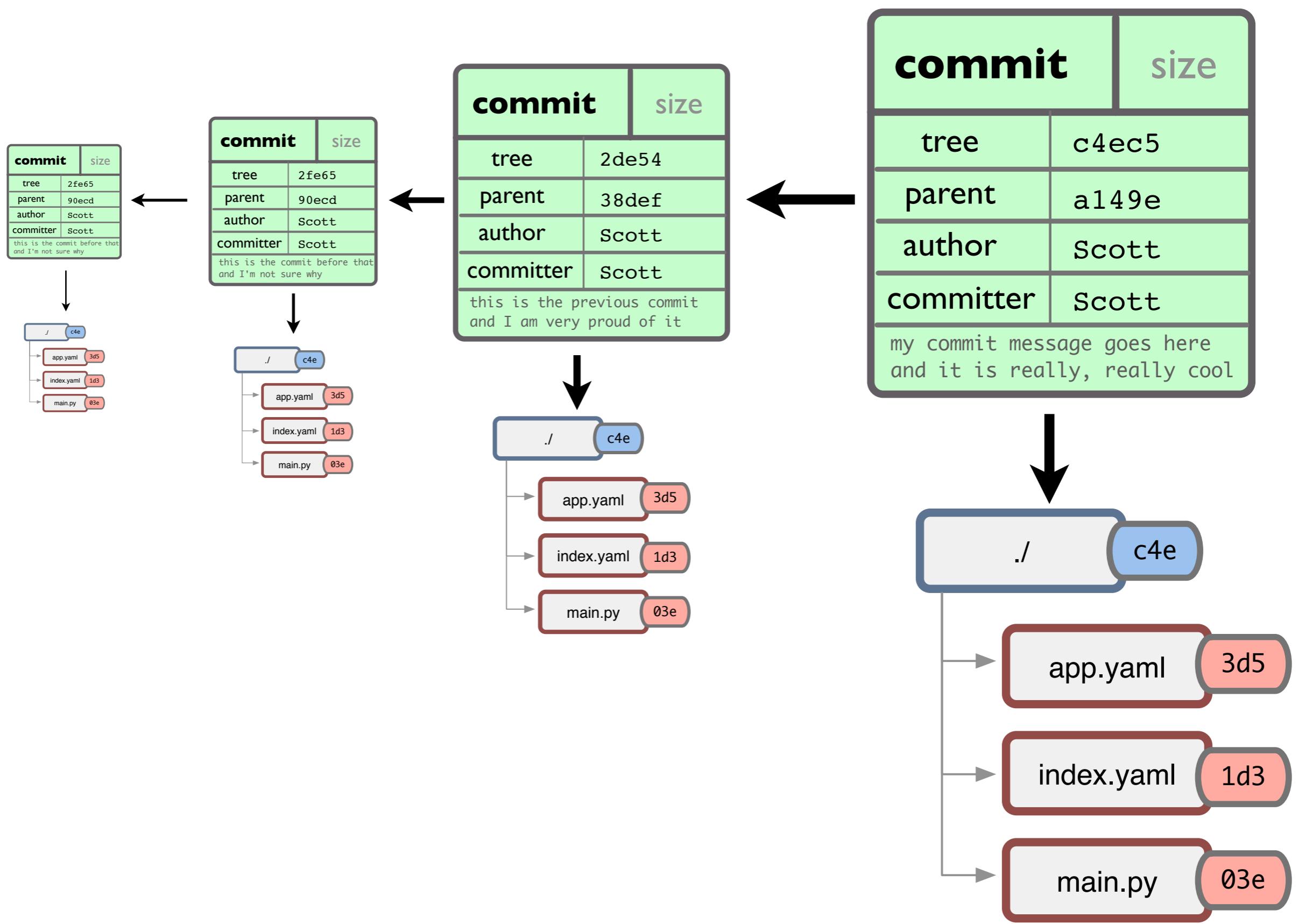
commit	size
tree	2de54
parent	38def
author	Scott
committer	Scott
this is the previous commit and I am very proud of it	

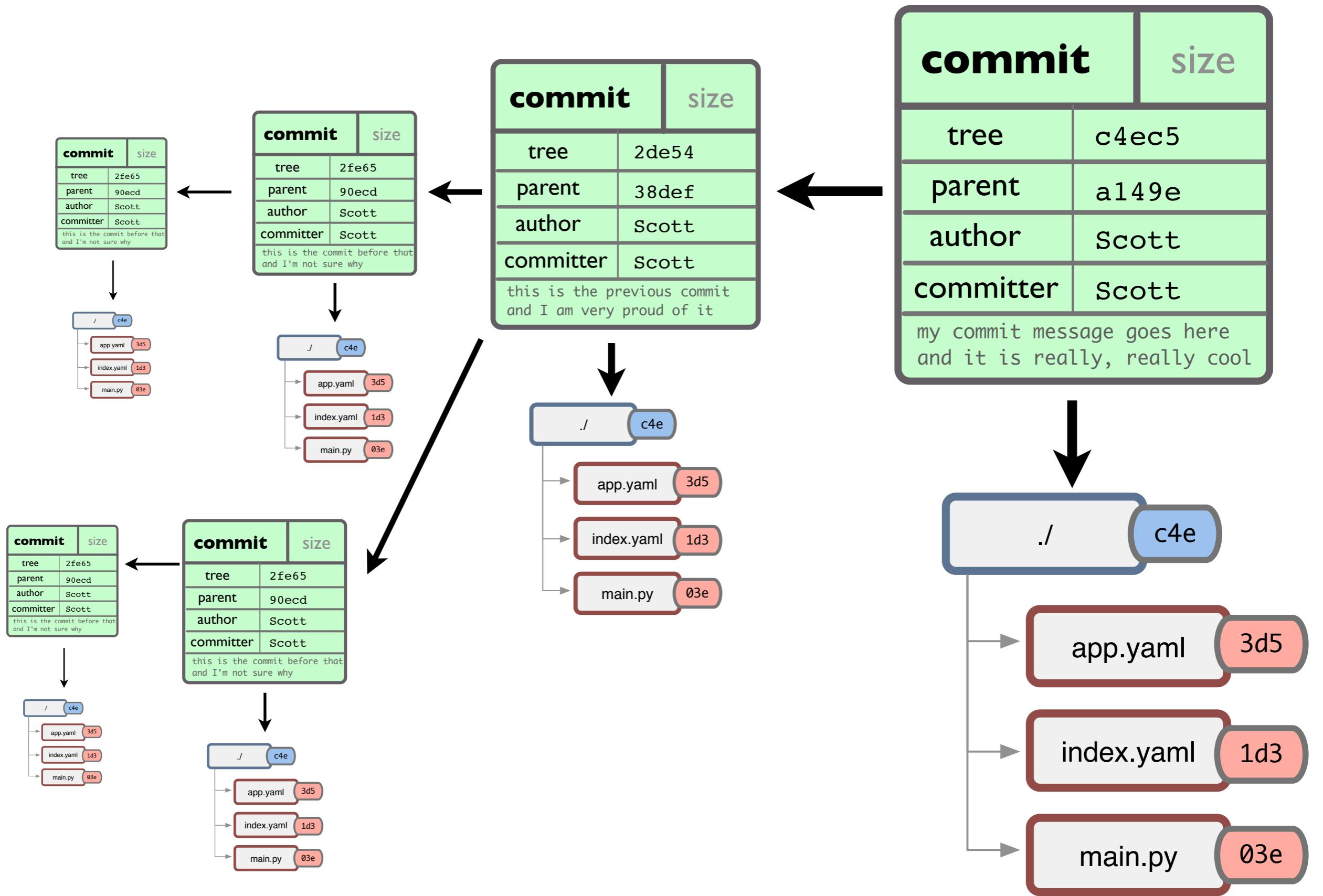


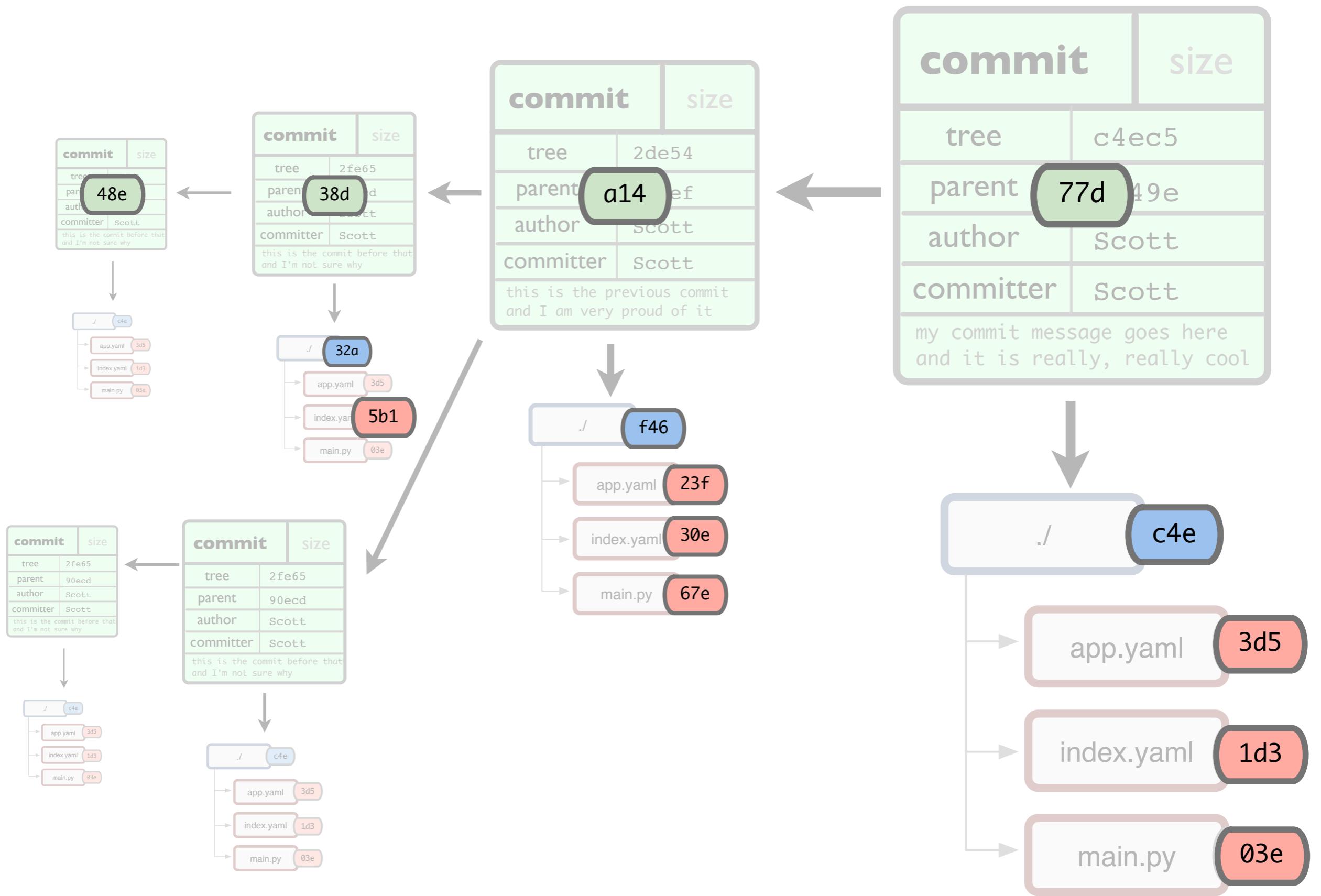
commit		size
tree	2de54	
parent	38def	
author	Scott	
committer	Scott	
this is the previous commit and I am very proud of it		

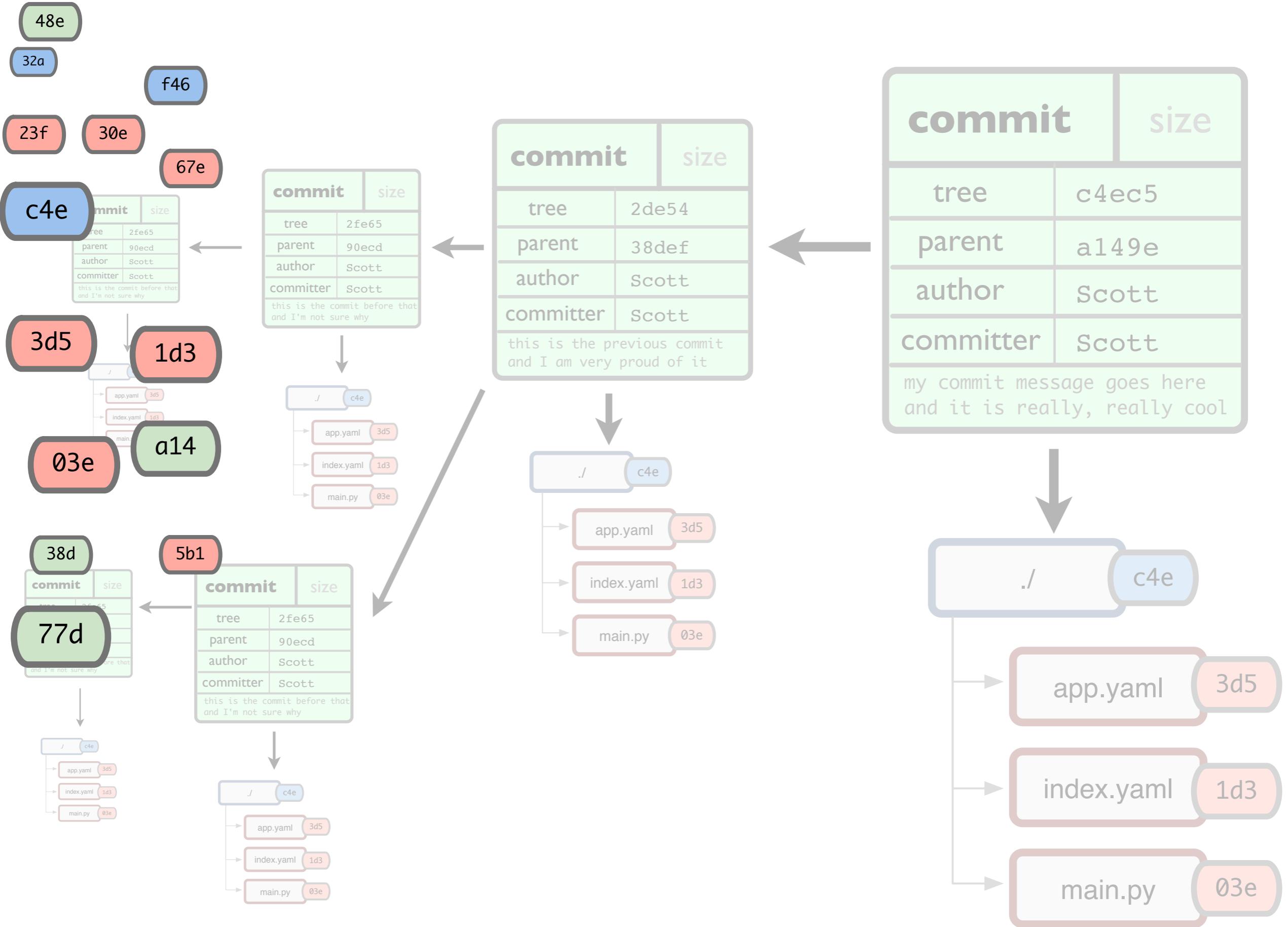
commit		size
tree	c4ec5	
parent	a149e	
author	Scott	
committer	Scott	
my commit message goes here and it is really, really cool		











Repository

5b1	32a	c36
1d3	f46	3d4
ffe	6fe	ae6
38d	23f	03e
254	30e	5b1
a14	67e	1d3
3d5	735	d23
c4e	c4e	48e
77d	de3	2d3

Repository

5b1	32a	c36
1d3	f46	3d4
ffe	6fe	ae6
38d	23f	03e
254	30e	5b1
a14	67e	1d3
3d5	735	d23
c4e	c4e	48e
77d	de3	2d3

git checkout ae635

Repository

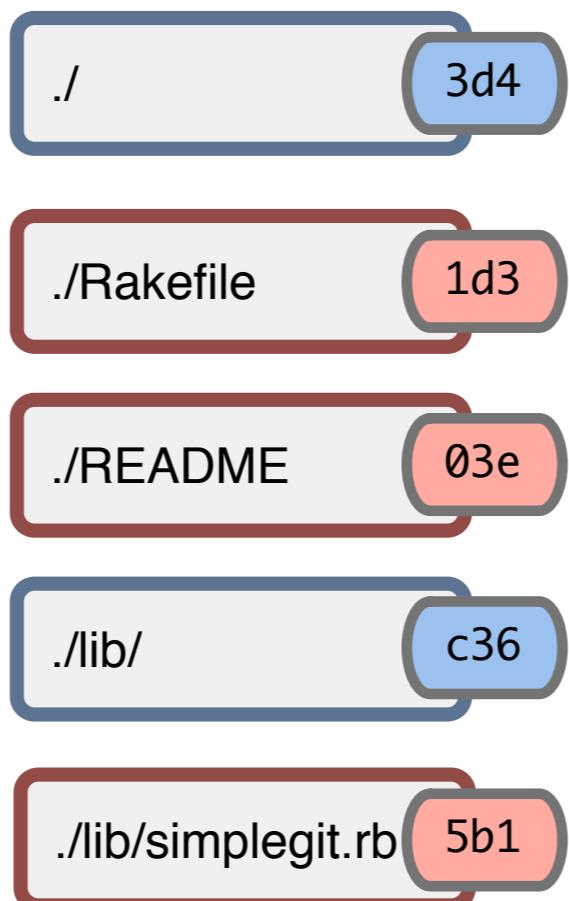
5b1	32a	c36
1d3	f46	3d4
ffe	6fe	ae6
38d	23f	03e
254	30e	5b1
a14	67e	1d3
3d5	735	d23
c4e	c4e	48e
77d	de3	2d3

git checkout ae635

Repository



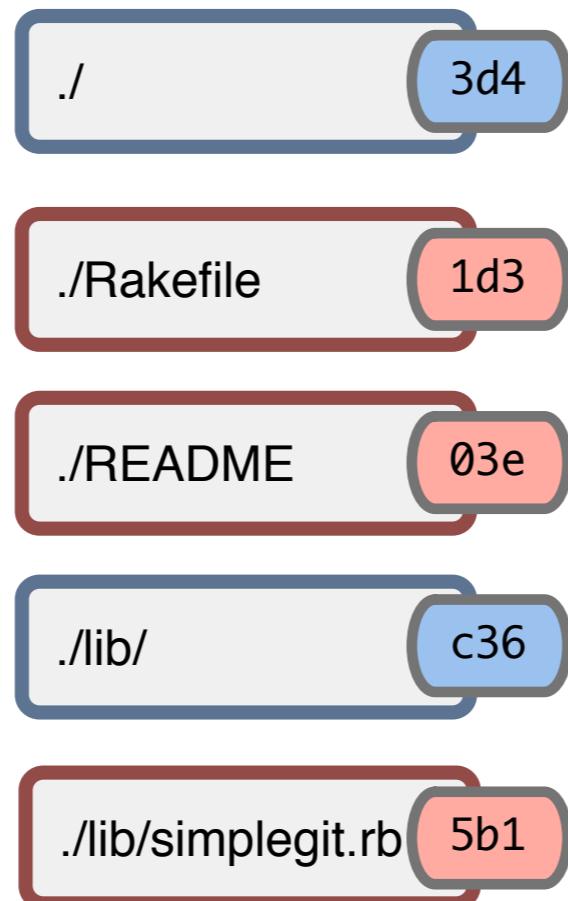
Index



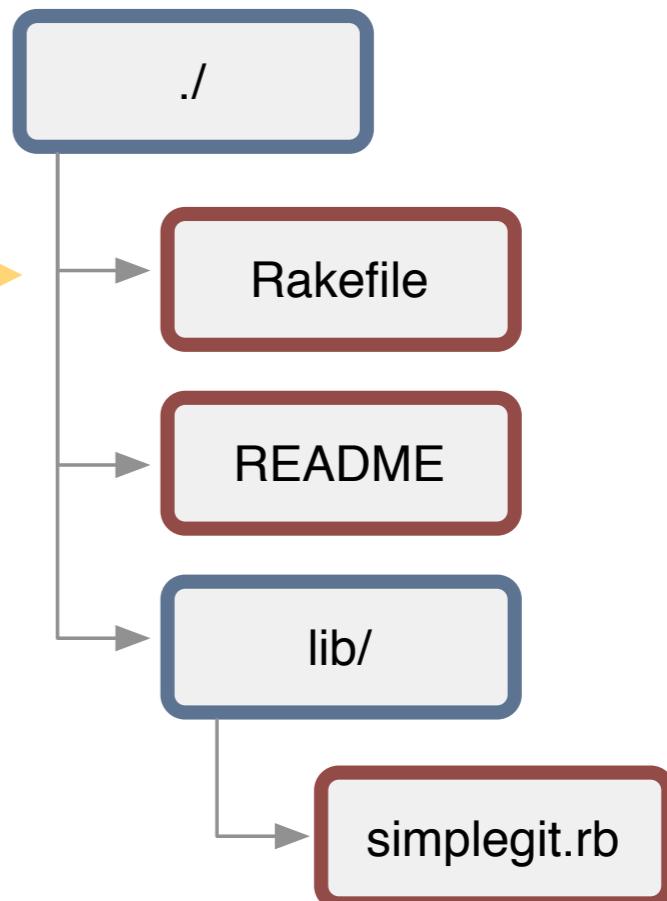
Repository



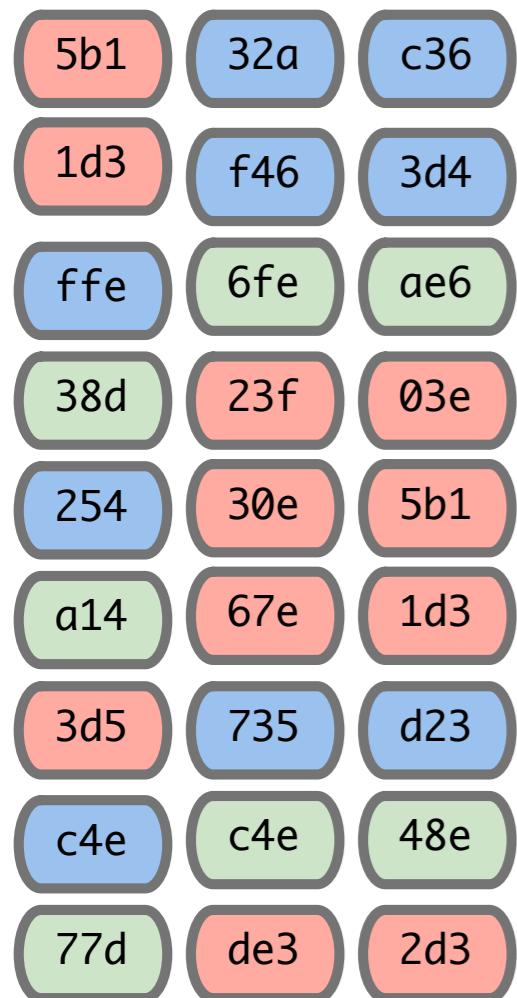
Index



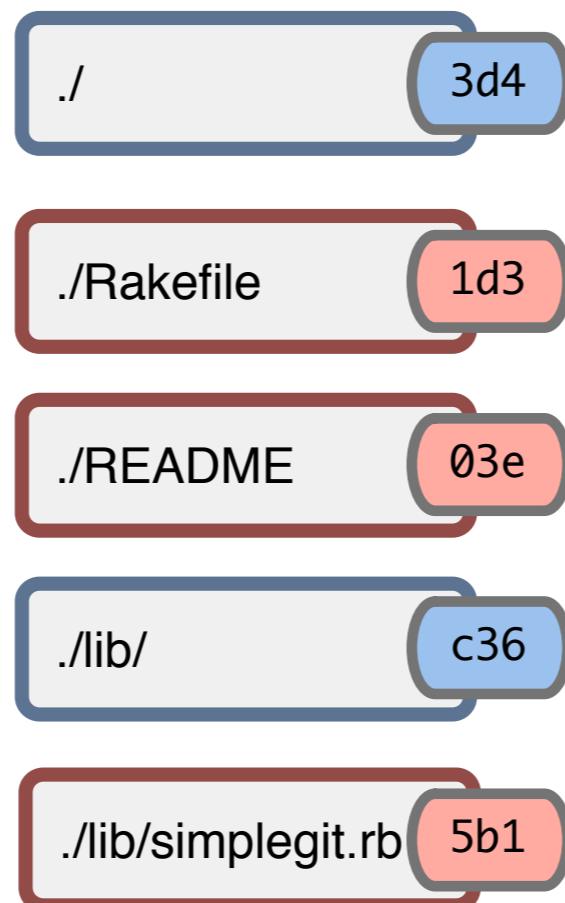
Working Directory



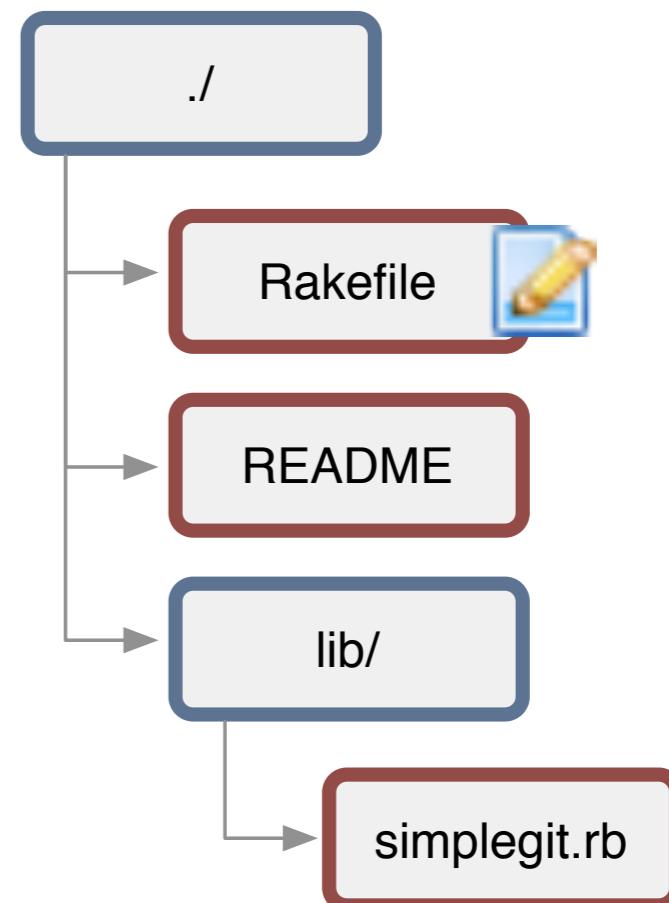
Repository



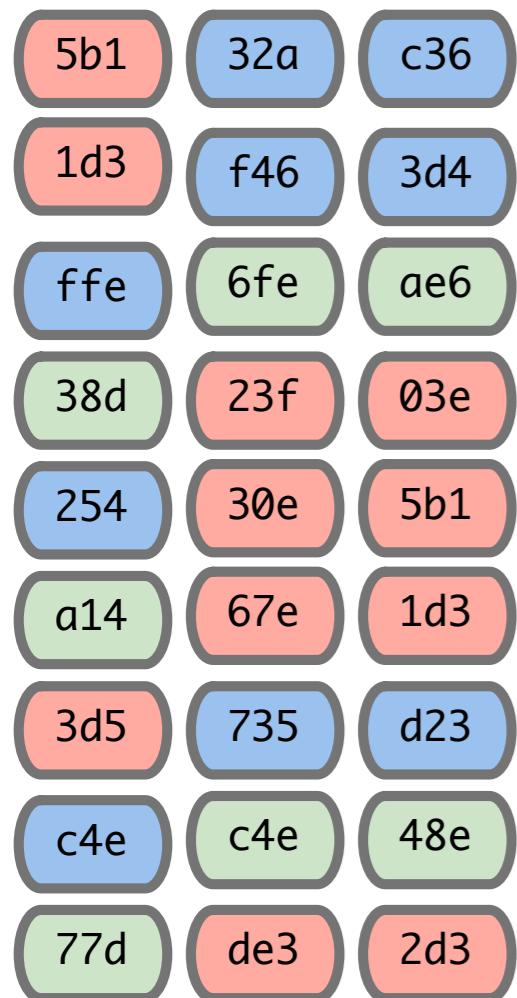
Index



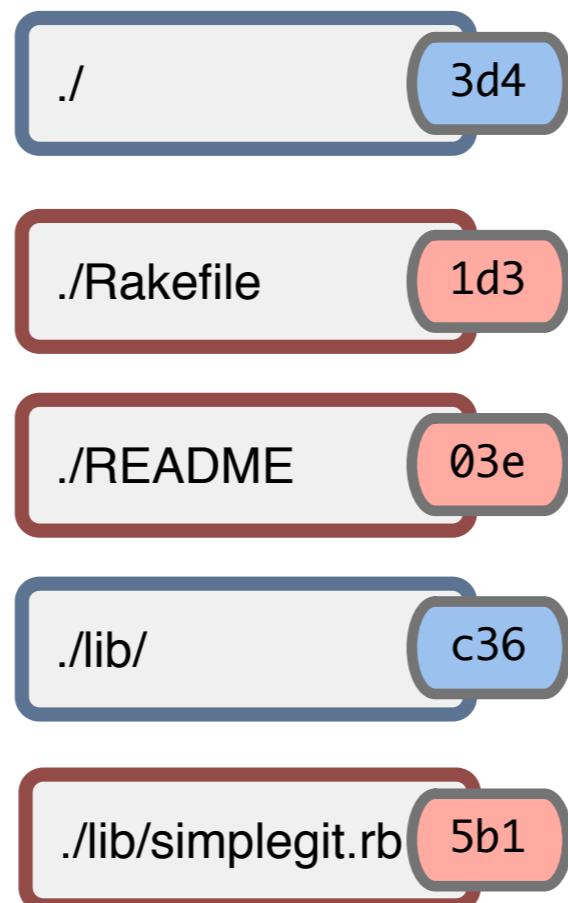
Working Directory



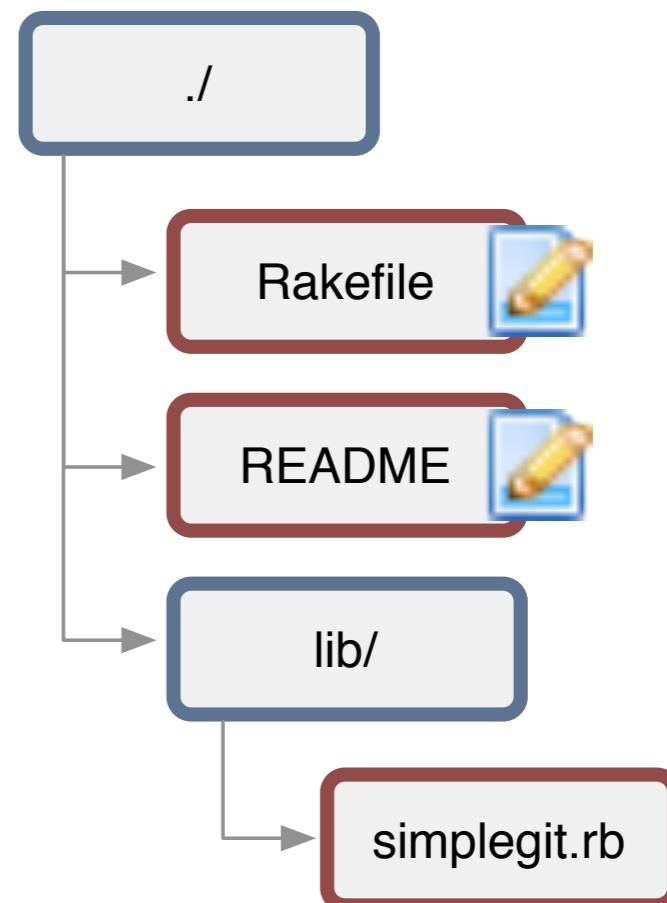
Repository



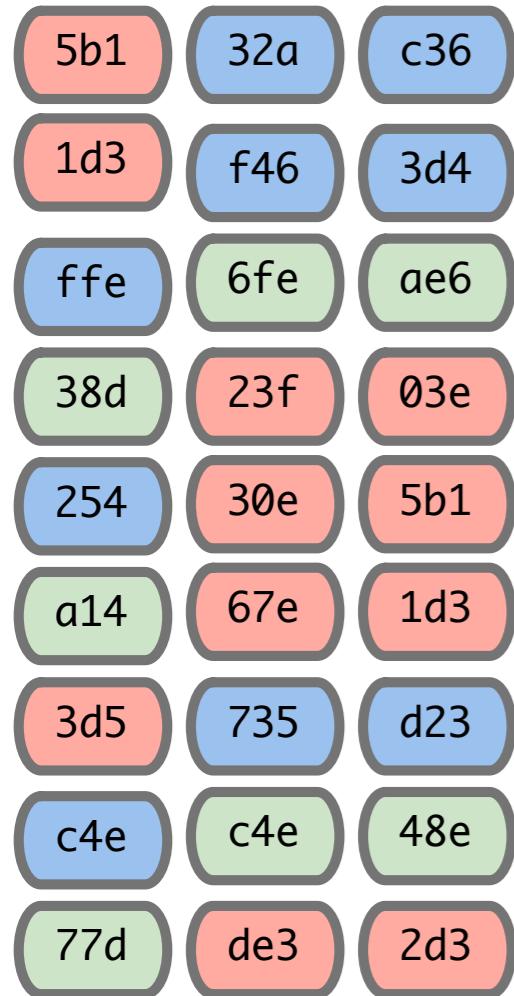
Index



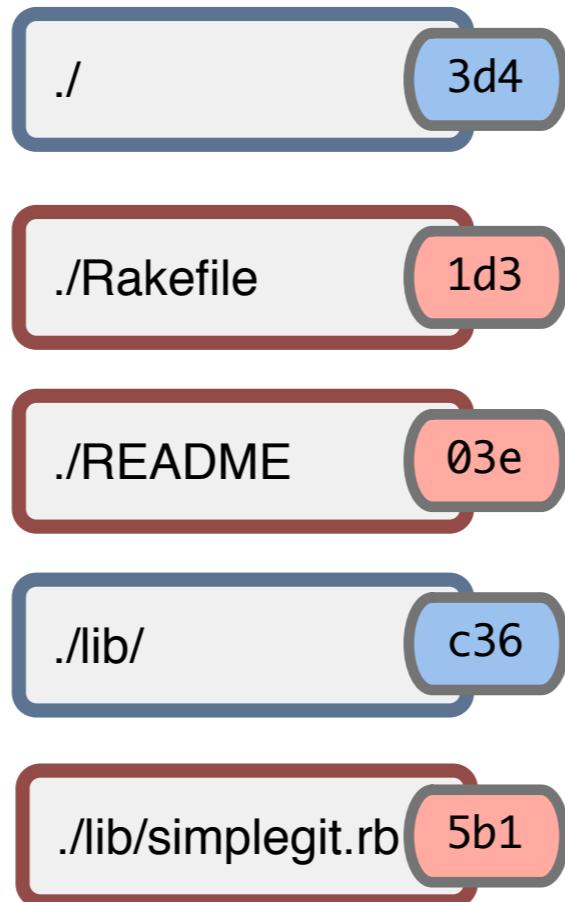
Working Directory



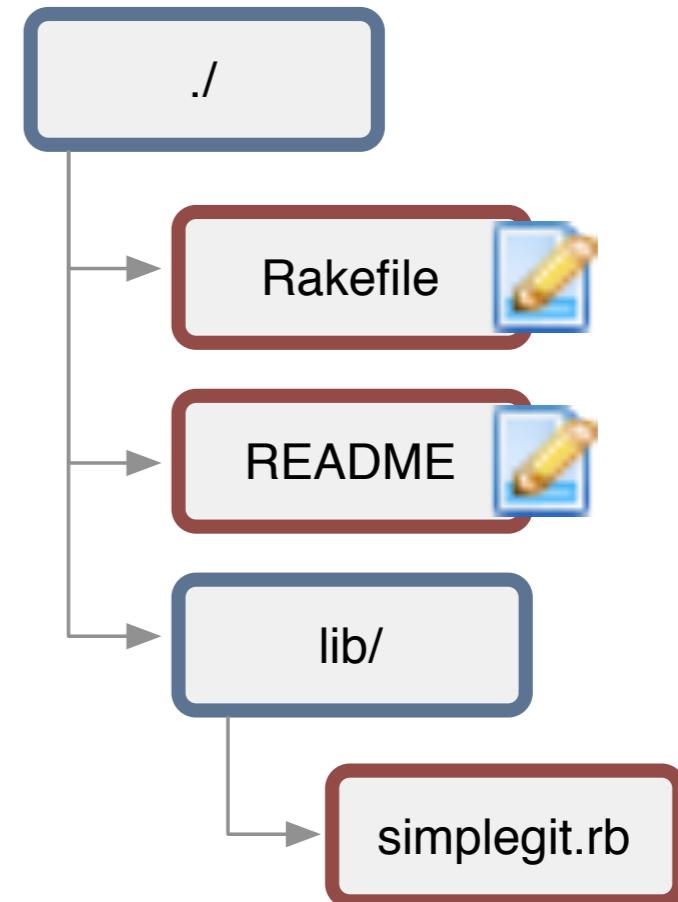
Repository



Index



Working Directory

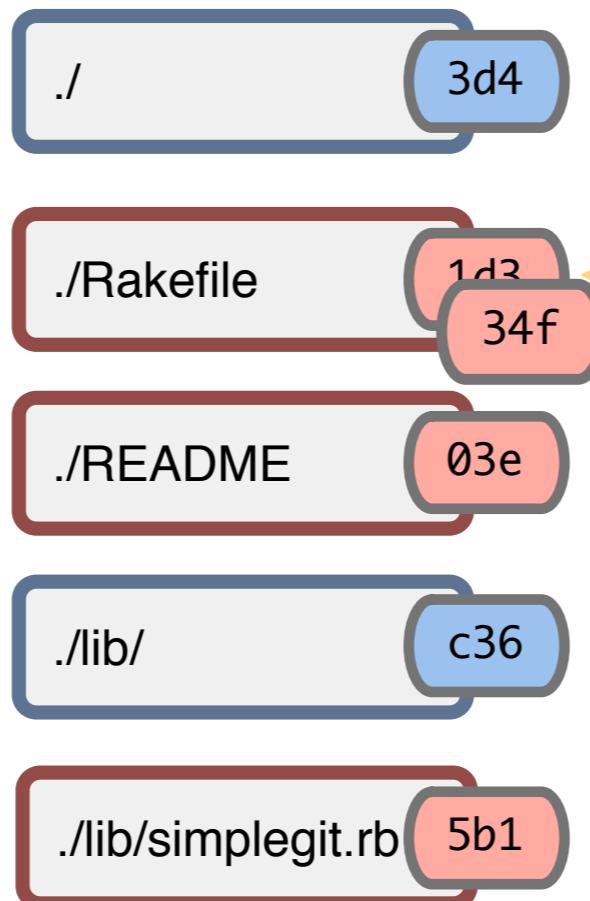


git add

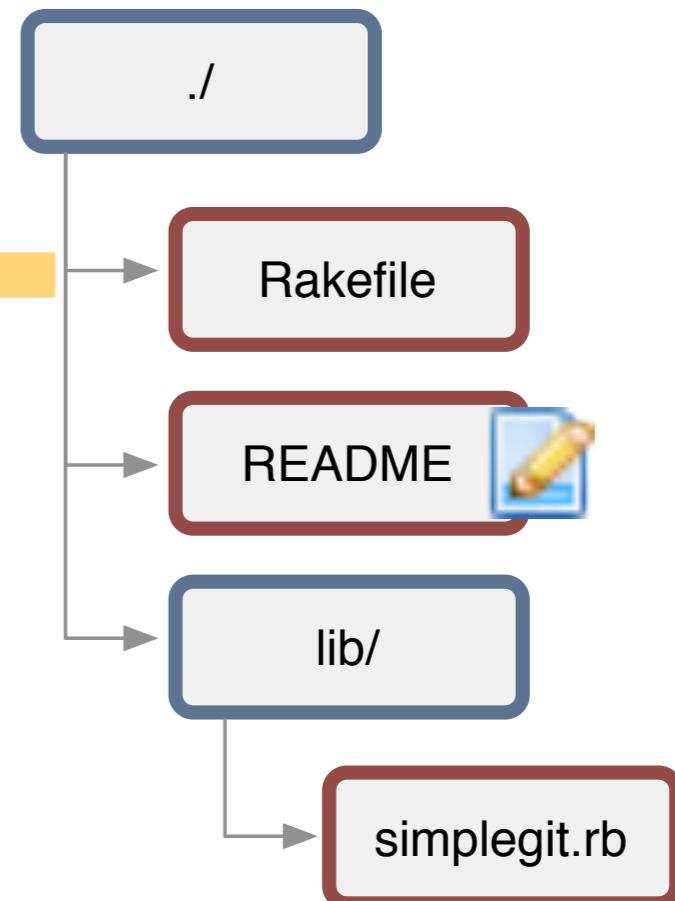
Repository



Index



Working Directory



git add

Repository

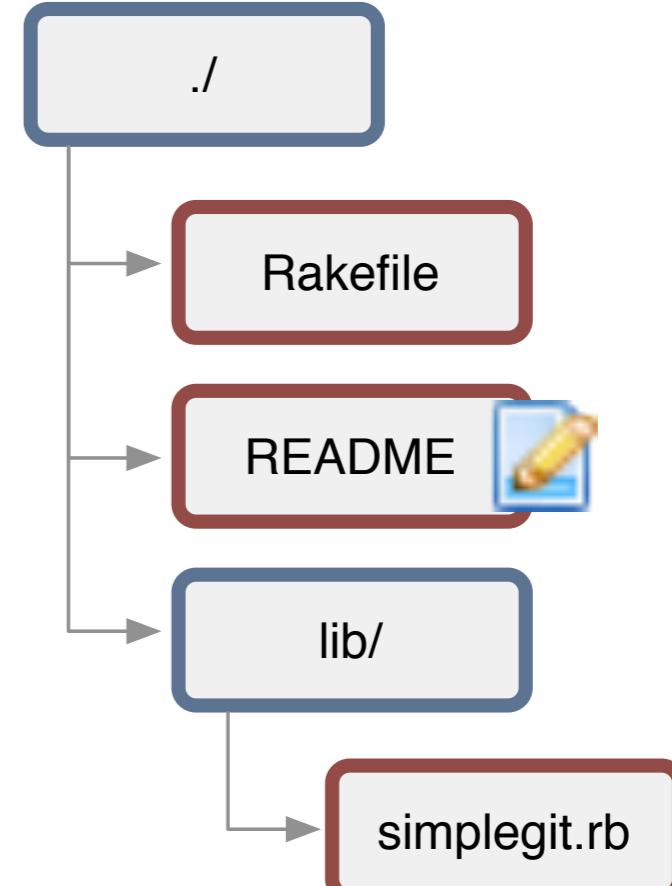


Index



git commit

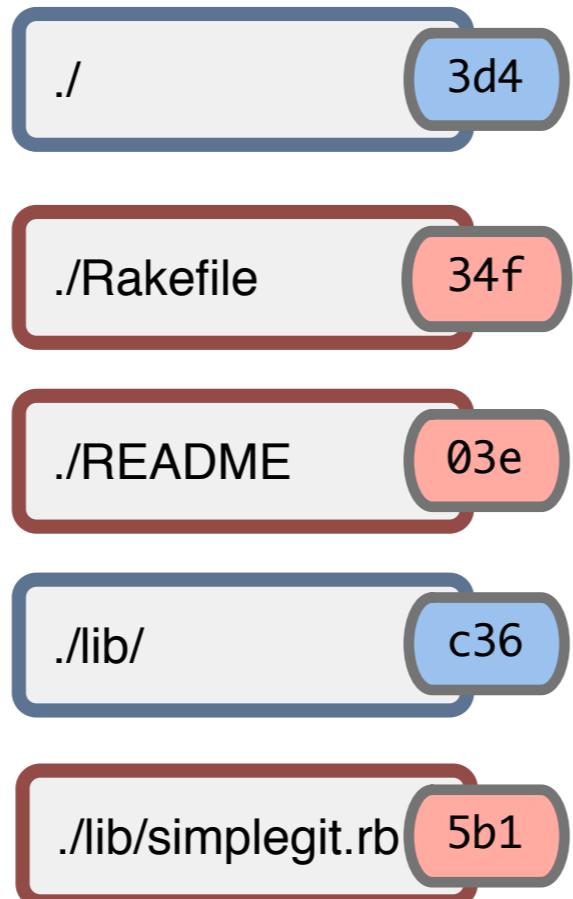
Working Directory



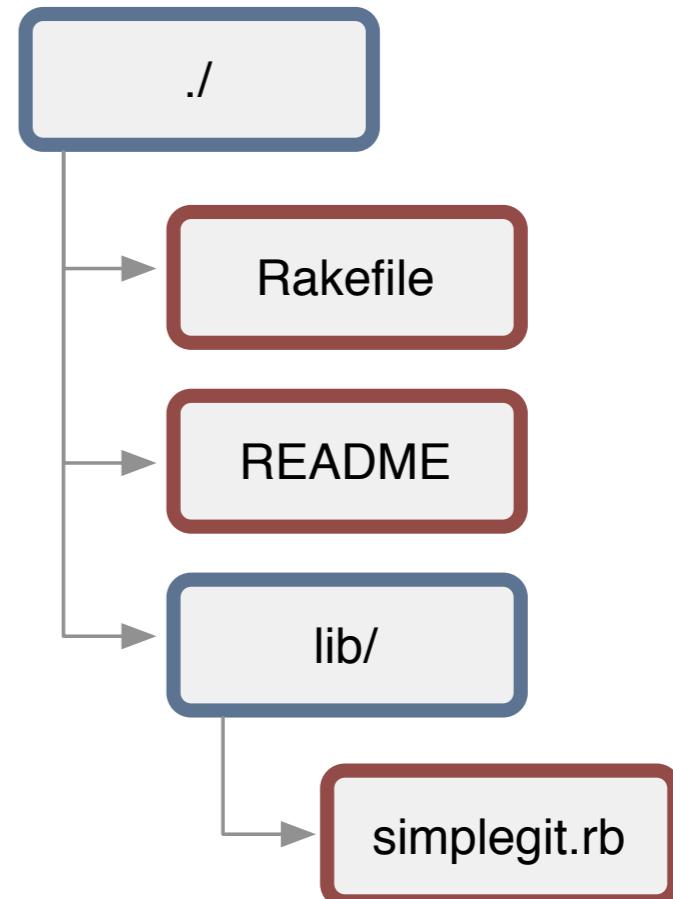
Repository



Index



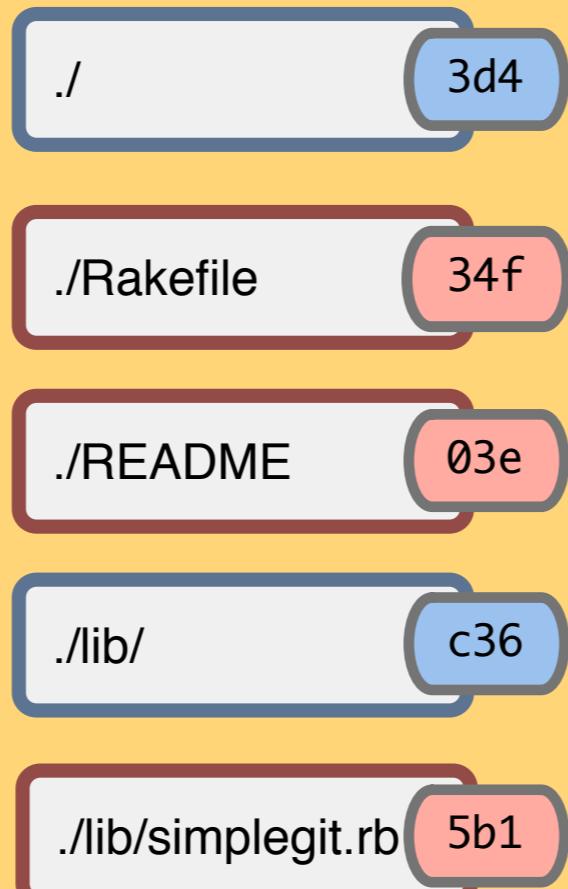
Working Directory



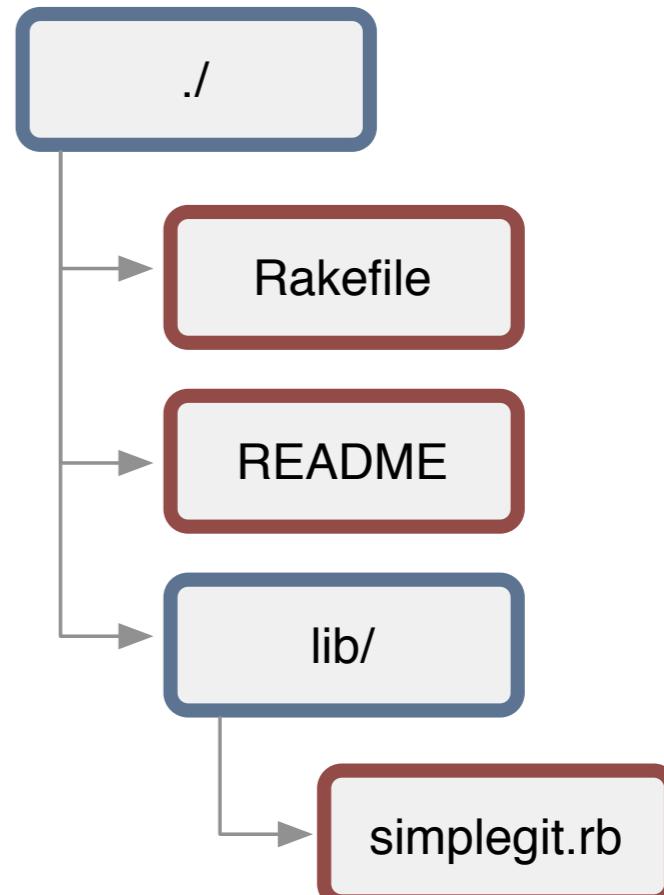
Repository



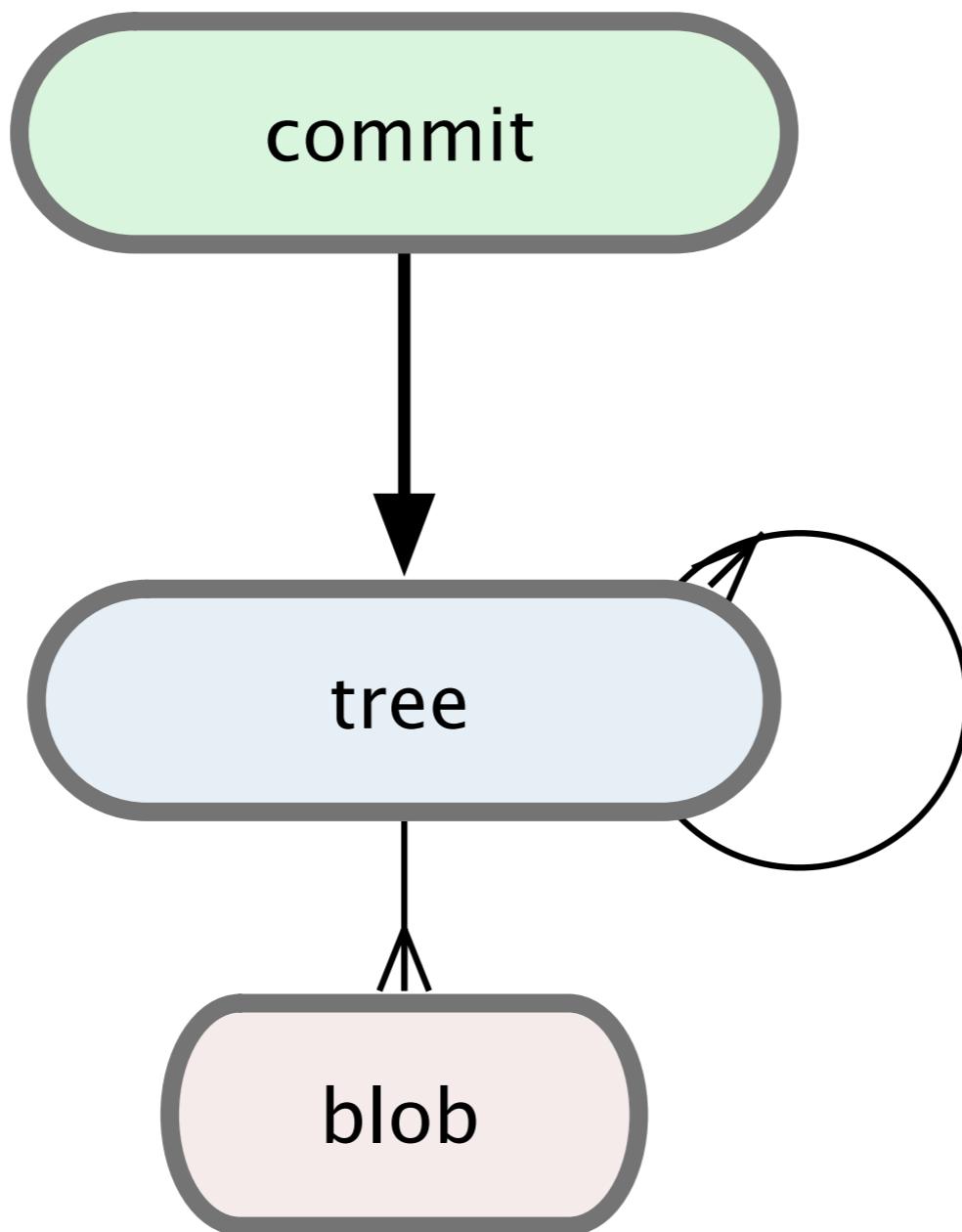
Index



Working Directory



object model



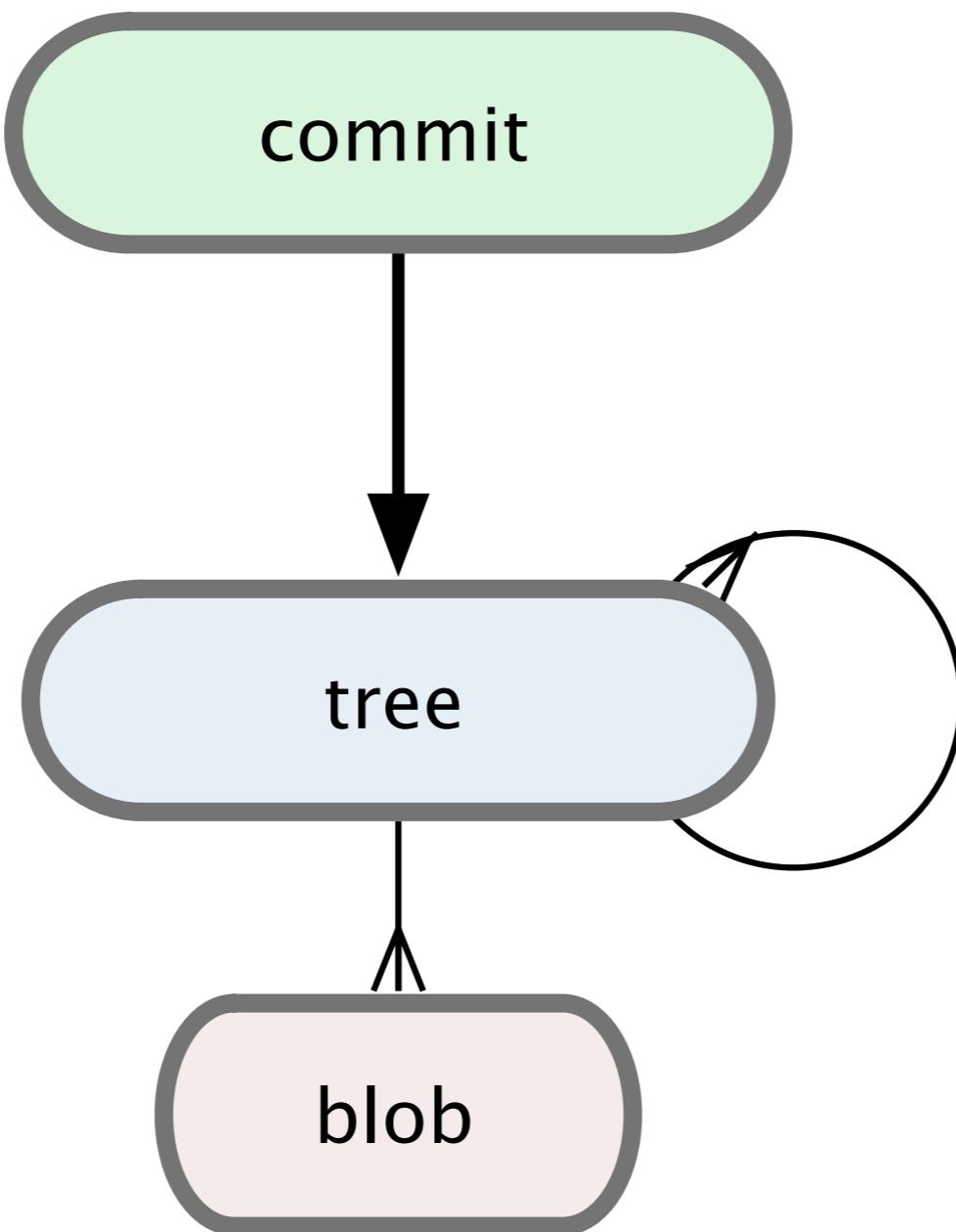
Branching and Merging

object model

pointer to a
snapshot

directory list

file contents

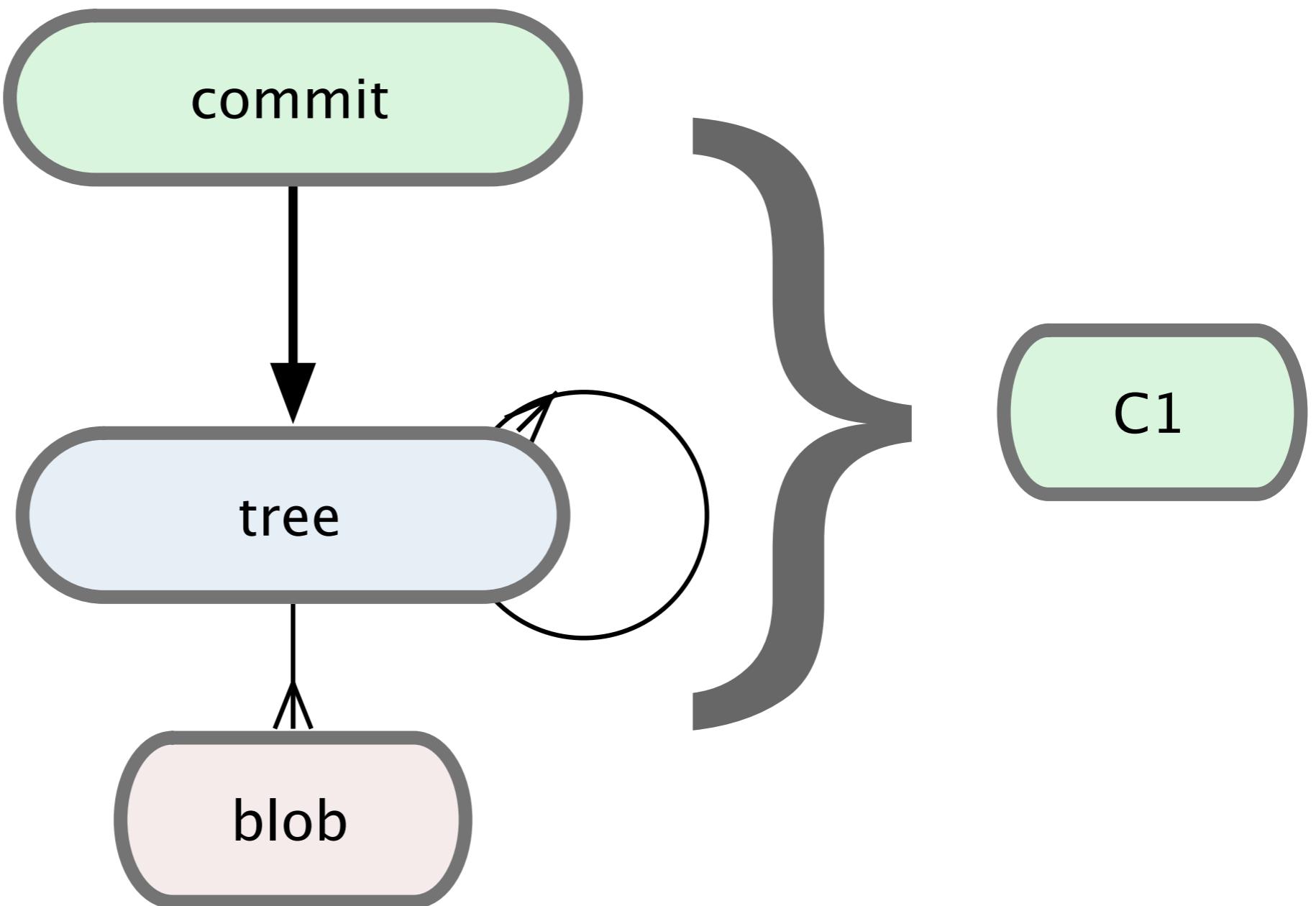


object model

pointer to a
snapshot

directory list

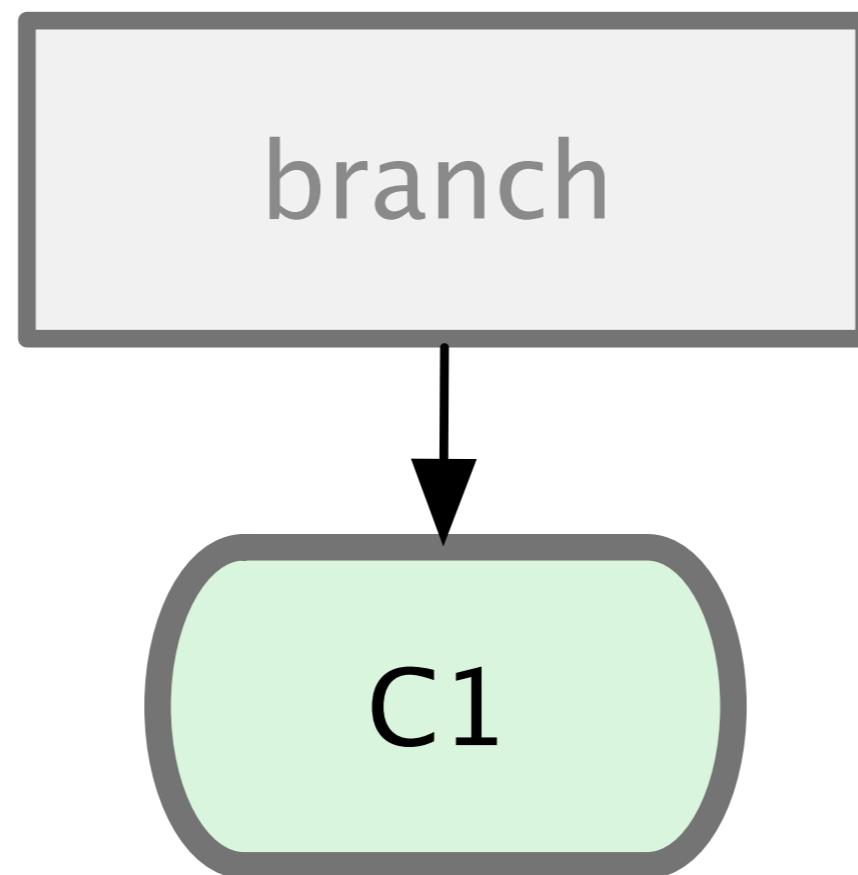
file contents



branches

branches

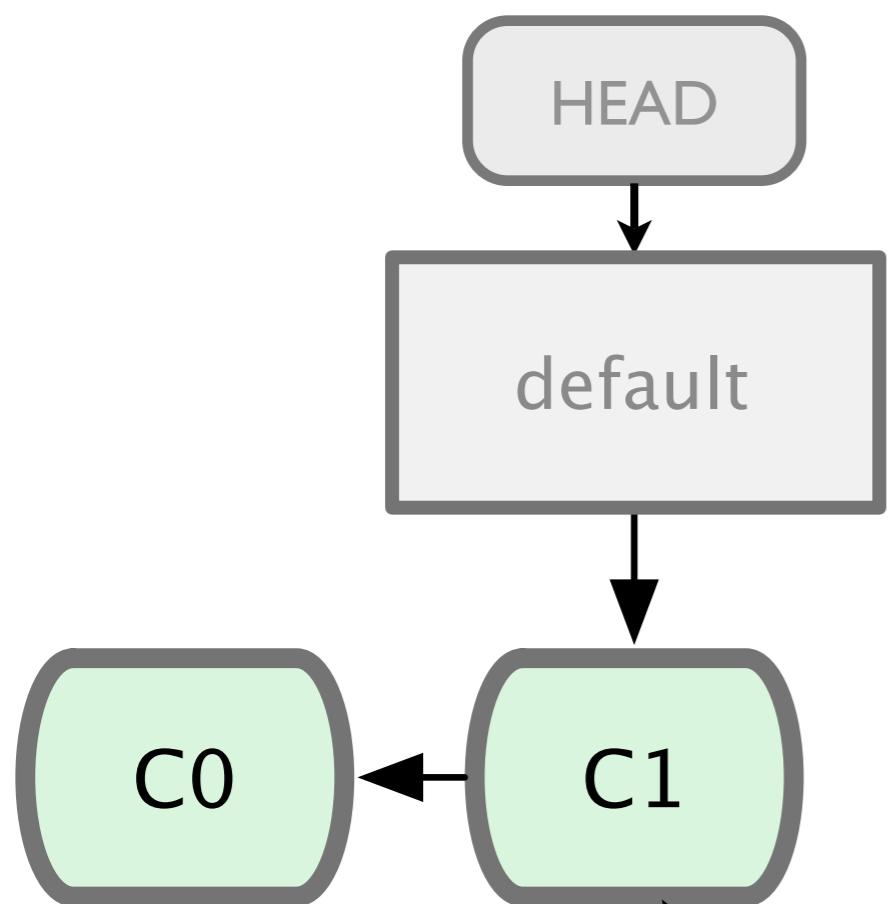
lightweight, movable
pointers to a commit

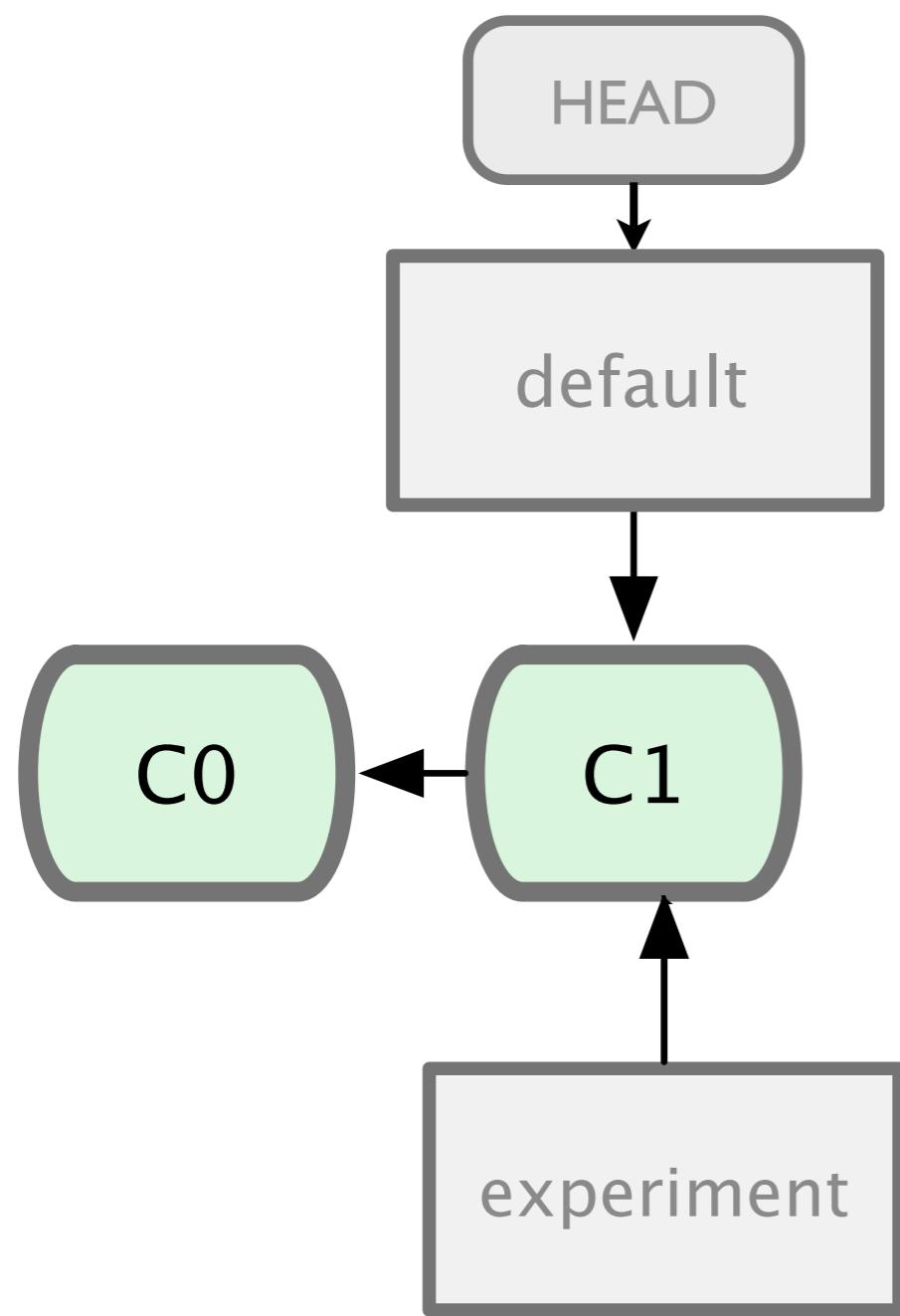


branching

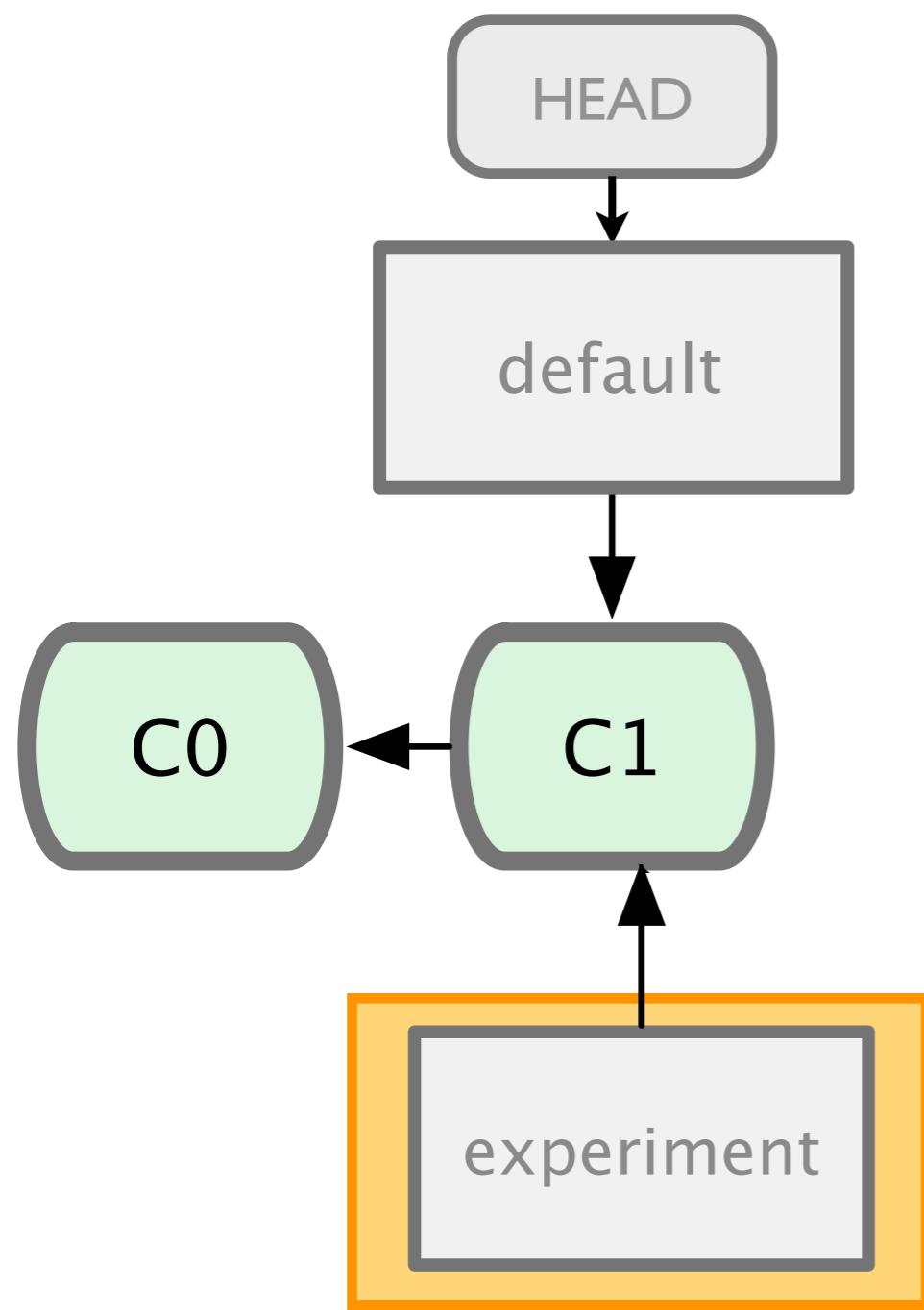
git branch

git checkout

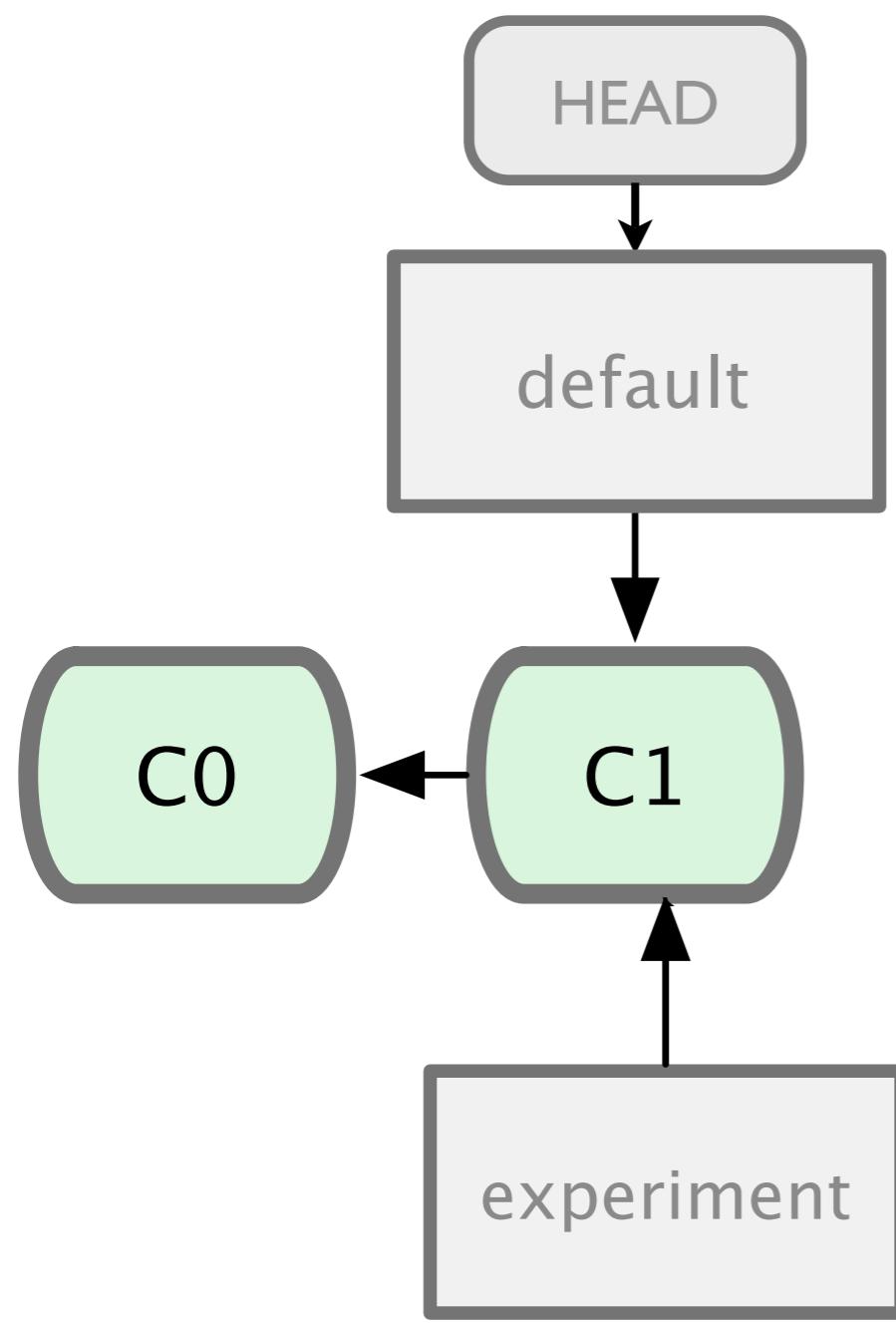




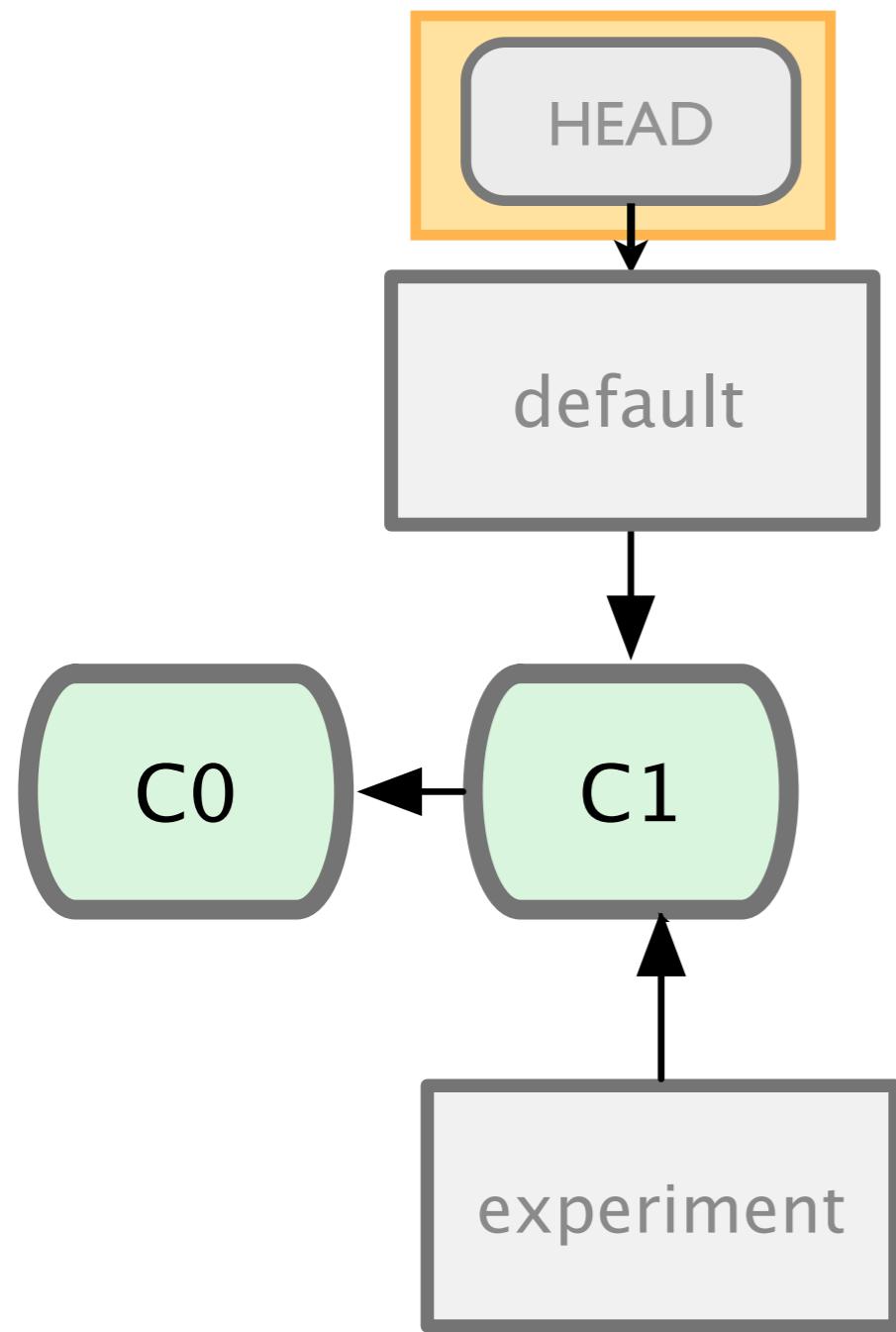
git branch experiment



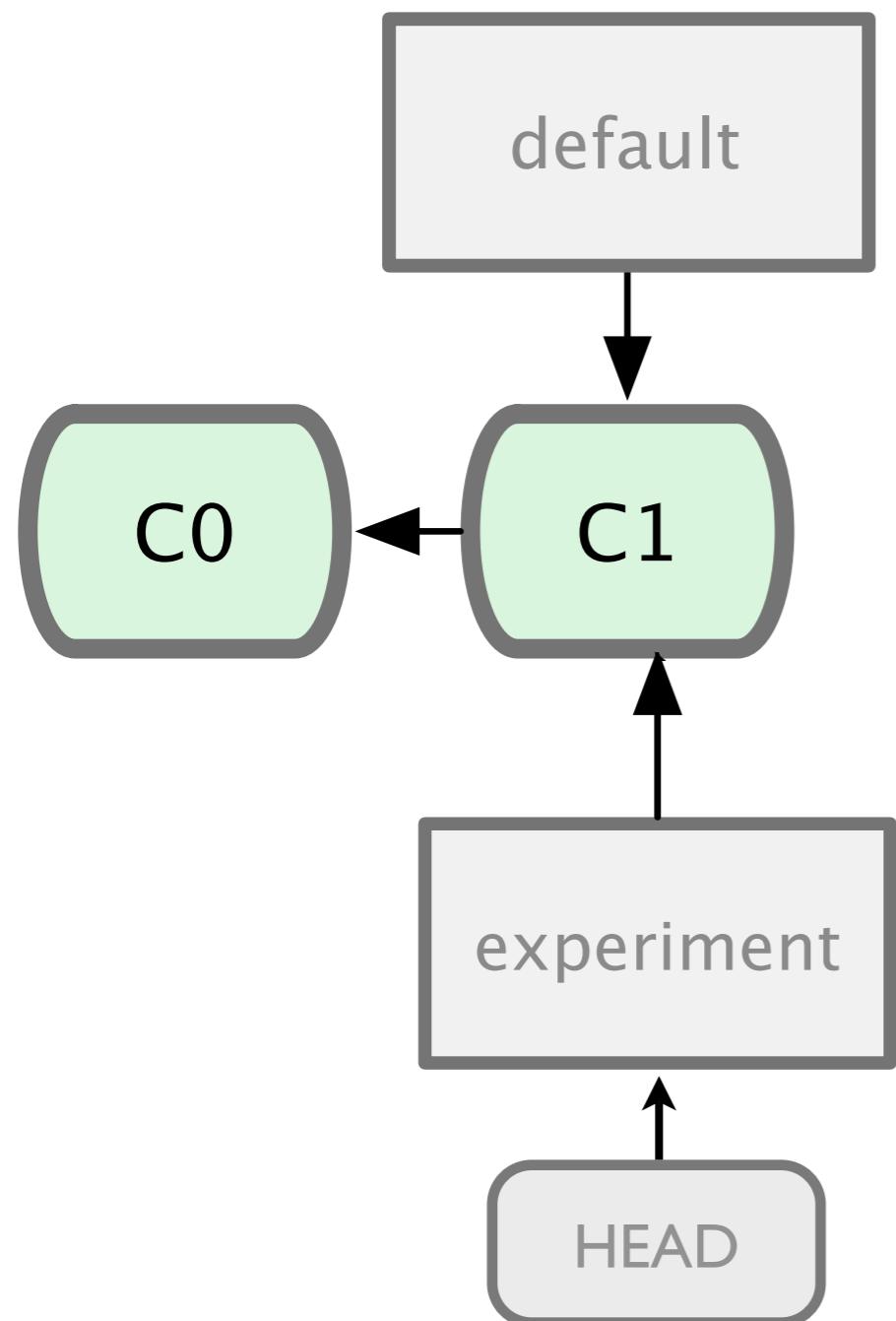
git branch experiment



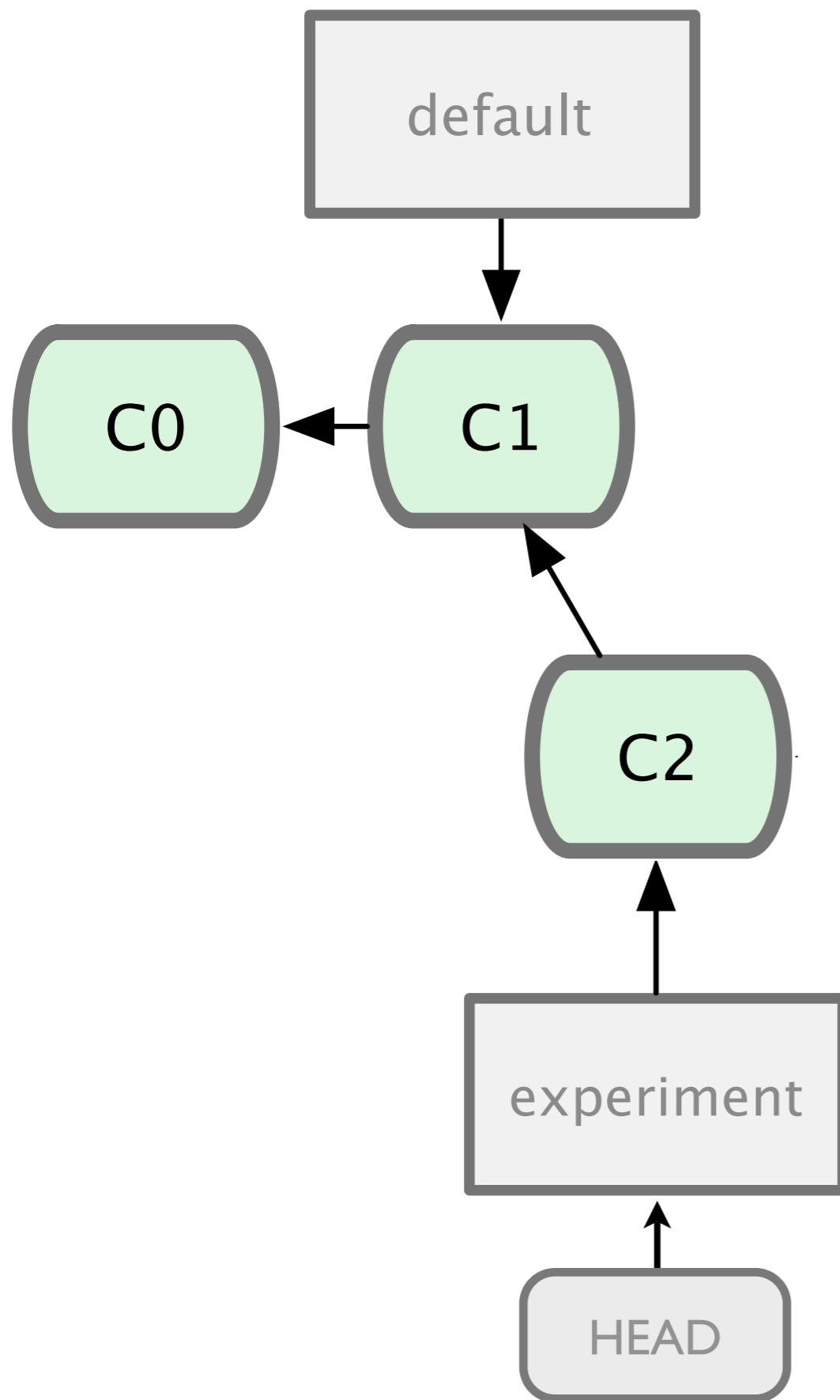
```
$ git branch  
* default  
  experiment
```



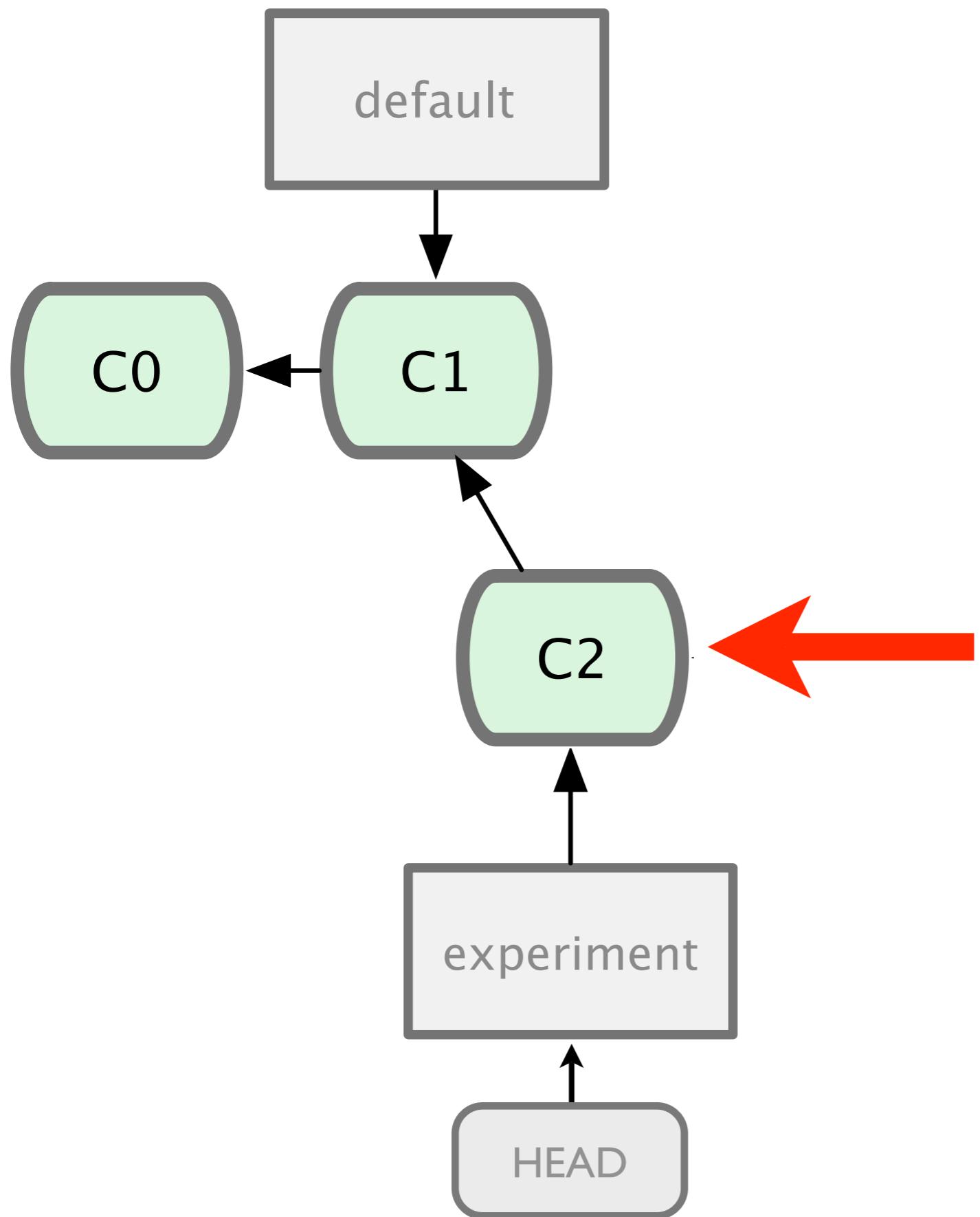
```
$ git branch  
* default  
  experiment
```



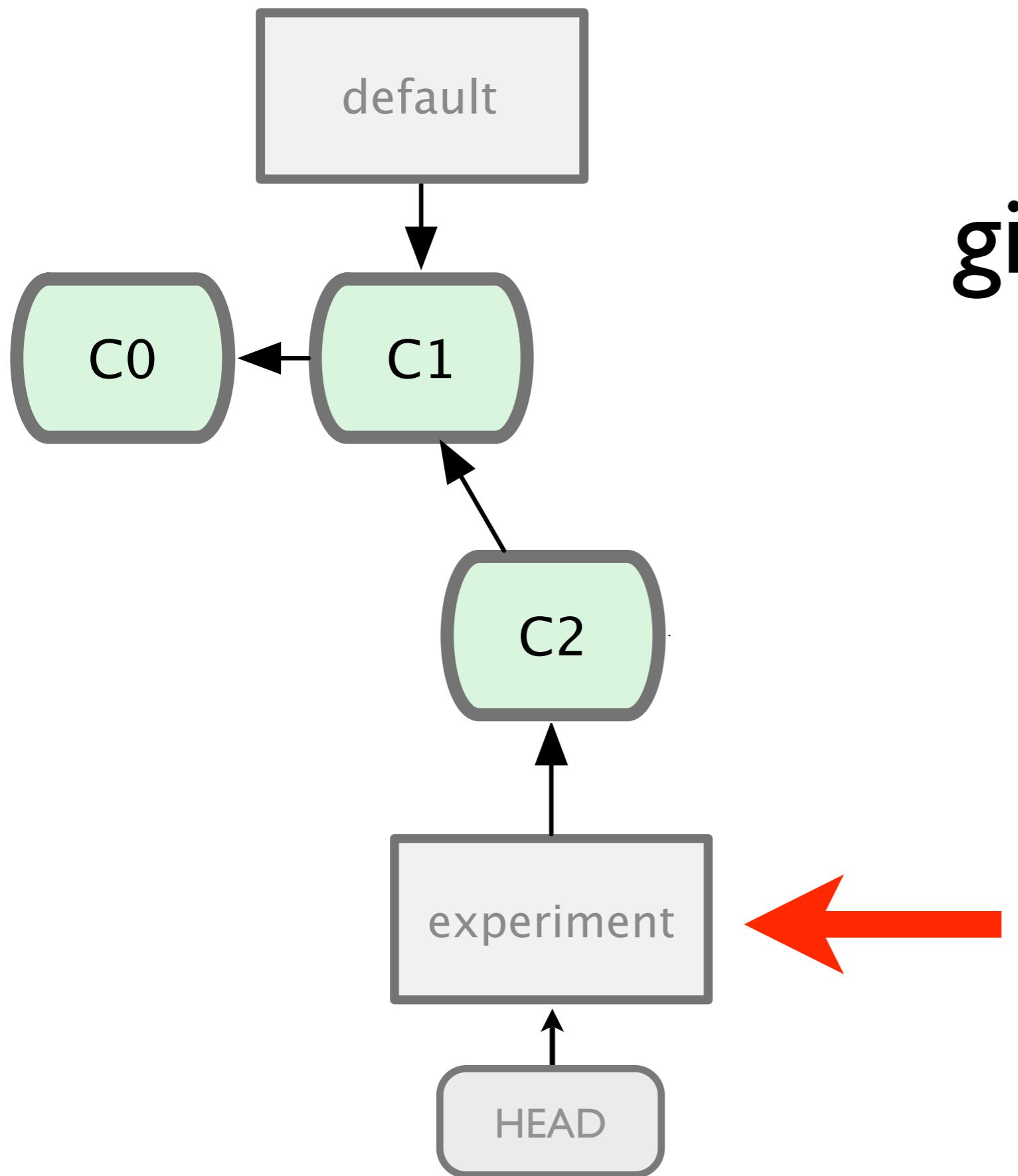
git checkout experiment



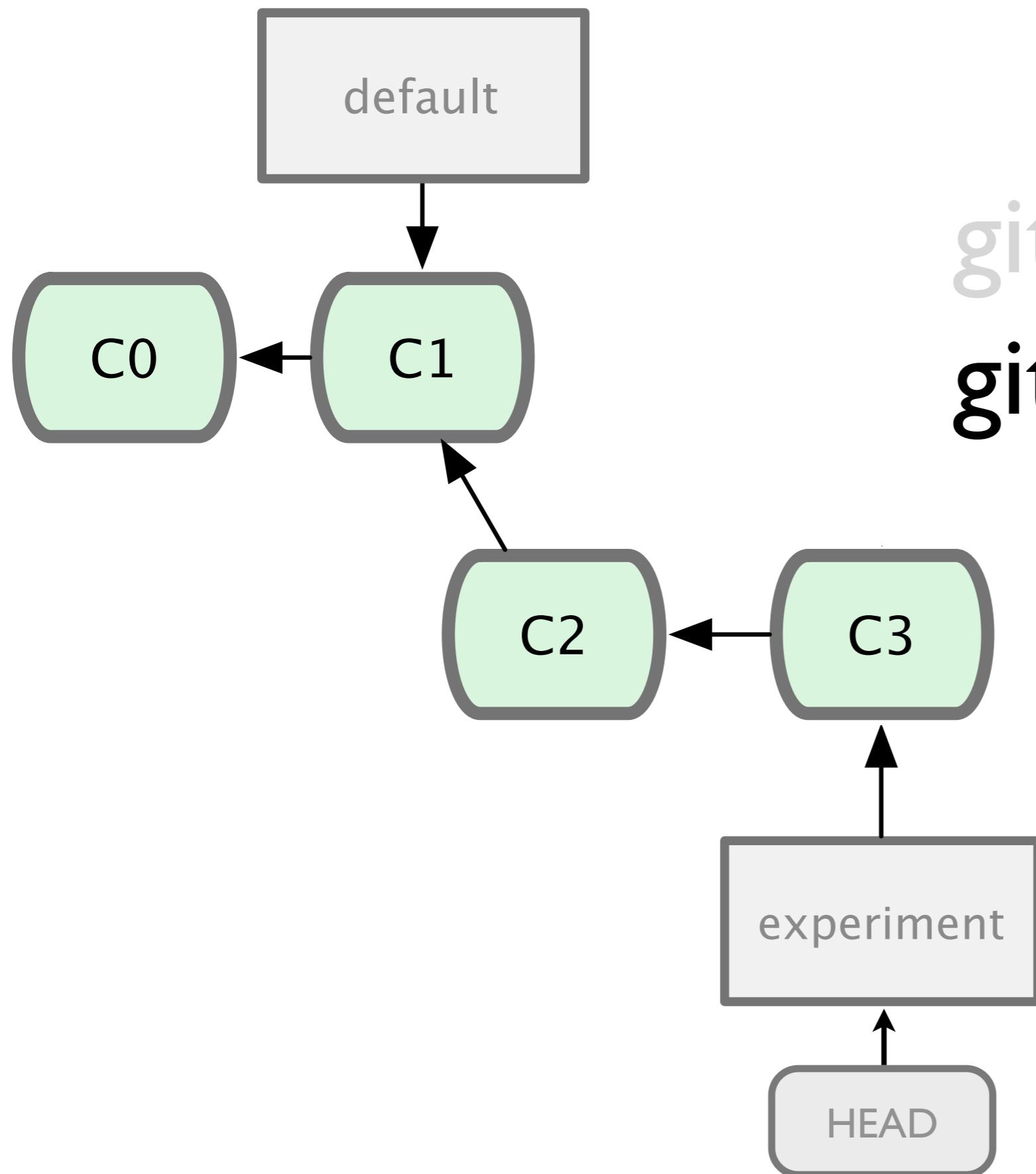
git commit



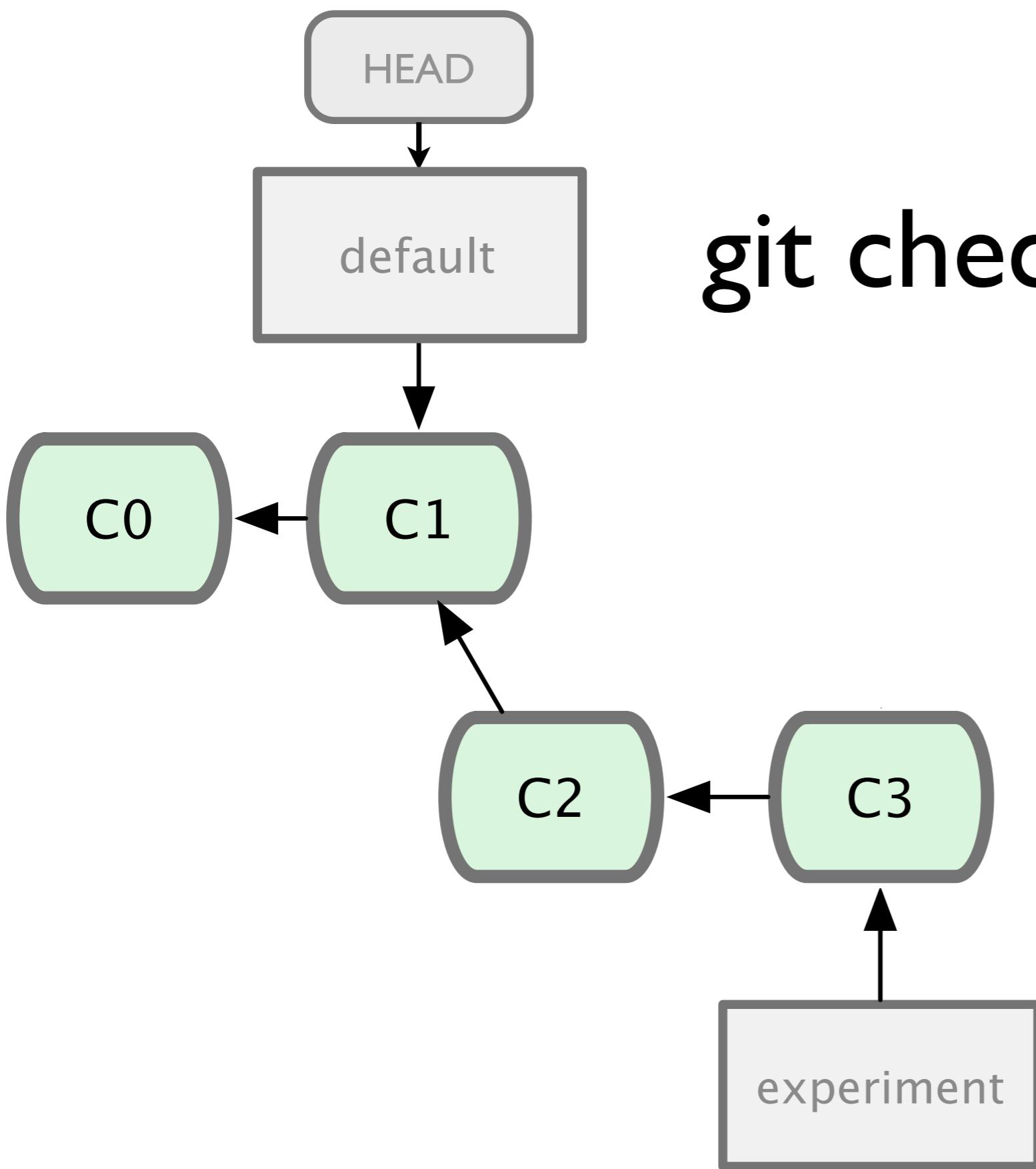
git commit



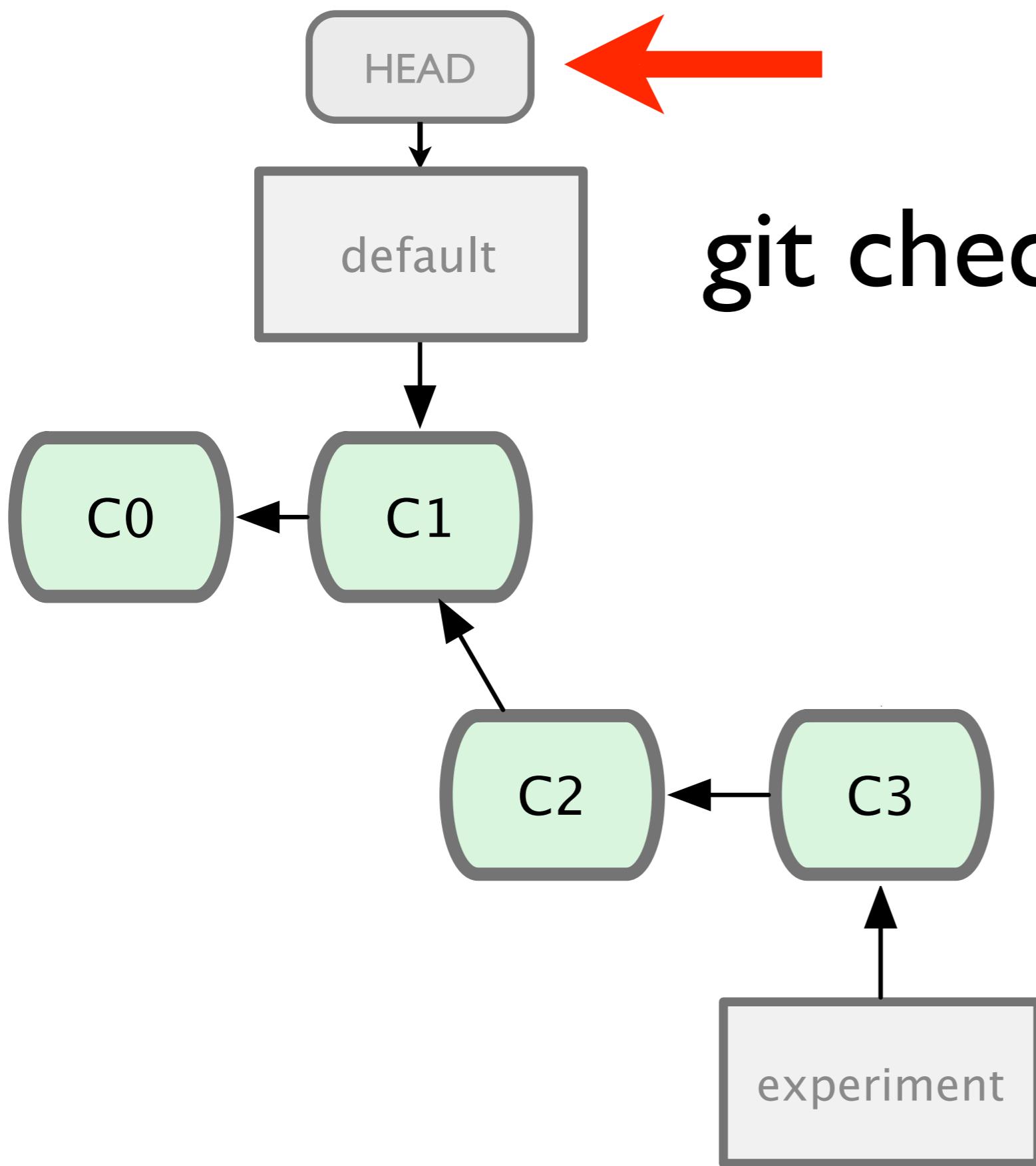
git commit



git commit
git commit

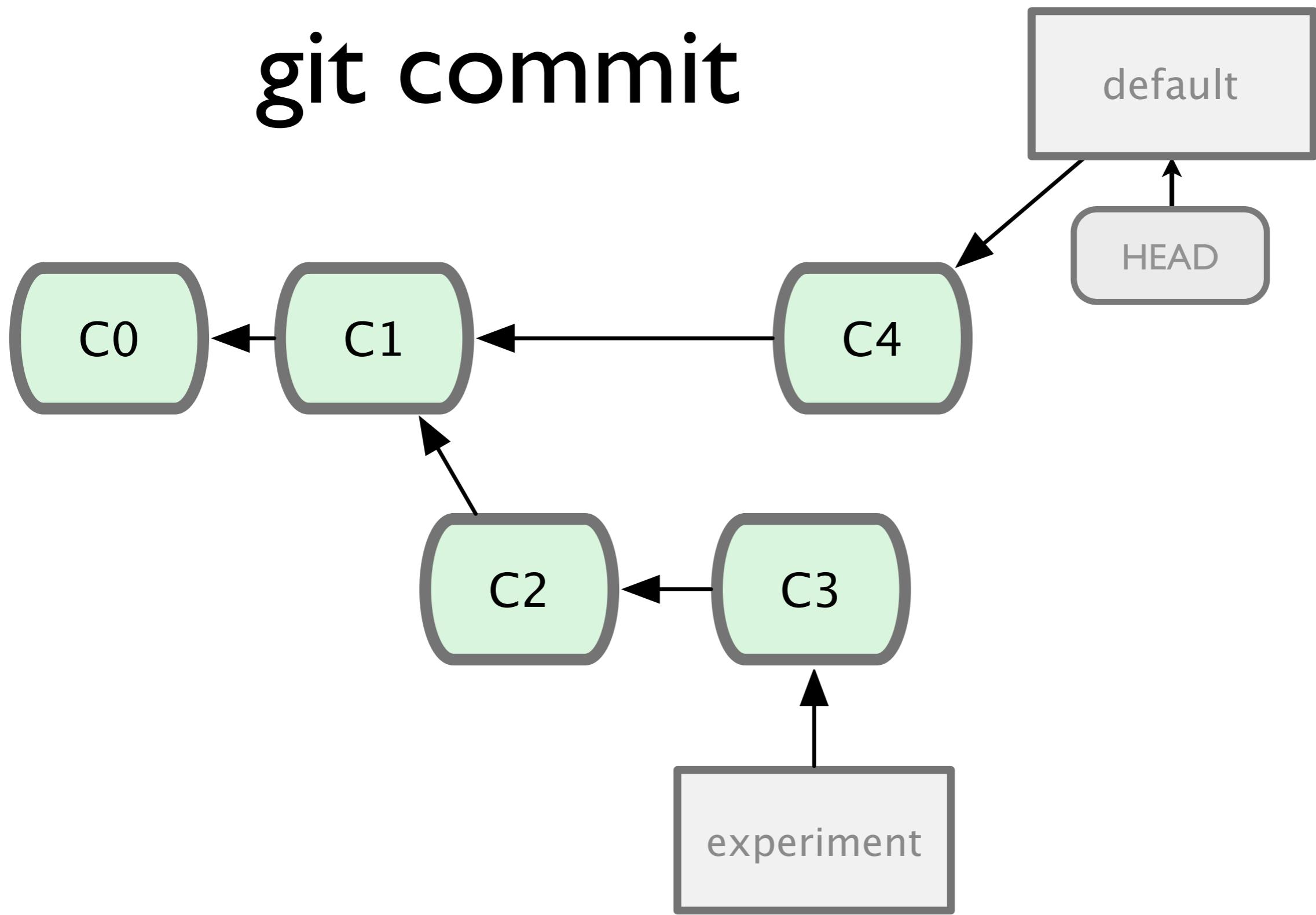


git checkout default

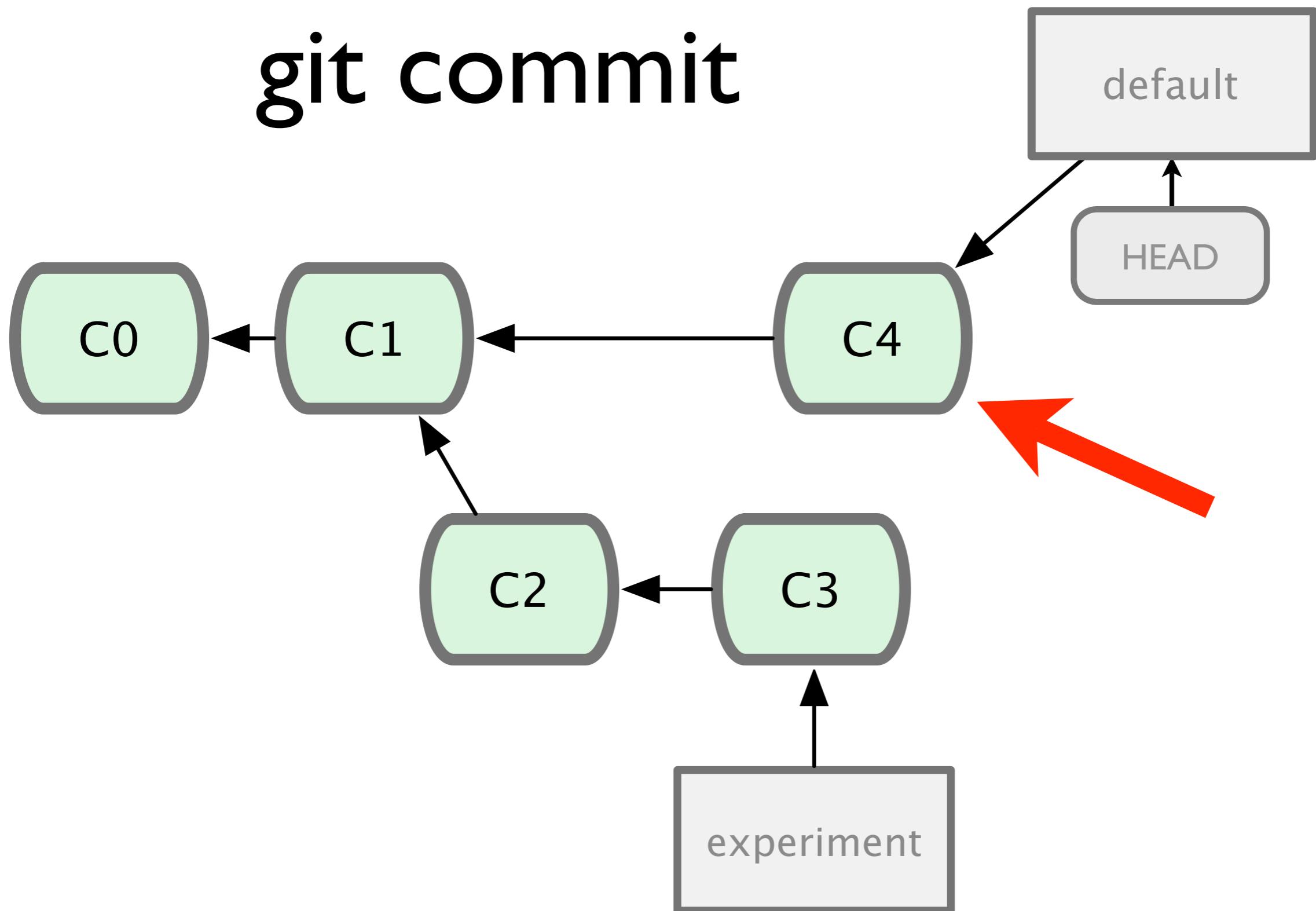


git checkout default

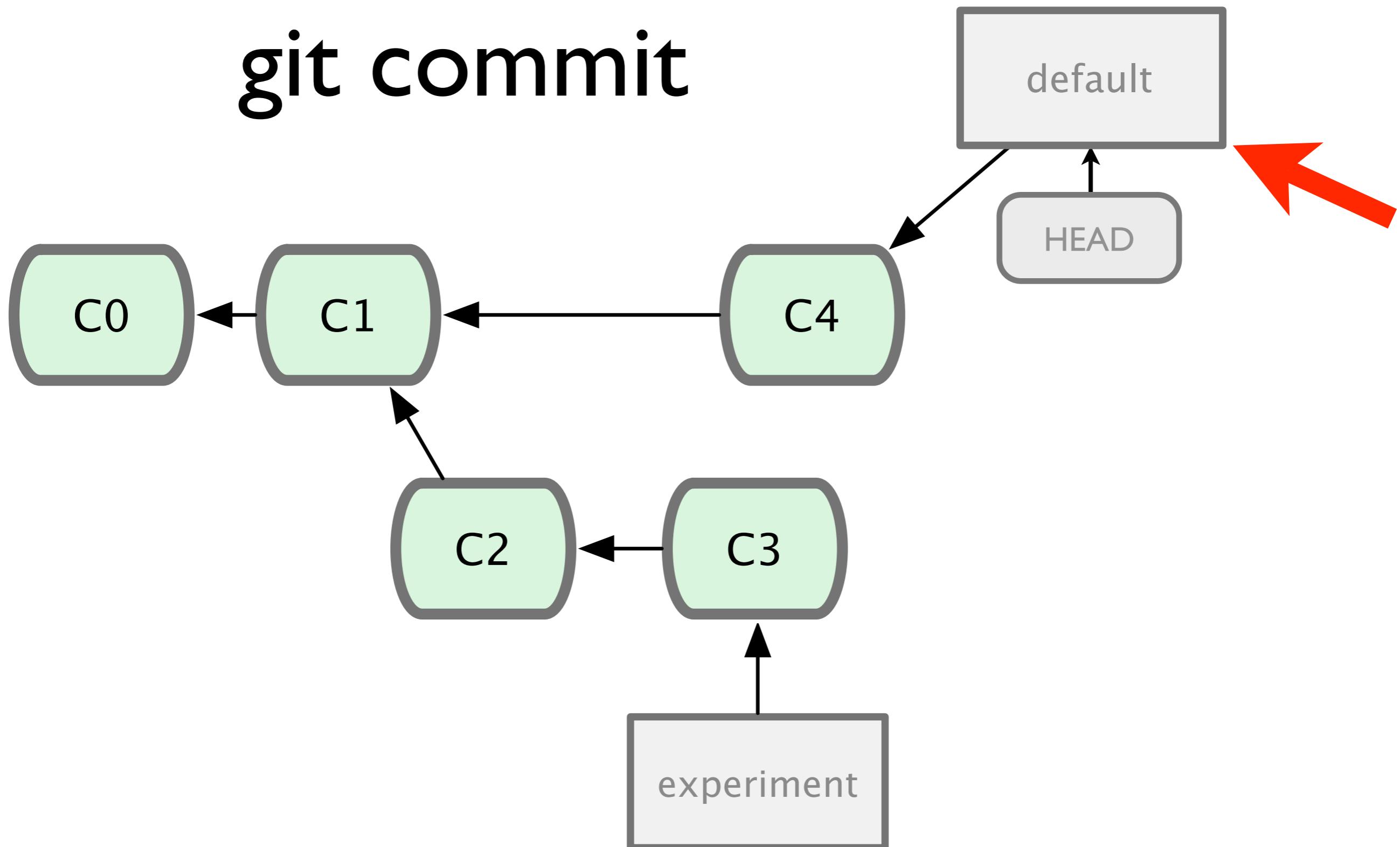
git commit

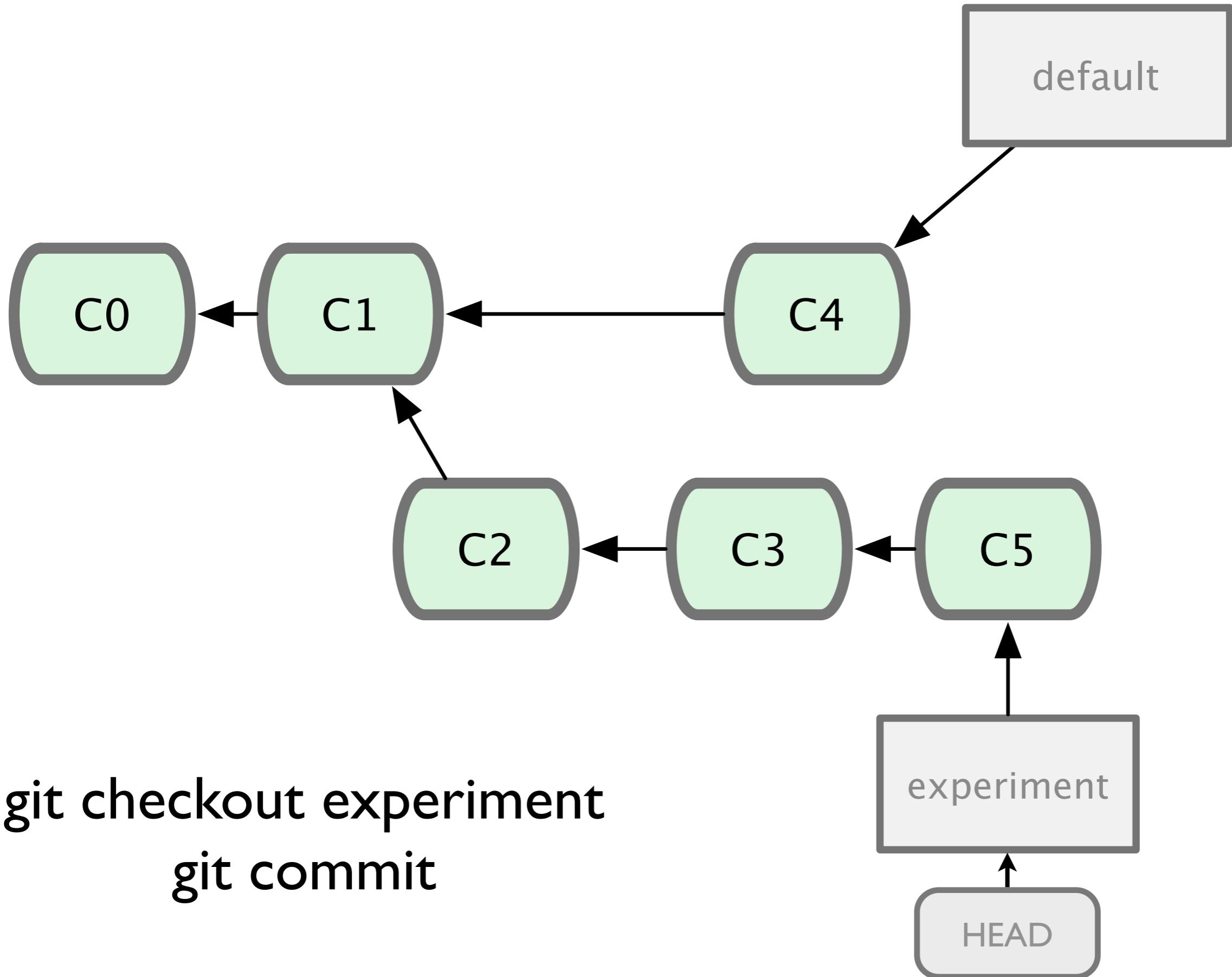


git commit



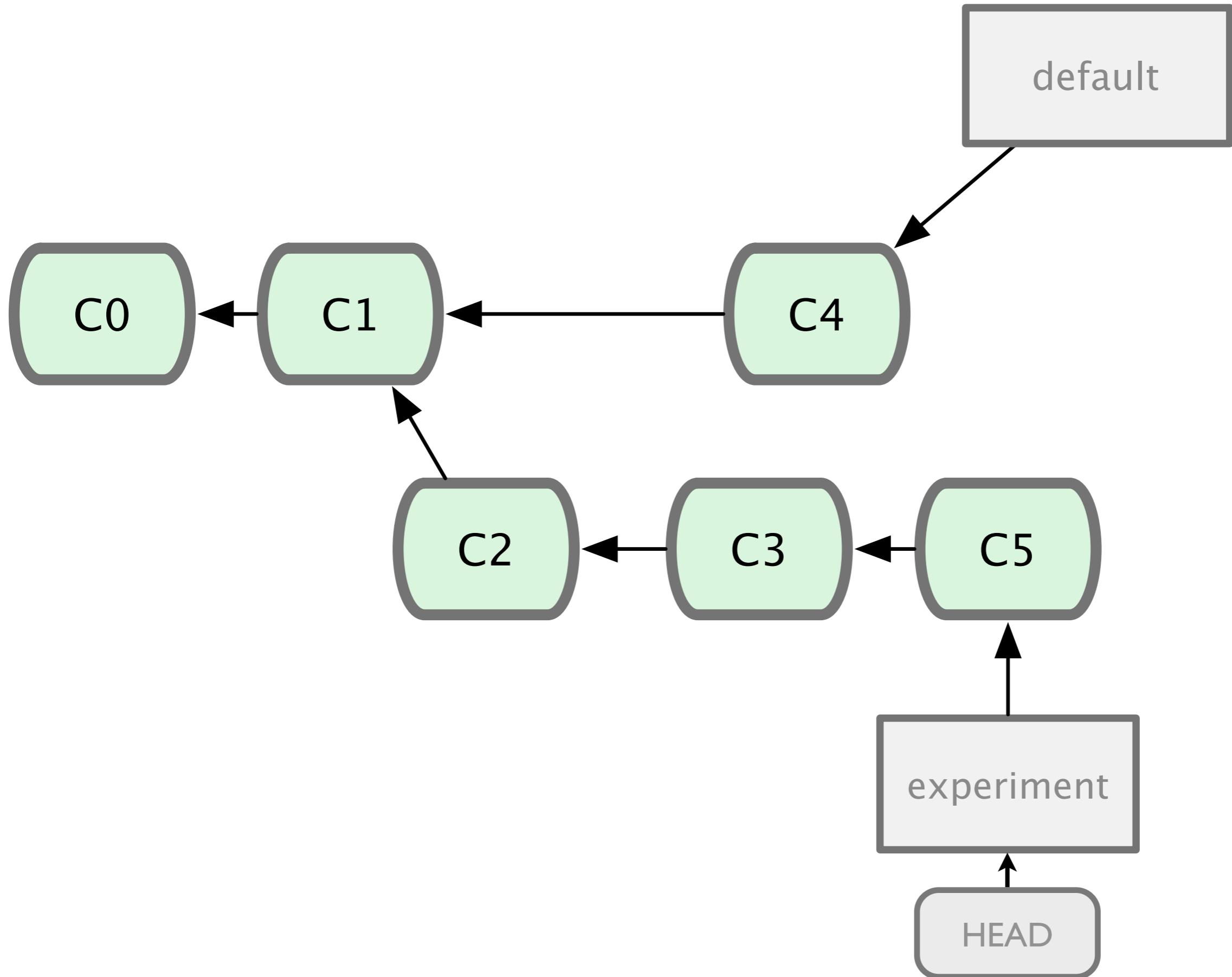
git commit

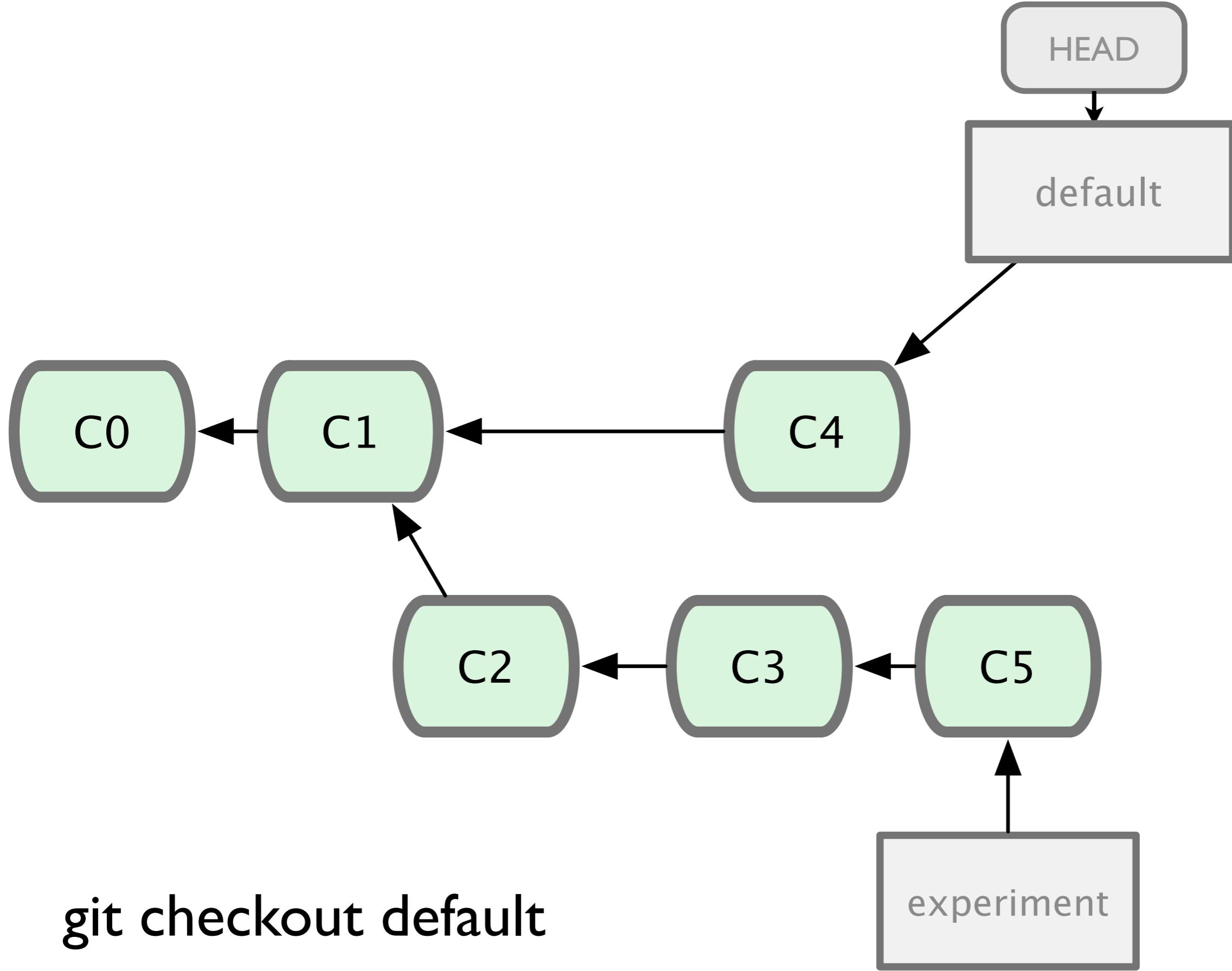


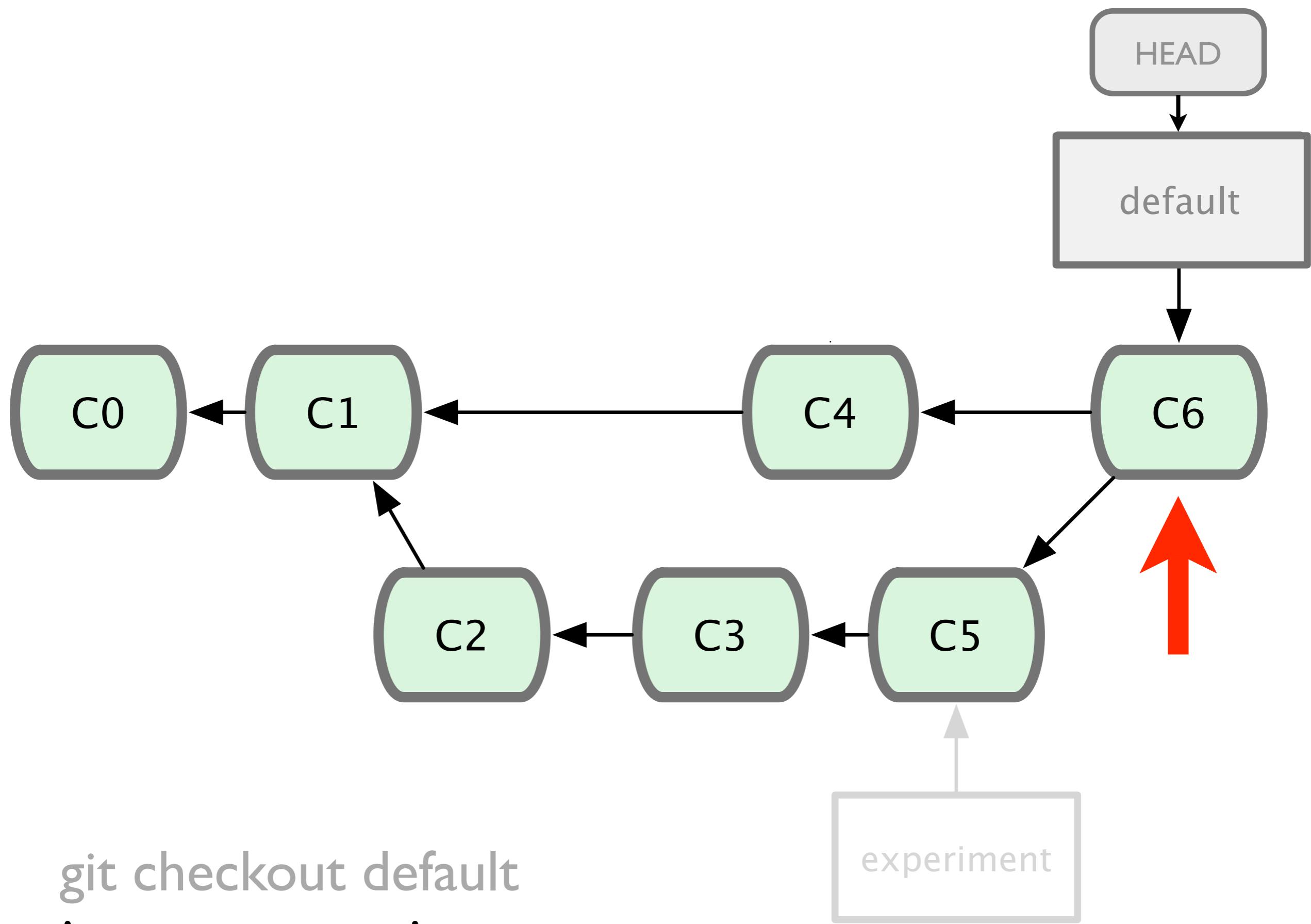


merging

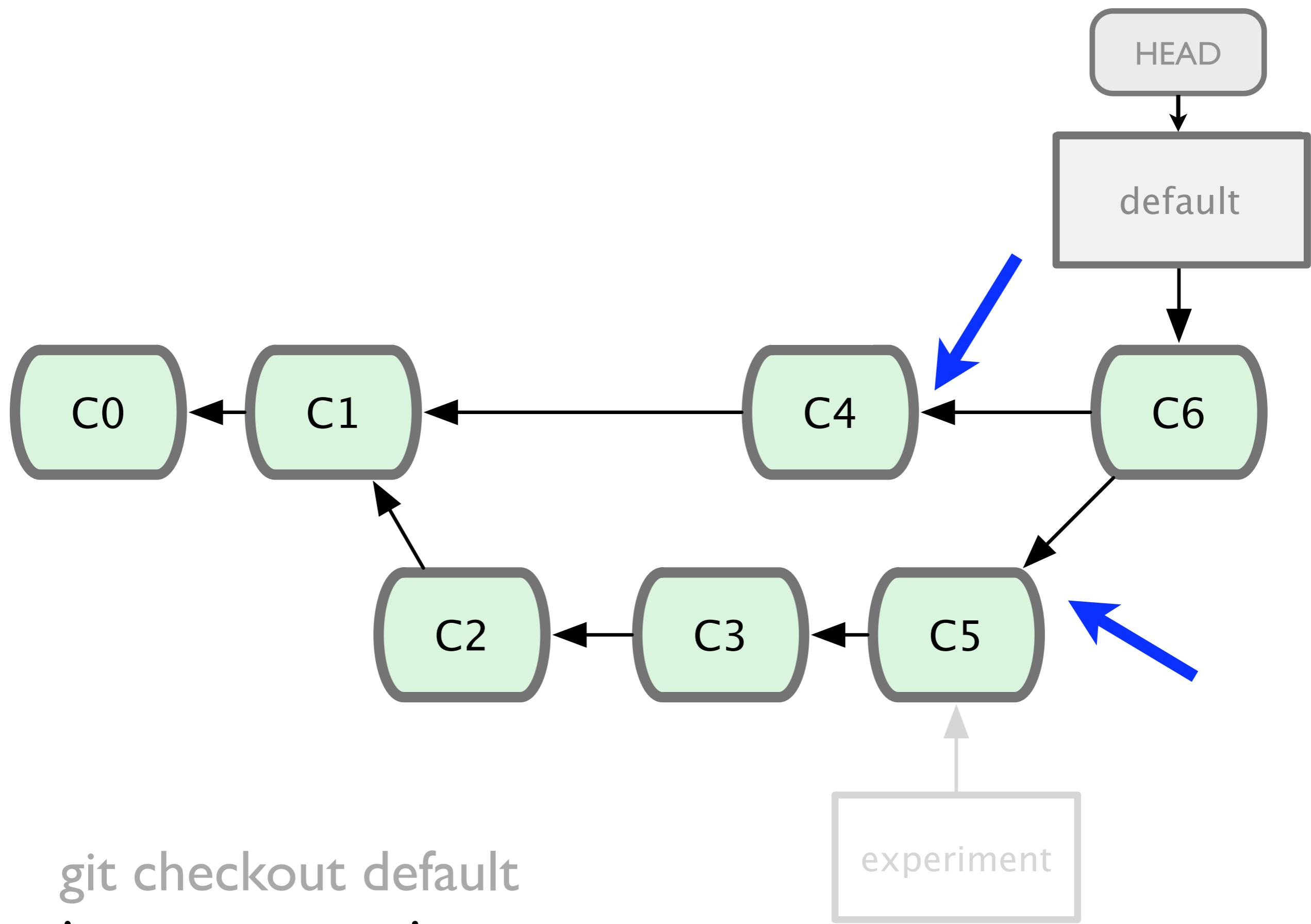
git merge



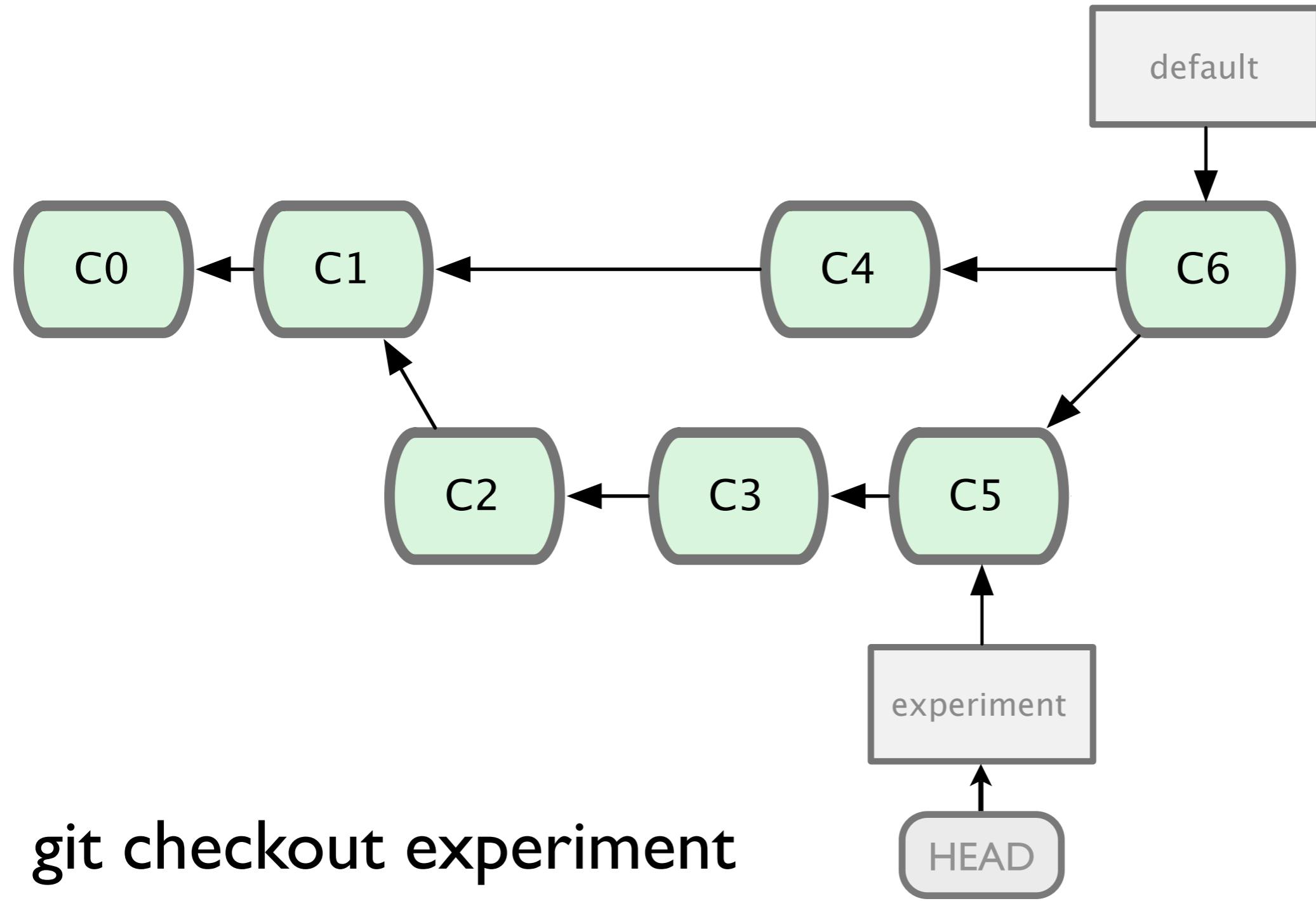


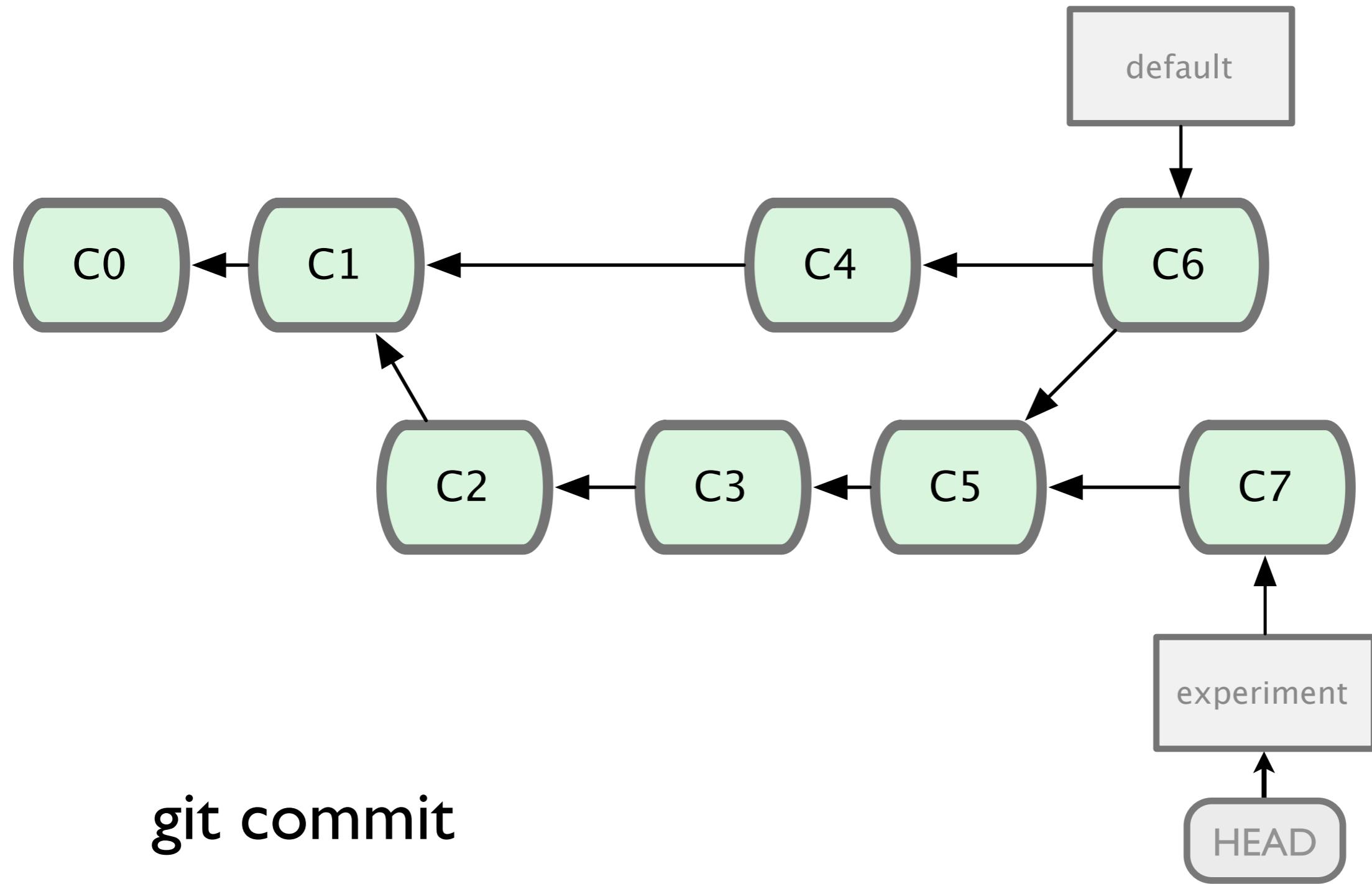


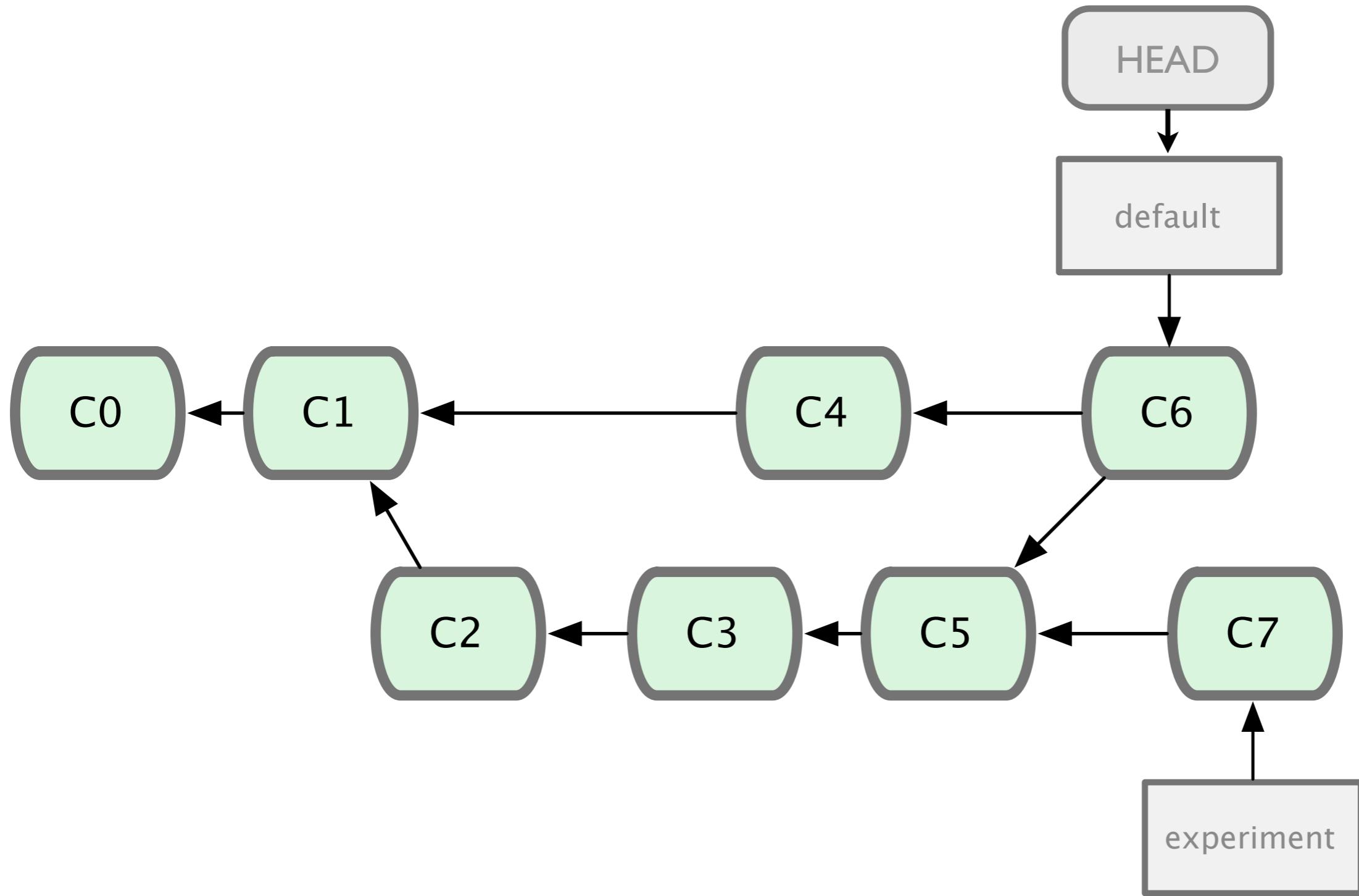
`git checkout default`
`git merge experiment`



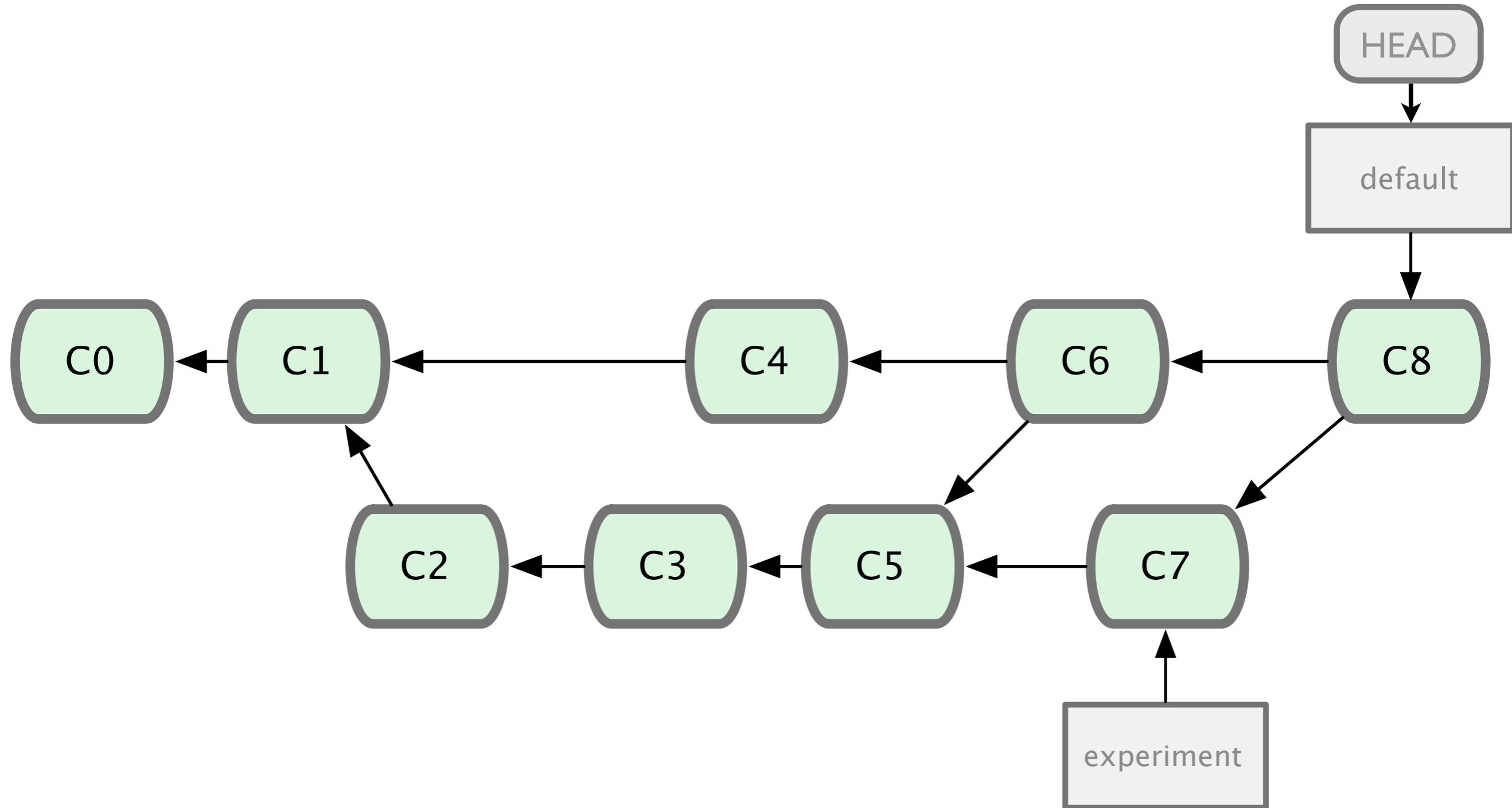
git checkout default
git merge experiment





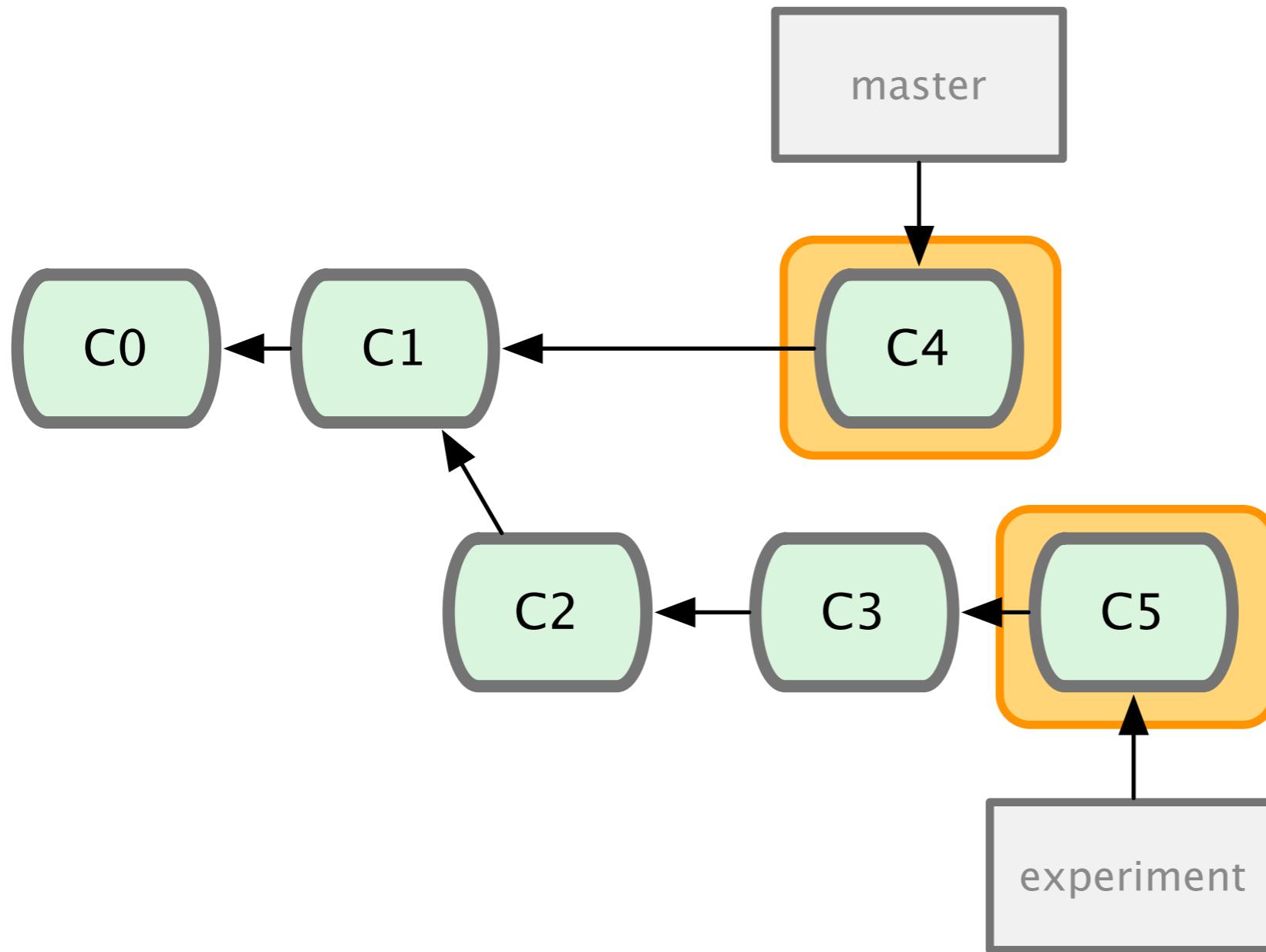


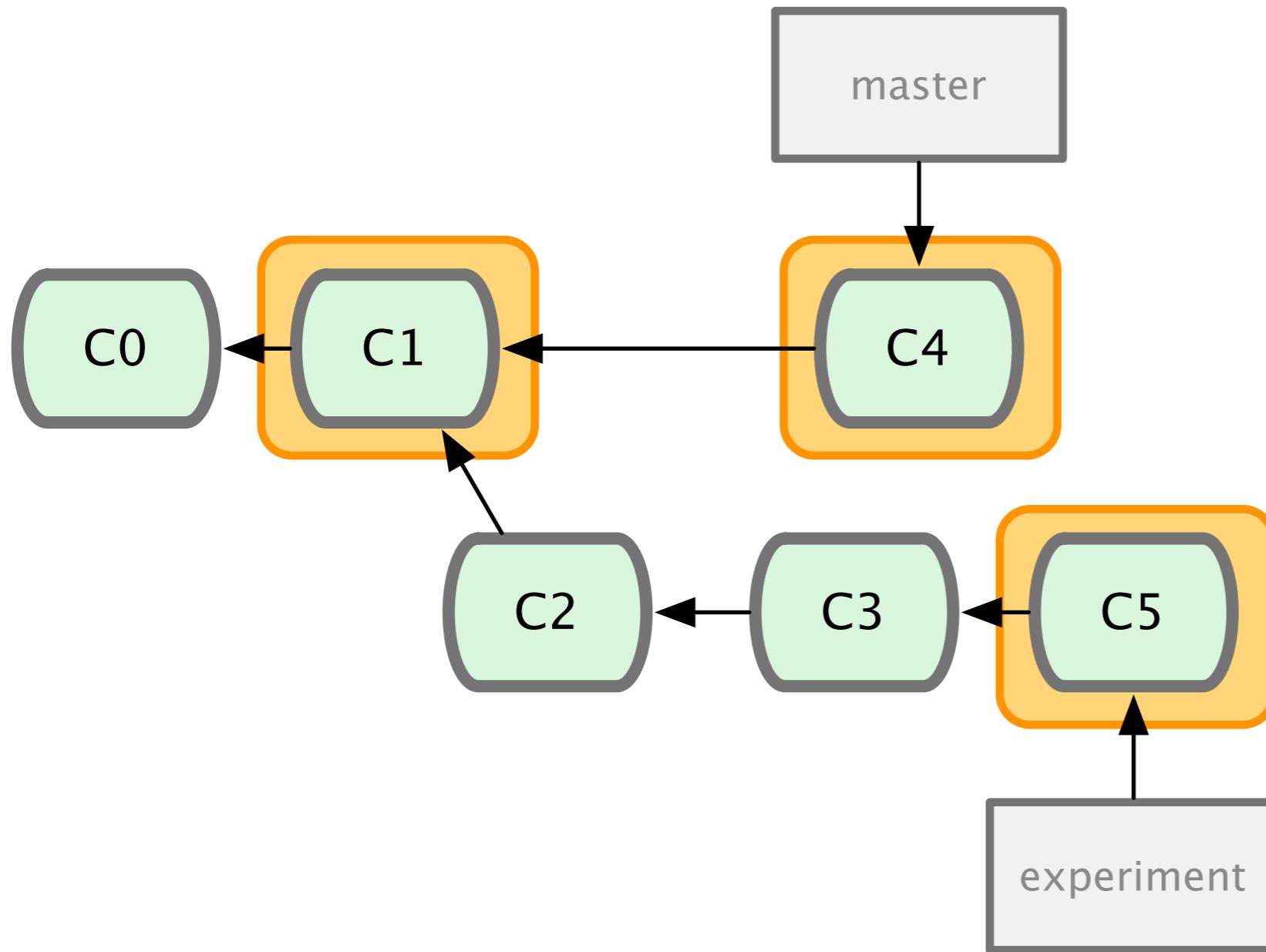
git checkout default

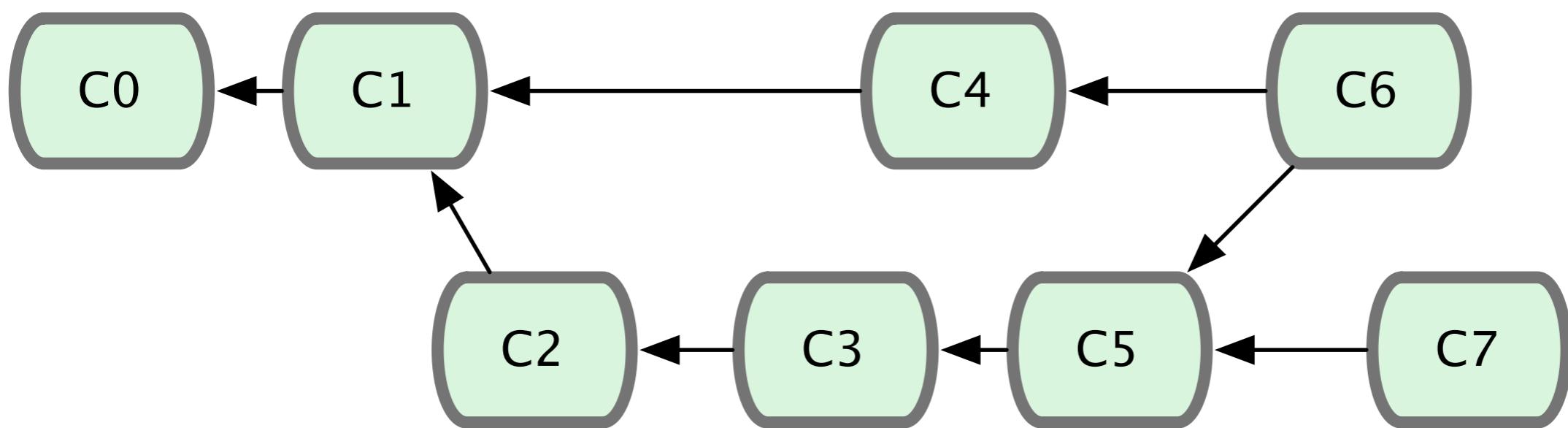


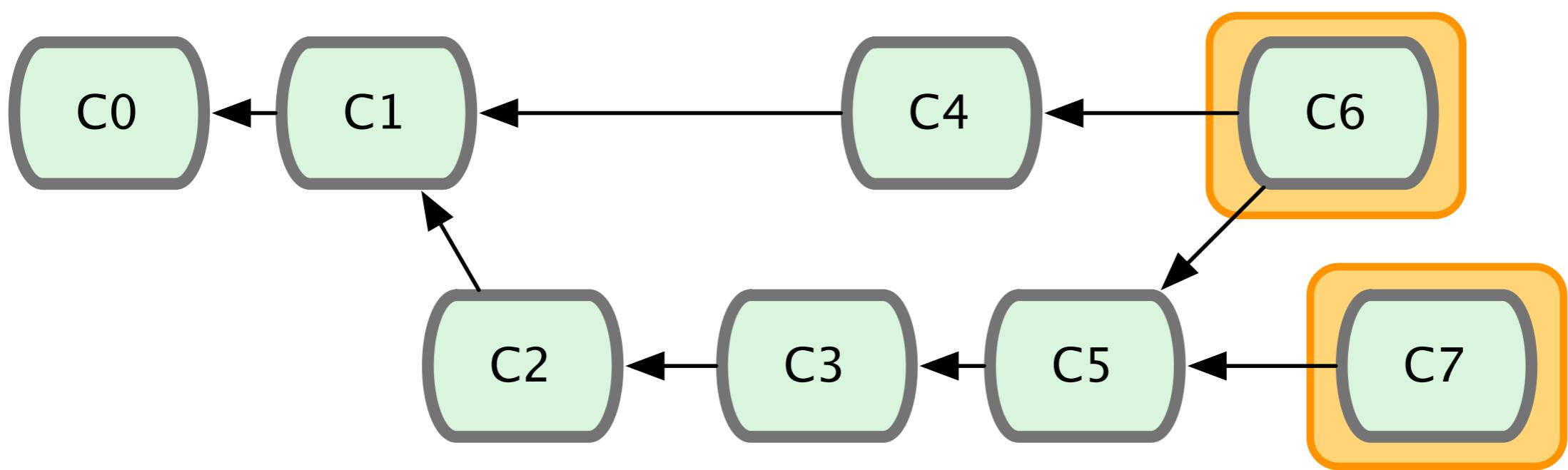
git merge experiment

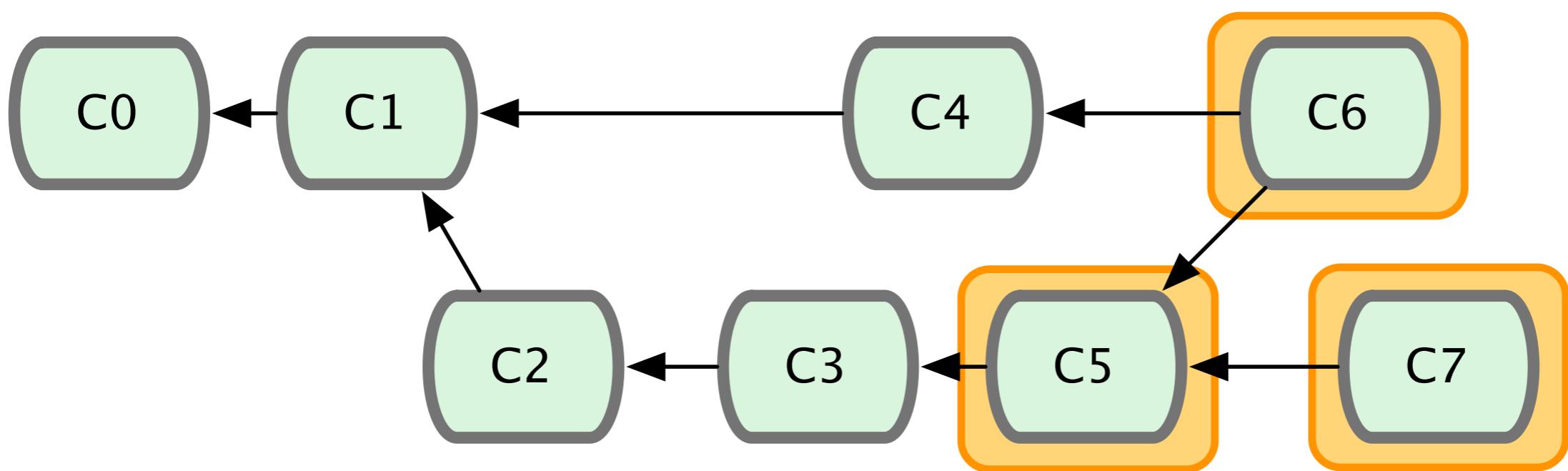
Hot
3 Way Merge
Action











</git-crash-course>

Lesser Known Git Tricks

Lesser Known Git Tricks

Revision Selection

Lesser Known Git Tricks

Revision Selection

Log Formatting and Subsets

Lesser Known Git Tricks

Revision Selection

Log Formatting and Subsets

Revision Diffs

Lesser Known Git Tricks

Revision Selection

Log Formatting and Subsets

Revision Diffs

Annotation

Lesser Known Git Tricks

Revision Selection

Log Formatting and Subsets

Revision Diffs

Annotation

Rewriting History

Lesser Known Git Tricks

Revision Selection

Log Formatting and Subsets

Revision Diffs

Annotation

Rewriting History

Cherry Picking

Lesser Known Git Tricks

Revision Selection

Log Formatting and Subsets

Revision Diffs

Annotation

Rewriting History

Cherry Picking

Undoing

Lesser Known Git Tricks

Revision Selection

Log Formatting and Subsets

Revision Diffs

Annotation

Rewriting History

Cherry Picking

Undoing

Bisecting

Revision Selection

Revision Selection

alternate ways to refer to
objects or ranges of objects

Revision Selection

full sha- l

partial sha- l

branch or tag name

caret parent

tilde spec

blob spec

relative specs

ranges

Full SHA1

6e453f523fa1da50ecb04431101112b3611c6a4d

Partial SHA1

6e453f523fa1da50ecb04431101112b3611c6a4d

6e453f523fa1da50

6e453

Branch, Remote or Tag Name

v1.0

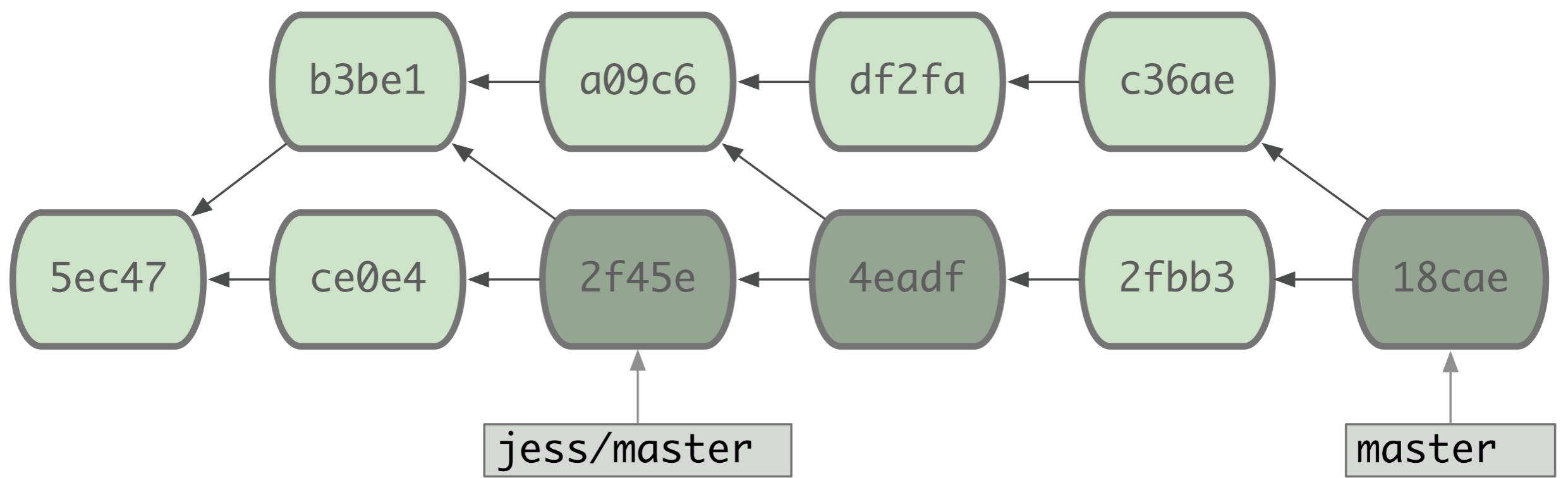
default

m/cupcake

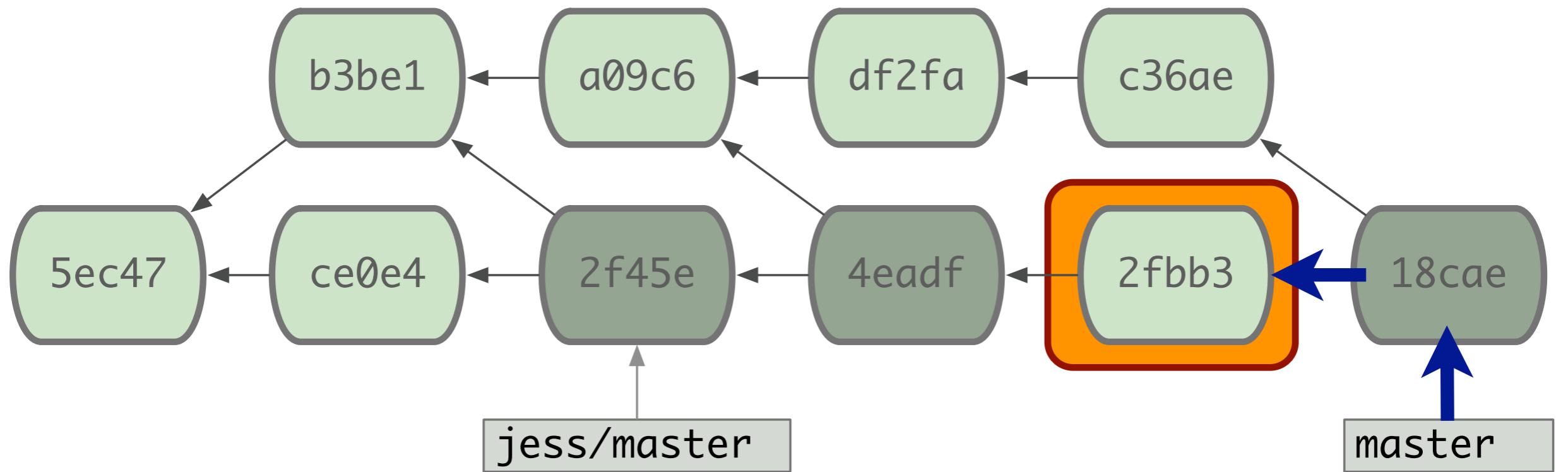
Caret Parent

default^{^2}

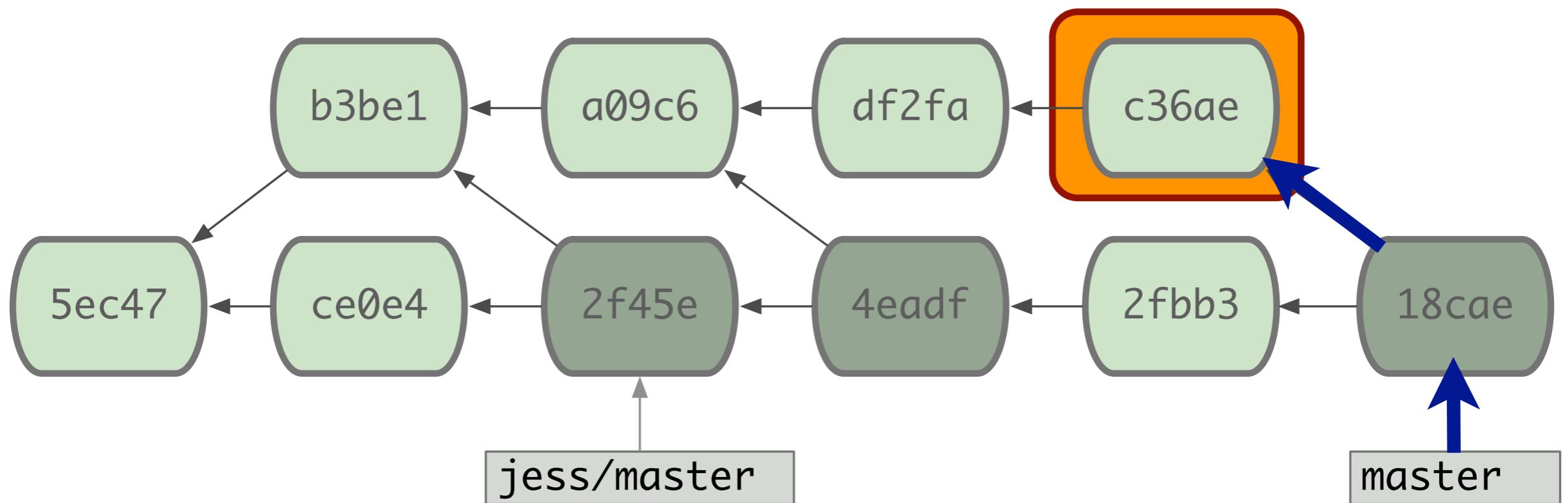
2nd parent of ‘default’



master^Λ



master^{^2}

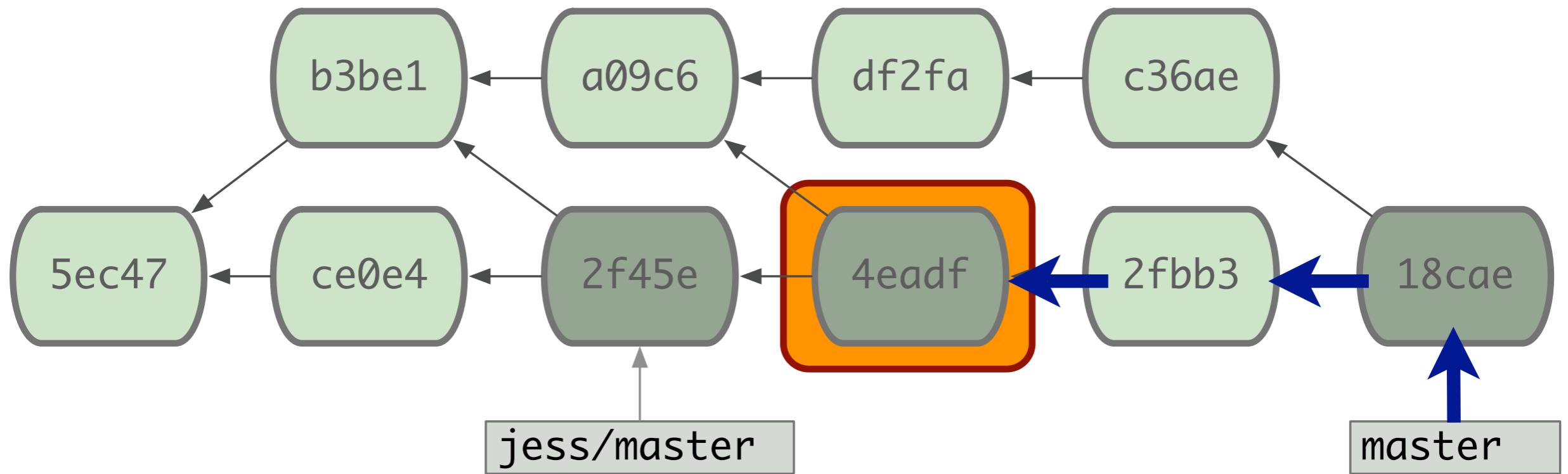


Tilde Spec

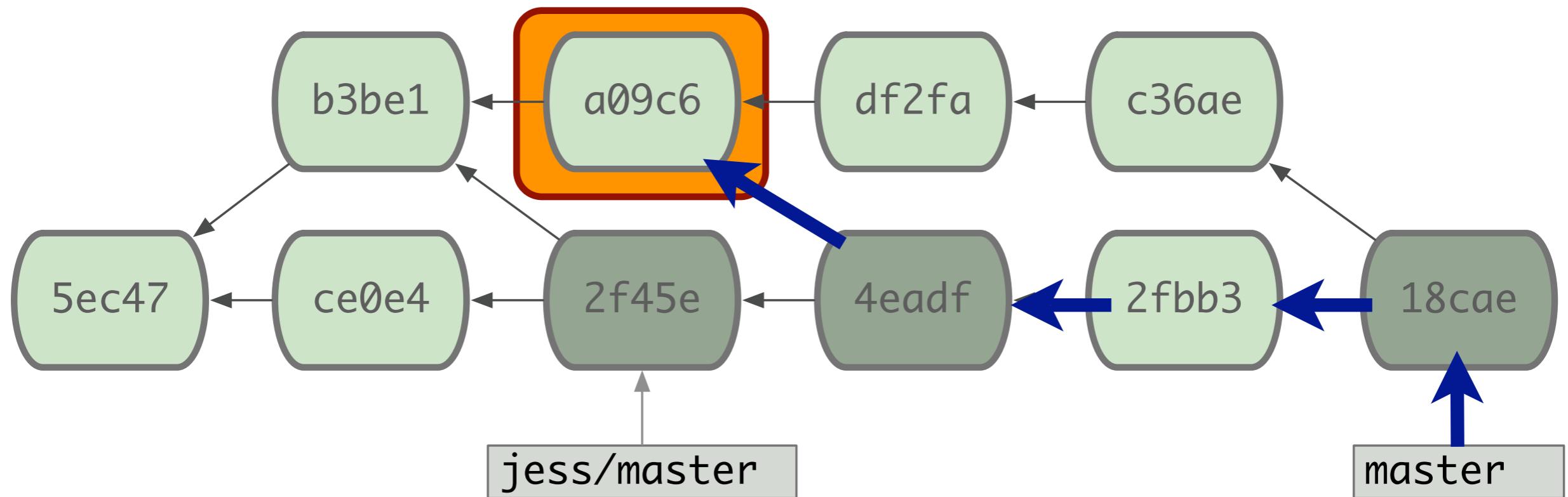
default~2

grandparent of ‘default’
(parent of the parent)

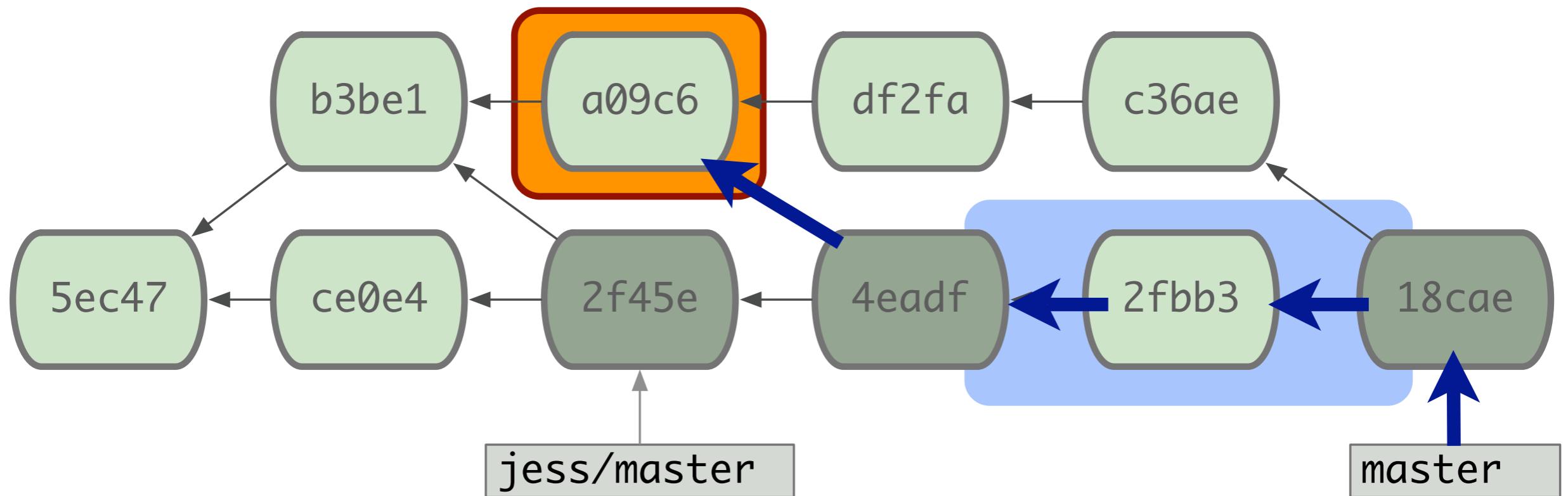
master~2



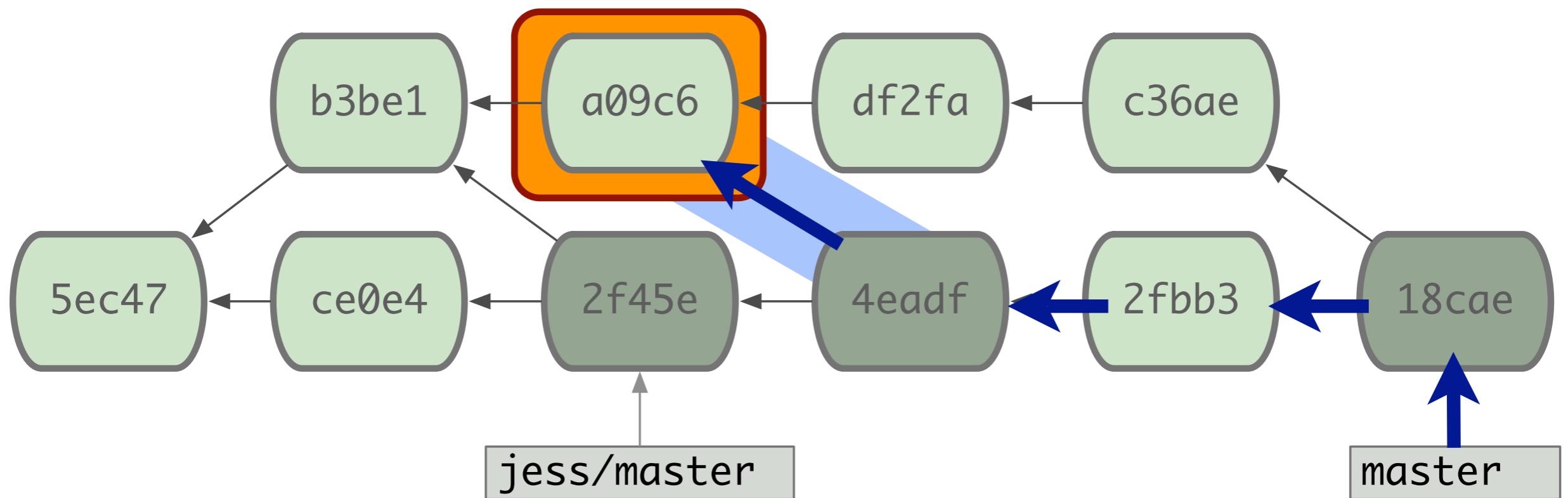
master^{^2} master~2^2



master^{^2}
master~2^2



master^{^2}
master~2^{^2}



Blob Spec

default:path/to/file

blob of that file in ‘default’ commit

Relative Specs

master@{yesterday}

the commit ‘master’ was at yesterday

Relative Specs

master@{5}

the 5th prior value of ‘master’ (locally)

Ranges

`ce0e4 .. e4272`

every commit reachable by e4272
that is not reachable by ce034

Ranges

[old] . . [new]

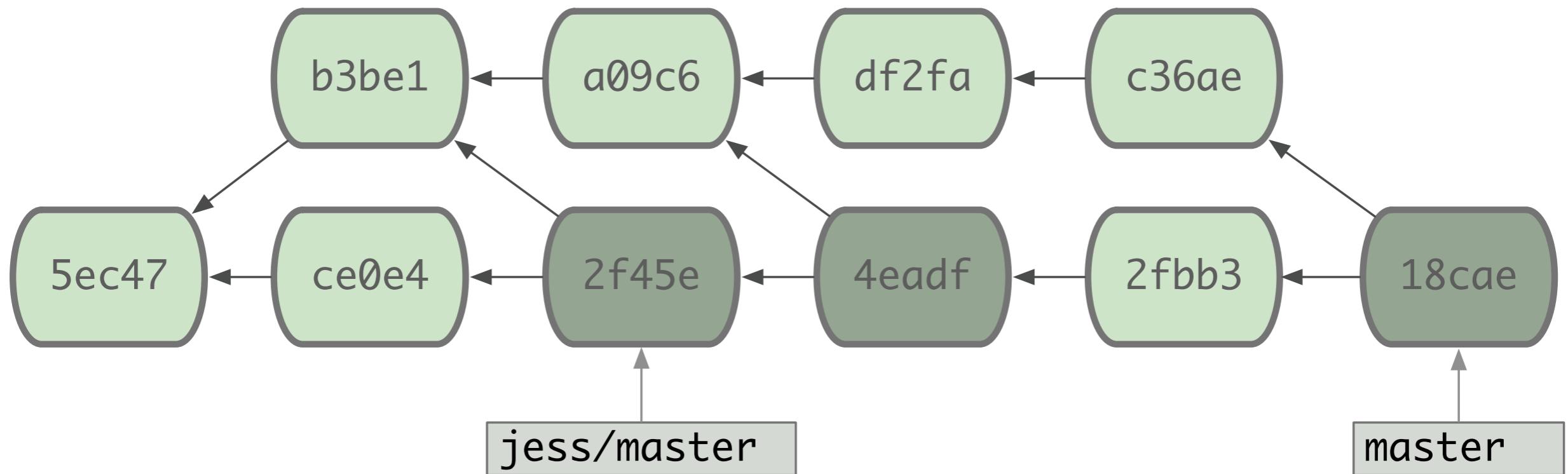
every commit reachable by [new]
that is not reachable by [old]

Ranges

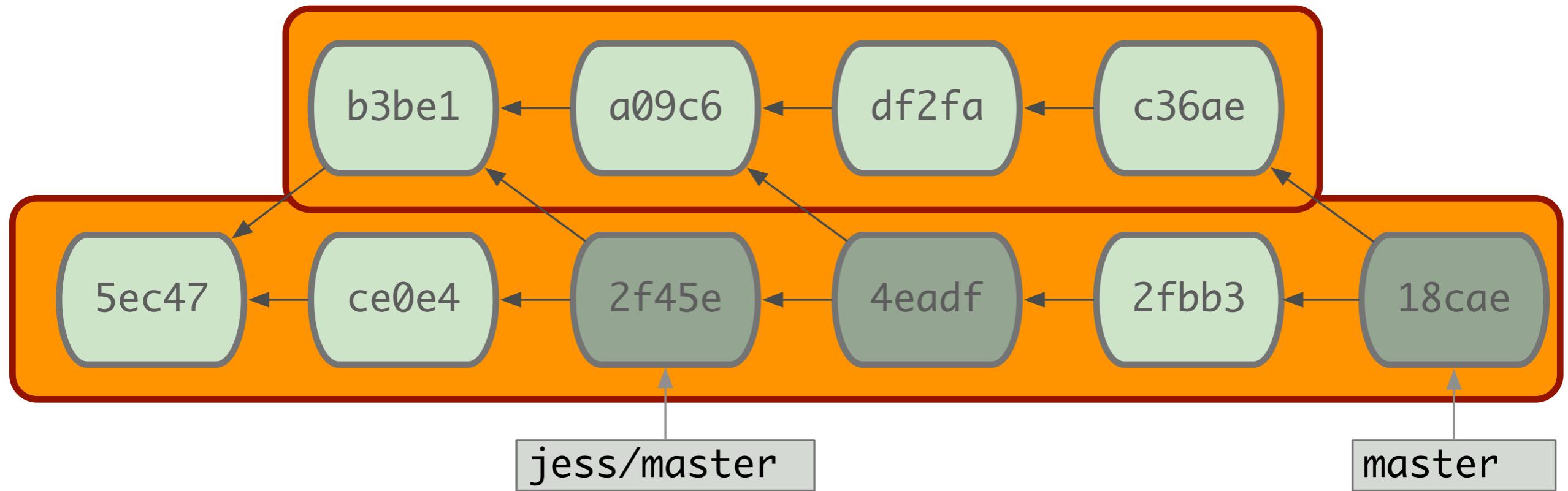
ce0e4 ..

everything since a commit

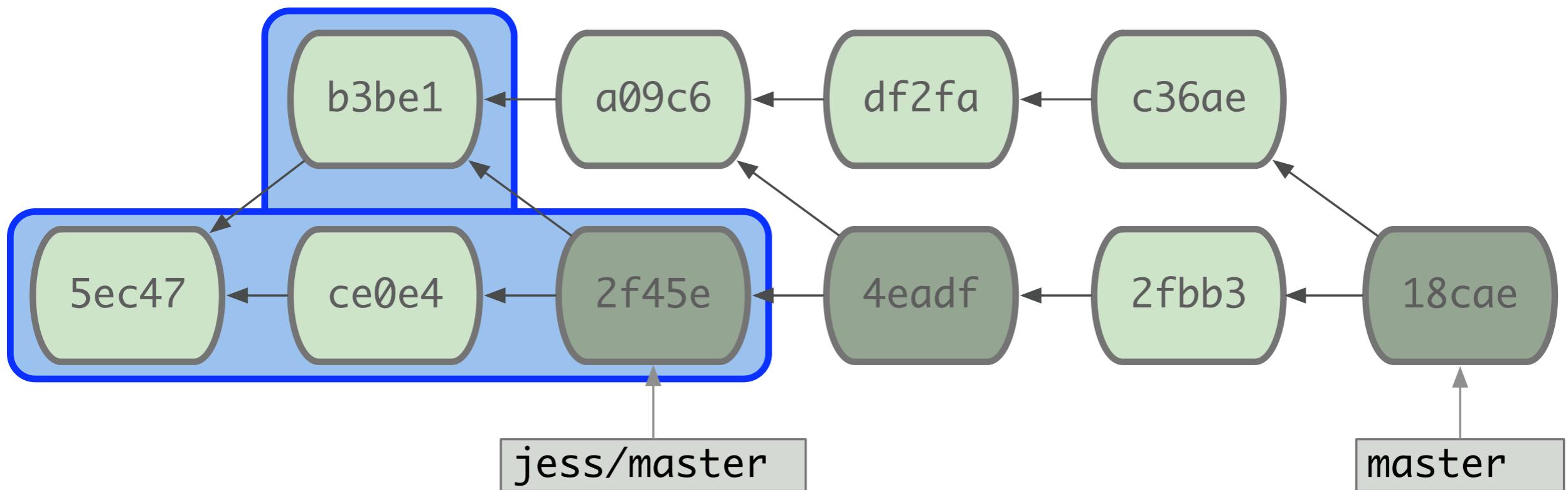
jess/master.master



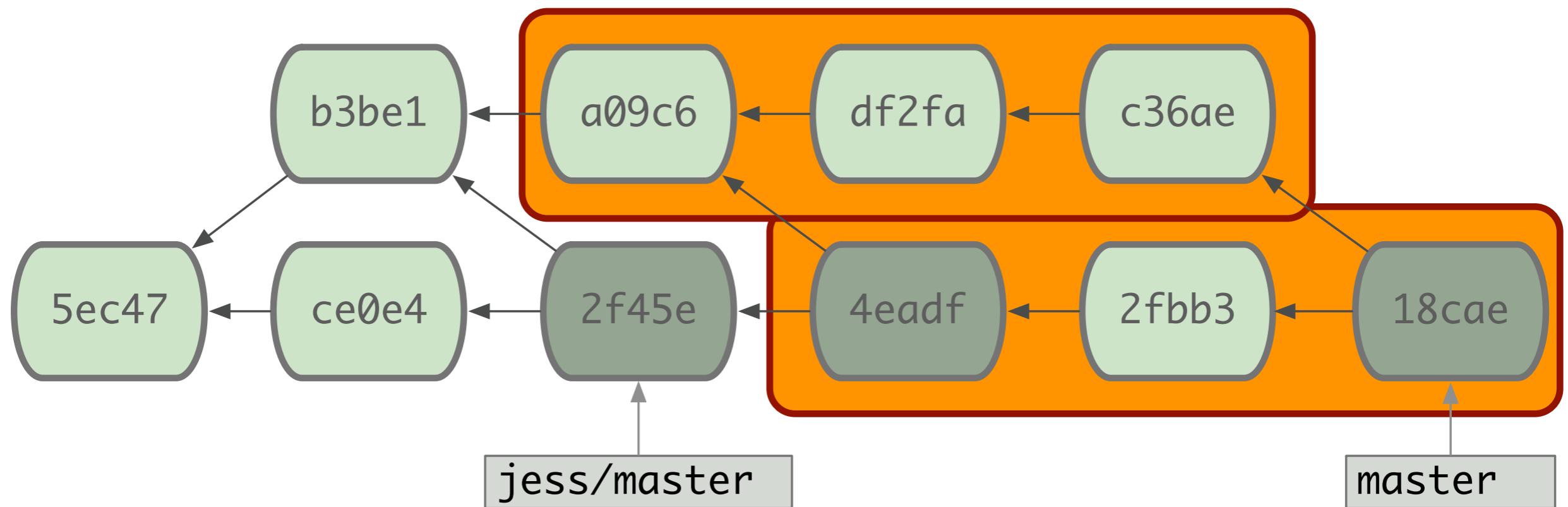
jess/master..master



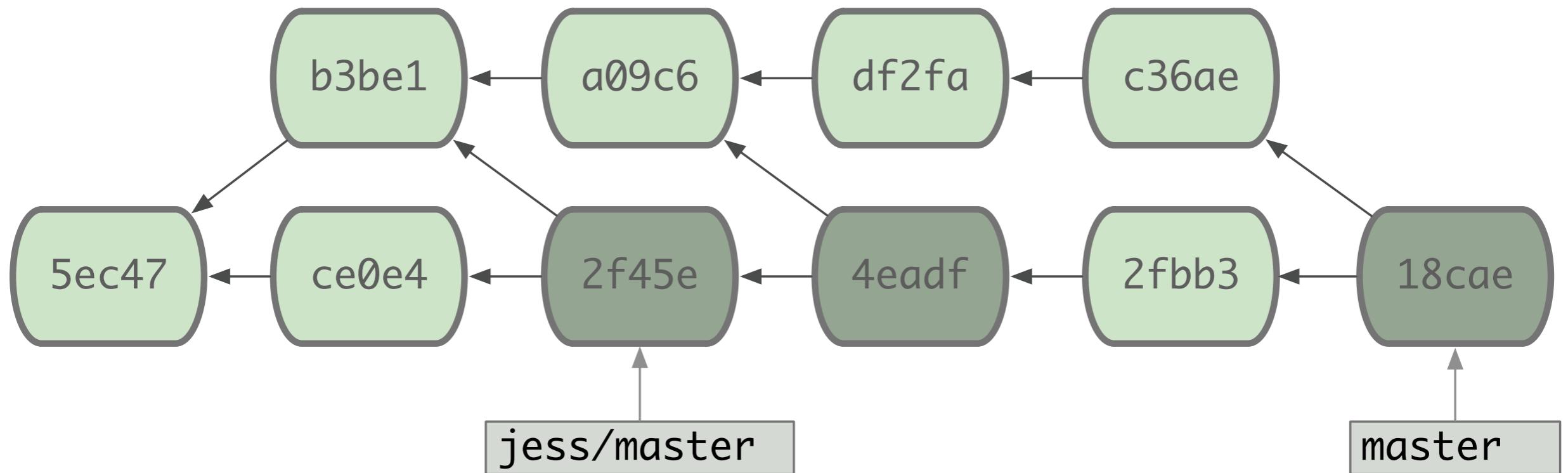
jess/master.master



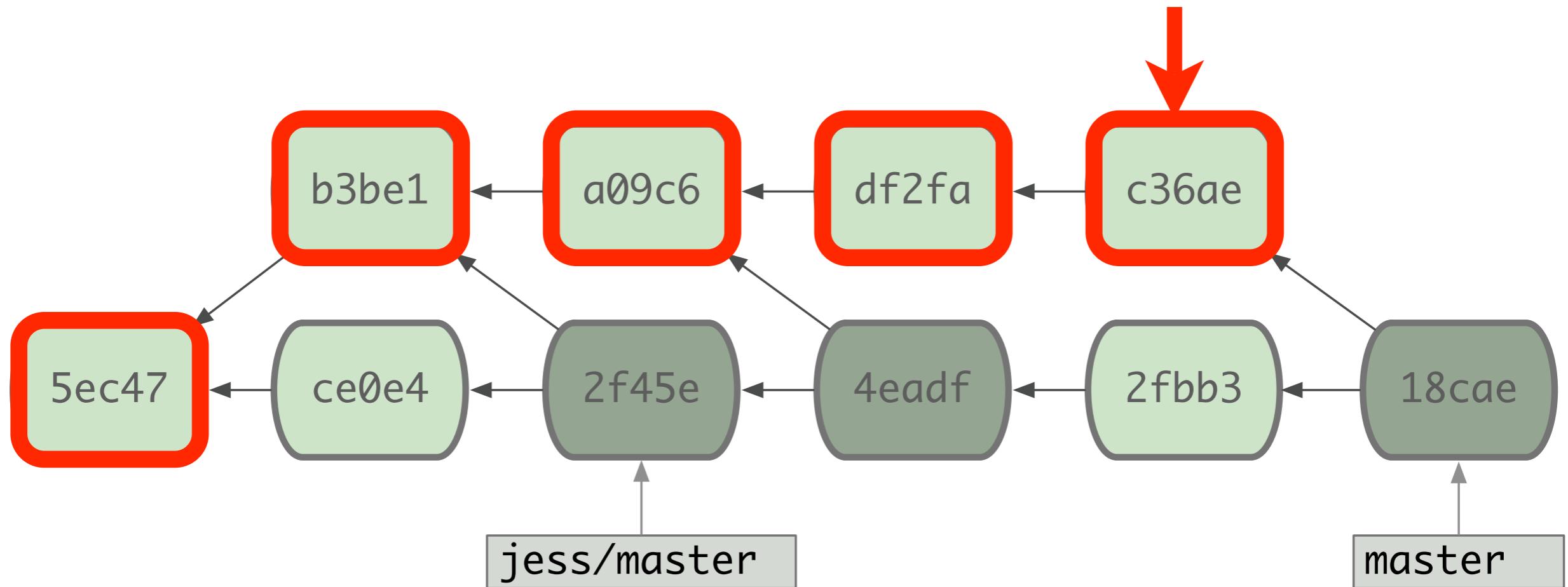
jess/master..master



jess/master..c36ae

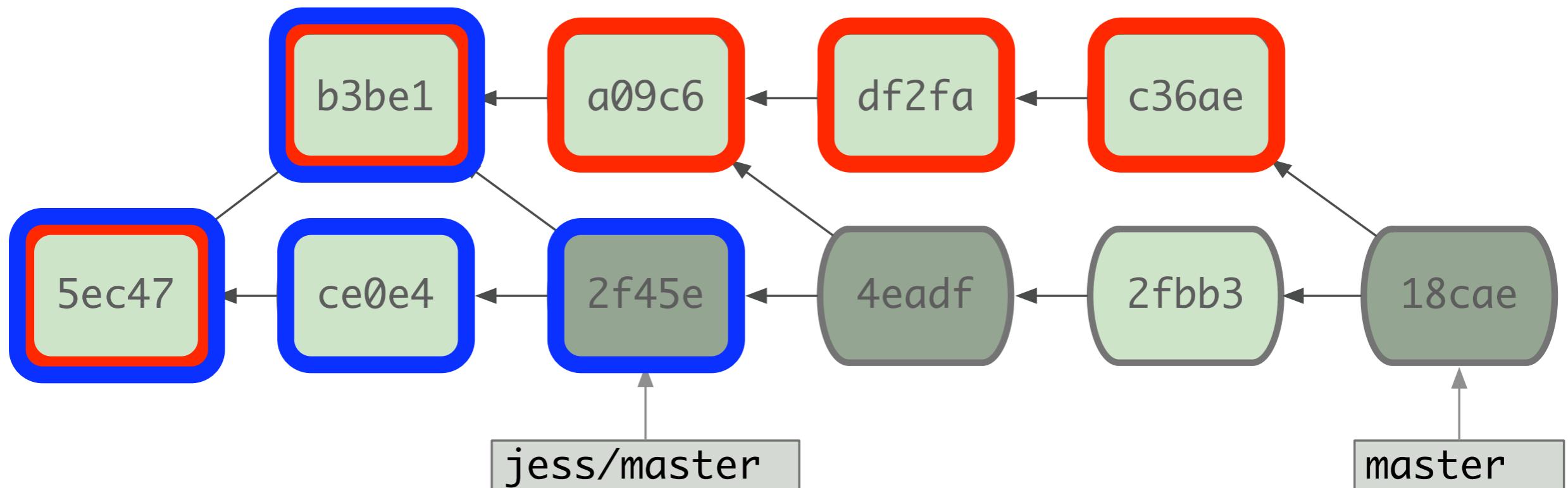


jess/master..c36ae

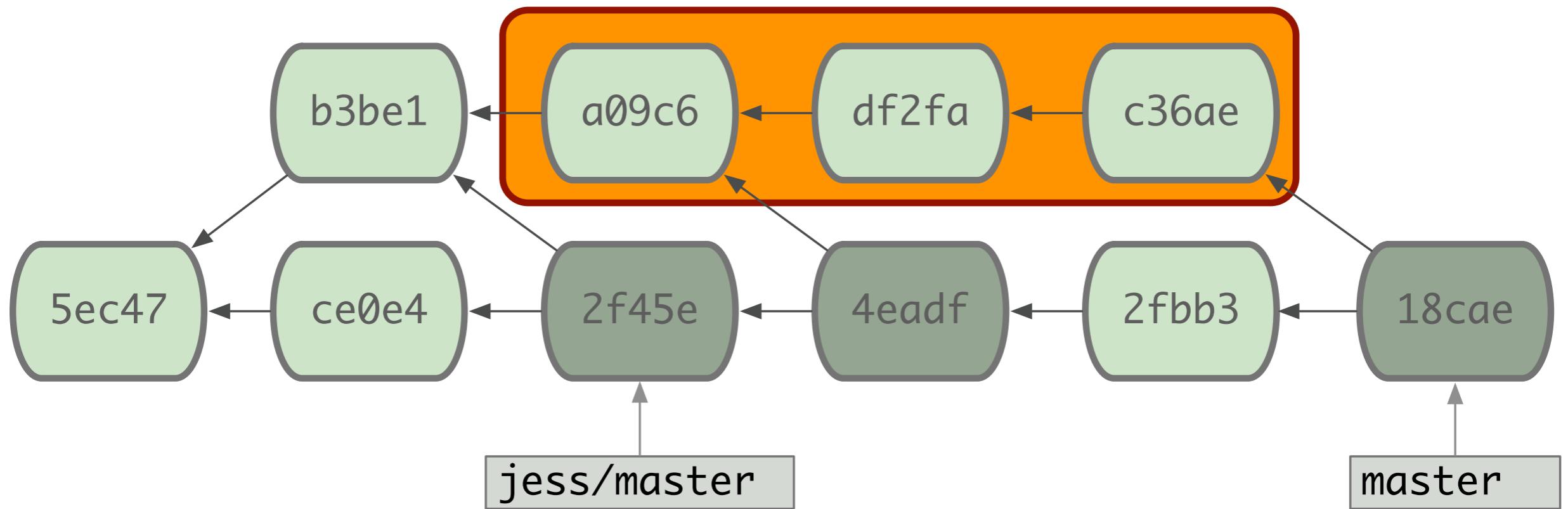




jess/master..c36ae



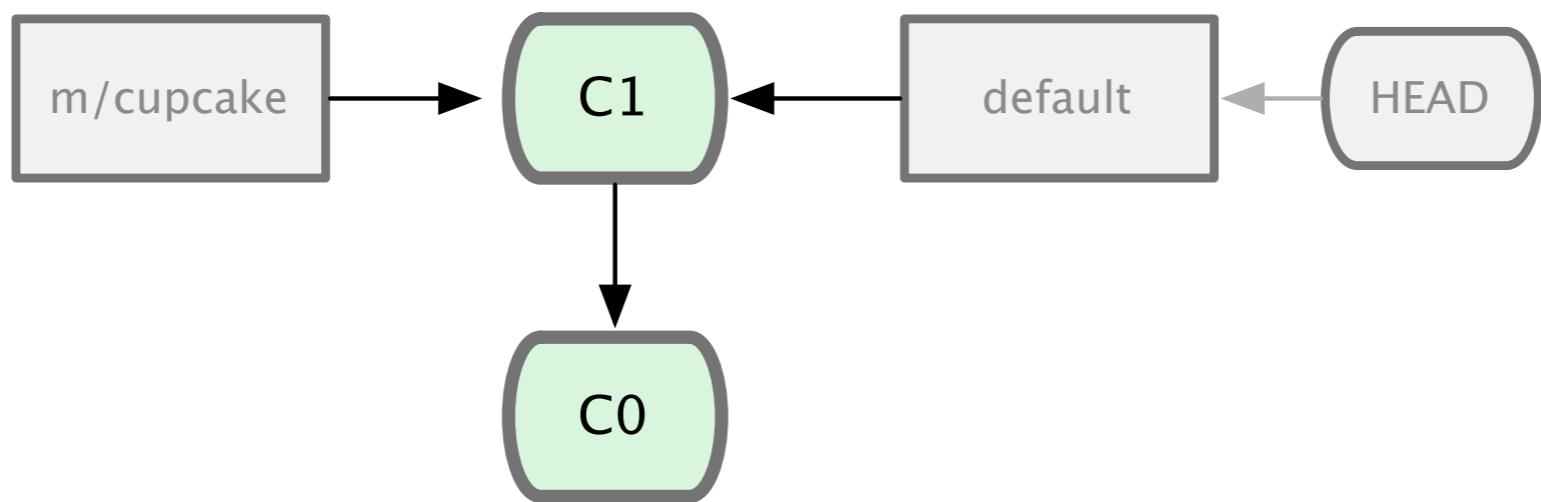
jess/master..c36ae

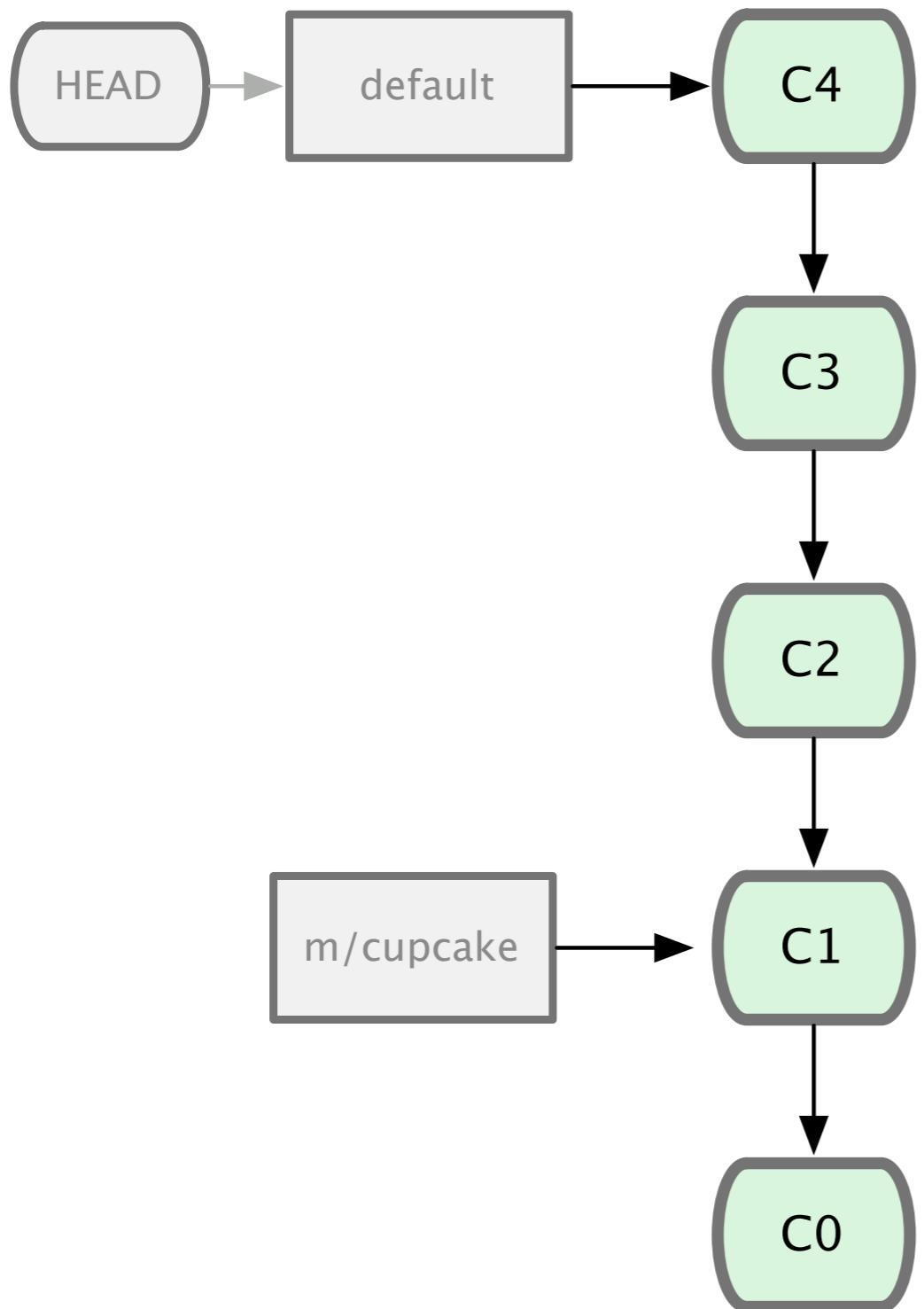


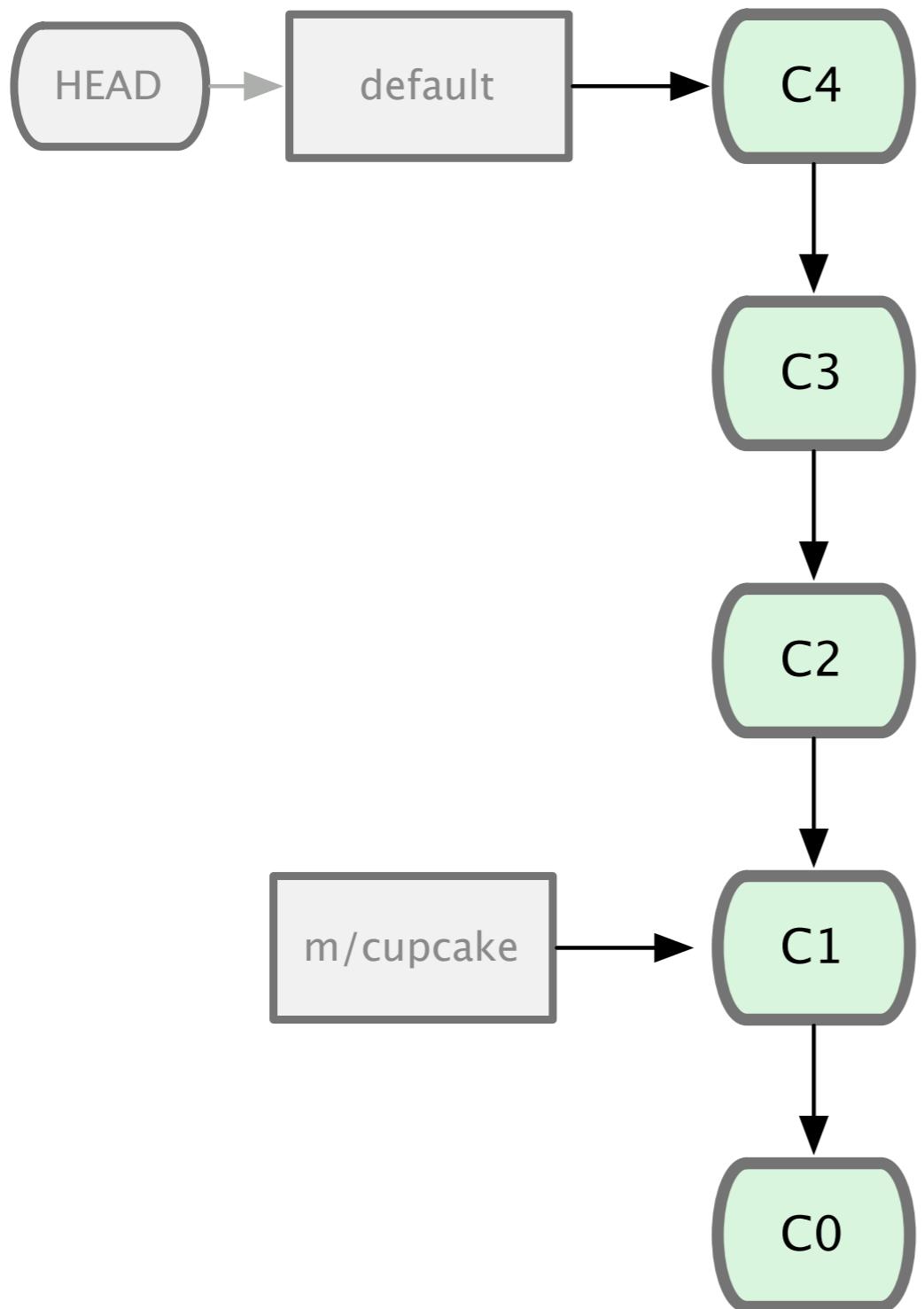
Advanced Log

Log Subsets

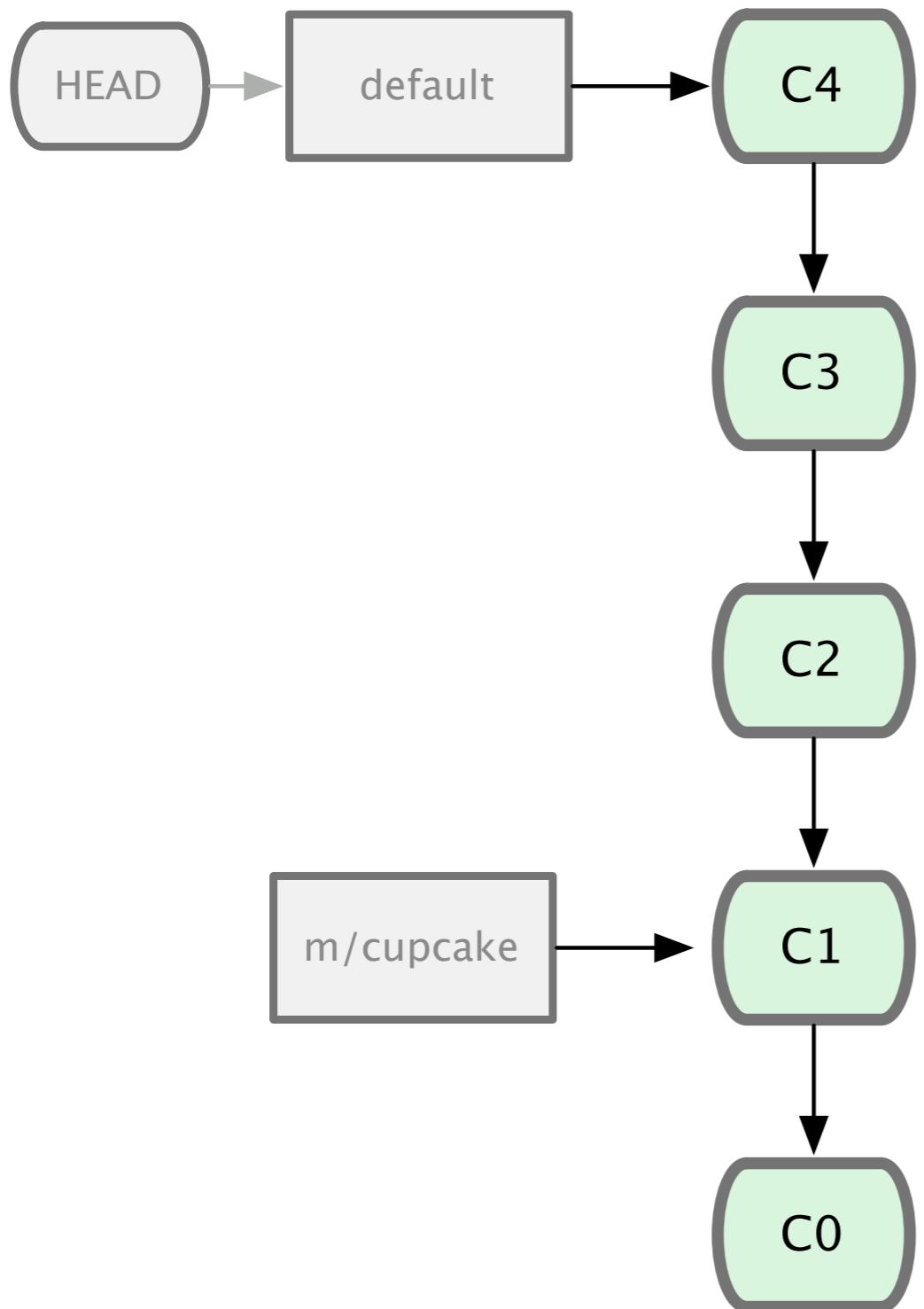
```
git log m/cupcake..
```



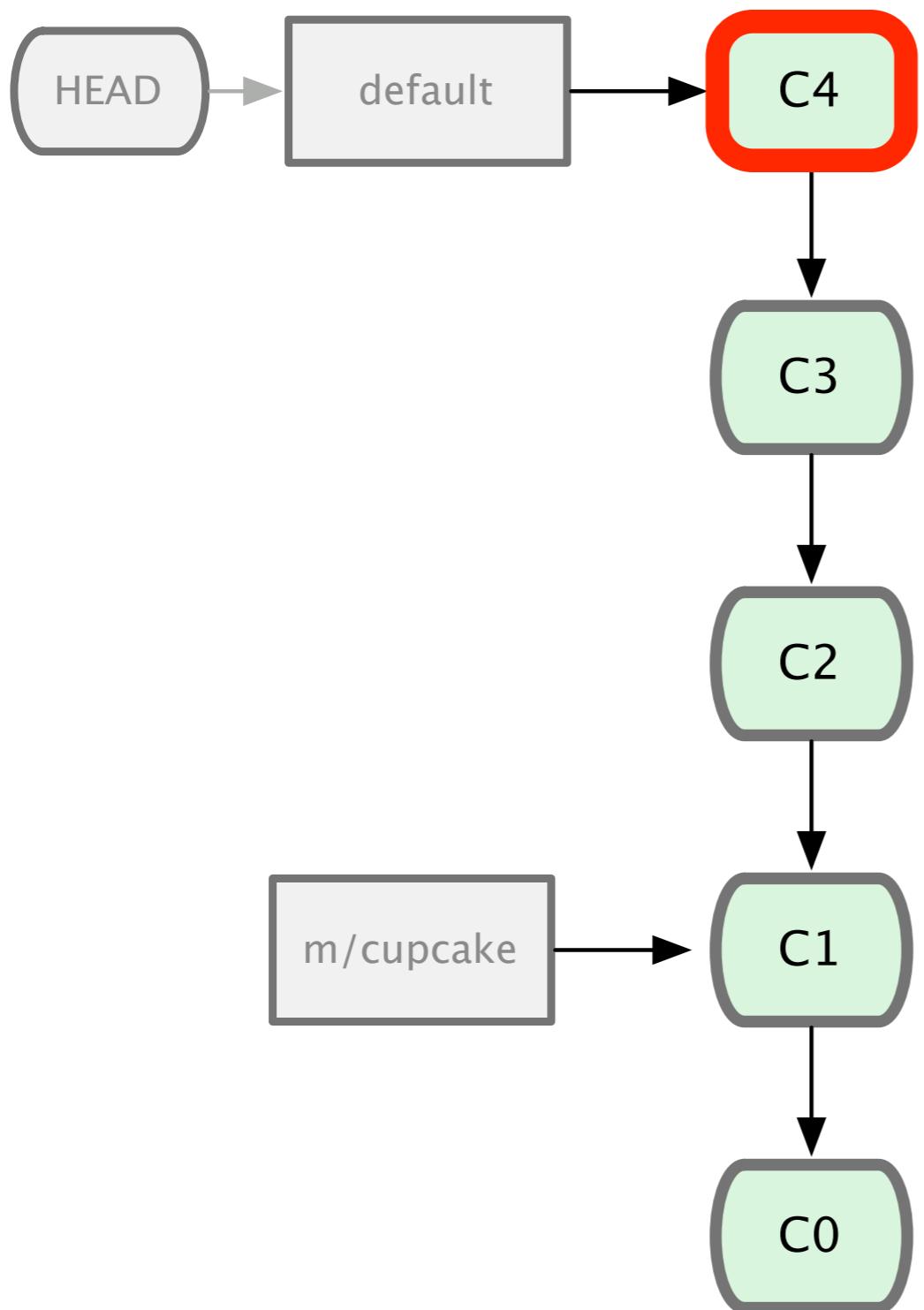




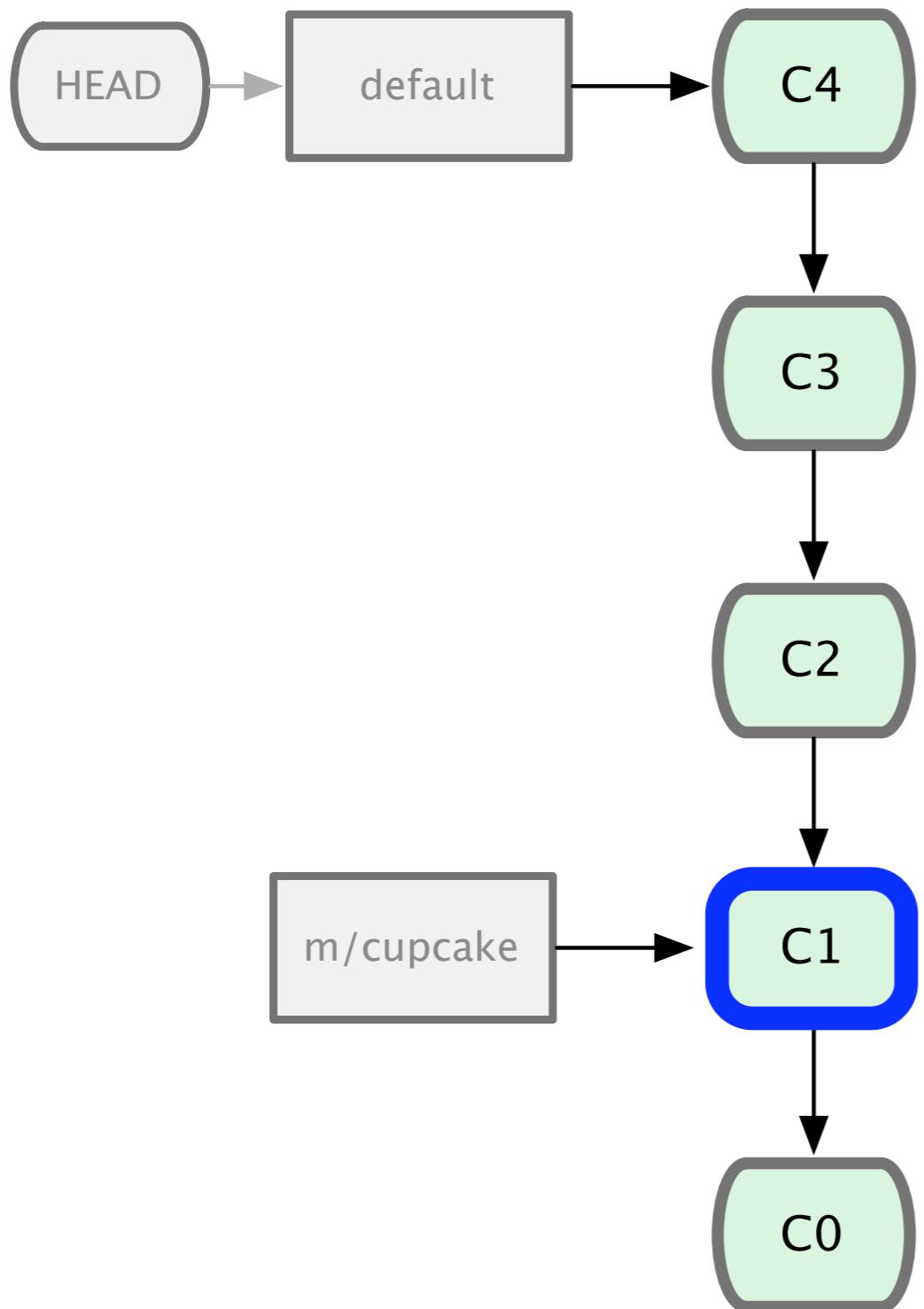
git log m/cupcake..



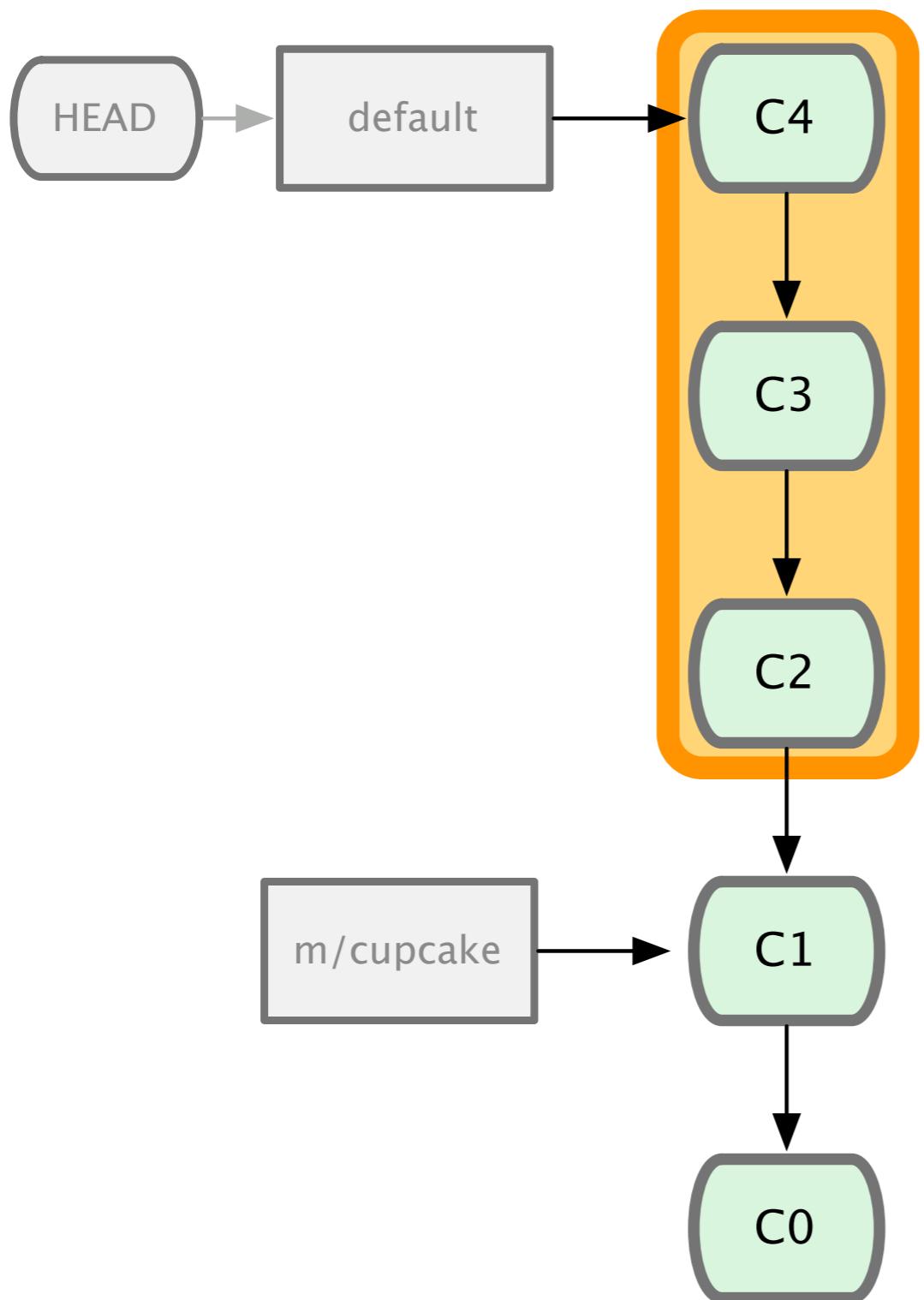
git log m/cupcake..HEAD



git log m/cupcake..HEAD



git log **m/cupcake**..HEAD



git log **m/cupcake..HEAD**

commit 72d404debaa804fca82fd9cf710fbde48c7305c6

Author: Shawn O. Pearce <spearce@spearce.org>

Date: Sun Oct 12 13:13:59 2008 -0700

test-lib: fix broken printf

b8eecaf888d219633f4c29e8b6a90fc21a46dfd introduced usage of
printf without a format string.

Signed-off-by: Shawn O. Pearce <spearce@spearce.org>

commit 969c877506cf8cc760c7b251fef6c5b6850bfc19

Author: Jeff King <peff@peff.net>

Date: Sun Oct 12 00:06:11 2008 -0400

git apply --directory broken for new files

We carefully verify that the input to git-apply is sane,
including cross-checking that the filenames we see in "+++"
headers match what was provided on the command line of "diff
--git". When --directory is used, however, we ended up
comparing the unadorned name to one with the prepended root,
causing us to complain about a mismatch.

We simply need to prepend the root directory, if any, when
pulling the name out of the git header.

commit ff74126c03a8dfd04e7533573a5c420f2a7112ac

Author: Johannes Schindelin <Johannes.Schindelin@gmx.de>

Date: Fri Oct 10 13:42:12 2008 +0200

rebase -i: do not fail when there is no commit to cherry-pick

C4

commit 72d404debaa804fca82fd9cf710fbde48c7305c6
Author: Shawn O. Pearce <spearce@spearce.org>
Date: Sun Oct 12 13:13:59 2008 -0700

test-lib: fix broken printf

b8eecaf888d219633f4c29e8b6a90fc21a46dfd introduced usage of printf without a format string.

Signed-off-by: Shawn O. Pearce <spearce@spearce.org>

C3

commit 969c877506cf8cc760c7b251fef6c5b6850bfc19
Author: Jeff King <peff@peff.net>
Date: Sun Oct 12 00:06:11 2008 -0400

git apply --directory broken for new files

We carefully verify that the input to git-apply is sane, including cross-checking that the filenames we see in "+++" headers match what was provided on the command line of "diff --git". When --directory is used, however, we ended up comparing the unadorned name to one with the prepended root, causing us to complain about a mismatch.

We simply need to prepend the root directory, if any, when pulling the name out of the git header.

C2

commit ff74126c03a8dfd04e7533573a5c420f2a7112ac
Author: Johannes Schindelin <Johannes.Schindelin@gmx.de>
Date: Fri Oct 10 13:42:12 2008 +0200

rebase -i: do not fail when there is no commit to cherry-pick


```
git log m/cupcake..
```

```
git log m/cupcake..
```

```
git log m/cupcake..HEAD
```

```
git log m/cupcake..
```

```
git log m/cupcake..HEAD
```

```
git log HEAD ^m/cupcake
```

```
git log m/cupcake..
```

```
git log m/cupcake..HEAD
```

```
git log HEAD ^m/cupcake
```

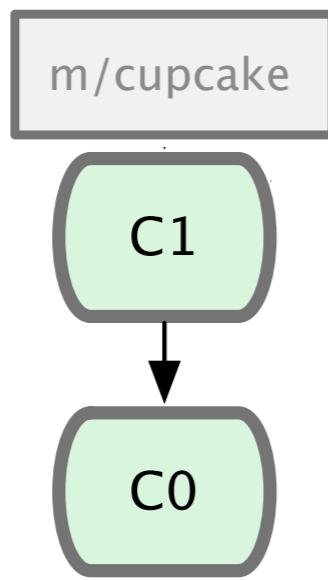
```
git log 35ae2 ^9fe42
```

```
git log 3e45a ^573eb
```

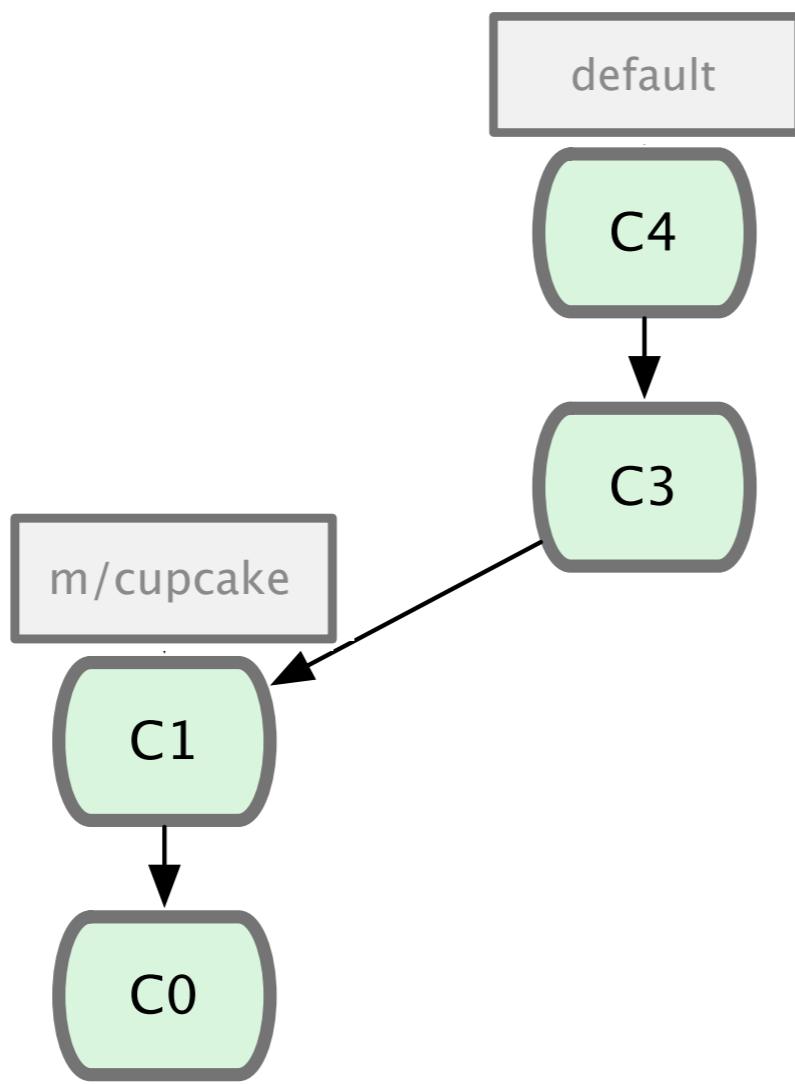
```
git log 3e45a ^573eb
```

```
git log 08cb1 ca374 ^c0bf9
```

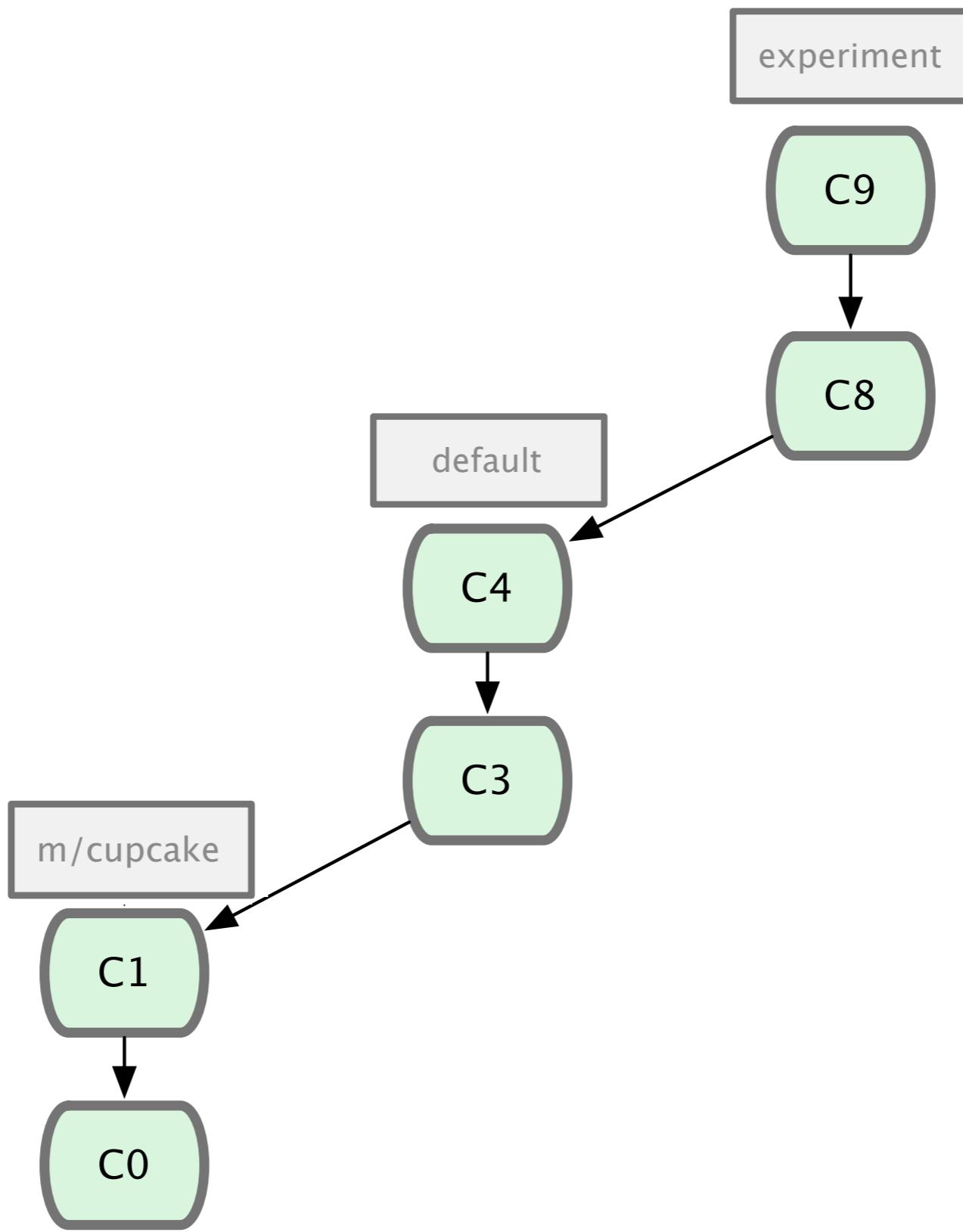
```
git log c5 c7 ^c9
```



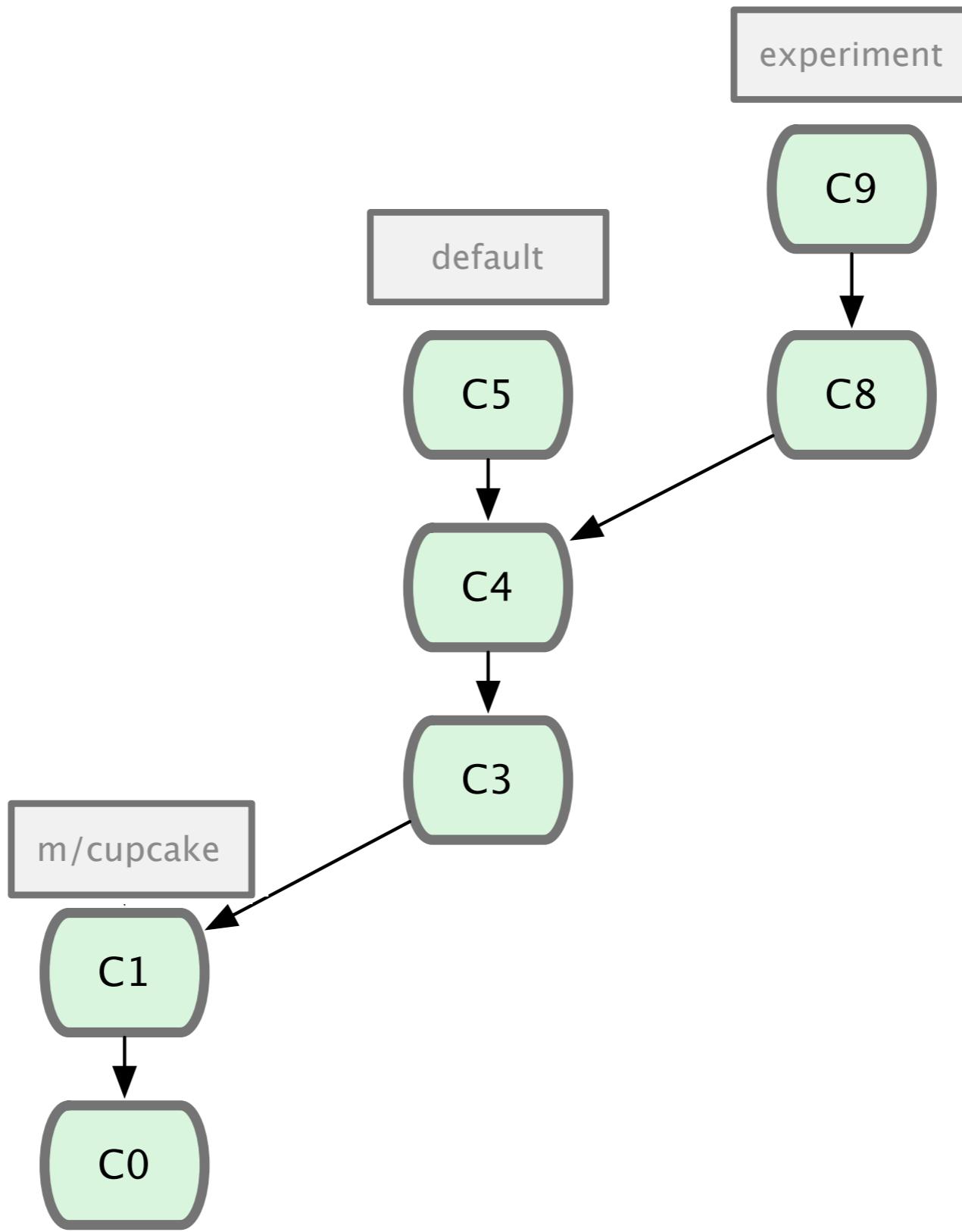
```
git log c5 c7 ^c9
```



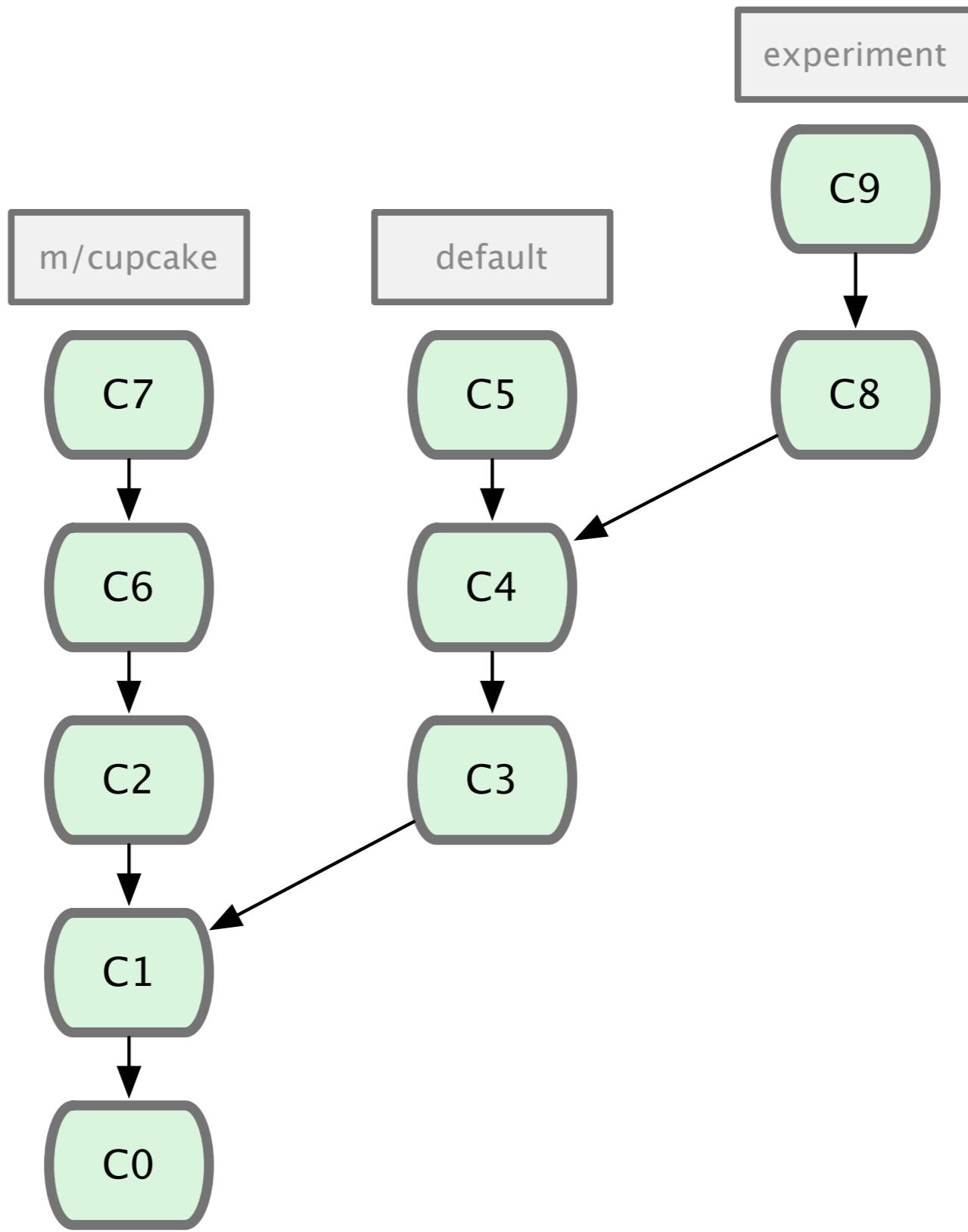
```
git log c5 c7 ^c9
```



```
git log c5 c7 ^c9
```

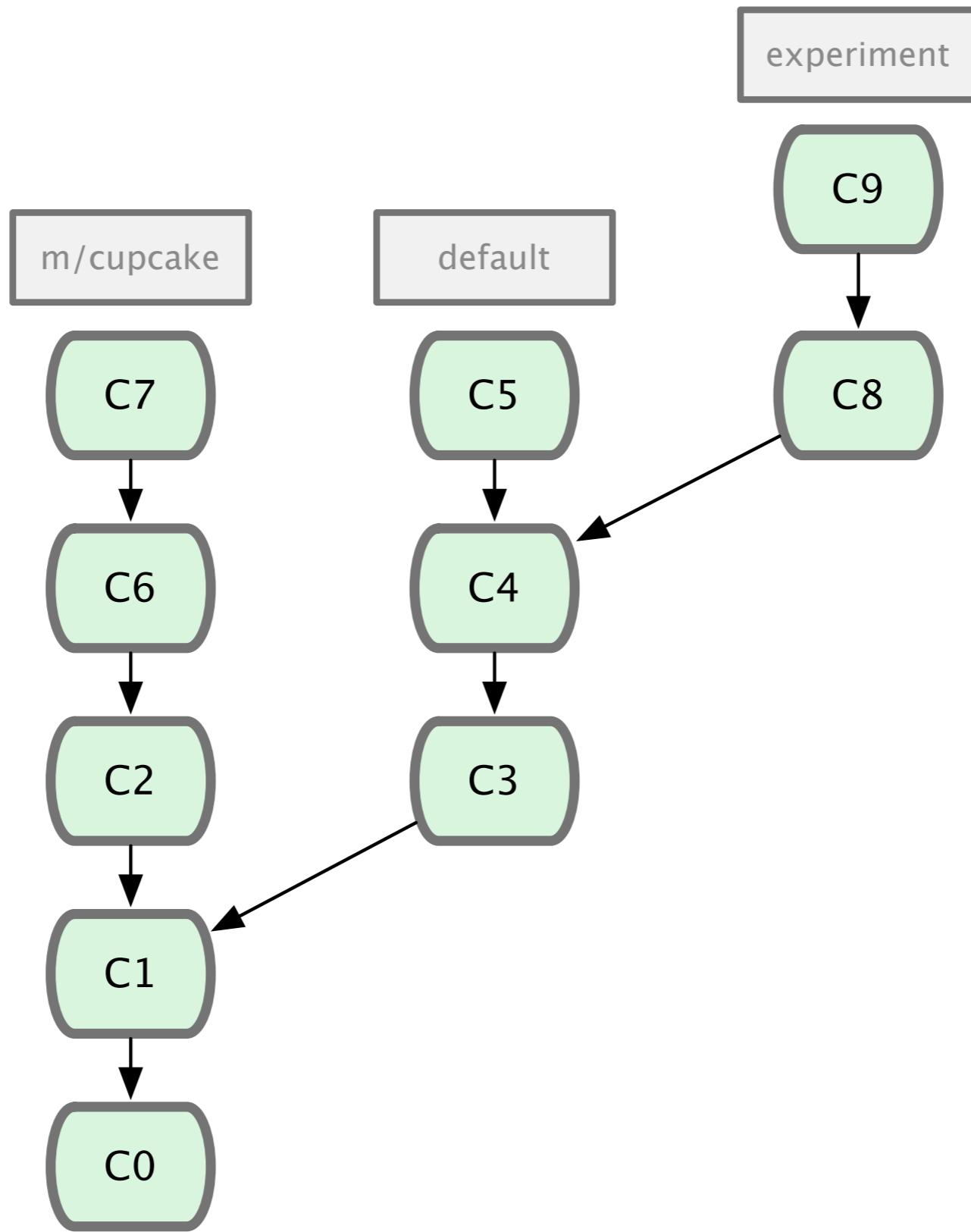


git log c5 c7 ^c9

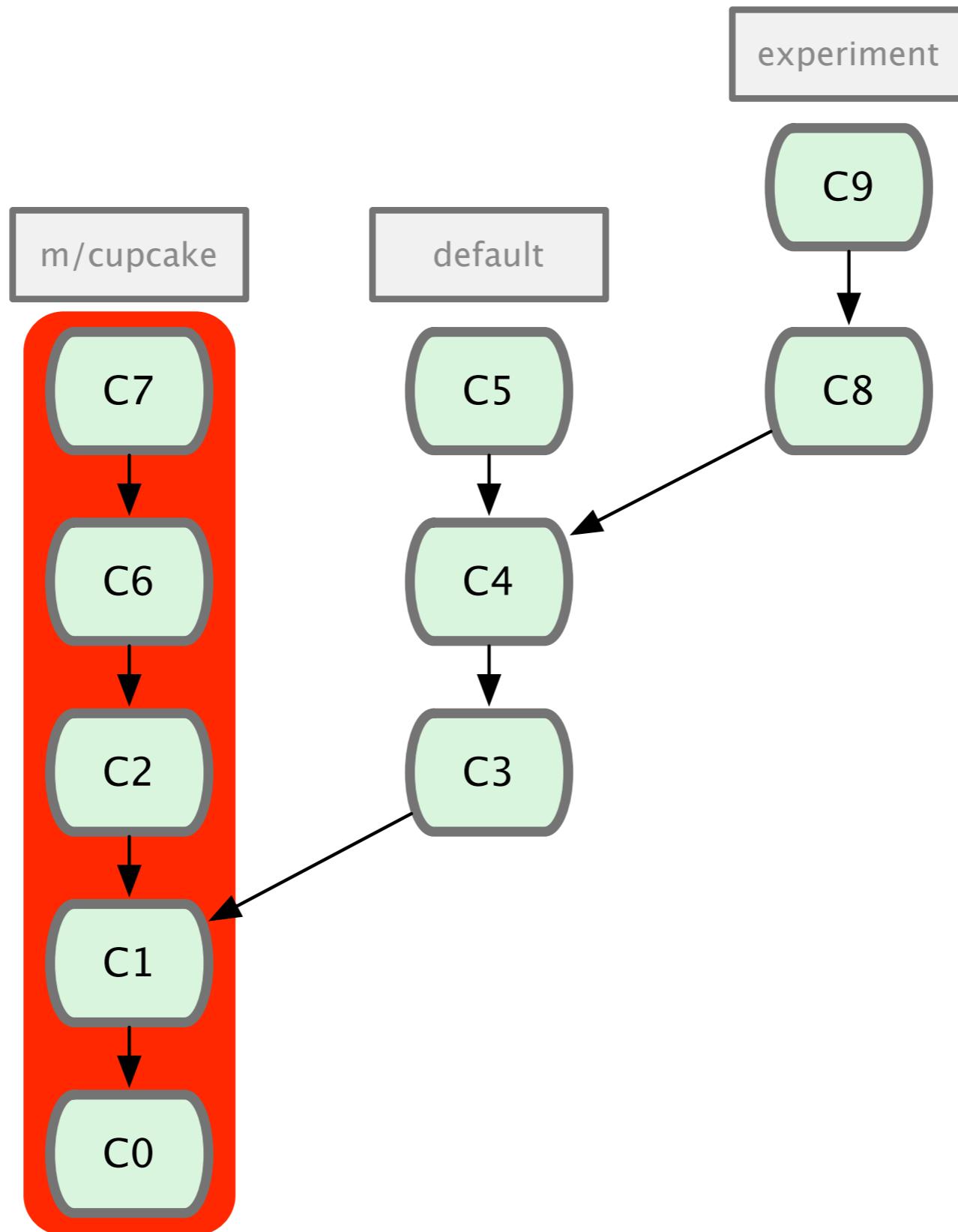


**which changes aren't in
experiment?**

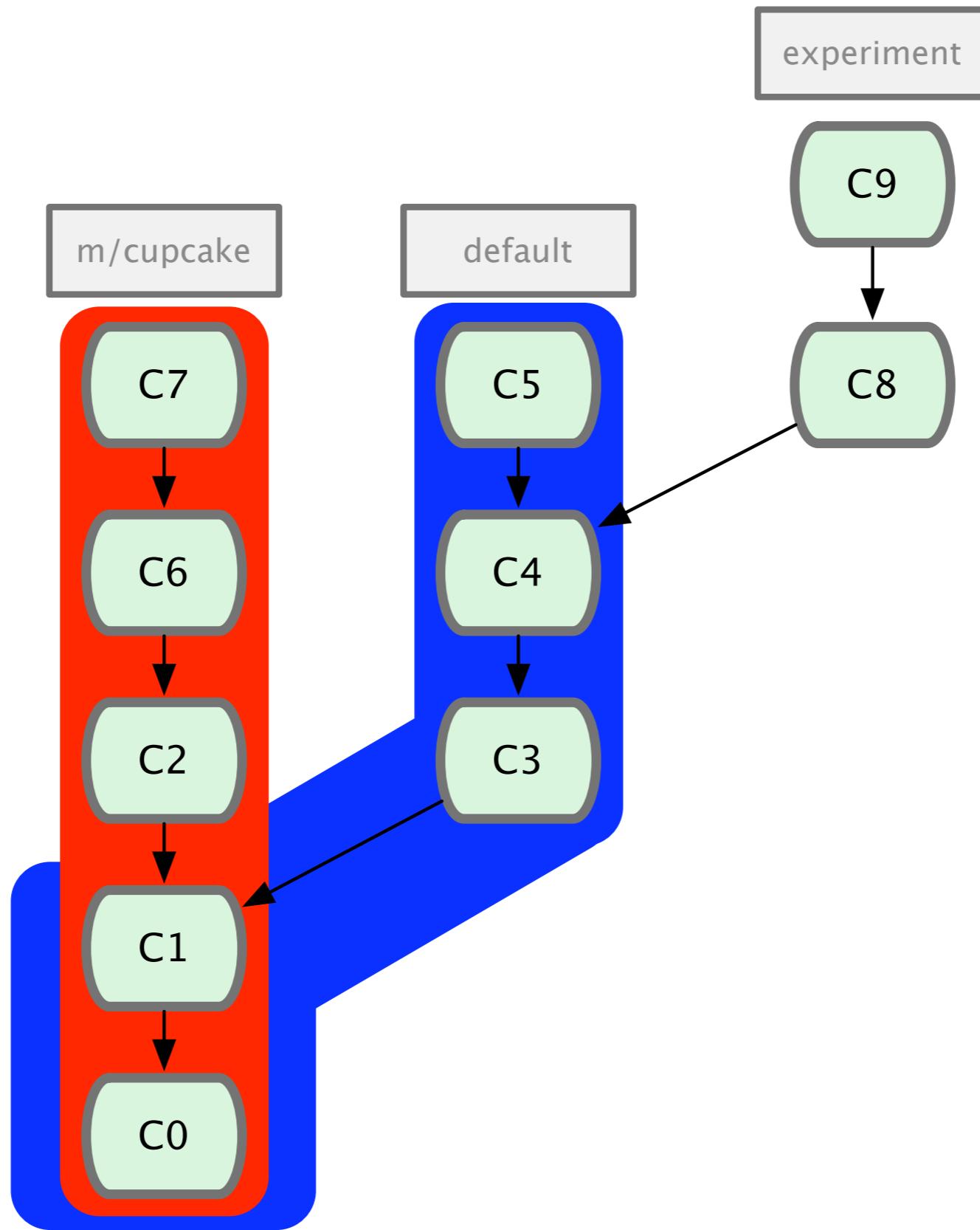
```
git log m/cupcake default ^experiment
```



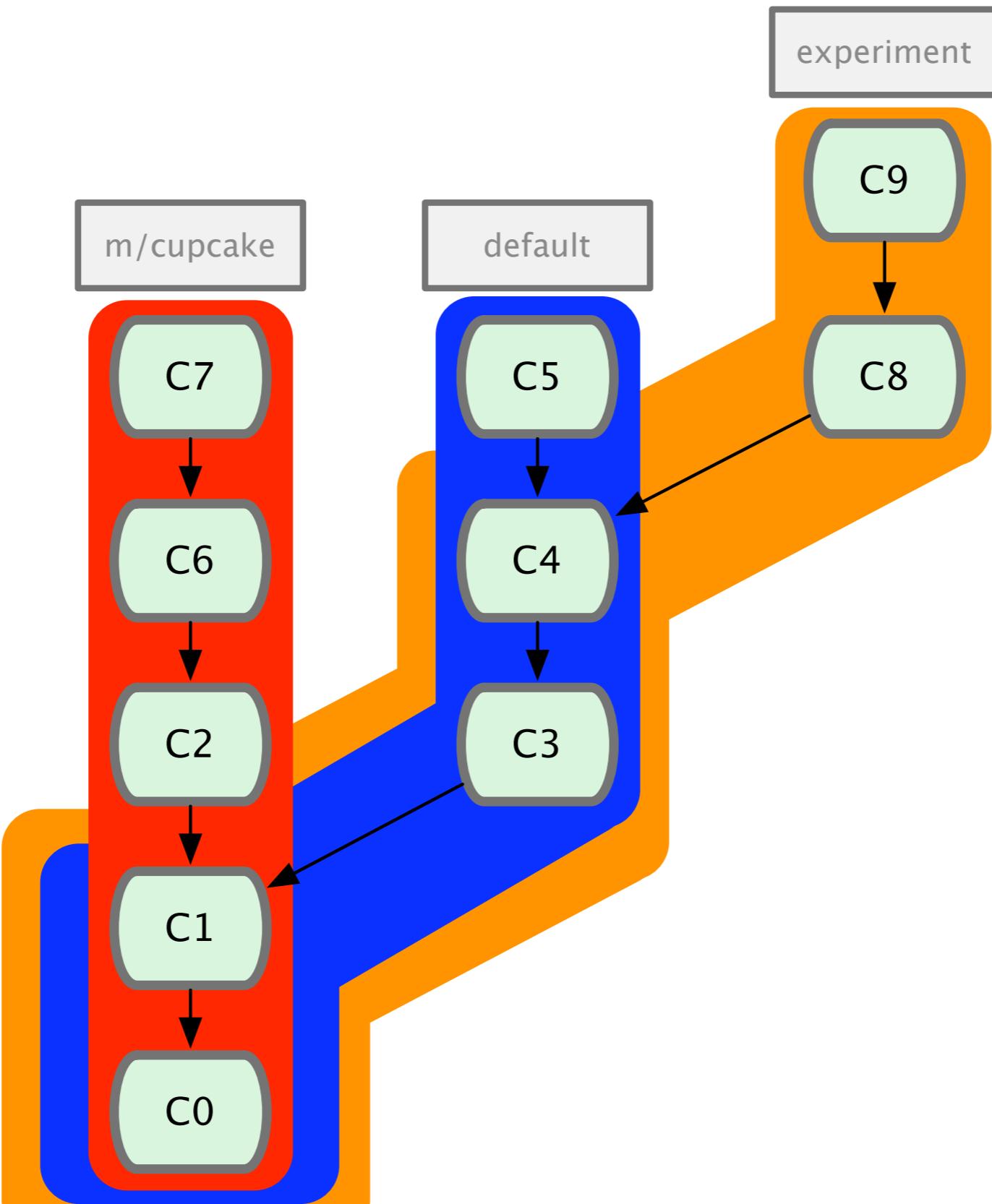
```
git log m/cupcake default ^experiment
```



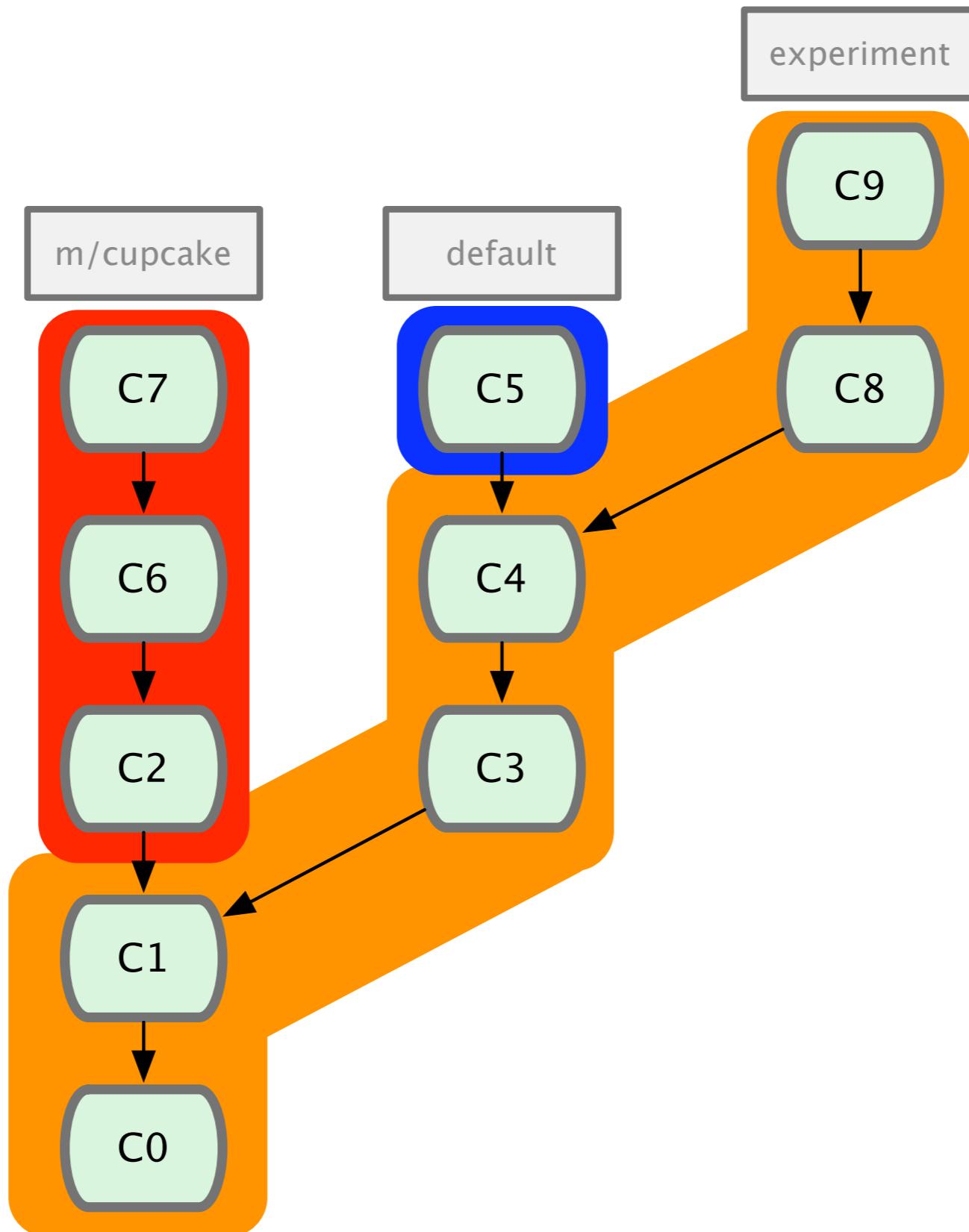
```
git log m/cupcake default ^experiment
```



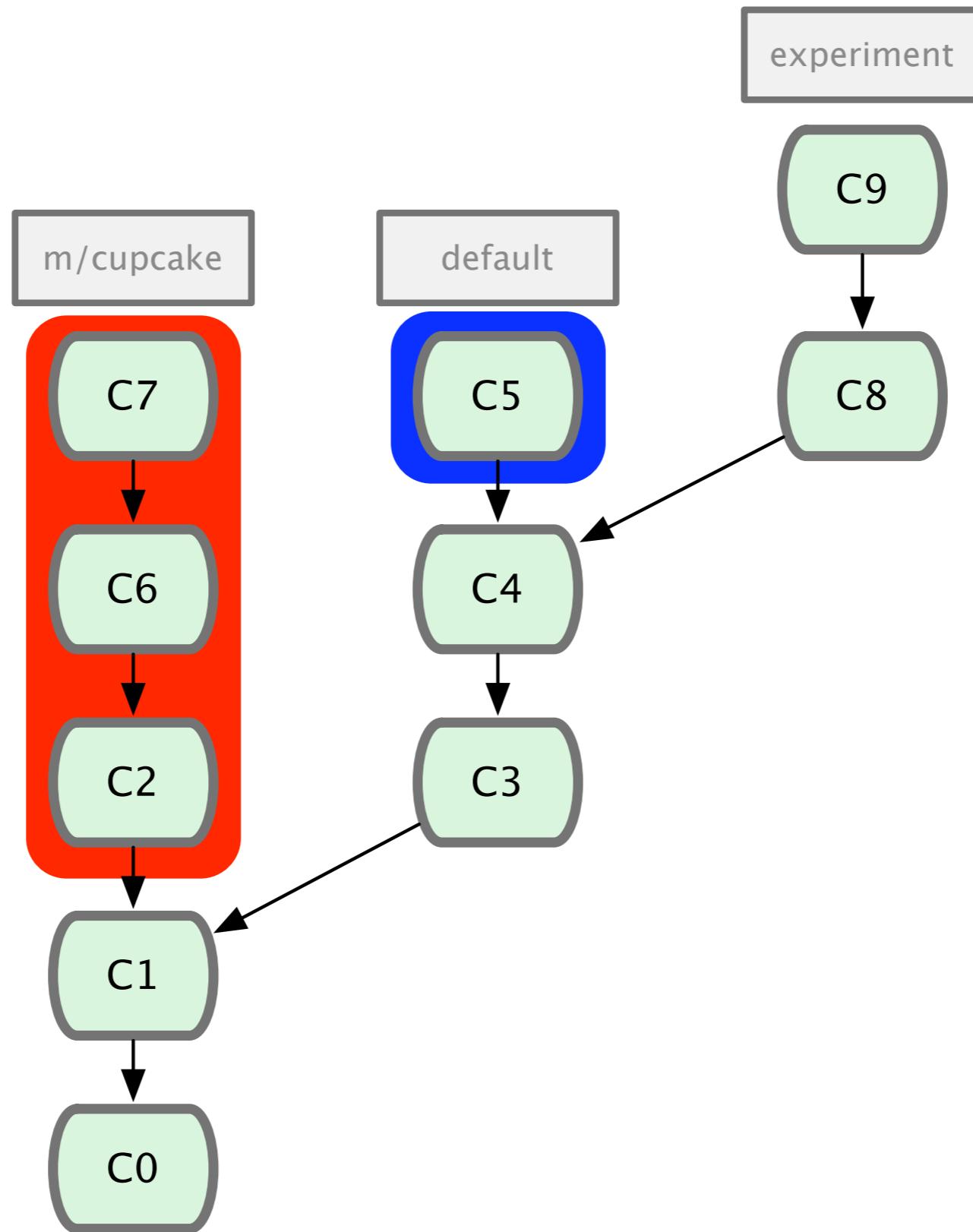
```
git log m/cupcake default ^experiment
```



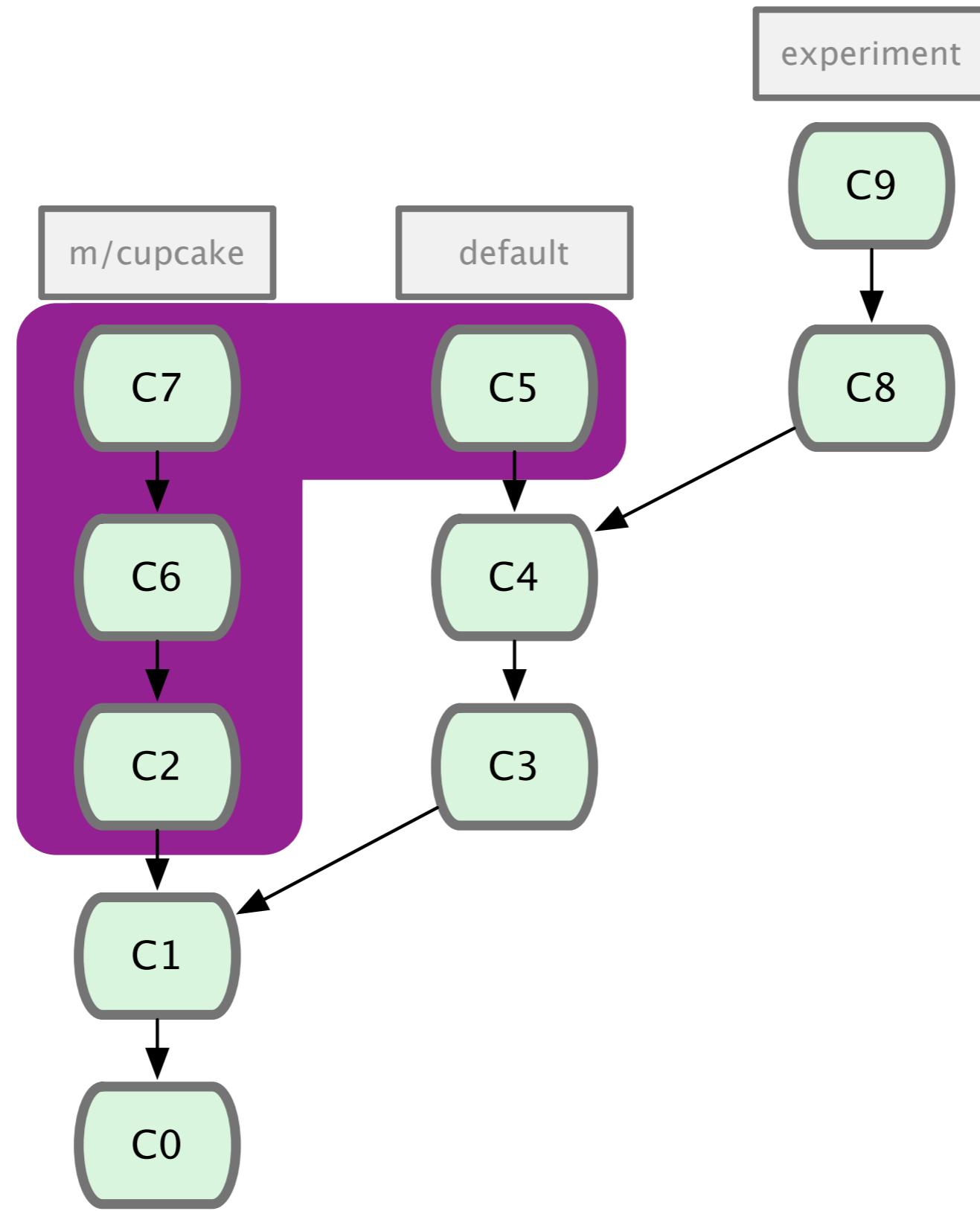
```
git log m/cupcake default ^experiment
```



```
git log m/cupcake default ^experiment
```



```
git log m/cupcake default ^experiment
```



Log Formatting

git log -P

```
commit 964f16d36dfccde844893cac5b347e7b3d44abbc
Author: Magnus Chacon <mchacon@gmail.com>
Date: Sun Apr 27 20:34:23 2008 -0700
```

commit the todo

```
diff --git a/TODO b/TODO
index da208f1..eaad748 100644
--- a/TODO
+++ b/TODO
@@ -3,3 +3,4 @@ Git Functions:
 * commit
 * rebase
 * merge
+* status
```

```
commit 8a5cbc430f1a9c3d00faaeffd07798508422908a
Author: Scott Chacon <schacon@gmail.com>
Date: Wed Sep 3 16:10:57 2008 -0700
```

updated readme

```
diff --git a/README b/README
index 8ea1587..c526f88 100644
--- a/README
+++ b/README
@@ -1,9 +1,10 @@
 SimpleGit Ruby Library
 =====
```

```
-This library calls git commands and returns the output.
+This library calls git commands and returns the output. Sweet.
```

It is an example for the Git Peepcode book that I'm currently writing.

```
Author : Scott Chacon (schacon@gmail.com)
+ Orange Peel Chacon (opchacon@gmail.com)
Magnus O. Chacon (mchacon@gmail.com)
```

```
commit a5f4a0daa9b13bb85283461ddaba6e589e34ccb5
Author: Magnus Chacon <mchacon@gmail.com>
Date: Sun Apr 13 10:51:45 2008 -0700
```

added cat-file

```
commit 964f16d36dfccde844893cac5b347e7b3d44abbc
Author: Magnus Chacon <mchacon@gmail.com>
Date:   Sun Apr 27 20:34:23 2008 -0700

    commit the todo
```

```
diff --git a/TODO b/TODO
index da208f1..eaad748 100644
--- a/TODO
+++ b/TODO
@@ -3,3 +3,4 @@ Git Functions:
 * commit
 * rebase
 * merge
+* status
```

```
commit 8a5cbc430f1a9c3d00faaeffd07798508422908a
Author: Scott Chacon <schacon@gmail.com>
Date:   Wed Sep 3 16:10:57 2008 -0700
```

updated readme

```
diff --git a/README b/README
index 8ea1587..c526f88 100644
--- a/README
+++ b/README
@@ -1,9 +1,10 @@
 SimpleGit Ruby Library
 =====
```

```
-This library calls git commands and returns the output.
+This library calls git commands and returns the output. Sweet.
```

It is an example for the Git Peepcode book that I'm currently writing.

```
Author : Scott Chacon (schacon@gmail.com)
+      Orange Peel Chacon (opchacon@gmail.com)
      Magnus O. Chacon (mchacon@gmail.com)
```

```
commit a5f4a0daa9b13bb85283461ddaba6e589e34ccb5
Author: Magnus Chacon <mchacon@gmail.com>
Date:   Sun Apr 13 10:51:45 2008 -0700
```

added cat-file

```
commit 964f16d36dfccde844893cac5b347e7b3d44abbc
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 27 20:34:23 2008 -0700
```

```
commit the todo
```

```
diff --git a/TODO b/TODO
index da208f1..eaad748 100644
--- a/TODO
+++ b/TODO
@@ -3,3 +3,4 @@ Git Functions:
 * commit
 * rebase
 * merge
+* status
```

```
commit 8a5cbc430f1a9c3d00faaeffd07798508422908a
```

```
Author: Scott Chacon <schacon@gmail.com>
```

```
Date: Wed Sep 3 16:10:57 2008 -0700
```

```
updated readme
```

```
diff --git a/README b/README
index 8ea1587..c526f88 100644
--- a/README
+++ b/README
@@ -1,9 +1,10 @@
 SimpleGit Ruby Library
 =====
```

```
-This library calls git commands and returns the output.
```

```
+This library calls git commands and returns the output. Sweet.
```

```
It is an example for the Git Peepcode book that I'm currently writing.
```

```
Author : Scott Chacon (schacon@gmail.com)
```

```
+ Orange Peel Chacon (opchacon@gmail.com)
```

```
Magnus O. Chacon (mchacon@gmail.com)
```

```
commit a5f4a0daa9b13bb85283461ddaba6e589e34ccb5
```

```
Author: Magnus Chacon <mchacon@gmail.com>
```

```
Date: Sun Apr 13 10:51:45 2008 -0700
```

```
added cat-file
```

```
commit 964f16d36dfccde844893cac5b347e7b3d44abbc
Author: Magnus Chacon <mchacon@gmail.com>
Date: Sun Apr 27 20:34:23 2008 -0700
```

commit the todo

```
diff --git a/TODO b/TODO
index da208f1..eaad748 100644
--- a/TODO
+++ b/TODO
@@ -3,3 +3,4 @@ Git Functions:
 * commit
 * rebase
 * merge
+* status
```

```
commit 8a5cbc430f1a9c3d00faaeffd07798508422908a
Author: Scott Chacon <schacon@gmail.com>
Date: Wed Sep 3 16:10:57 2008 -0700
```

updated readme

```
diff --git a/README b/README
index 8ea1587..c526f88 100644
--- a/README
+++ b/README
@@ -1,9 +1,10 @@
 SimpleGit Ruby Library
 =====
```

```
-This library calls git commands and returns the output.
+This library calls git commands and returns the output. Sweet.
```

It is an example for the Git Peepcode book that I'm currently writing.

```
Author : Scott Chacon (schacon@gmail.com)
+ Orange Peel Chacon (opchacon@gmail.com)
Magnus O. Chacon (mchacon@gmail.com)
```

```
commit a5f4a0daa9b13bb85283461ddaba6e589e34ccb5
Author: Magnus Chacon <mchacon@gmail.com>
Date: Sun Apr 13 10:51:45 2008 -0700
```

added cat-file

```
commit 964f16d36dfccde844893cac5b347e7b3d44abbc
Author: Magnus Chacon <mchacon@gmail.com>
Date: Sun Apr 27 20:34:23 2008 -0700
```

commit the todo

```
diff --git a/TODO b/TODO
index da208f1..eaad748 100644
--- a/TODO
+++ b/TODO
@@ -3,3 +3,4 @@ Git Functions:
 * commit
 * rebase
 * merge
+* status
```

```
commit 8a5cbc430f1a9c3d00faaeffd07798508422908a
Author: Scott Chacon <schacon@gmail.com>
Date: Wed Sep 3 16:10:57 2008 -0700
```

updated readme

```
diff --git a/README b/README
index 8ea1587..c526f88 100644
--- a/README
+++ b/README
@@ -1,9 +1,10 @@
 SimpleGit Ruby Library
 =====

-This library calls git commands and returns the output.
+This library calls git commands and returns the output. Sweet.
```

It is an example for the Git Peepcode book that I'm currently writing.

```
Author : Scott Chacon (schacon@gmail.com)
+ Orange Peel Chacon (opchacon@gmail.com)
Magnus O. Chacon (mchacon@gmail.com)
```

```
commit a5f4a0daa9b13bb85283461ddaba6e589e34ccb5
Author: Magnus Chacon <mchacon@gmail.com>
Date: Sun Apr 13 10:51:45 2008 -0700
```

added cat-file

git log -p

```
commit 964f16d36dfccde844893cac5b347e7b3d44abbc
Author: Magnus Chacon <mchacon@gmail.com>
Date:   Sun Apr 27 20:34:23 2008 -0700
```

commit the todo

```
diff --git a/TODO b/TODO
index da208f1..eaad748 100644
--- a/TODO
+++ b/TODO
@@ -3,3 +3,4 @@ Git Functions:
 * commit
 * rebase
 * merge
+* status
```

```
commit 8a5cbc430f1a9c3d00faaeffd07798508422908a
Author: Scott Chacon <schacon@gmail.com>
Date:   Wed Sep 3 16:10:57 2008 -0700
```

updated readme

```
diff --git a/README b/README
index 8ea1587..c526f88 100644
--- a/README
+++ b/README
@@ -1,9 +1,10 @@
 SimpleGit Ruby Library
 =====
```

```
-This library calls git commands and returns the output.
+This library calls git commands and returns the output. Sweet.
```

It is an example for the Git Peepcode book that I'm currently writing.

```
Author : Scott Chacon (schacon@gmail.com)
+      Orange Peel Chacon (opchacon@gmail.com)
      Magnus O. Chacon (mchacon@gmail.com)
```

```
commit a5f4a0daa9b13bb85283461ddaba6e589e34ccb5
Author: Magnus Chacon <mchacon@gmail.com>
Date:   Sun Apr 13 10:51:45 2008 -0700
```

added cat-file

git log --pretty

```
$>git log --pretty=oneline
```

```
$>git log --pretty=oneline
```

```
3e625e2139a71bfbaa85b055c139bceafab7e2c0 updated rakefile
4fefbaafd51ff3e79b84e9ee1a0dac2848e21a98 commit the todo
c4f8fd68ae3897d1d10456ed2b93d87cdc53d6e8 updated README
a5f4a0daa9b13bb85283461ddaba6e589e34ccb5 added cat-file
310154e3c7db47d8bac935c2c43aee6afac11aae updated README formatting
f7f3f6dd8fd3fa40f052427c32785a0fa01aaa5f changed my name a bit
710f0f8d2cdf5af87033b9ec08859a505f9a6af5 added ls-files
c110d7ff8cfb86fd5cce9a8aee462678dbb4ef9b made the ls-tree function
ce9b0d5551762048735dd67917046b44176317e0 limiting log to 30
e22e01e39c39521edc8ccd3aa5df228209e7ad2d -added todo options
2c0d4d7e3d36a8cd5daleab3e0c1e83faa3bfd9f added limit to log functio
47c668a9e8f93f65ff9019c587f3b2204bd6c98d fixed conflict
ca28a28e64192af4527ea63665dceaa9bdb4a4b new version 0.1.2
f30ba8480d9aba88580696651e80d9aab8e15a2 new version 0.2.0
1edee6b1d61823a2de3b09c160d7080b8d1b3a40 added a new function
4b0780c5b7a4ae1217f4da742a172e7d787ca3eb rakefile and todo file add
0d57a4eb03d35e1e1041857d1cb01199eeb23cc4 Revert "rakefile and todo
f65a297eb0ace63e5b302e849064f37ded909c0b rakefile and todo file add
c035c611b559537d725d875cfa10c05c17d48e4f Merge branch 'newfunc'
d6fad7dd6f98481cab874b192aaa79d03490d4a0 added more description
1a8c32ea9725886aaadc35dbf4d68468d3af0abe added lmtree function
40e5c10f67fa01c01da5687d08bd3bc21bb6a44b added email to readme
cf25cc3bfb0ece7dc3609b8dc0cc4a1e19ffbcd4 committing all changes
0c8a9ec46029a4e92a428cb98c9693f09f69a3ff changed the verison number
```

```
$>git log --pretty=format:"%h %an %ar - %s"__
```

```
$>git log --pretty=format:"%h %an %ar - %s"
3e625e2 Magnus Chacon 47 minutes ago - updated rakefile
4fefbaa Magnus Chacon 56 minutes ago - commit the todo
c4f8fd6 Magnus Chacon 60 minutes ago - updated README
a5f4a0d Magnus Chacon 2 weeks ago - added cat-file
310154e Scott Chacon 2 weeks ago - updated README formatting and a
f7f3f6d Magnus Chacon 2 weeks ago - changed my name a bit
710f0f8 Magnus Chacon 2 weeks ago - added ls-files
c110d7f Scott Chacon 3 weeks ago - made the ls-tree function recur
ce9b0d5 Scott Chacon 3 weeks ago - limiting log to 30
e22e01e Scott Chacon 3 weeks ago - -added todo options
2c0d4d7 Scott Chacon 3 weeks ago - added limit to log function
47c668a Scott Chacon 3 weeks ago - fixed conflict
ca28a28 Scott Chacon 3 weeks ago - new version 0.1.2
f30ba84 Scott Chacon 3 weeks ago - new version 0.2.0
1edee6b Scott Chacon 3 weeks ago - added a new function
4b0780c Scott Chacon 3 weeks ago - rakefile and todo file added
0d57a4e Scott Chacon 3 weeks ago - Revert "rakefile and todo file
f65a297 Scott Chacon 3 weeks ago - rakefile and todo file added
c035c61 Scott Chacon 3 weeks ago - Merge branch 'newfunc'
d6fad7d Scott Chacon 3 weeks ago - added more description
1a8c32e Scott Chacon 3 weeks ago - added lstree function
40e5c10 Scott Chacon 3 weeks ago - added email to readme
cf25cc3 Scott Chacon 6 weeks ago - committing all changes
0c8a9ec Scott Chacon 6 weeks ago - changed the verison number
0576fac Scott Chacon 6 weeks ago - my second commit which is bett
```

```
$>git log --pretty=format:"%h %an %ar - %s"
3e625e2 Magnus Chacon 47 minutes ago - updated rakefile
4fefbaa Magnus Chacon 56 minutes ago - commit the todo
c4f8fd6 Magnus Chacon 60 minutes ago - updated README
a5f4a0d Magnus Chacon 2 weeks ago - added cat-file
310154e Scott Chacon 2 weeks ago - updated README formatting and a
f7f3f6d Magnus Chacon 2 weeks ago - changed my name a bit
710f0f8 Magnus Chacon 2 weeks ago - added ls-files
c110d7f Scott Chacon 3 weeks ago - made the ls-tree function recur
ce9b0d5 Scott Chacon 3 weeks ago - limiting log to 30
e22e01e Scott Chacon 3 weeks ago - -added todo options
2c0d4d7 Scott Chacon 3 weeks ago - added limit to log function
47c668a Scott Chacon 3 weeks ago - fixed conflict
ca28a28 Scott Chacon 3 weeks ago - new version 0.1.2
f30ba84 Scott Chacon 3 weeks ago - new version 0.2.0
1edee6b Scott Chacon 3 weeks ago - added a new function
4b0780c Scott Chacon 3 weeks ago - rakefile and todo file added
0d57a4e Scott Chacon 3 weeks ago - Revert "rakefile and todo file
f65a297 Scott Chacon 3 weeks ago - rakefile and todo file added
c035c61 Scott Chacon 3 weeks ago - Merge branch 'newfunc'
d6fad7d Scott Chacon 3 weeks ago - added more description
1a8c32e Scott Chacon 3 weeks ago - added lstree function
40e5c10 Scott Chacon 3 weeks ago - added email to readme
cf25cc3 Scott Chacon 6 weeks ago - committing all changes
0c8a9ec Scott Chacon 6 weeks ago - changed the verison number
0576fac Scott Chacon 6 weeks ago - my second commit which is bett
```

```
$>git log --pretty=format:"%h %an %ar - %s"
3e625e2 Magnus Chacon 47 minutes ago - updated rakefile
4fefbaa Magnus Chacon 56 minutes ago - commit the todo
c4f8fd6 Magnus Chacon 60 minutes ago - updated README
a5f4a0d Magnus Chacon 2 weeks ago - added cat-file
310154e Scott Chacon 2 weeks ago - updated README formatting and a
f7f3f6d Magnus Chacon 2 weeks ago - changed my name a bit
710f0f8 Magnus Chacon 2 weeks ago - added ls-files
c110d7f Scott Chacon 3 weeks ago - made the ls-tree function recur
ce9b0d5 Scott Chacon 3 weeks ago - limiting log to 30
e22e01e Scott Chacon 3 weeks ago - -added todo options
2c0d4d7 Scott Chacon 3 weeks ago - added limit to log function
47c668a Scott Chacon 3 weeks ago - fixed conflict
ca28a28 Scott Chacon 3 weeks ago - new version 0.1.2
f30ba84 Scott Chacon 3 weeks ago - new version 0.2.0
1edee6b Scott Chacon 3 weeks ago - added a new function
4b0780c Scott Chacon 3 weeks ago - rakefile and todo file added
0d57a4e Scott Chacon 3 weeks ago - Revert "rakefile and todo file
f65a297 Scott Chacon 3 weeks ago - rakefile and todo file added
c035c61 Scott Chacon 3 weeks ago - Merge branch 'newfunc'
d6fad7d Scott Chacon 3 weeks ago - added more description
1a8c32e Scott Chacon 3 weeks ago - added lstree function
40e5c10 Scott Chacon 3 weeks ago - added email to readme
cf25cc3 Scott Chacon 6 weeks ago - committing all changes
0c8a9ec Scott Chacon 6 weeks ago - changed the verison number
0576fac Scott Chacon 6 weeks ago - my second commit which is bett
```

`git log --graph`

```
git log --pretty=oneline --graph
```

```
* 20d4111 - fix copy strategy for production
* 86ef476 - confirm daemon servering is on for public imports
* 990bc8b - Merge branch 'friendly_delete'
| \
| * ed555b7 - better deletion flow
| * eec7491 - verify private forks aren't deleted
| * d65fc92 - simple fork deletion logic
| * e9370f4 - one more test for the gipper
| * 1d7c276 - remove busted test, add deleted template test
| * c0bf99f - let forks know their private network was deleted
* | ef2a5e2 - out of sight, out of mind
* | d352768 - guess we can turn this off now
* | d0f25c0 - use default timeout
* | 2e6eeae - convert executes to AR:B
* | e6b8628 - Merge branch 'mirrors'
| \
| * | ca3744f - couple mirror bug fixes
| * | 56b8868 - mirroring
* | | 08cb112 - bang. ya dead.
| / /
* | c153b21 - let AR:B figure it out
* | 19e66a2 - raw private gists are ssl'd
* | fdda4ac - Merge branch 'master' of git@github.com:defunkt/github
| \
| * | 0e9cf1 - x add slice39 to the deploy role
| * | 82703eb - Revert "update newrelic plugin"
| * | 77faff1 - turn on tracer
| * | 3807239 - update newrelic plugin
* | | 0aaehh2 - sexy UP inbox messages
```

gitk

gitk: github

fix copy strategy for production
confirm daemon servering is on for public imports
Merge branch 'friendly_delete'
better deletion flow
verify private forks aren't deleted
simple fork deletion logic
one more test for the gipper
remove busted test, add deleted template test
let forks know their private network was deleted
out of sight, out of mind
guess we can turn this off now
use default timeout
convert executes to AR:B
Merge branch 'mirrors'
couple mirror bug fixes
mirroring
bang. ya dead.
let AR:B figure it out
raw private gists are ssl'd
Merge branch 'master' of git@github.com:defunkt/github
x add slice39 to the deploy role
Report "undate no url fix plugin"

PJ Hyett <pjhyett@gmail.com> 2008-09-30 16:49:34
PJ Hyett <pjhyett@gmail.com> 2008-09-29 18:17:19
PJ Hyett <pjhyett@gmail.com> 2008-09-29 16:07:04
PJ Hyett <pjhyett@gmail.com> 2008-09-29 16:05:17
PJ Hyett <pjhyett@gmail.com> 2008-09-29 15:31:02
PJ Hyett <pjhyett@gmail.com> 2008-09-29 15:13:51
PJ Hyett <pjhyett@gmail.com> 2008-09-22 17:03:25
PJ Hyett <pjhyett@gmail.com> 2008-09-22 16:52:40
PJ Hyett <pjhyett@gmail.com> 2008-09-19 18:31:50
PJ Hyett <pjhyett@gmail.com> 2008-09-29 12:46:18
PJ Hyett <pjhyett@gmail.com> 2008-09-29 12:09:16
PJ Hyett <pjhyett@gmail.com> 2008-09-29 11:57:49
PJ Hyett <pjhyett@gmail.com> 2008-09-26 17:46:50
PJ Hyett <pjhyett@gmail.com> 2008-09-26 15:32:34
PJ Hyett <pjhyett@gmail.com> 2008-09-26 15:32:09
PJ Hyett <pjhyett@gmail.com> 2008-09-26 15:17:37
Chris Wanstrath <chris@ozmm.org> 2008-09-26 14:46:15
PJ Hyett <pjhyett@gmail.com> 2008-09-24 18:29:54
Chris Wanstrath <chris@ozmm.org> 2008-09-24 14:31:31
Chris Wanstrath <chris@ozmm.org> 2008-09-23 18:48:04
PJ Hyett <pjhyett@gmail.com> 2008-09-23 17:35:29
PJ Hyett <pjhyett@gmail.com> 2008-09-23 17:28:58

SHA1 ID: 20d4113a586b0ed05f530e6f7f4b895540da0f0

Find next prev commit containing: Exact All fields

Search Patch Tree

Diff Old version New version Lines of context: 3 Ignore space change

Author: PJ Hyett <pjhyett@gmail.com> 2008-09-30 16:49:34
Committer: PJ Hyett <pjhyett@gmail.com> 2008-09-30 16:49:34
Parent: [86ef47642bf2a1c67820de999e15437601e872bb](#) (confirm daemon servering is on for public
Branches: master, remotes/github/master, solr, training, wiki2
Follows:
Precedes:

fix copy strategy for production

----- app/models/repository.rb -----
index 09b8659..574ab6b 100644
@@ -82,11 +82,7 @@ class Repository < ActiveRecord::Base
end

gitk 6d13f..41aba

gitk: github

The screenshot shows a commit history for the master branch of the GitHub repository. The commits are listed in chronological order from top to bottom:

- master (solr) made the solr client a config setting and integrated it with the code search function
- first pass at a site-wide code search function
- remotes/github/master fixed #877 - gist private clone broken
- specifying embeded font color that we are assuming is black
- fixed #904 - fogbugz and friendfeed should work fine now

Each commit is associated with a timestamp and the author's email address (Scott Chacon <schacon@gmail.com>). The commit IDs are: 41aba2bc4065c96097550a2f1d053a3bd69c3a69, 2008-10-13 21:00:56; 2008-10-13 17:39:43, 2008-10-09 18:24:58, 2008-10-09 18:04:34, and 2008-10-09 16:34:51 respectively.

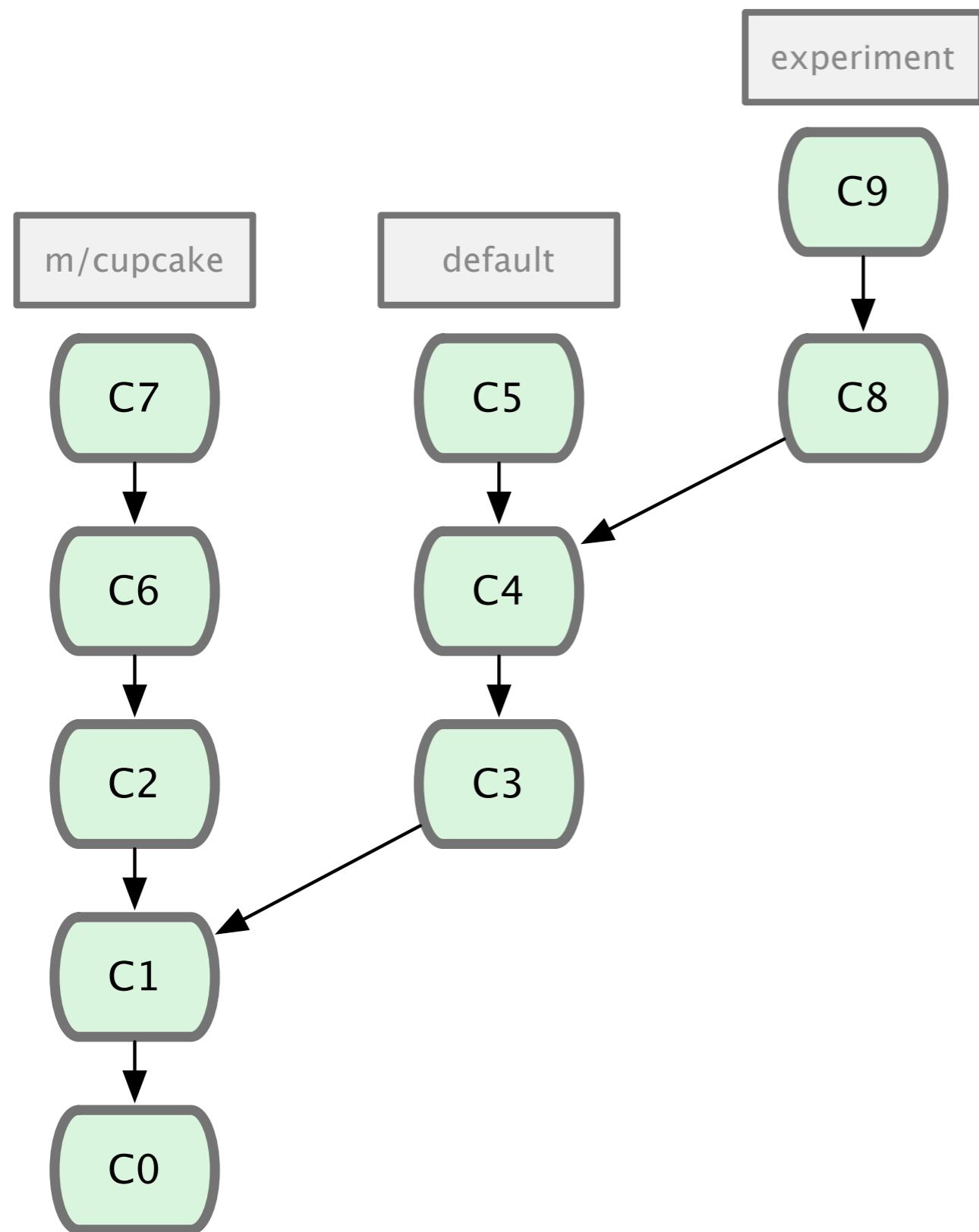
Below the commit list, there are navigation buttons: SHA1 ID, Find, next, prev, commit, containing, Row, 1, 5, Exact, All fields, and a search bar. On the right side, there are tabs for Patch and Tree, with Patch selected. The patch view shows the diff between two versions of the file app/controllers/codesearch_controller.rb. The diff highlights changes in the code, such as the addition of a configuration setting and the removal of a specific line related to Solr connection.

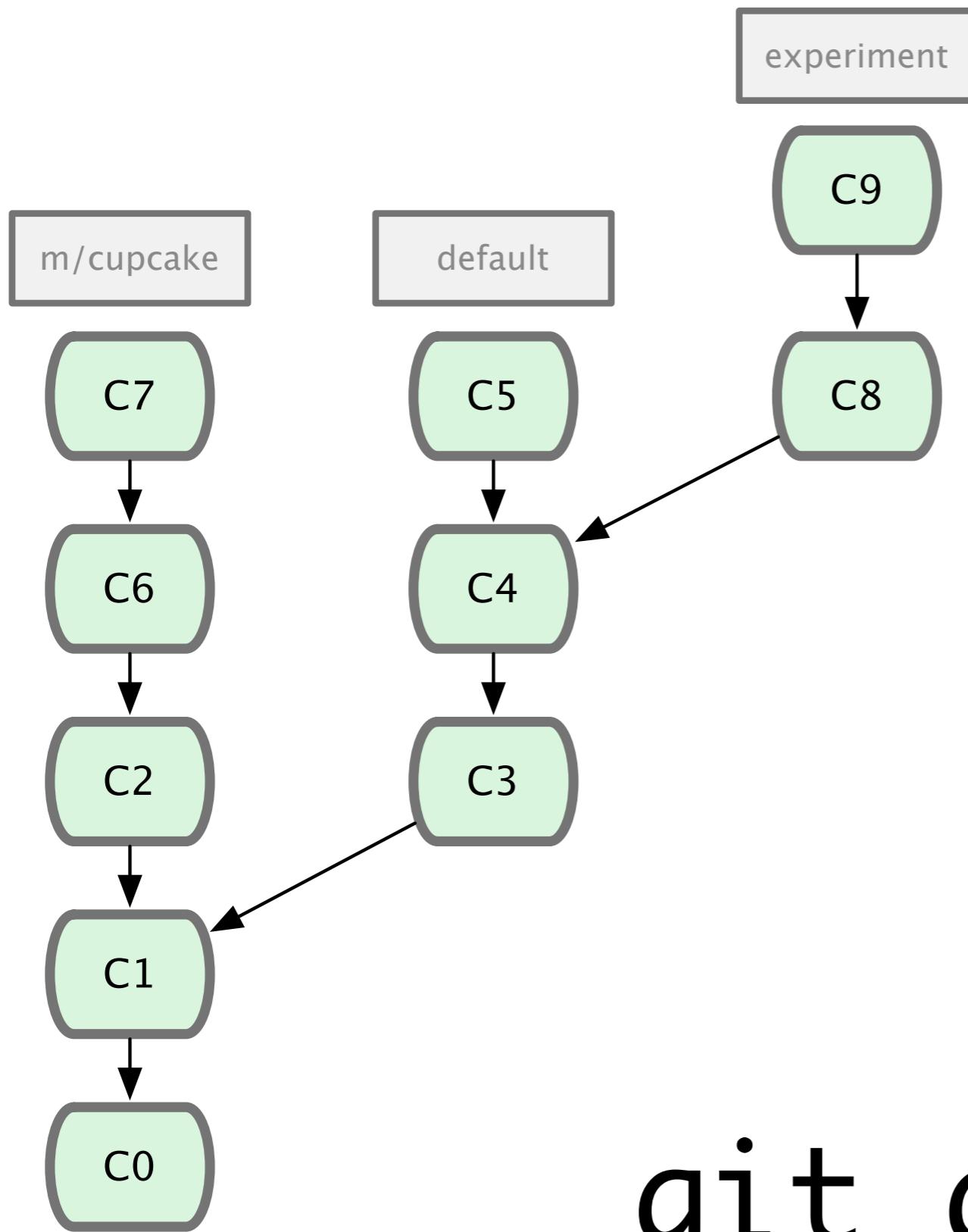
Comments for the commit are listed on the right:

- app/controllers/codesearch_controller.rb
- app/controllers/tree_controller.rb
- app/views/codesearch/_results.html.erb
- app/views/codesearch/index.html.erb
- app/views/repositories/_details.html.erb
- app/views/tree/search_code.html.erb
- config/initializers/solr.rb
- config/solr.yml

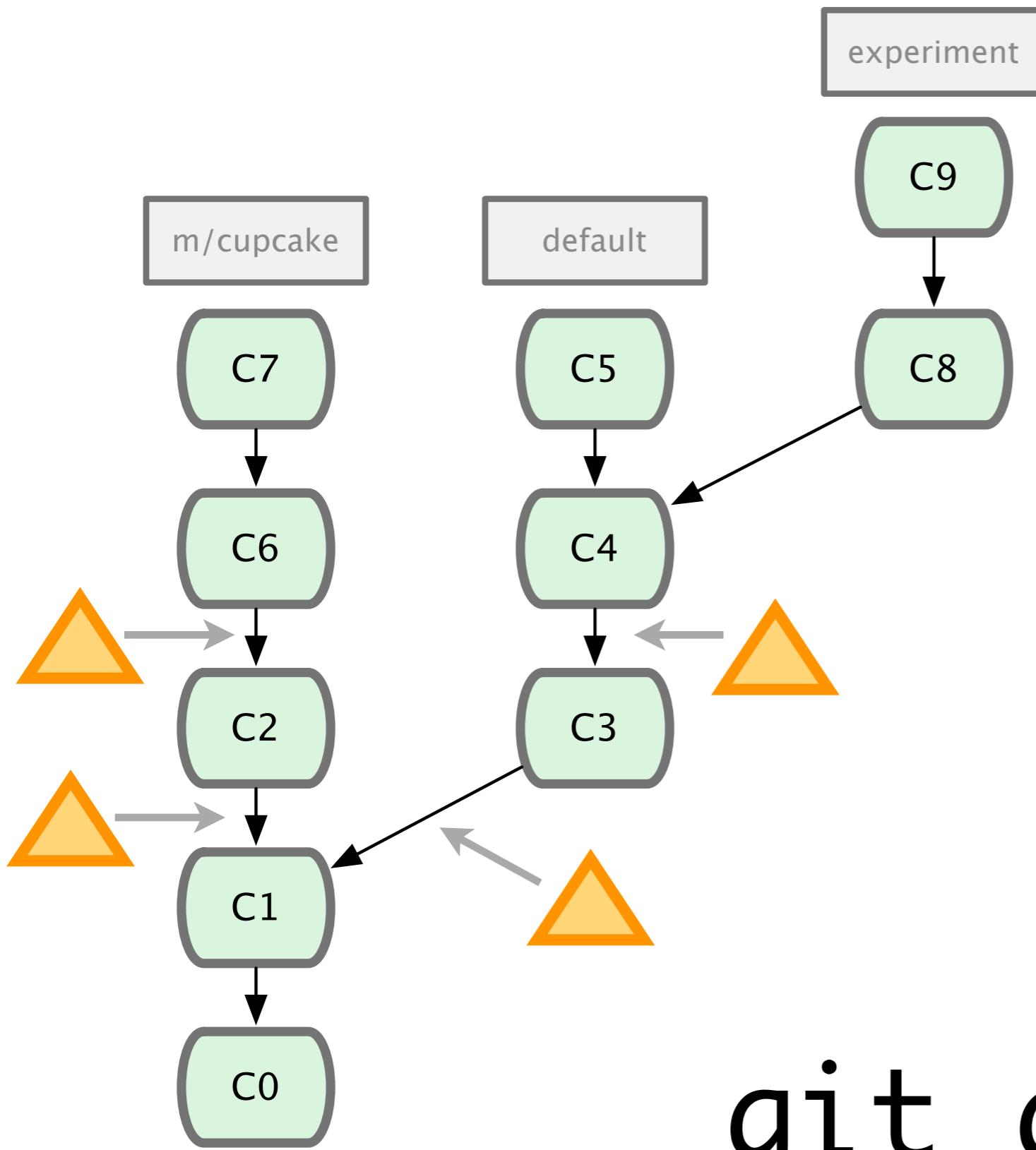
Revision Diffs

```
git diff revA revB
```

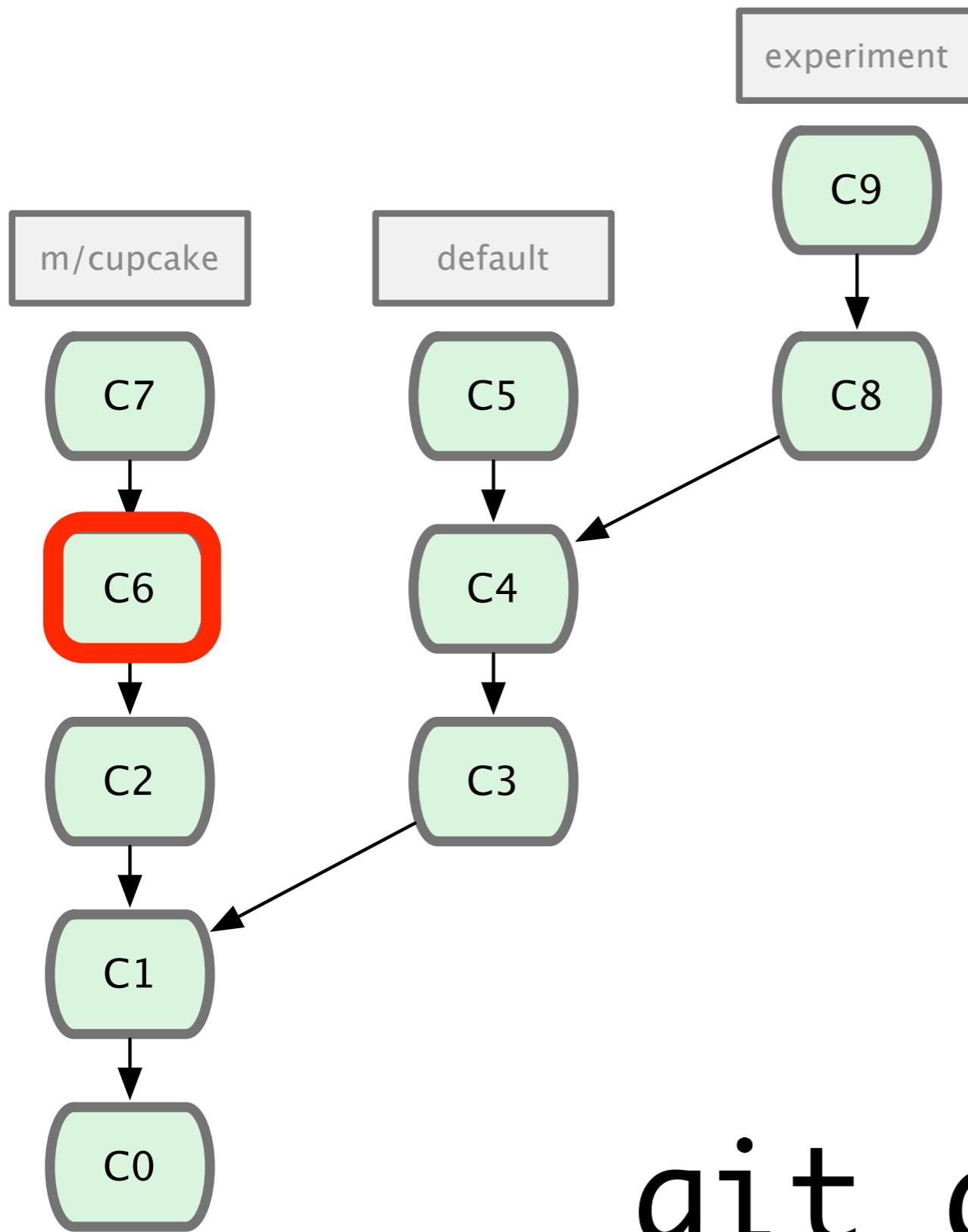




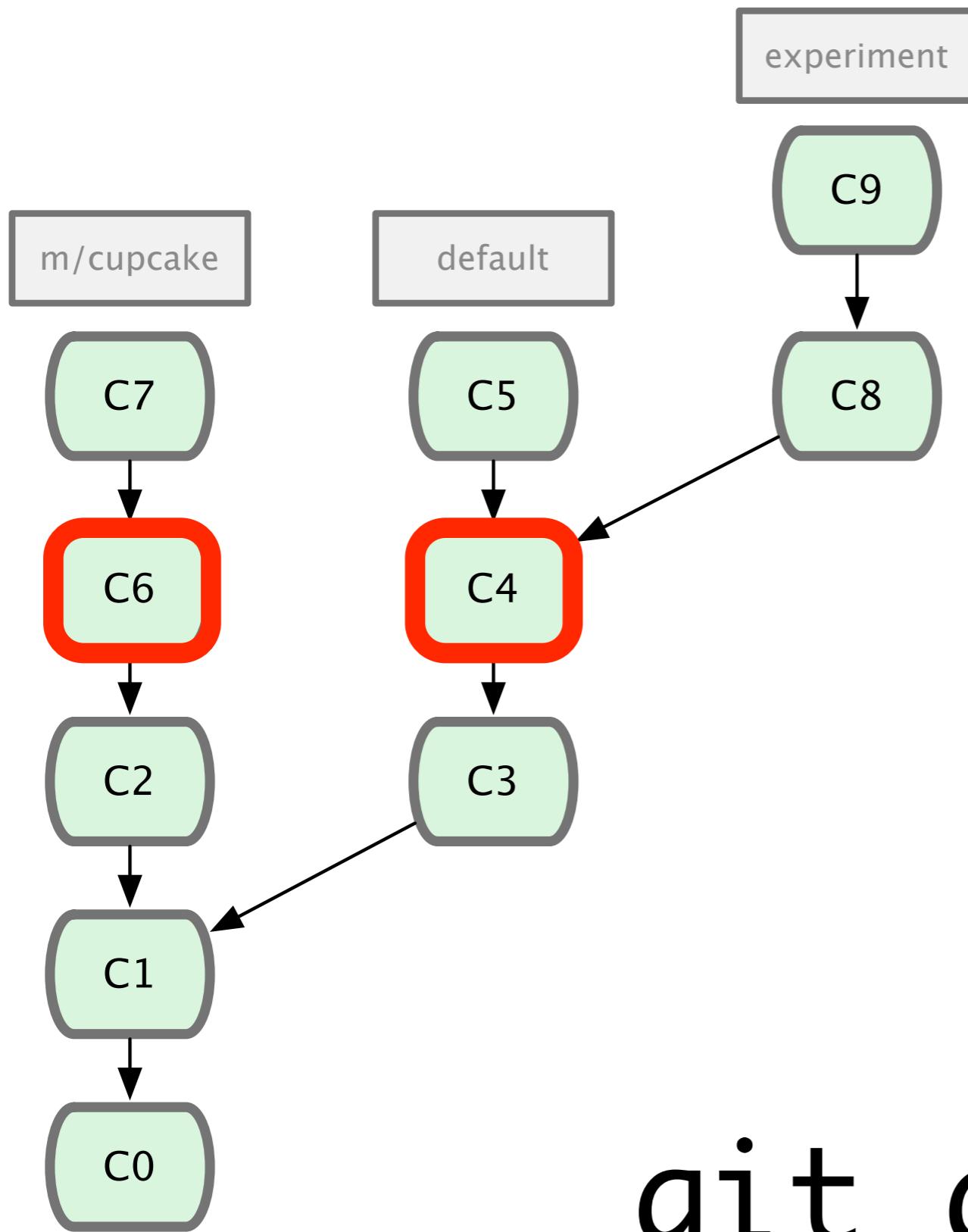
git diff c6 c4



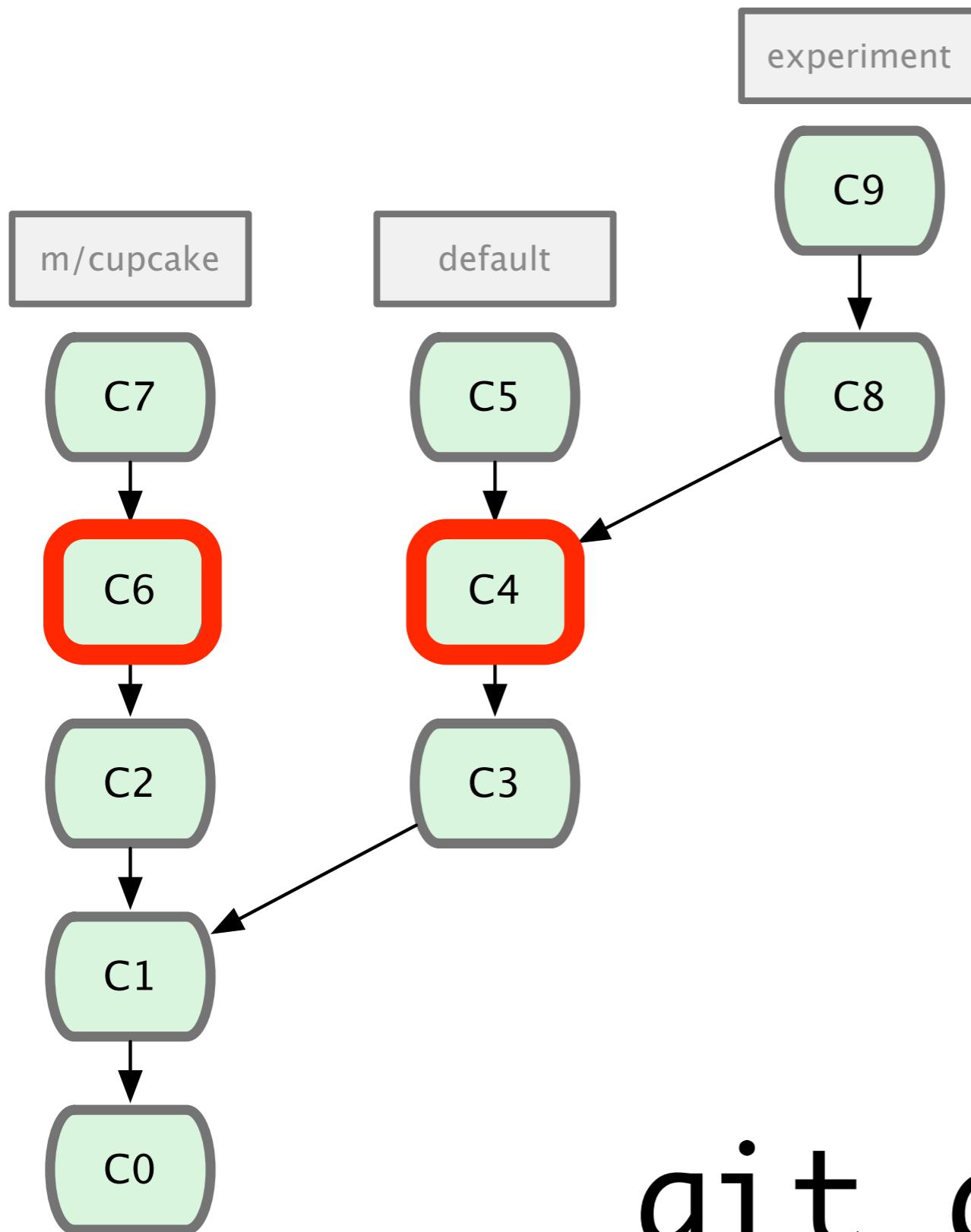
git diff c6 c4



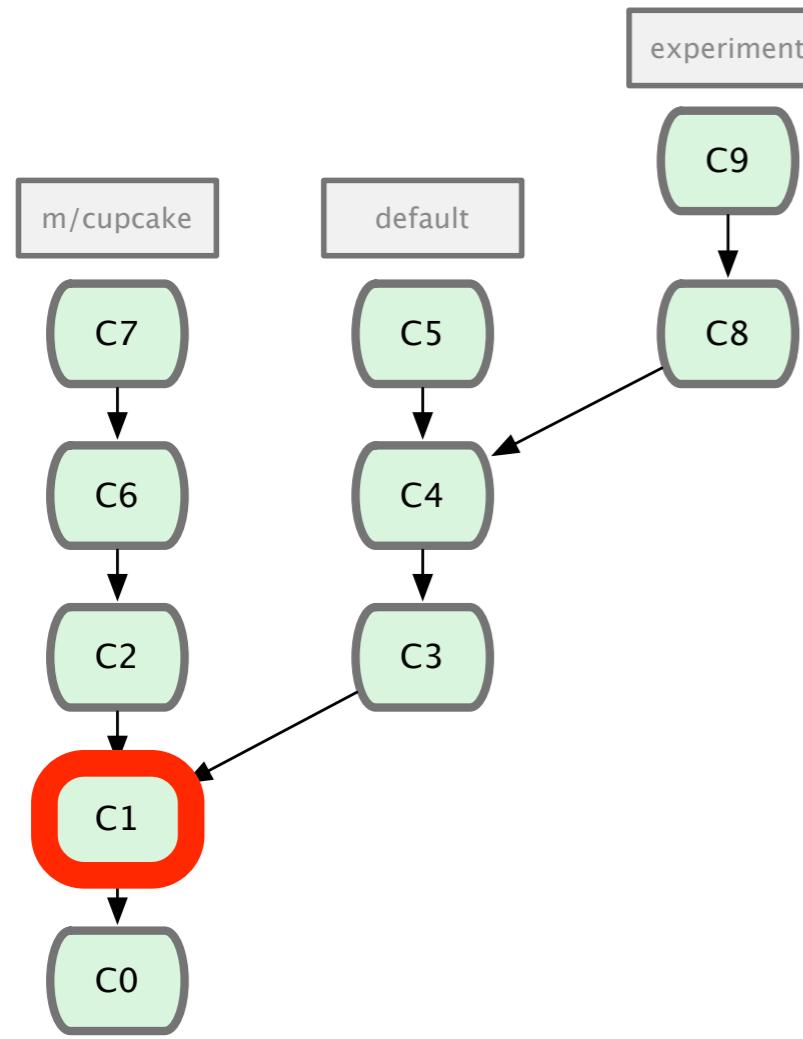
git diff c6 c4



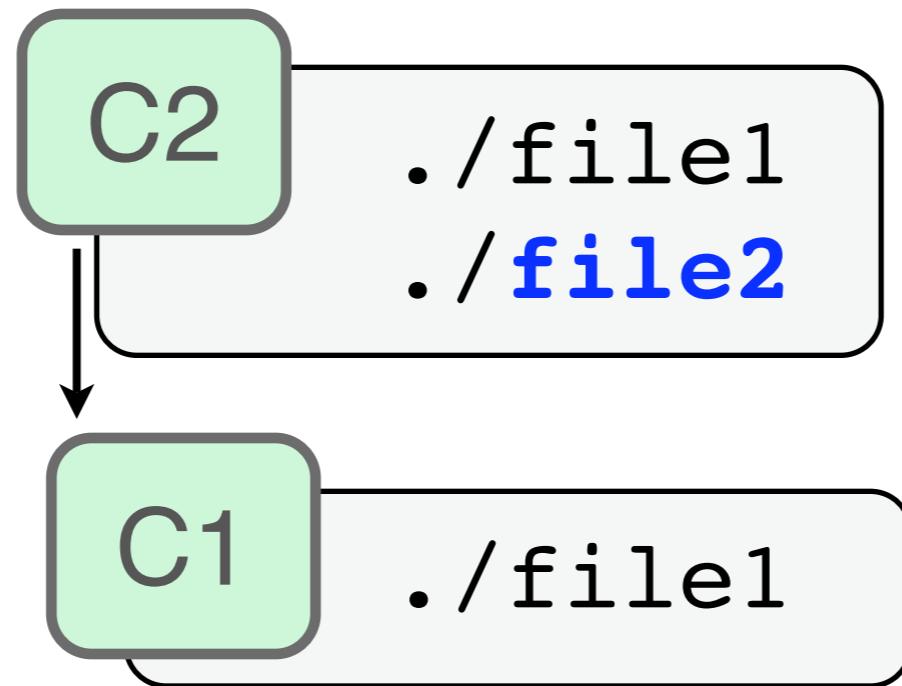
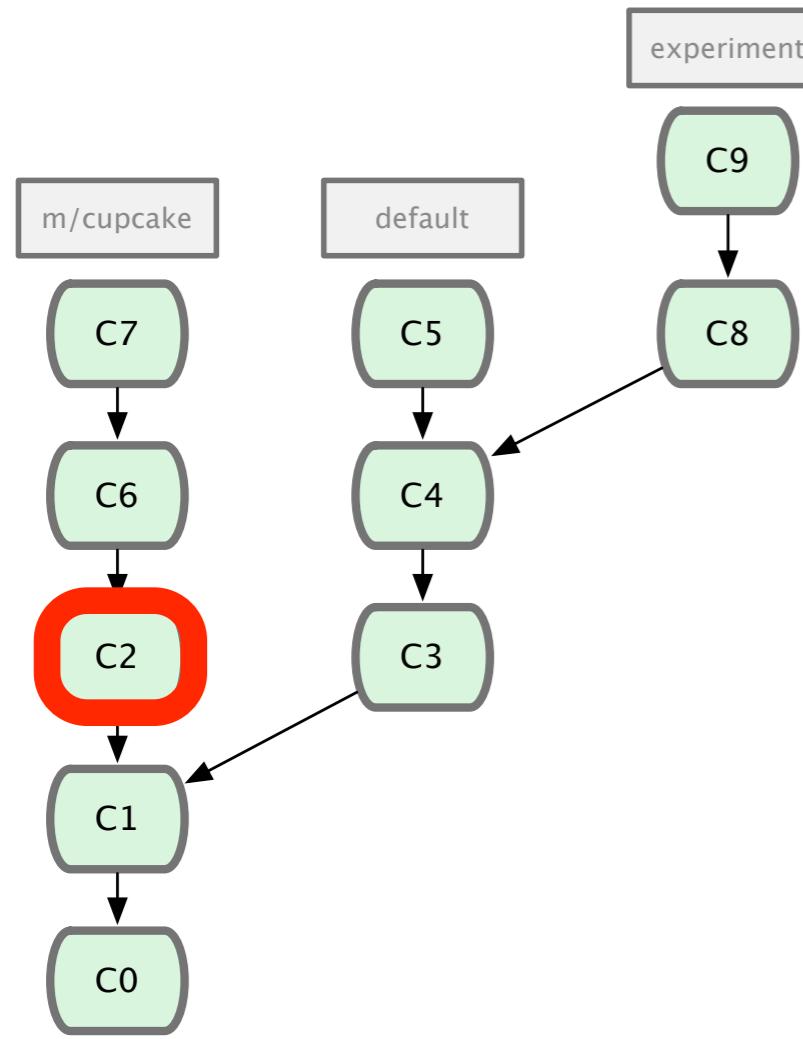
git diff c6 c4

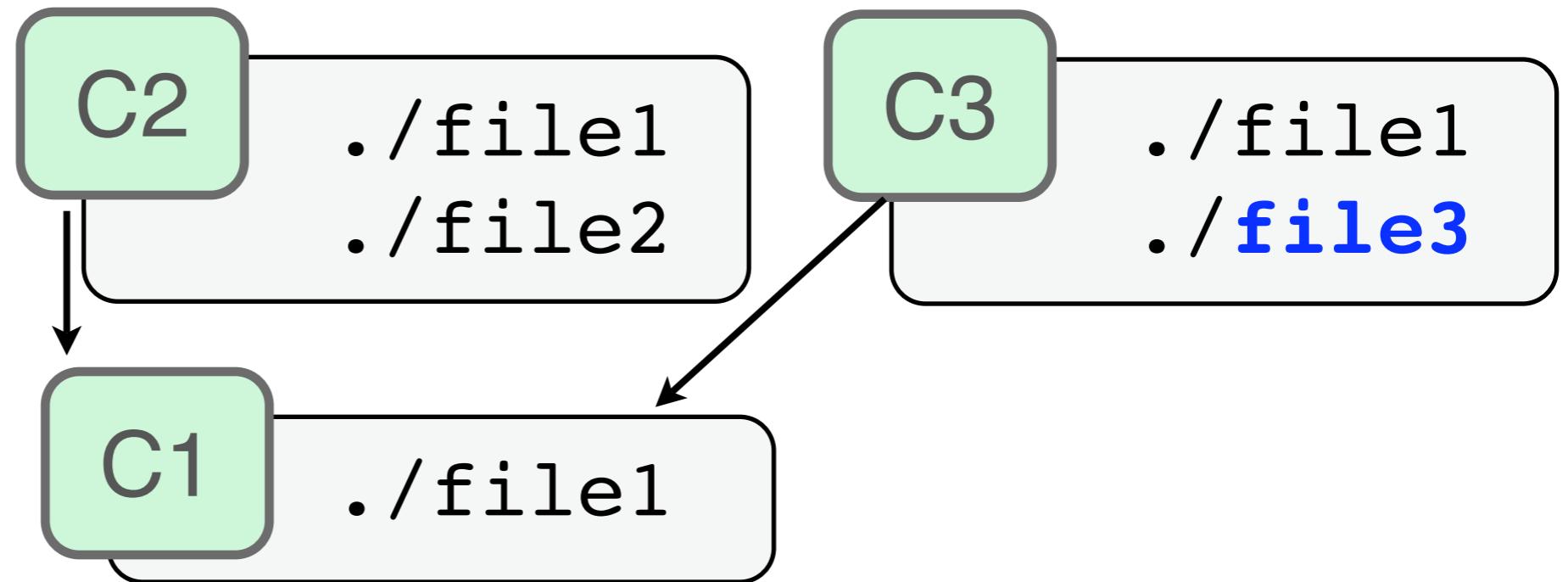
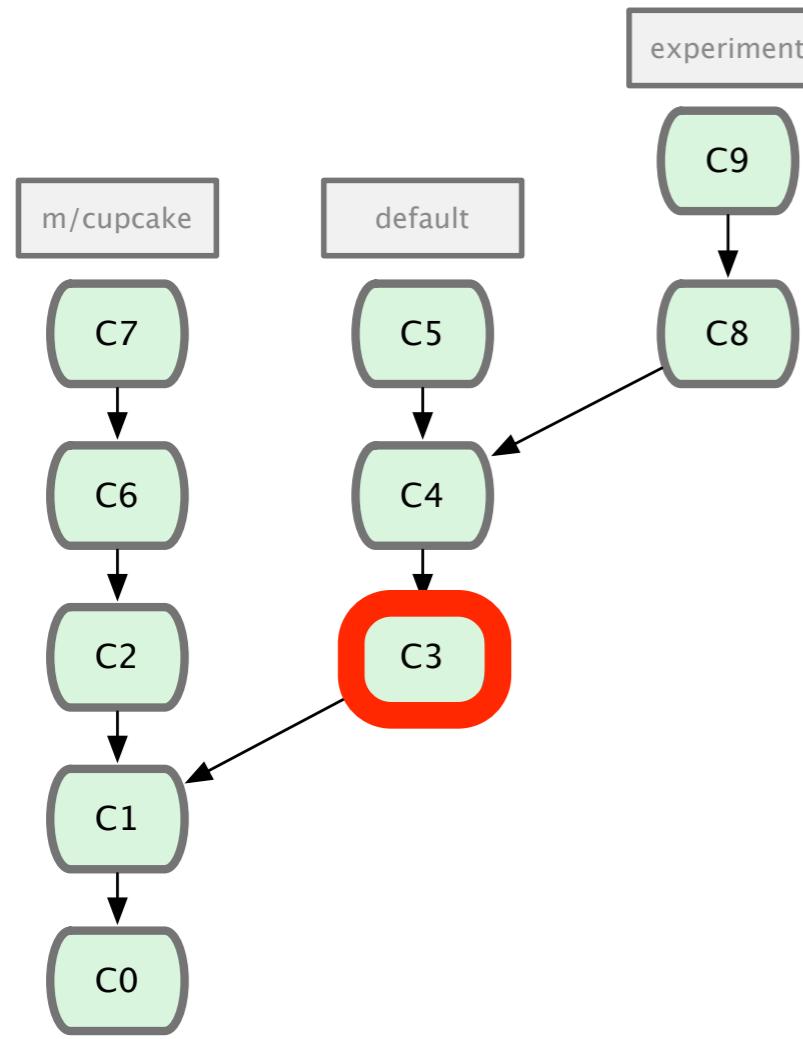


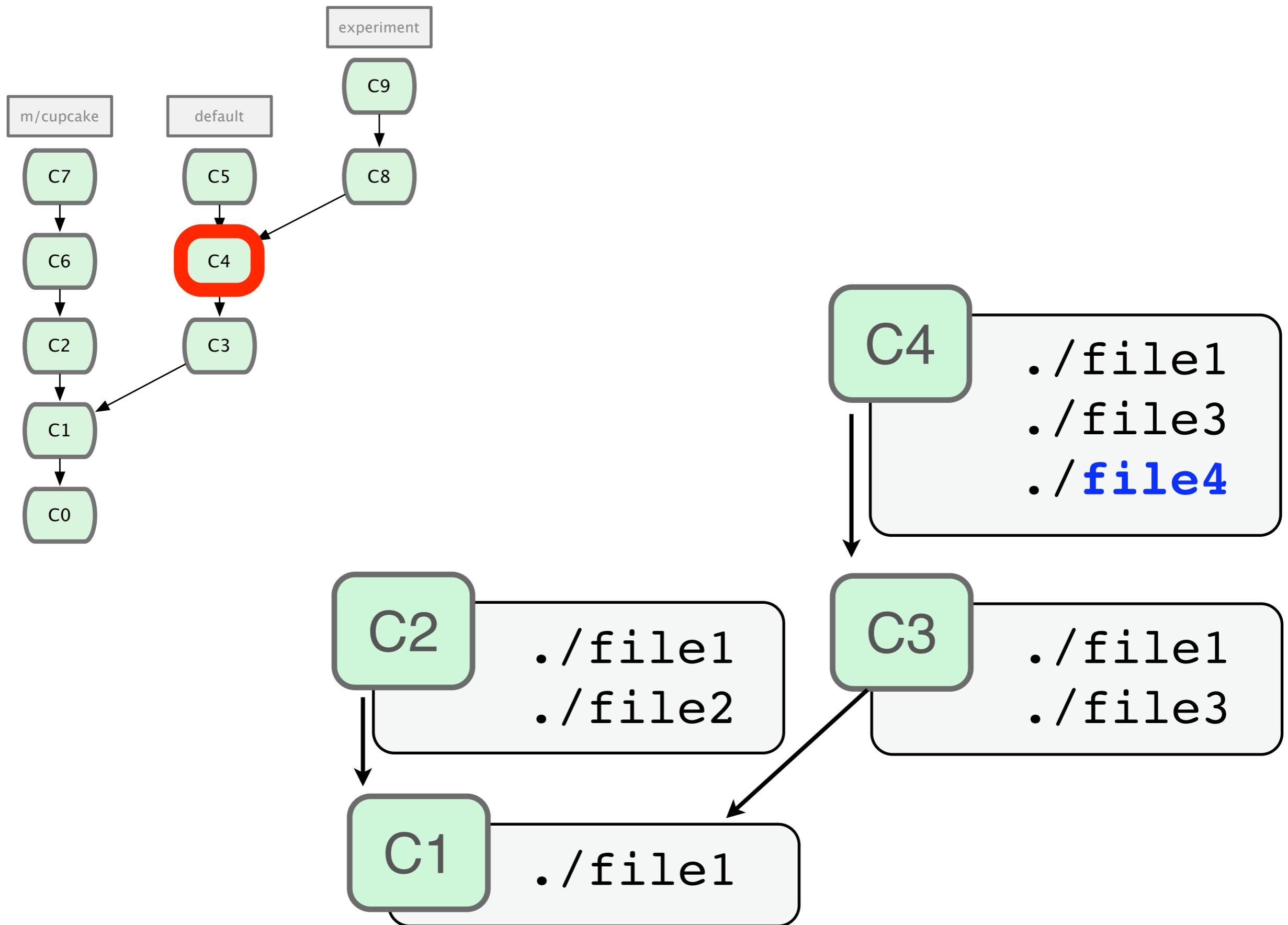
git diff c6 c4
this may not be what you want!

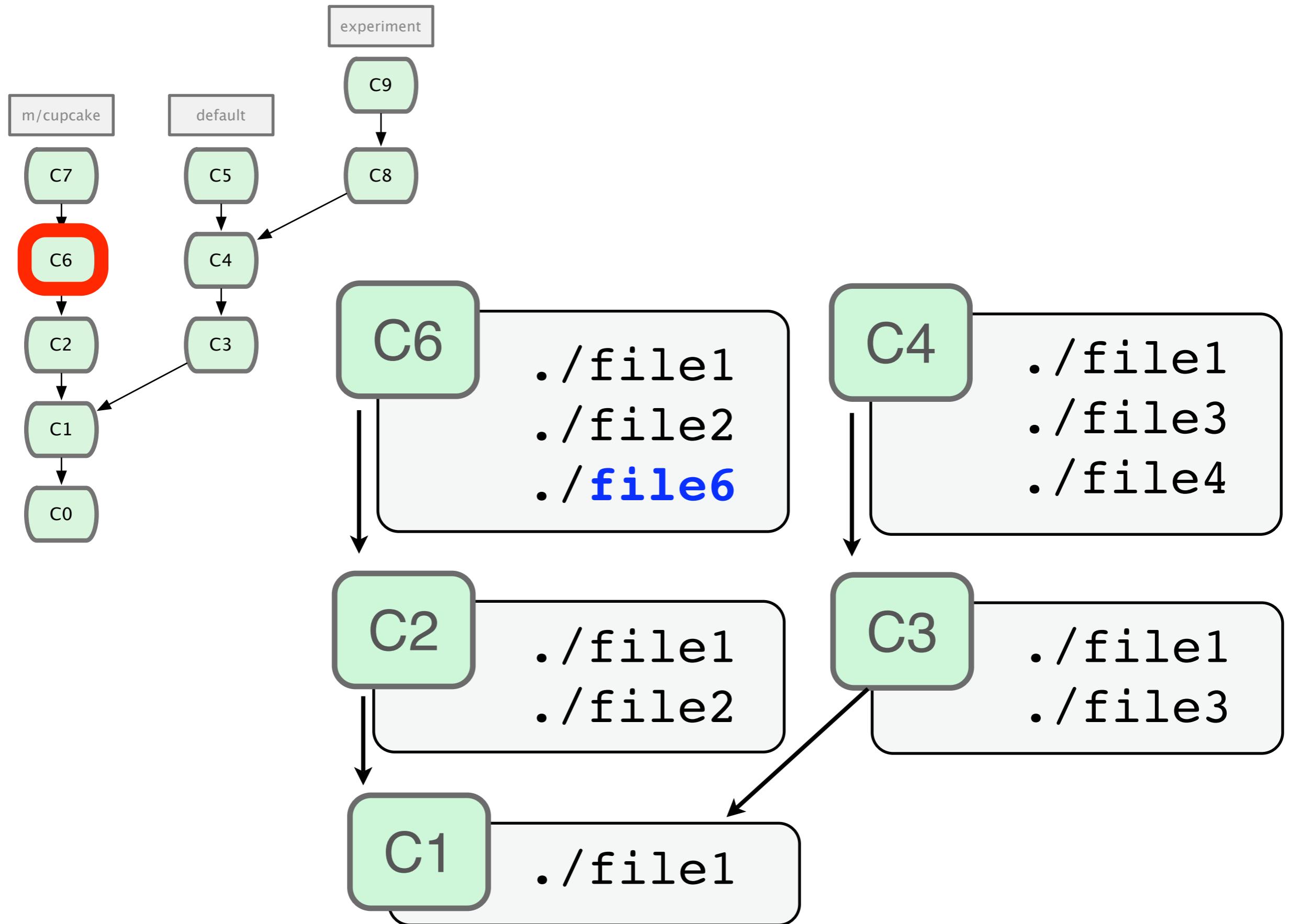


C1 . /file1



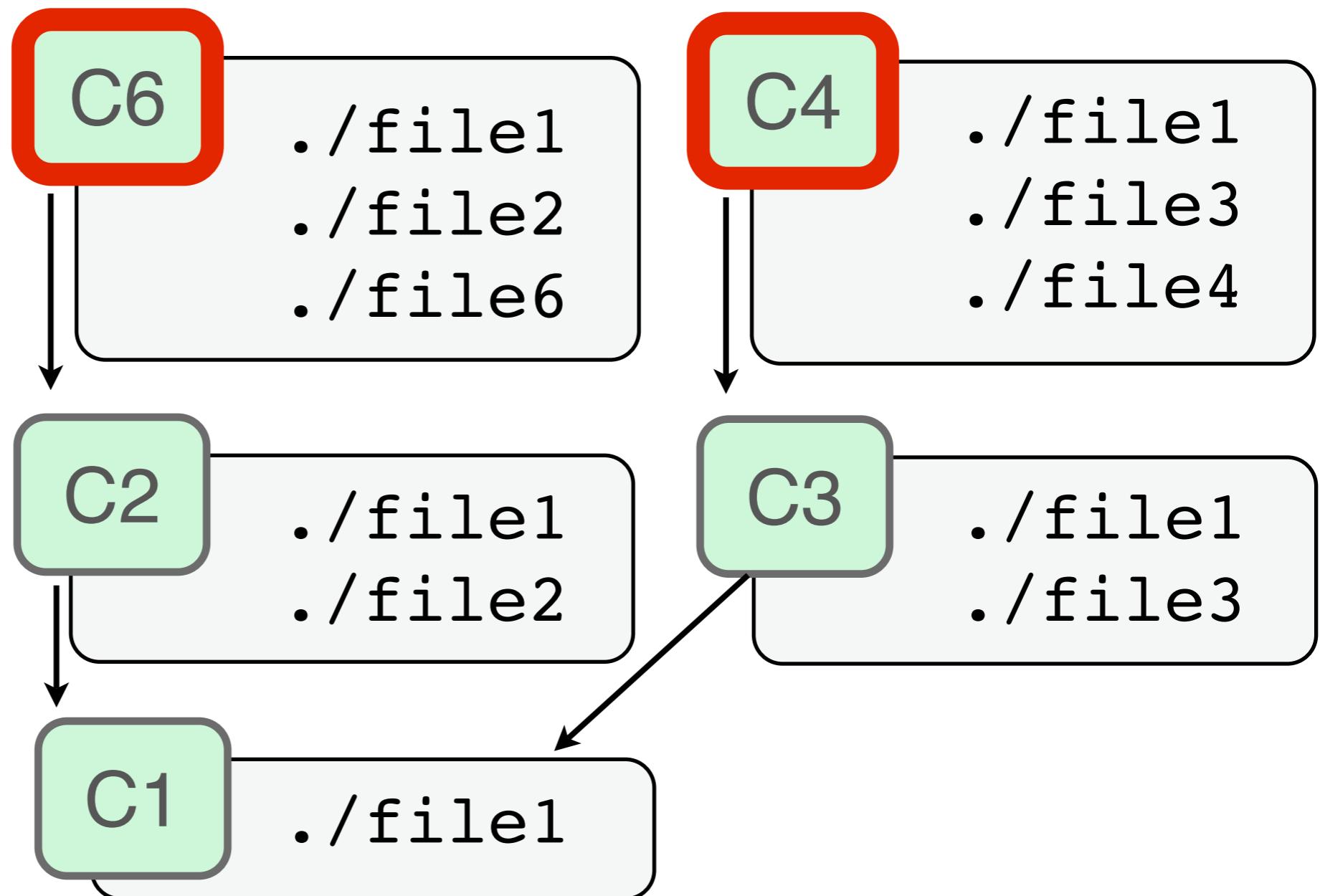


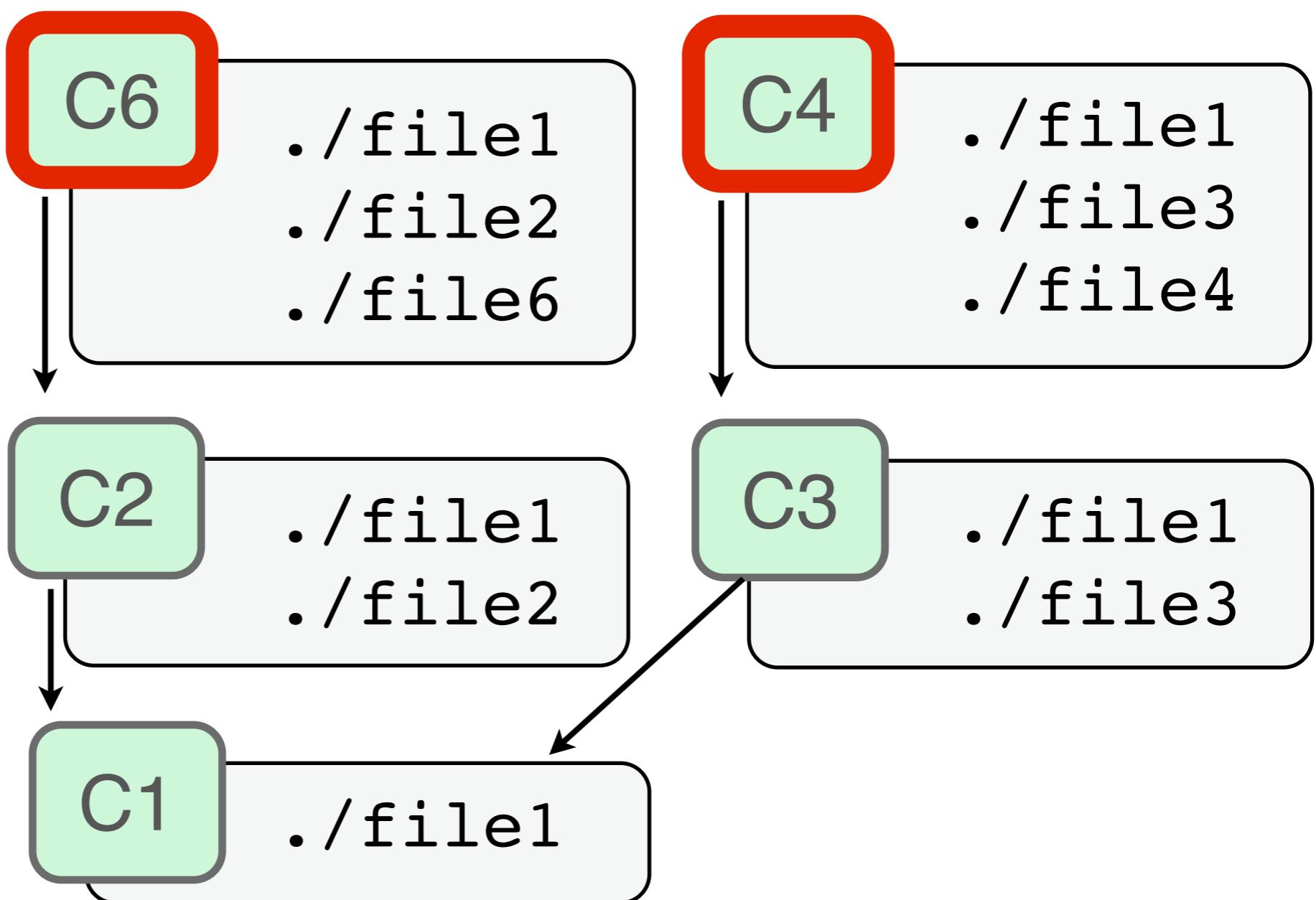
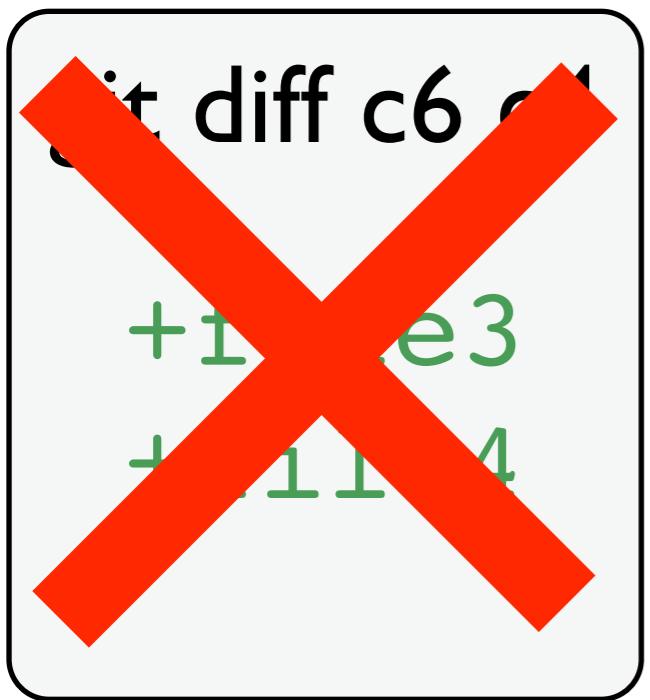




```
git diff c6 c4
```

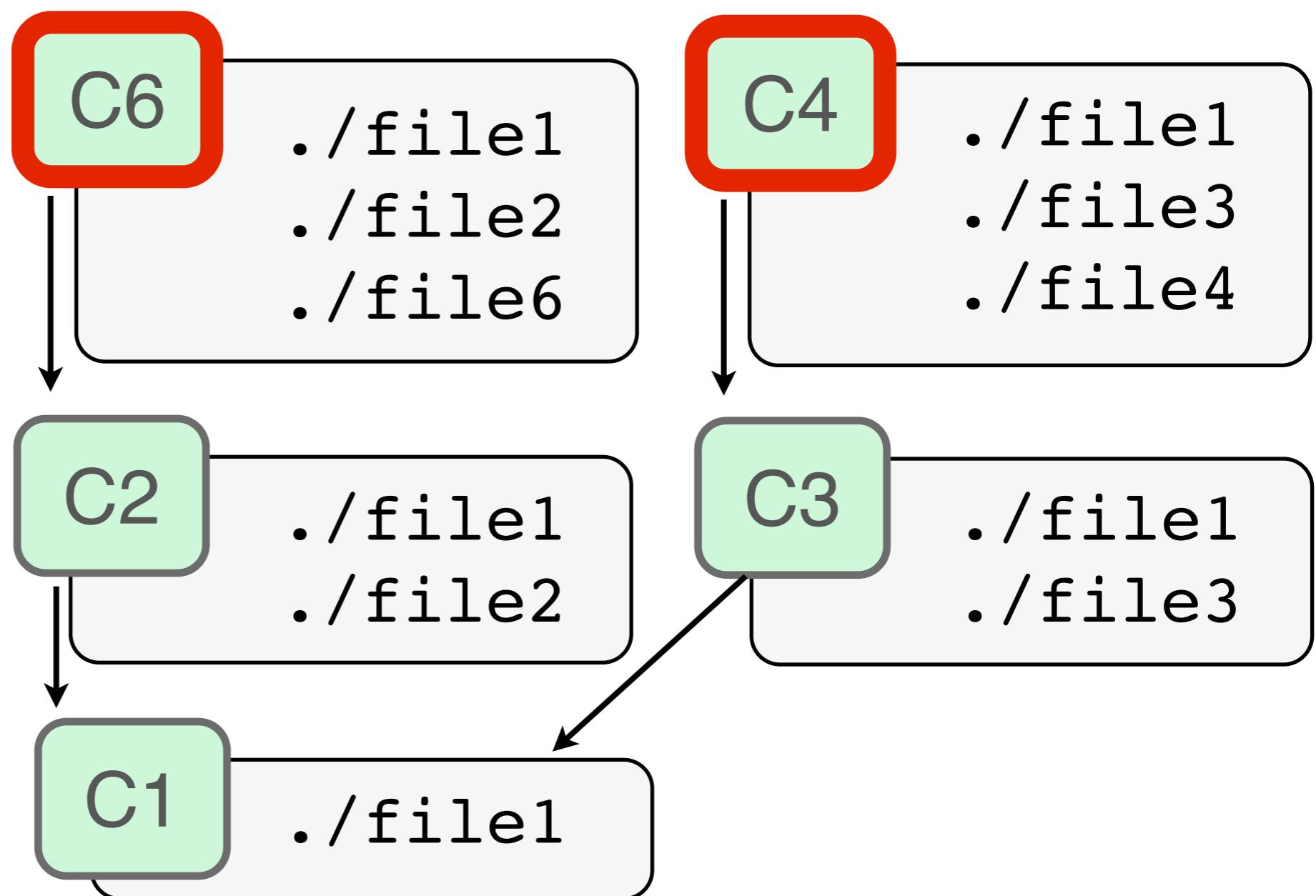
```
+file3  
+file4
```





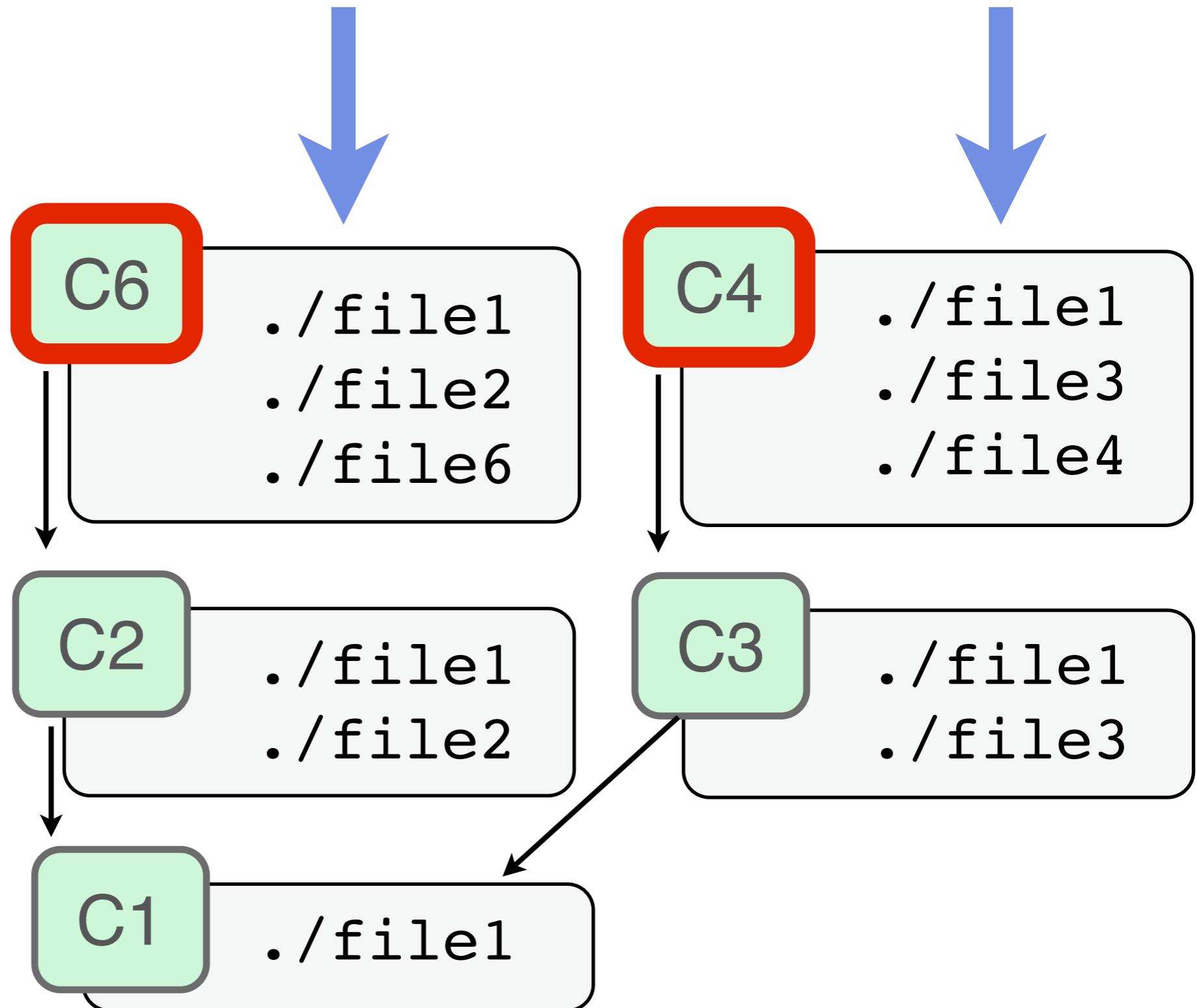
```
git diff c6 c4
```

```
+file3  
+file4  
-file2  
-file6
```



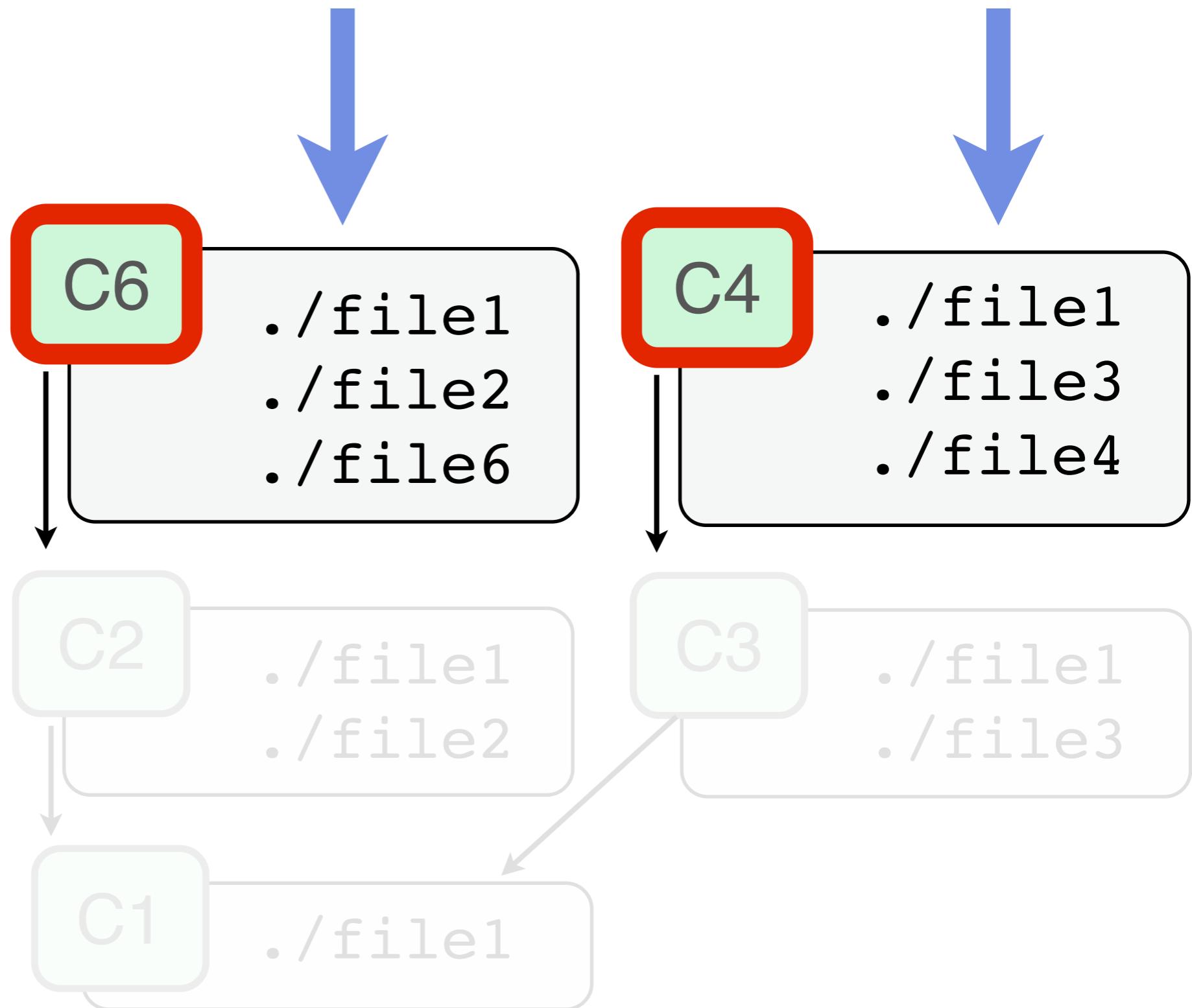
```
git diff c6 c4
```

```
+file3  
+file4  
-file2  
-file6
```



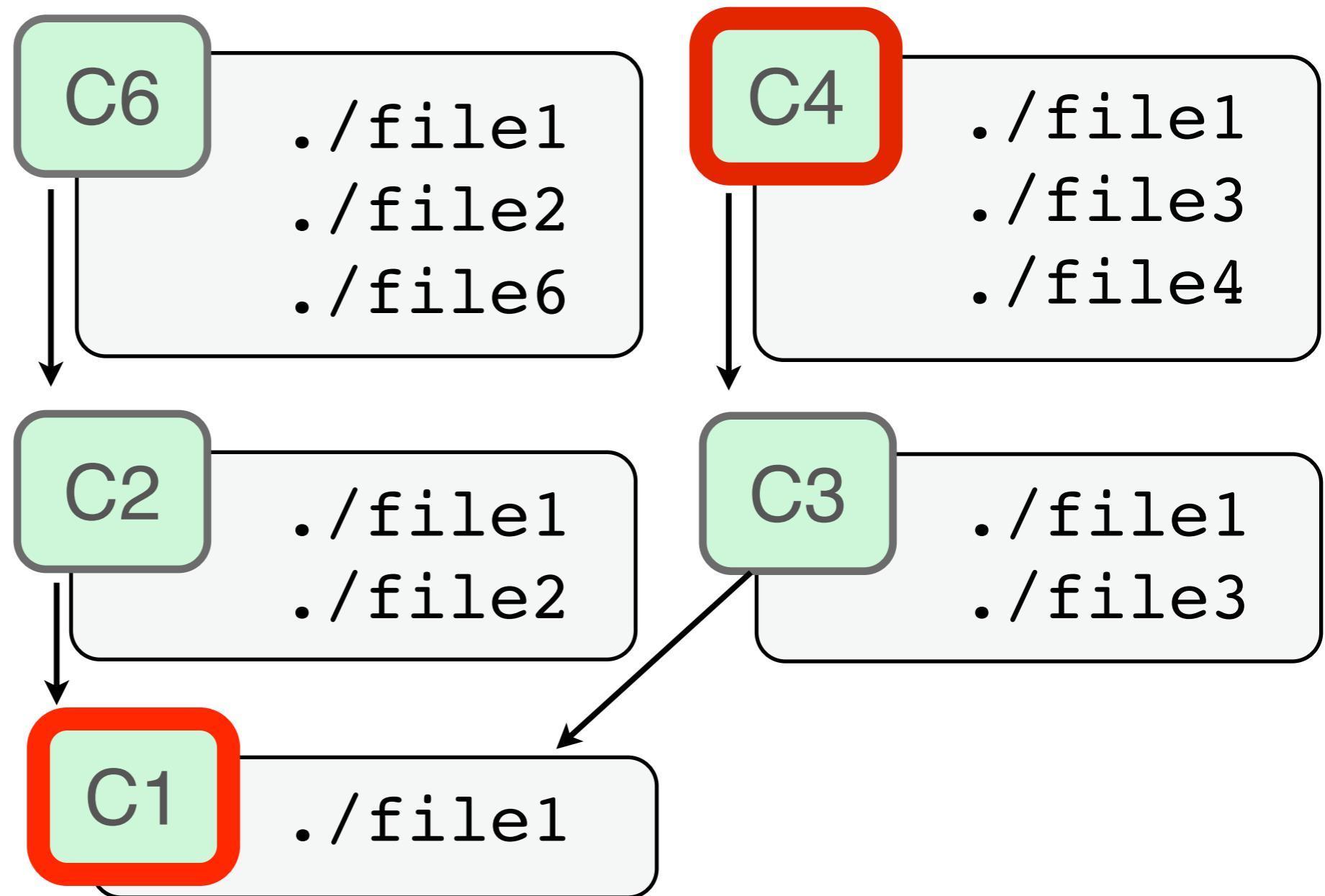
```
git diff c6 c4
```

```
+file3  
+file4  
-file2  
-file6
```



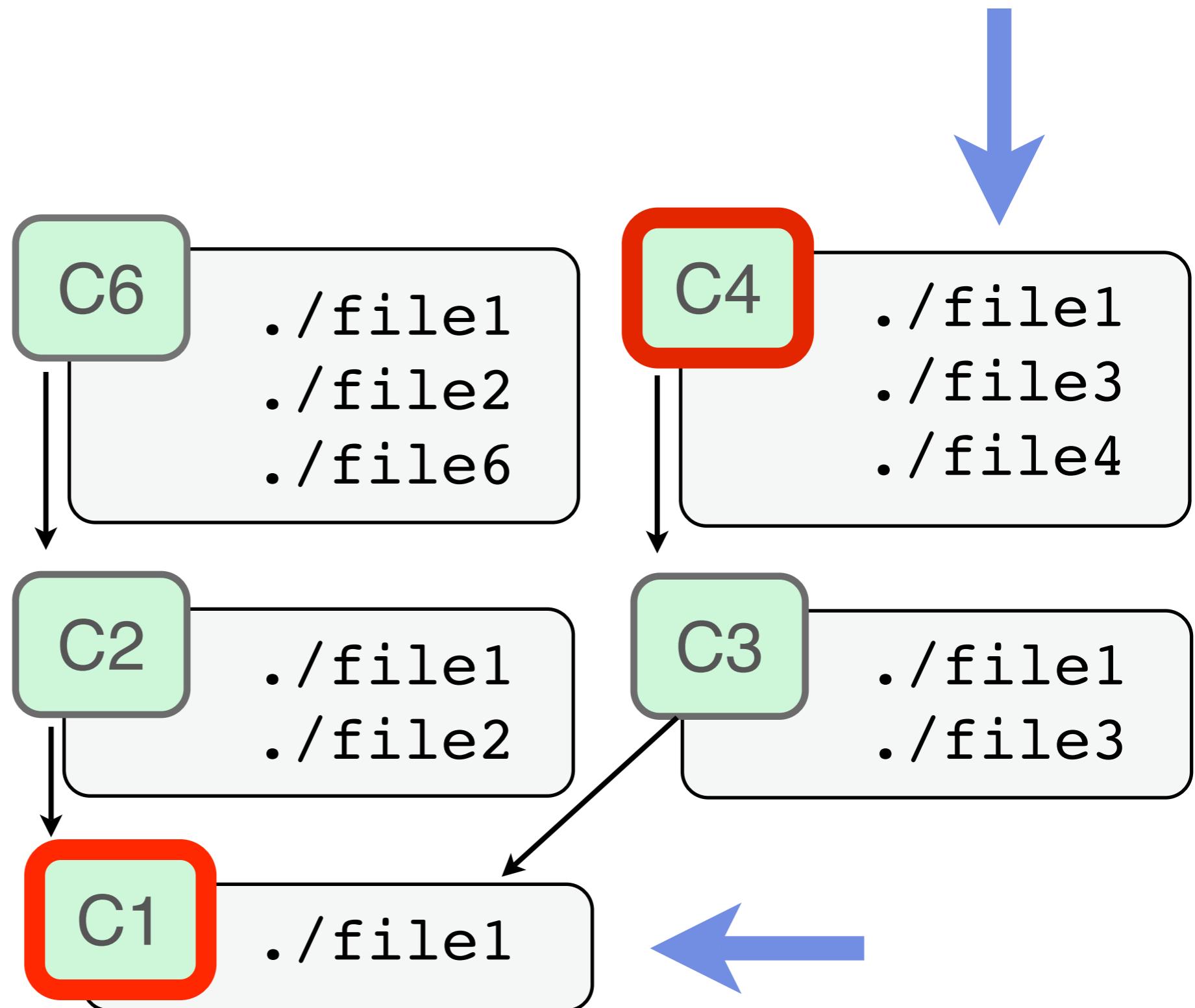
```
git diff c1 c4
```

+file3
+file4



git diff c6...c4

+file3
+file4



Annotation

git blame

git blame

ie: “what dumbass did this? oh, it was me...”

git blame daemon.c

979e32fa (Randal L. Schwartz	2005-10-25 16:29:09 -0700	1) #include "cache.h"
85023577 (Junio C Hamano	2006-12-19 14:34:12 -0800	2) #include "pkt-line.h"
77cb17e9 (Michal Ostrowski	2006-01-10 21:12:17 -0500	3) #include "exec_cmd.h"
49ba83fb (Jon Loeliger	2006-09-19 20:31:51 -0500	4) #include " interpolate.h"
f8ff0c06 (Petr Baudis	2005-09-22 11:25:28 +0200	5)
85023577 (Junio C Hamano	2006-12-19 14:34:12 -0800	6) #include <syslog.h>
85023577 (Junio C Hamano	2006-12-19 14:34:12 -0800	7)
695dffe2 (Johannes Schindelin	2006-09-28 12:00:35 +0200	8) #ifndef HOST_NAME_MAX
695dffe2 (Johannes Schindelin	2006-09-28 12:00:35 +0200	9) #define HOST_NAME_MAX 256
695dffe2 (Johannes Schindelin	2006-09-28 12:00:35 +0200	10) #endif
695dffe2 (Johannes Schindelin	2006-09-28 12:00:35 +0200	11)
415e7b87 (Patrick Welche	2007-10-18 18:17:39 +0100	12) #ifndef NI_MAXSERV
415e7b87 (Patrick Welche	2007-10-18 18:17:39 +0100	13) #define NI_MAXSERV 32
415e7b87 (Patrick Welche	2007-10-18 18:17:39 +0100	14) #endif
415e7b87 (Patrick Welche	2007-10-18 18:17:39 +0100	15)
9048fe1c (Petr Baudis	2005-09-24 16:13:01 +0200	16) static int log_syslog;
f8ff0c06 (Petr Baudis	2005-09-22 11:25:28 +0200	17) static int verbose;
1955fabf (Mark Wooding	2006-02-03 20:27:04 +0000	18) static int reuseaddr;
f8ff0c06 (Petr Baudis	2005-09-22 11:25:28 +0200	19)
960deccb (H. Peter Anvin	2005-10-19 14:27:01 -0700	20) static const char daemon_usage[] =
1b1dd23f (Stephan Beyer	2008-07-13 15:36:15 +0200	21) "git daemon [--verbose] [--syslog] [
3bd62c21 (Stephen R. van den Berg	2008-08-14 20:02:20 +0200	22) " --timeout=n] [--init-ti
3bd62c21 (Stephen R. van den Berg	2008-08-14 20:02:20 +0200	23) " --strict-paths] [--base
73a7a656 (Jens Axboe	2007-07-27 14:00:29 -0700	24) " --user-path --user-pa
49ba83fb (Jon Loeliger	2006-09-19 20:31:51 -0500	25) " --interpolated-path=pat
678dac6b (Tilman Sauerbeck	2006-08-22 19:37:41 +0200	26) " --reuseaddr] [--detach]
d9edcbd6 (Junio C Hamano	2006-09-07 01:40:04 -0700	27) " --[enable disable allow
dd467629 (Jon Loeliger	2006-09-26 09:47:43 -0500	28) " --inetd [--listen=hos
dd467629 (Jon Loeliger	2006-09-26 09:47:43 -0500	29) " --user=user
dd467629 (Jon Loeliger	2006-09-26 09:47:43 -0500	30) " [directory...]";
4ae95682 (H. Peter Anvin	2005-09-26 19:10:55 -0700	31)
4ae95682 (H. Peter Anvin	2005-09-26 19:10:55 -0700	32) /* List of acceptable pathname prefi
96f1e58f (David Rienties	2006-08-15 10:23:48 -0700	33) static char **ok_paths:

git blame daemon.c

979e32fa	(Randal L. Schwartz	2005-10-25 16:29:09 -0700	1) #include "cache.h"
85023577	(Junio C Hamano	2006-12-19 14:34:12 -0800	2) #include "pkt-line.h"
77cb17e9	(Michal Ostrowski	2006-01-10 21:12:17 -0500	3) #include "exec_cmd.h"
49ba83fb	(Jon Loeliger	2006-09-19 20:31:51 -0500	4) #include " interpolate.h"
f8ff0c06	(Petr Baudis	2005-09-22 11:25:28 +0200	5)
85023577	(Junio C Hamano	2006-12-19 14:34:12 -0800	6) #include <syslog.h>
85023577	(Junio C Hamano	2006-12-19 14:34:12 -0800	7)
695dffe2	(Johannes Schindelin	2006-09-28 12:00:35 +0200	8) #ifndef HOST_NAME_MAX
695dffe2	(Johannes Schindelin	2006-09-28 12:00:35 +0200	9) #define HOST_NAME_MAX 256
695dffe2	(Johannes Schindelin	2006-09-28 12:00:35 +0200	10) #endif
695dffe2	(Johannes Schindelin	2006-09-28 12:00:35 +0200	11)
415e7b87	(Patrick Welche	2007-10-18 18:17:39 +0100	12) #ifndef NI_MAXSERV
415e7b87	(Patrick Welche	2007-10-18 18:17:39 +0100	13) #define NI_MAXSERV 32
415e7b87	(Patrick Welche	2007-10-18 18:17:39 +0100	14) #endif
415e7b87	(Patrick Welche	2007-10-18 18:17:39 +0100	15)
9048fe1c	(Petr Baudis	2005-09-24 16:13:01 +0200	16) static int log_syslog;
f8ff0c06	(Petr Baudis	2005-09-22 11:25:28 +0200	17) static int verbose;
1955fabf	(Mark Wooding	2006-02-03 20:27:04 +0000	18) static int reuseaddr;
f8ff0c06	(Petr Baudis	2005-09-22 11:25:28 +0200	19)
960deccb	(H. Peter Anvin	2005-10-19 14:27:01 -0700	20) static const char daemon_usage[] =
1b1dd23f	(Stephan Beyer	2008-07-13 15:36:15 +0200	21) "git daemon [--verbose] [--syslog] [
3bd62c21	(Stephen R. van den Berg	2008-08-14 20:02:20 +0200	22) " --timeout=n] [--init-ti
3bd62c21	(Stephen R. van den Berg	2008-08-14 20:02:20 +0200	23) " --strict-paths] [--base
73a7a656	(Jens Axboe	2007-07-27 14:00:29 -0700	24) " --user-path --user-pa
49ba83fb	(Jon Loeliger	2006-09-19 20:31:51 -0500	25) " --interpolated-path=pat
678dac6b	(Tilman Sauerbeck	2006-08-22 19:37:41 +0200	26) " --reuseaddr] [--detach]
d9edcbd6	(Junio C Hamano	2006-09-07 01:40:04 -0700	27) " --[enable disable allow
dd467629	(Jon Loeliger	2006-09-26 09:47:43 -0500	28) " --inetd [--listen=hos
dd467629	(Jon Loeliger	2006-09-26 09:47:43 -0500	29) " [--user=user
dd467629	(Jon Loeliger	2006-09-26 09:47:43 -0500	30) " [directory...]";
4ae95682	(H. Peter Anvin	2005-09-26 19:10:55 -0700	31)
4ae95682	(H. Peter Anvin	2005-09-26 19:10:55 -0700	32) /* List of acceptable pathname prefi
96f1e58f	(David Rientjes	2006-08-15 10:23:48 -0700	33) static char **ok_paths:

git blame daemon.c

979e32fa	(Randal L. Schwartz	2005-10-25 16:29:09 -0700	1) #include "cache.h"
85023577	(Junio C Hamano	2006-12-19 14:34:12 -0800	2) #include "pkt-line.h"
77cb17e9	(Michal Ostrowski	2006-01-10 21:12:17 -0500	3) #include "exec_cmd.h"
49ba83fb	(Jon Loeliger	2006-09-19 20:31:51 -0500	4) #include " interpolate.h"
f8ff0c06	(Petr Baudis	2005-09-22 11:25:28 +0200	5)
85023577	(Junio C Hamano	2006-12-19 14:34:12 -0800	6) #include <syslog.h>
85023577	(Junio C Hamano	2006-12-19 14:34:12 -0800	7)
695dffe2	(Johannes Schindelin	2006-09-28 12:00:35 +0200	8) #ifndef HOST_NAME_MAX
695dffe2	(Johannes Schindelin	2006-09-28 12:00:35 +0200	9) #define HOST_NAME_MAX 256
695dffe2	(Johannes Schindelin	2006-09-28 12:00:35 +0200	10) #endif
695dffe2	(Johannes Schindelin	2006-09-28 12:00:35 +0200	11)
415e7b87	(Patrick Welche	2007-10-18 18:17:39 +0100	12) #ifndef NI_MAXSERV
415e7b87	(Patrick Welche	2007-10-18 18:17:39 +0100	13) #define NI_MAXSERV 32
415e7b87	(Patrick Welche	2007-10-18 18:17:39 +0100	14) #endif
415e7b87	(Patrick Welche	2007-10-18 18:17:39 +0100	15)
9048fe1c	(Petr Baudis	2005-09-24 16:13:01 +0200	16) static int log_syslog;
f8ff0c06	(Petr Baudis	2005-09-22 11:25:28 +0200	17) static int verbose;
1955fabf	(Mark Wooding	2006-02-03 20:27:04 +0000	18) static int reuseaddr;
f8ff0c06	(Petr Baudis	2005-09-22 11:25:28 +0200	19)
960deccb	(H. Peter Anvin	2005-10-19 14:27:01 -0700	20) static const char daemon_usage[] =
1b1dd23f	(Stephan Beyer	2008-07-13 15:36:15 +0200	21) "git daemon [--verbose] [--syslog] [
3bd62c21	(Stephen R. van den Berg	2008-08-14 20:02:20 +0200	22) " --timeout=n] [--init-ti
3bd62c21	(Stephen R. van den Berg	2008-08-14 20:02:20 +0200	23) " --strict-paths] [--base
73a7a656	(Jens Axboe	2007-07-27 14:00:29 -0700	24) " --user-path --user-pa
49ba83fb	(Jon Loeliger	2006-09-19 20:31:51 -0500	25) " --interpolated-path=pat
678dac6b	(Tilman Sauerbeck	2006-08-22 19:37:41 +0200	26) " --reuseaddr] [--detach]
d9edcbd6	(Junio C Hamano	2006-09-07 01:40:04 -0700	27) " --[enable disable allow
dd467629	(Jon Loeliger	2006-09-26 09:47:43 -0500	28) " --inetd [--listen=hos
dd467629	(Jon Loeliger	2006-09-26 09:47:43 -0500	29) " --user=user
dd467629	(Jon Loeliger	2006-09-26 09:47:43 -0500	30) " [directory...]";
4ae95682	(H. Peter Anvin	2005-09-26 19:10:55 -0700	31)
4ae95682	(H. Peter Anvin	2005-09-26 19:10:55 -0700	32) /* List of acceptable pathname prefi
96f1e58f	(David Rientjes	2006-08-15 10:23:48 -0700	33) static char **ok_paths:

git blame daemon.c

```
979e32fa (Randal L. Schwartz
85023577 (Junio C Hamano
77cb17e9 (Michal Ostrowski
49ba83fb (Jon Loeliger
f8ff0c06 (Petr Baudis
85023577 (Junio C Hamano
85023577 (Junio C Hamano
695dffe2 (Johannes Schindelin
695dffe2 (Johannes Schindelin
695dffe2 (Johannes Schindelin
695dffe2 (Johannes Schindelin
415e7b87 (Patrick Welche
415e7b87 (Patrick Welche
415e7b87 (Patrick Welche
415e7b87 (Patrick Welche
9048fe1c (Petr Baudis
f8ff0c06 (Petr Baudis
1955fabf (Mark Wooding
f8ff0c06 (Petr Baudis
960deccb (H. Peter Anvin
1b1dd23f (Stephan Beyer
3bd62c21 (Stephen R. van den Berg
3bd62c21 (Stephen R. van den Berg
73a7a656 (Jens Axboe
49ba83fb (Jon Loeliger
678dac6b (Tilman Sauerbeck
d9edcbd6 (Junio C Hamano
dd467629 (Jon Loeliger
dd467629 (Jon Loeliger
dd467629 (Jon Loeliger
4ae95682 (H. Peter Anvin
4ae95682 (H. Peter Anvin
96f1e58f (David Rientjes
```

```
2005-10-25 16:29:09 -0700
2006-12-19 14:34:12 -0800
2006-01-10 21:12:17 -0500
2006-09-19 20:31:51 -0500
2005-09-22 11:25:28 +0200
2006-12-19 14:34:12 -0800
2006-12-19 14:34:12 -0800
2006-09-28 12:00:35 +0200
2006-09-28 12:00:35 +0200
2006-09-28 12:00:35 +0200
2006-09-28 12:00:35 +0200
2007-10-18 18:17:39 +0100
2007-10-18 18:17:39 +0100
2007-10-18 18:17:39 +0100
2007-10-18 18:17:39 +0100
2005-09-24 16:13:01 +0200
2005-09-22 11:25:28 +0200
2006-02-03 20:27:04 +0000
2005-09-22 11:25:28 +0200
2005-10-19 14:27:01 -0700
2008-07-13 15:36:15 +0200
2008-08-14 20:02:20 +0200
2008-08-14 20:02:20 +0200
2007-07-27 14:00:29 -0700
2006-09-19 20:31:51 -0500
2006-08-22 19:37:41 +0200
2006-09-07 01:40:04 -0700
2006-09-26 09:47:43 -0500
2006-09-26 09:47:43 -0500
2006-09-26 09:47:43 -0500
2005-09-26 19:10:55 -0700
2005-09-26 19:10:55 -0700
2006-08-15 10:23:48 -0700
```

```
1) #include "cache.h"
2) #include "pkt-line.h"
3) #include "exec_cmd.h"
4) #include " interpolate.h"
5)
6) #include <syslog.h>
7)
8) #ifndef HOST_NAME_MAX
9) #define HOST_NAME_MAX 256
10) #endif
11)
12) #ifndef NI_MAXSERV
13) #define NI_MAXSERV 32
14) #endif
15)
16) static int log_syslog;
17) static int verbose;
18) static int reuseaddr;
19)
20) static const char daemon_usage[] =
21) "git daemon [--verbose] [--syslog] [
22) " --timeout=n ] [--init-ti
23) " --strict-paths] [--base
24) " --user-path | --user-pa
25) " --interpolated-path=pat
26) " --reuseaddr] [--detach]
27) " [--enable|disable|allow
28) " --inetd | [--listen=hos
29) " --user=user
30) " [directory...]";
31)
32) /* List of acceptable pathname prefi
33) static char **ok_paths;
```

git blame daemon.c

```
979e32fa (Randal L. Schwartz 2005-10-25 16:29:09 -0700
85023577 (Junio C Hamano 2006-12-19 14:34:12 -0800
77cb17e9 (Michal Ostrowski 2006-01-10 21:12:17 -0500
49ba83fb (Jon Loeliger 2006-09-19 20:31:51 -0500
f8ff0c06 (Petr Baudis 2005-09-22 11:25:28 +0200
85023577 (Junio C Hamano 2006-12-19 14:34:12 -0800
85023577 (Junio C Hamano 2006-12-19 14:34:12 -0800
695dffe2 (Johannes Schindelin 2006-09-28 12:00:35 +0200
415e7b87 (Patrick Welche 2007-10-18 18:17:39 +0100
9048fe1c (Petr Baudis 2005-09-24 16:13:01 +0200
f8ff0c06 (Petr Baudis 2005-09-22 11:25:28 +0200
1955fabf (Mark Wooding 2006-02-03 20:27:04 +0000
f8ff0c06 (Petr Baudis 2005-09-22 11:25:28 +0200
960deccb (H. Peter Anvin 2005-10-19 14:27:01 -0700
1b1dd23f (Stephan Beyer 2008-07-13 15:36:15 +0200
3bd62c21 (Stephen R. van den Berg 2008-08-14 20:02:20 +0200
3bd62c21 (Stephen R. van den Berg 2008-08-14 20:02:20 +0200
73a7a656 (Jens Axboe 2007-07-27 14:00:29 -0700
49ba83fb (Jon Loeliger 2006-09-19 20:31:51 -0500
678dac6b (Tilman Sauerbeck 2006-08-22 19:37:41 +0200
d9edcbd6 (Junio C Hamano 2006-09-07 01:40:04 -0700
dd467629 (Jon Loeliger 2006-09-26 09:47:43 -0500
dd467629 (Jon Loeliger 2006-09-26 09:47:43 -0500
dd467629 (Jon Loeliger 2006-09-26 09:47:43 -0500
4ae95682 (H. Peter Anvin 2005-09-26 19:10:55 -0700
4ae95682 (H. Peter Anvin 2005-09-26 19:10:55 -0700
96f1e58f (David Rientjes
```

```
2006-08-15 10:23:48 -0700
1) #include "cache.h"
2) #include "pkt-line.h"
3) #include "exec_cmd.h"
4) #include " interpolate.h"
5)
6) #include <syslog.h>
7)
8) #ifndef HOST_NAME_MAX
9) #define HOST_NAME_MAX 256
10) #endif
11)
12) #ifndef NI_MAXSERV
13) #define NI_MAXSERV 32
14) #endif
15)
16) static int log_syslog;
17) static int verbose;
18) static int reuseaddr;
19)
20) static const char daemon_usage[] =
21) "git daemon [--verbose] [--syslog] [
22) " --timeout=n ] [--init-ti
23) " --strict-paths] [--base
24) " --user-path | --user-pa
25) " --interpolated-path=pat
26) " --reuseaddr] [--detach]
27) " [--enable|disable|allow
28) " --inetd | [--listen=hos
29) " --user=user
30) " [directory...]";
31)
32) /* List of acceptable pathname prefi
33) static char **ok_paths;
```

git blame daemon.c

979e32fa (Randal L. Schwartz	2005-10-25 16:29:09 -0700	1) #include "cache.h"
85023577 (Junio C Hamano	2006-12-19 14:34:12 -0800	2) #include "pkt-line.h"
77cb17e9 (Michal Ostrowski	2006-01-10 21:12:17 -0500	3) #include "exec_cmd.h"
49ba83fb (Jon Loeliger	2006-09-19 20:31:51 -0500	4) #include " interpolate.h"
f8ff0c06 (Petr Baudis	2005-09-22 11:25:28 +0200	5)
85023577 (Junio C Hamano	2006-12-19 14:34:12 -0800	6) #include <syslog.h>
85023577 (Junio C Hamano	2006-12-19 14:34:12 -0800	7)
695dffe2 (Johannes Schindelin	2006-09-28 12:00:35 +0200	8) #ifndef HOST_NAME_MAX
695dffe2 (Johannes Schindelin	2006-09-28 12:00:35 +0200	9) #define HOST_NAME_MAX 256
695dffe2 (Johannes Schindelin	2006-09-28 12:00:35 +0200	10) #endif
695dffe2 (Johannes Schindelin	2006-09-28 12:00:35 +0200	11)
415e7b87 (Patrick Welche	2007-10-18 18:17:39 +0100	12) #ifndef NI_MAXSERV
415e7b87 (Patrick Welche	2007-10-18 18:17:39 +0100	13) #define NI_MAXSERV 32
415e7b87 (Patrick Welche	2007-10-18 18:17:39 +0100	14) #endif
415e7b87 (Patrick Welche	2007-10-18 18:17:39 +0100	15)
9048fe1c (Petr Baudis	2005-09-24 16:13:01 +0200	16) static int log_syslog;
f8ff0c06 (Petr Baudis	2005-09-22 11:25:28 +0200	17) static int verbose;
1955fabf (Mark Wooding	2006-02-03 20:27:04 +0000	18) static int reuseaddr;
f8ff0c06 (Petr Baudis	2005-09-22 11:25:28 +0200	19)
960deccb (H. Peter Anvin	2005-10-19 14:27:01 -0700	20) static const char daemon_usage[] =
1b1dd23f (Stephan Beyer	2008-07-13 15:36:15 +0200	21) "git daemon [--verbose] [--syslog] [
3bd62c21 (Stephen R. van den Berg	2008-08-14 20:02:20 +0200	22) " --timeout=n] [--init-ti
3bd62c21 (Stephen R. van den Berg	2008-08-14 20:02:20 +0200	23) " --strict-paths] [--base
73a7a656 (Jens Axboe	2007-07-27 14:00:29 -0700	24) " --user-path --user-pa
49ba83fb (Jon Loeliger	2006-09-19 20:31:51 -0500	25) " --interpolated-path=pat
678dac6b (Tilman Sauerbeck	2006-08-22 19:37:41 +0200	26) " --reuseaddr] [--detach]
d9edcbd6 (Junio C Hamano	2006-09-07 01:40:04 -0700	27) " --[enable disable allow
dd467629 (Jon Loeliger	2006-09-26 09:47:43 -0500	28) " --inetd [--listen=hos
dd467629 (Jon Loeliger	2006-09-26 09:47:43 -0500	29) " [--user=user
dd467629 (Jon Loeliger	2006-09-26 09:47:43 -0500	30) " [directory...]";
4ae95682 (H. Peter Anvin	2005-09-26 19:10:55 -0700	31)
4ae95682 (H. Peter Anvin	2005-09-26 19:10:55 -0700	32) /* List of acceptable pathname prefi
96f1e58f (David Rienties	2006-08-15 10:23:48 -0700	33) static char **ok_paths:

```
git blame -c GITPackUpload.m
```

git blame -C GITPackUpload.m

```
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 12)
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 13) #define PACK_SIGN
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 14) #define PACK_VERS
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 15)
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 16) @implementation G
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 17)
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 18) @synthesize gitRe
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 19) @synthesize needR
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 20) @synthesize gitSo
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 21) @synthesize refDi
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 22)
a2cbabf5 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-25 22:29:39 +0100 23) - (id) initWithGi
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 24) {
a2cbabf5 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-25 22:29:39 +0100 25)     gitRepo =
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 26)     needRefs =
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 27)     gitSocket
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 28)     return sel
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 29) }
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 30)
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 31) - (bool) uploadPa
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 32) {
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 33)     NSLog(@"up
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 34)     NSString *
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 35)     NSArray *t
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 36)
56ef2caf Source/Network/GITServerHandler.m (Scott Chacon 2009-01-05 21:44:26 -0800 37)     refDict =
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 38)
```

git blame -C GITPackUpload.m

f344f58d	Source/Network/GITServerHandler.m	(Scott Chacon 2009-01-04 18:59:04 -0800 12)
f344f58d	Source/Network/GITServerHandler.m	(Scott Chacon 2009-01-04 18:59:04 -0800 13) #define PACK_SIGN
f344f58d	Source/Network/GITServerHandler.m	(Scott Chacon 2009-01-04 18:59:04 -0800 14) #define PACK_VERS
f344f58d	Source/Network/GITServerHandler.m	(Scott Chacon 2009-01-04 18:59:04 -0800 15)
ad11ac80	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-24 18:32:50 +0100 16) @implementation G
ad11ac80	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-24 18:32:50 +0100 17)
ad11ac80	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-24 18:32:50 +0100 18) @synthesize gitRe
ad11ac80	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-24 18:32:50 +0100 19) @synthesize needR
ad11ac80	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-24 18:32:50 +0100 20) @synthesize gitSo
ad11ac80	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-24 18:32:50 +0100 21) @synthesize refDi
ad11ac80	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-24 18:32:50 +0100 22)
a2cbabf5	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-25 22:29:39 +0100 23) - (id) initWithGi
ad11ac80	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-24 18:32:50 +0100 24) {
a2cbabf5	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-25 22:29:39 +0100 25) gitRepo =
ad11ac80	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-24 18:32:50 +0100 26) needRefs =
ad11ac80	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-24 18:32:50 +0100 27) gitSocket
ad11ac80	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-24 18:32:50 +0100 28) return sel
ad11ac80	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-24 18:32:50 +0100 29) }
ad11ac80	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-24 18:32:50 +0100 30)
ad11ac80	Source/Network/GITPackUpload.m	(Scott Chacon 2009-03-24 18:32:50 +0100 31) - (bool) uploadPa
f344f58d	Source/Network/GITServerHandler.m	(Scott Chacon 2009-01-04 18:59:04 -0800 32) {
f344f58d	Source/Network/GITServerHandler.m	(Scott Chacon 2009-01-04 18:59:04 -0800 33) NSLog(@"up
f344f58d	Source/Network/GITServerHandler.m	(Scott Chacon 2009-01-04 18:59:04 -0800 34) NSString *
f344f58d	Source/Network/GITServerHandler.m	(Scott Chacon 2009-01-04 18:59:04 -0800 35) NSArray *t
f344f58d	Source/Network/GITServerHandler.m	(Scott Chacon 2009-01-04 18:59:04 -0800 36)
56ef2caf	Source/Network/GITServerHandler.m	(Scott Chacon 2009-01-05 21:44:26 -0800 37) refDict =
f344f58d	Source/Network/GITServerHandler.m	(Scott Chacon 2009-01-04 18:59:04 -0800 38)

git blame -C GITPackUpload.m

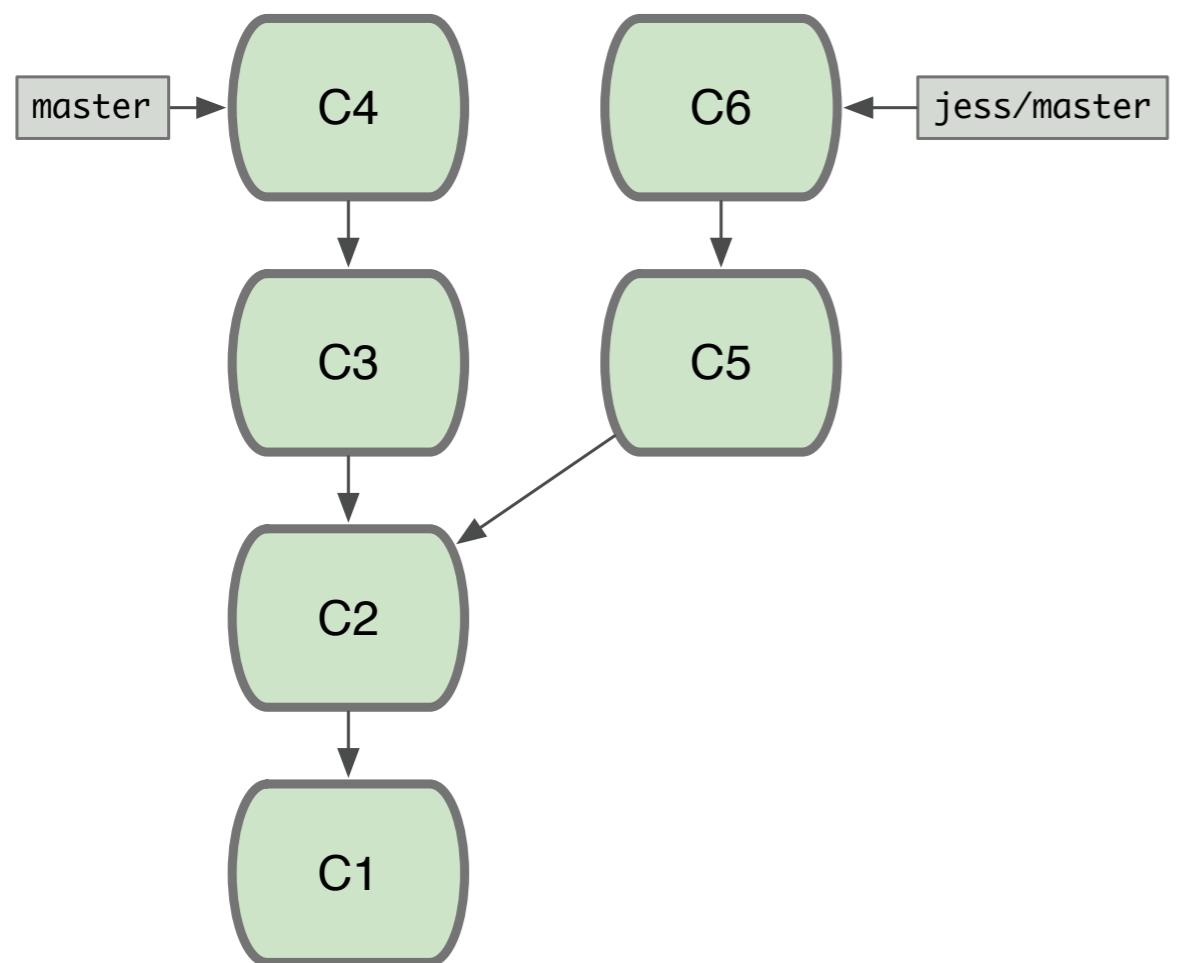
```
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 12)
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 13) #define PACK_SIGN
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 14) #define PACK_VERS
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 15)
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 16) @implementation G
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 17)
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 18) @synthesize gitRe
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 19) @synthesize needR
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 20) @synthesize gitSo
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 21) @synthesize refDi
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 22)
a2cbabf5 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-25 22:29:39 +0100 23) - (id) initWithGi
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 24) {
a2cbabf5 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-25 22:29:39 +0100 25)     gitRepo =
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 26)     needRefs =
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 27)     gitSocket
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 28)     return sel
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 29) }
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 30)
ad11ac80 Source/Network/GITPackUpload.m (Scott Chacon 2009-03-24 18:32:50 +0100 31) - (bool) uploadPa
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 32) {
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 33)     NSLog(@"up
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 34)     NSString *
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 35)     NSArray *t
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 36)
56ef2caf Source/Network/GITServerHandler.m (Scott Chacon 2009-01-05 21:44:26 -0800 37)     refDict =
f344f58d Source/Network/GITServerHandler.m (Scott Chacon 2009-01-04 18:59:04 -0800 38)
```

Rewriting History

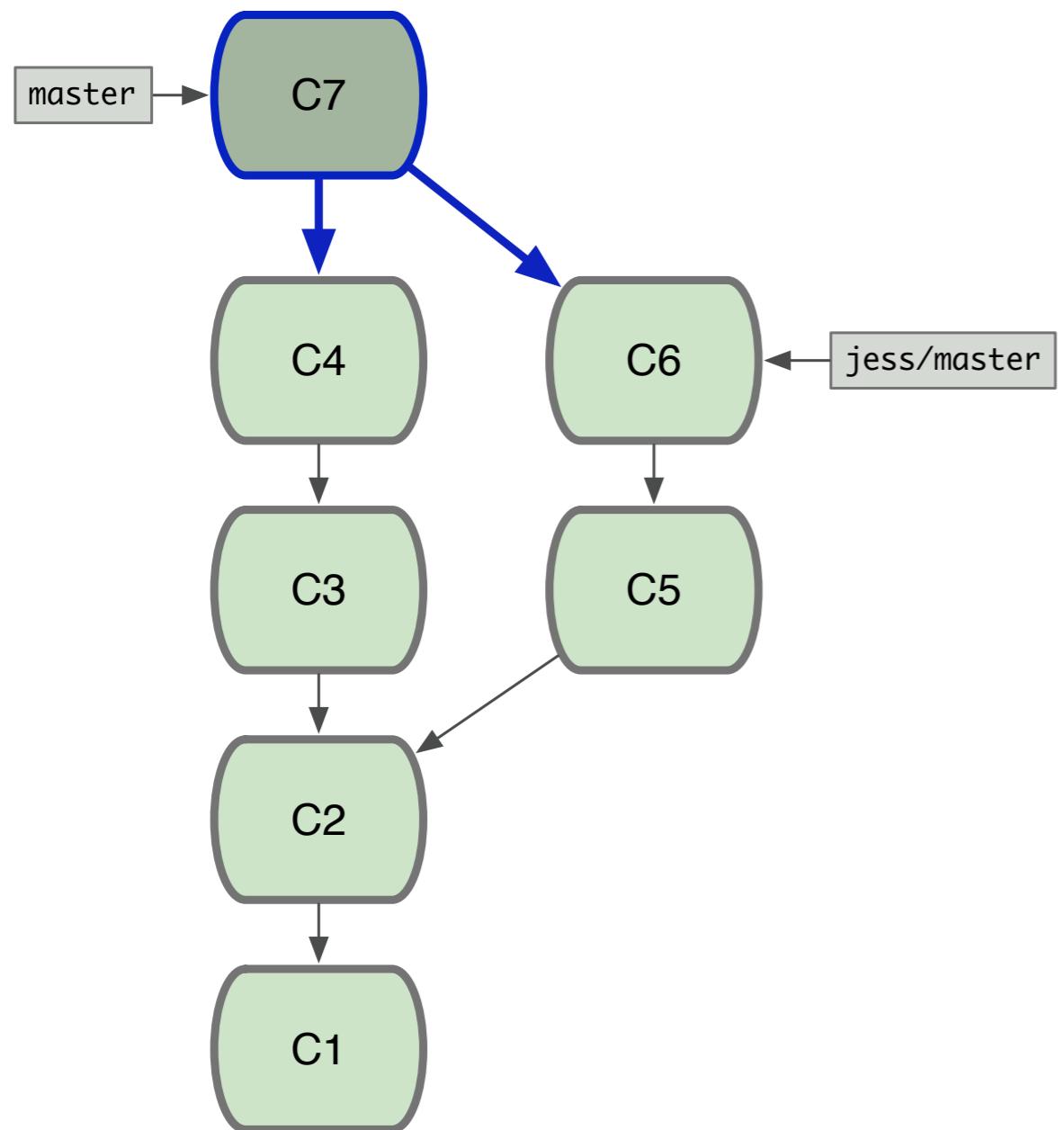
Modifying the last commit

git commit --amend

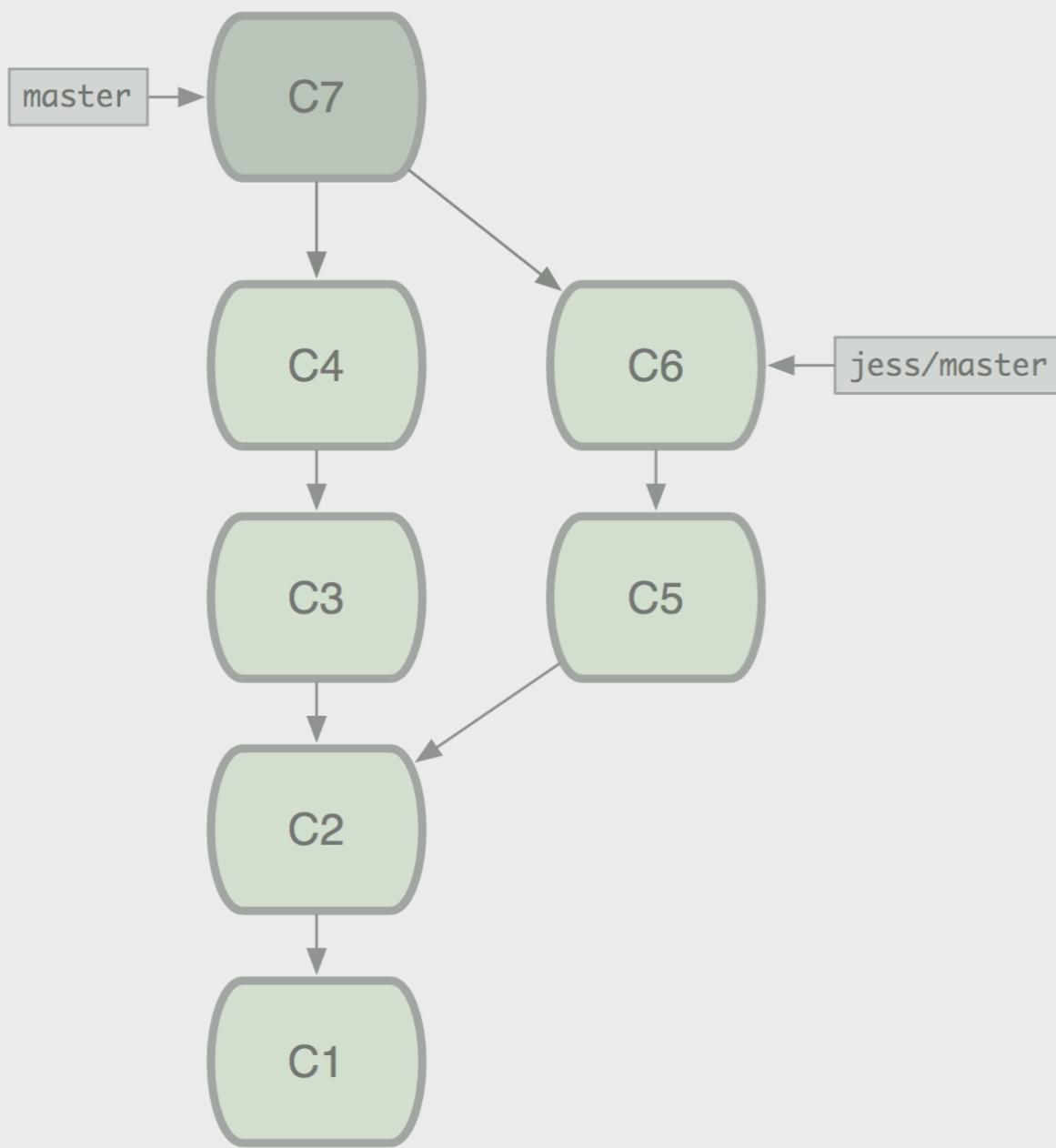
Rebasing



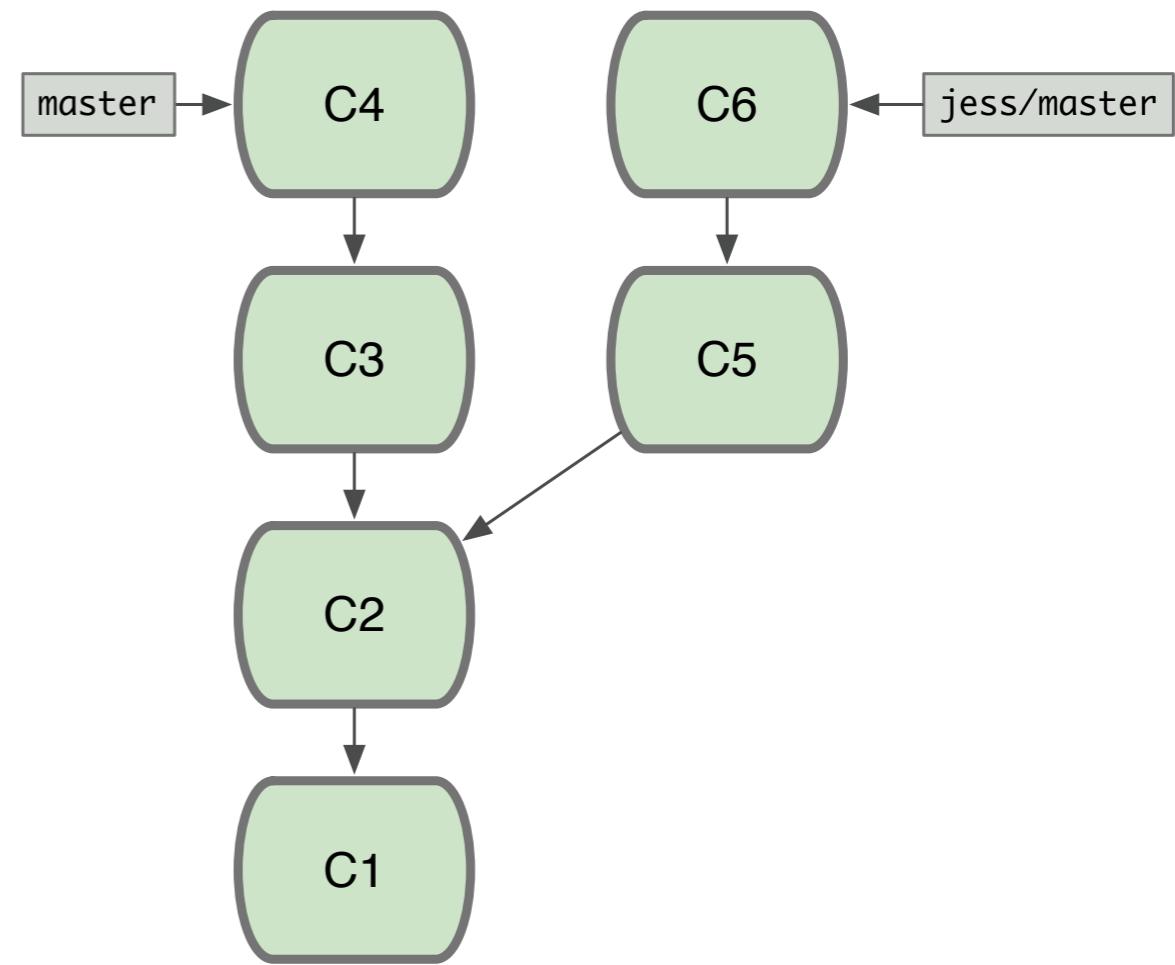
git merge



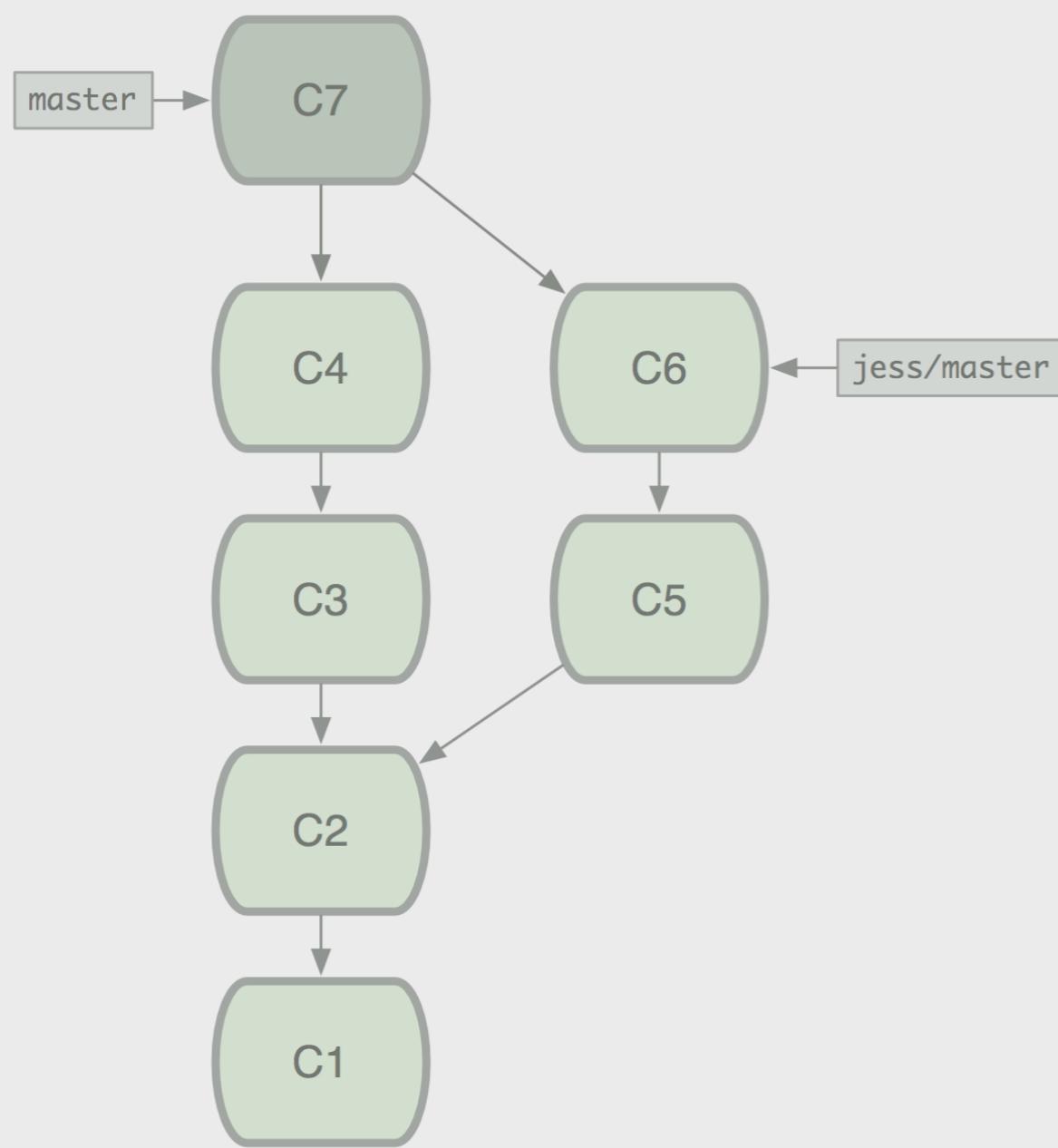
git merge



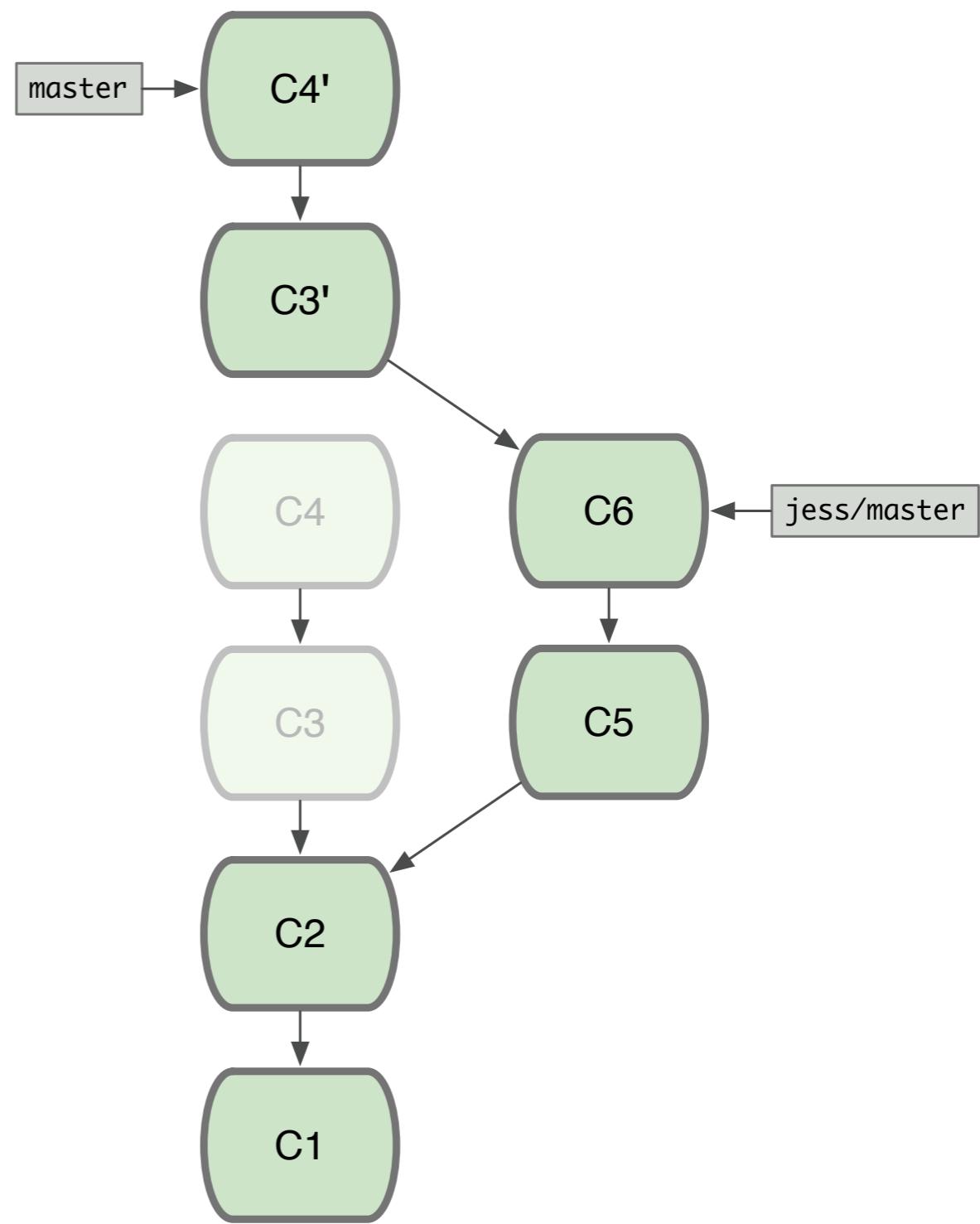
git rebase



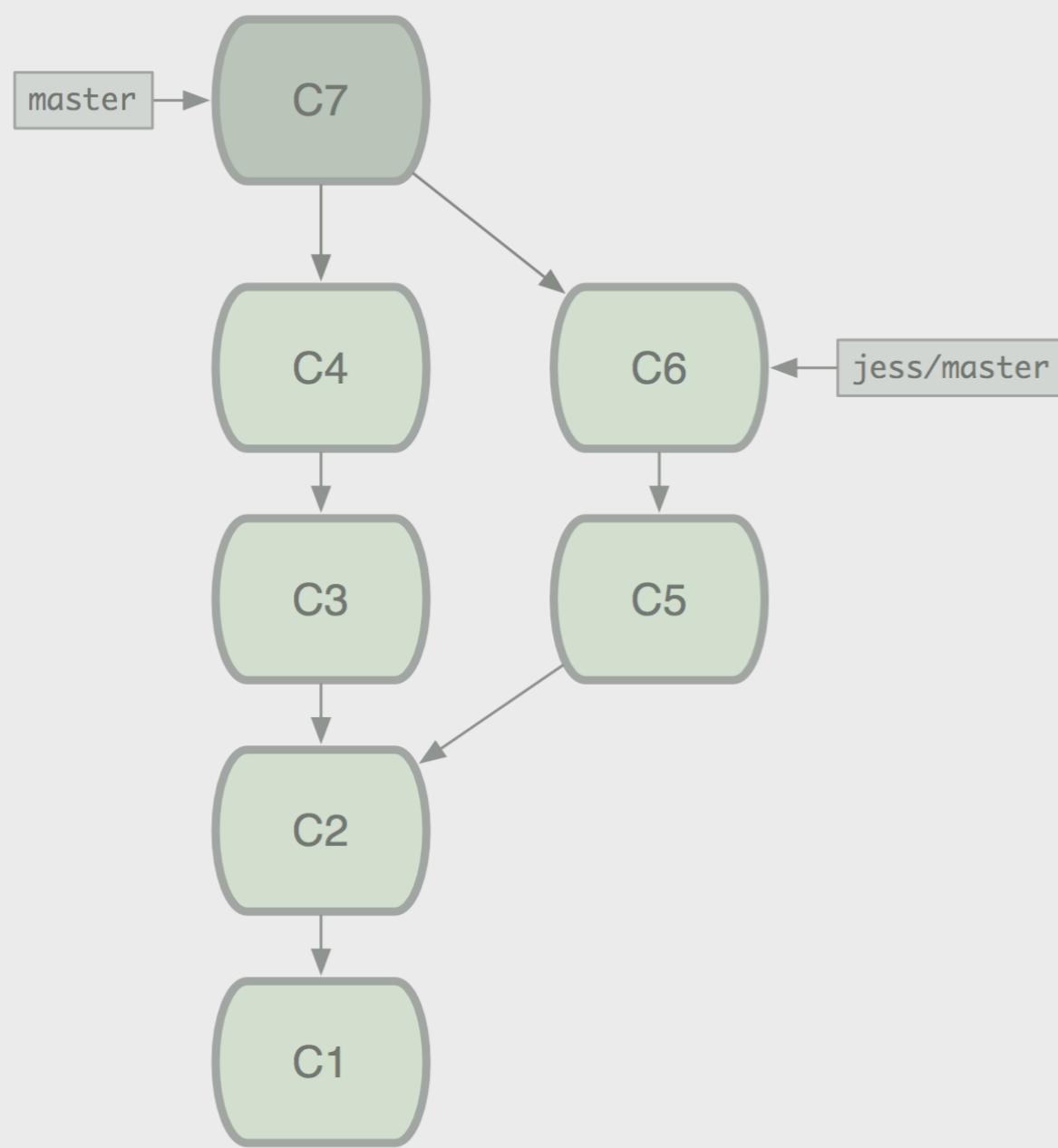
git merge



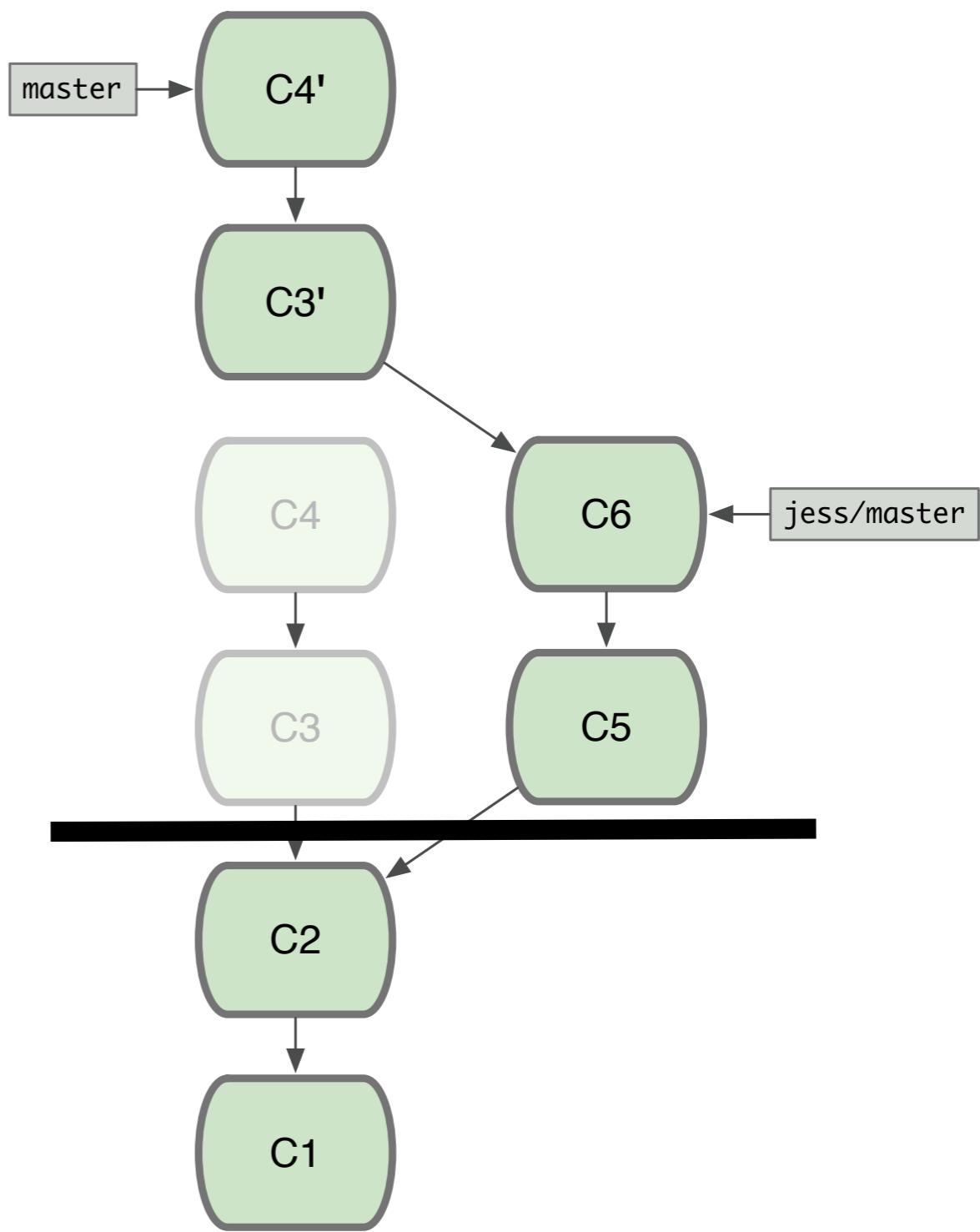
git rebase



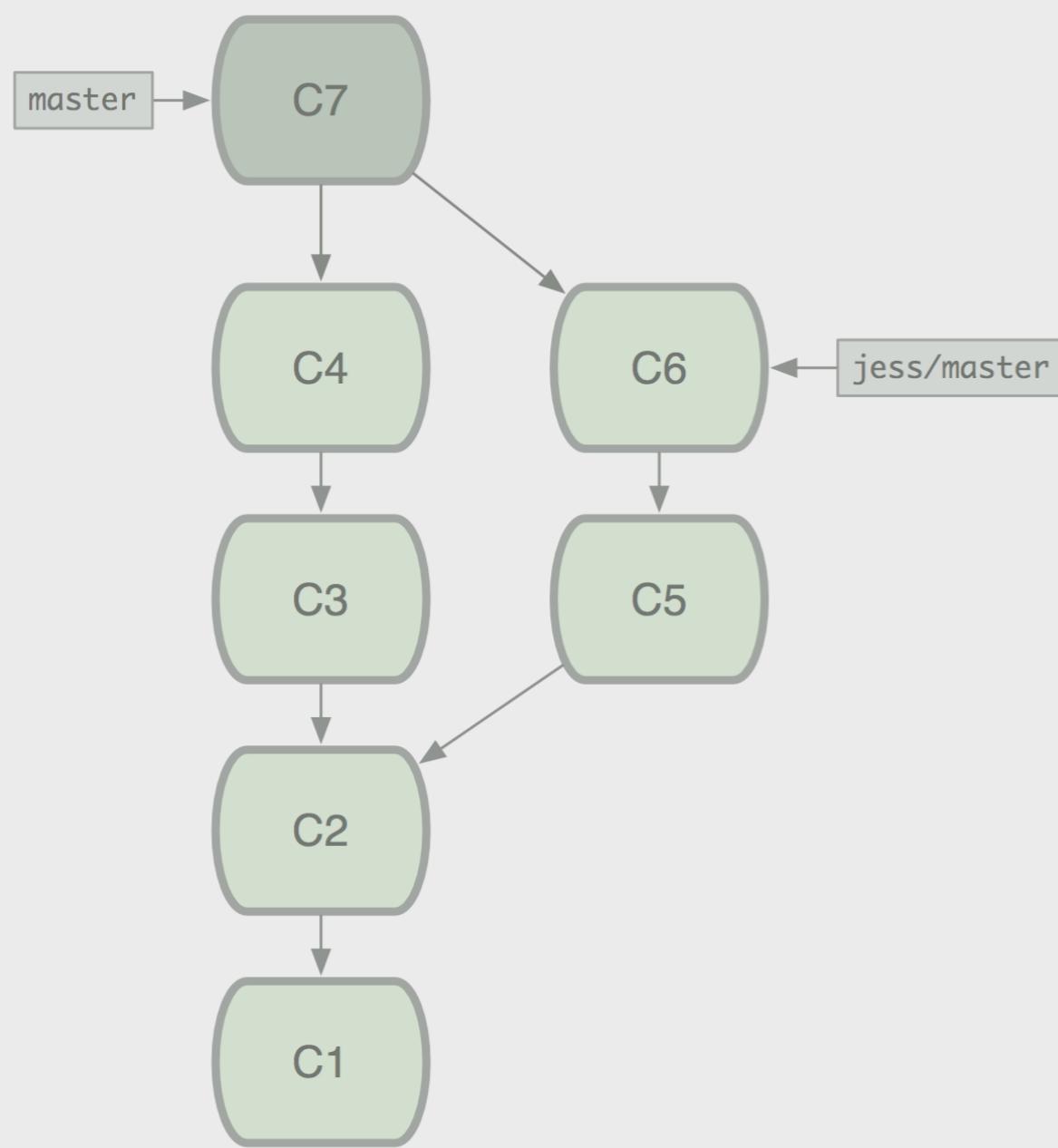
git merge



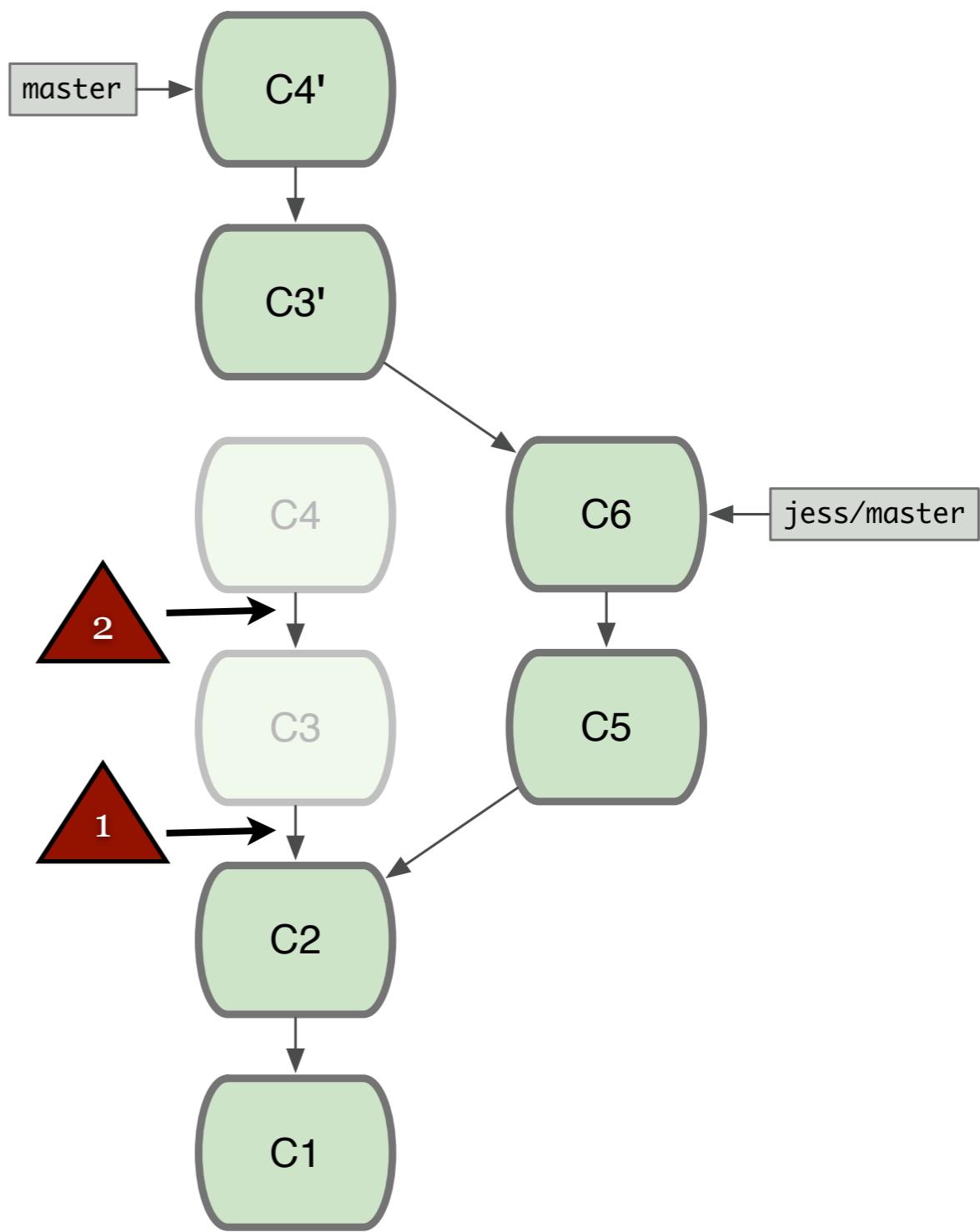
git rebase



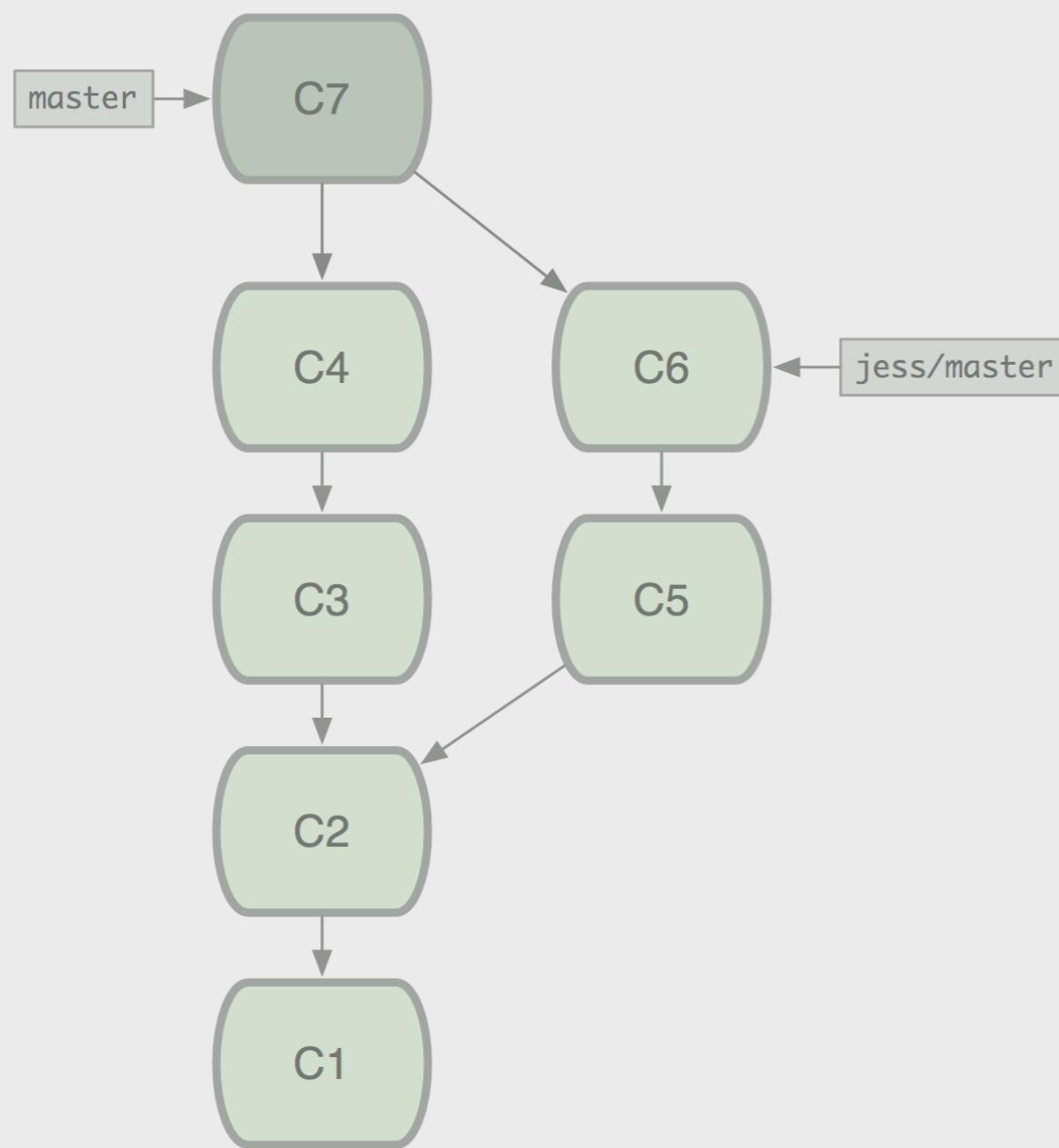
git merge



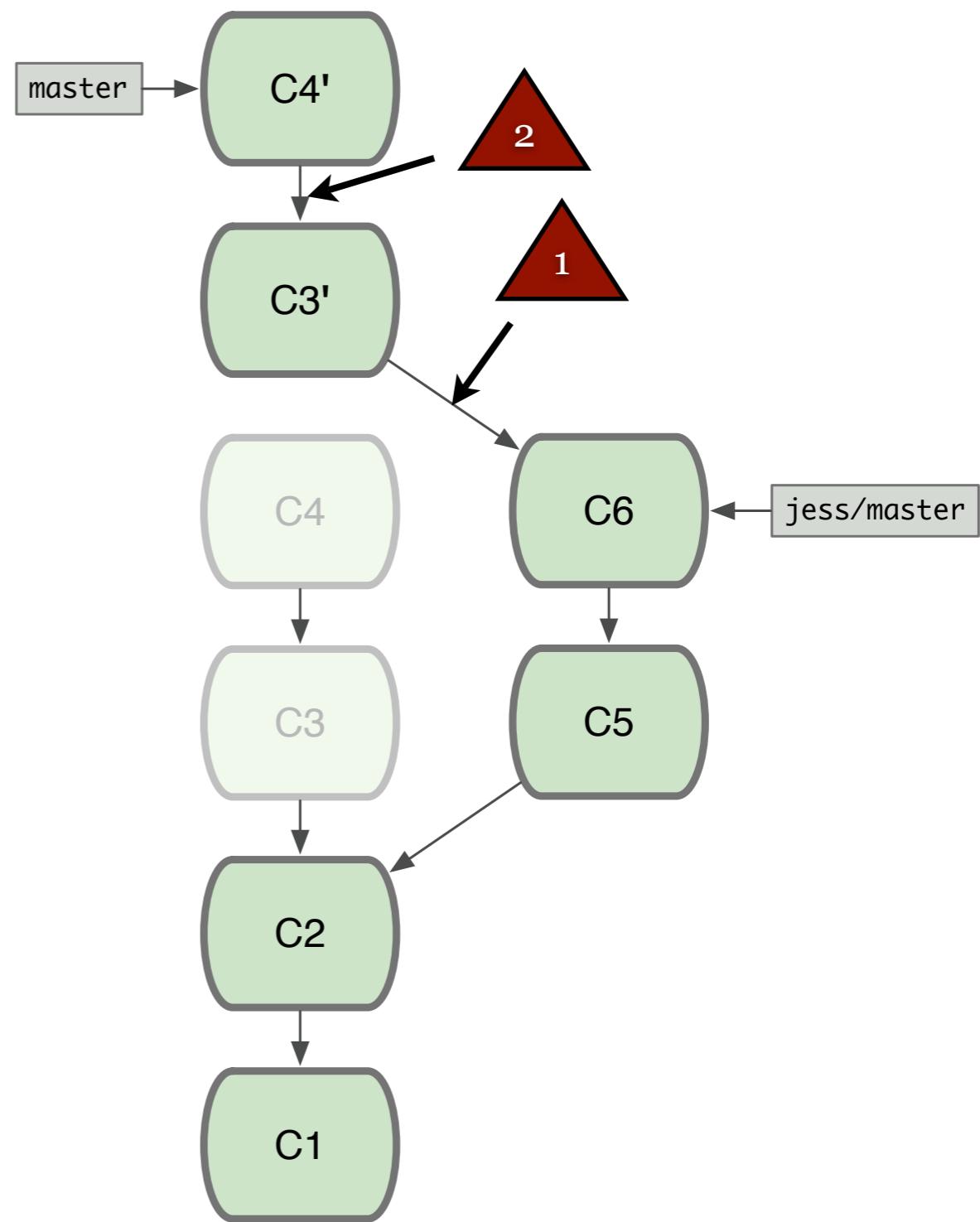
git rebase



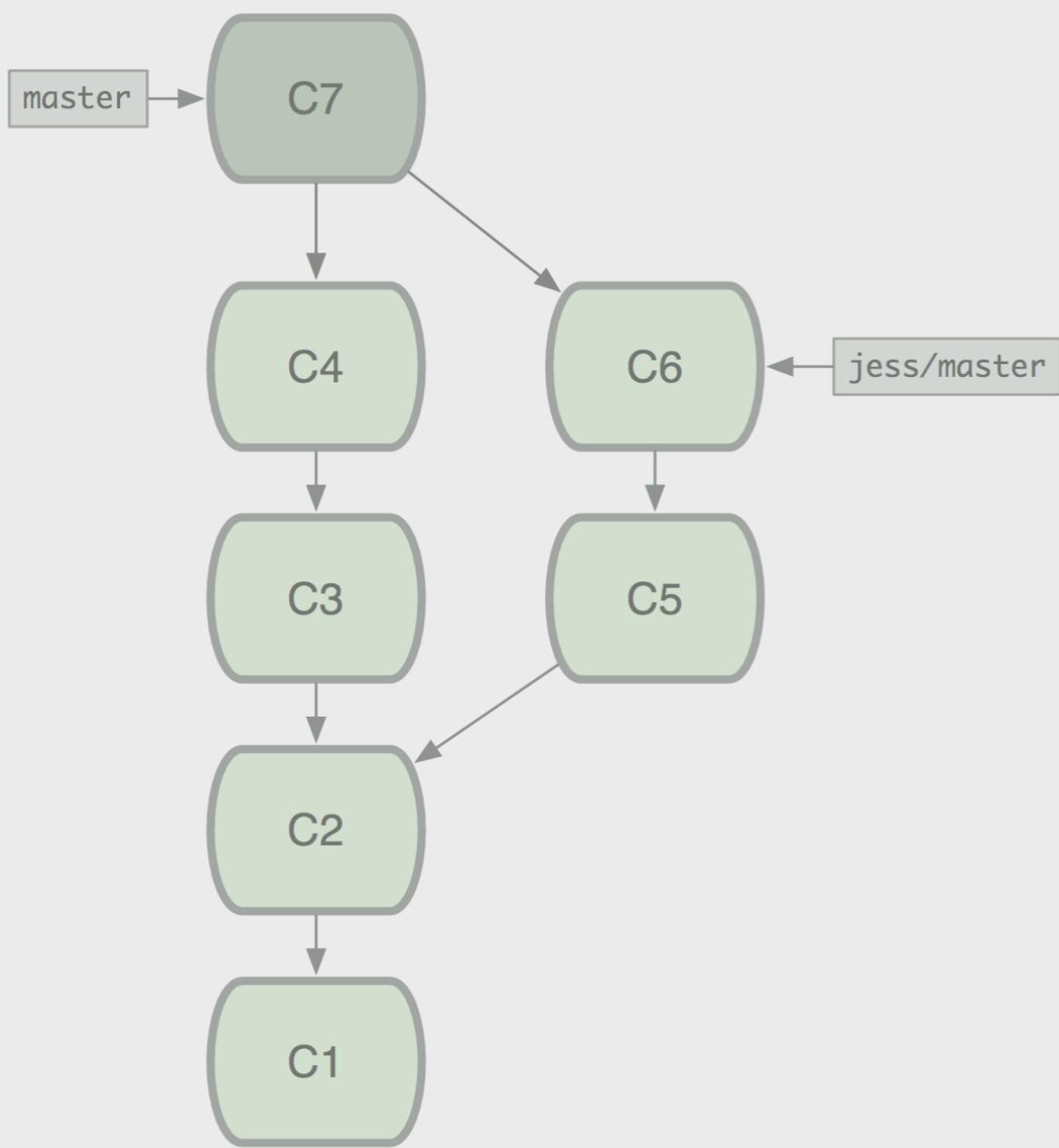
git merge



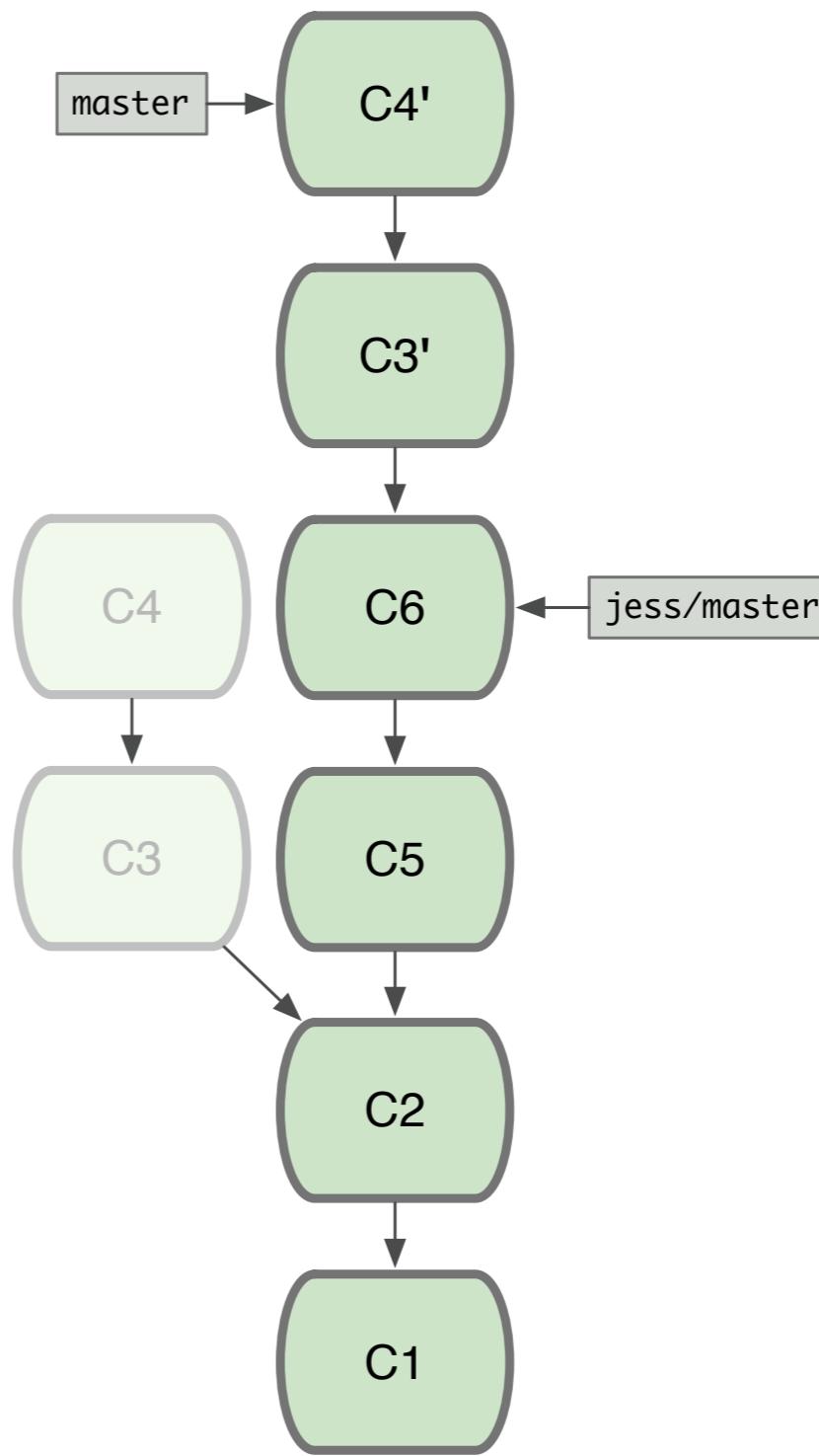
git rebase



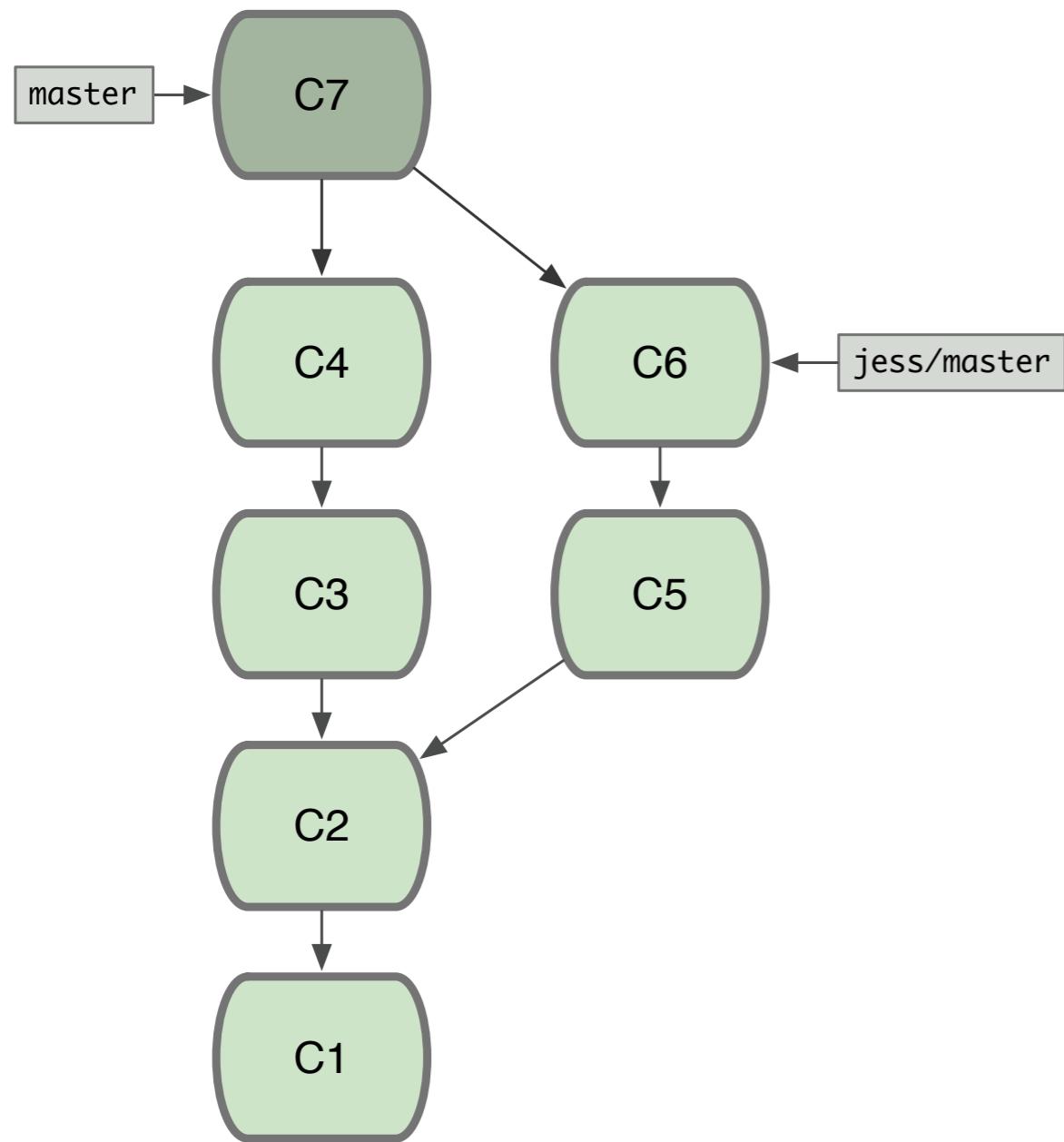
git merge



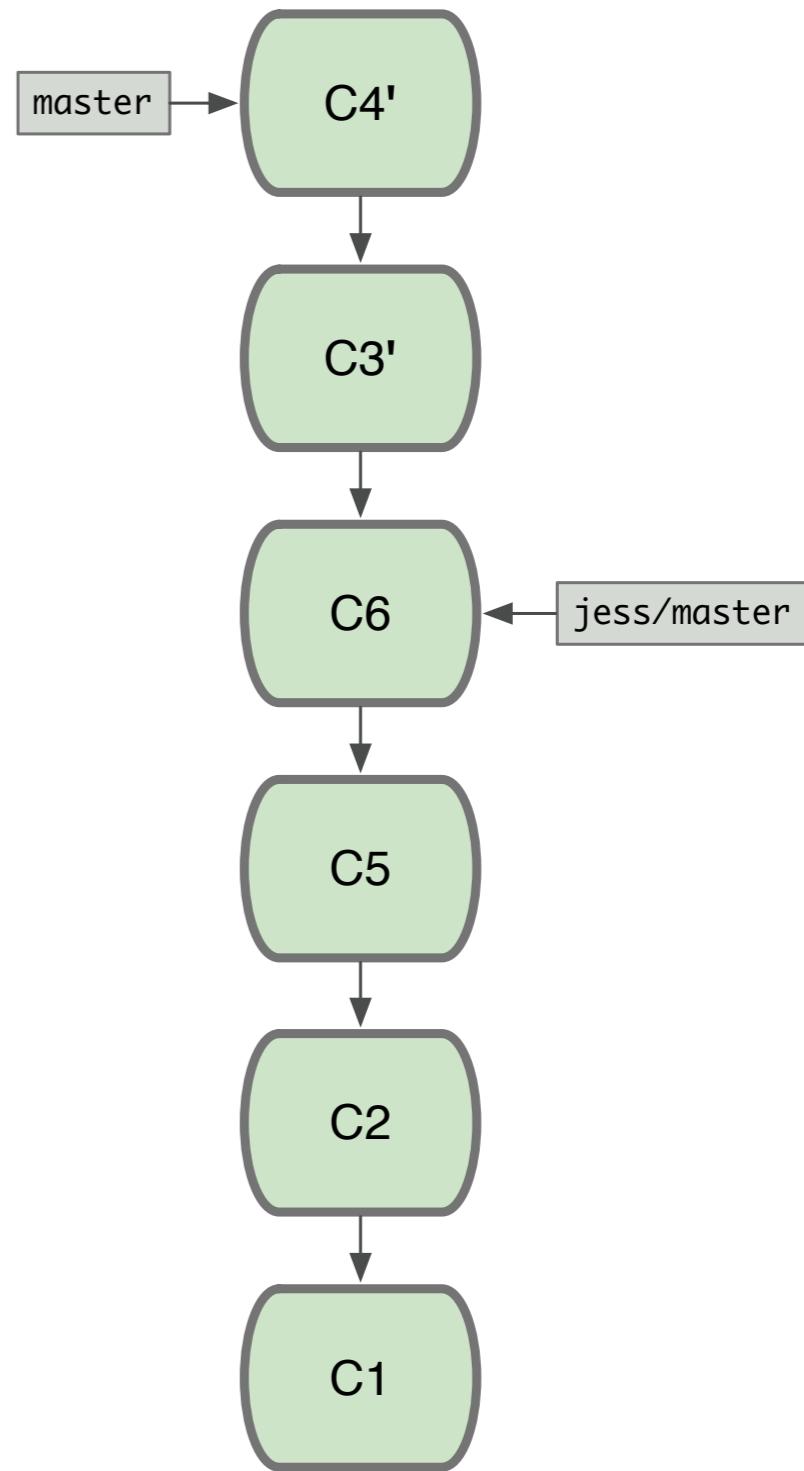
git rebase



git merge



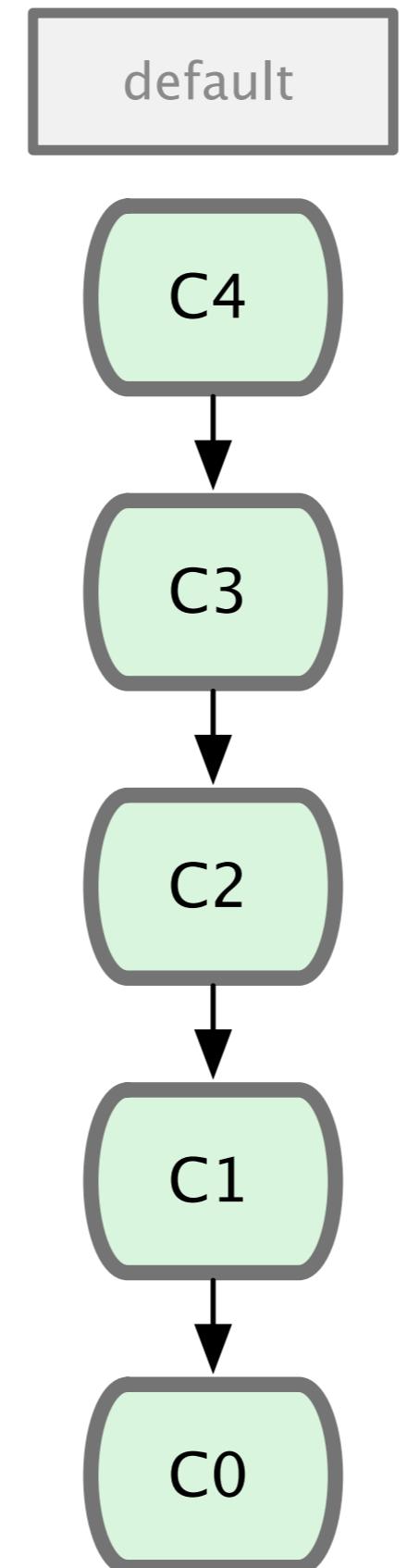
git rebase

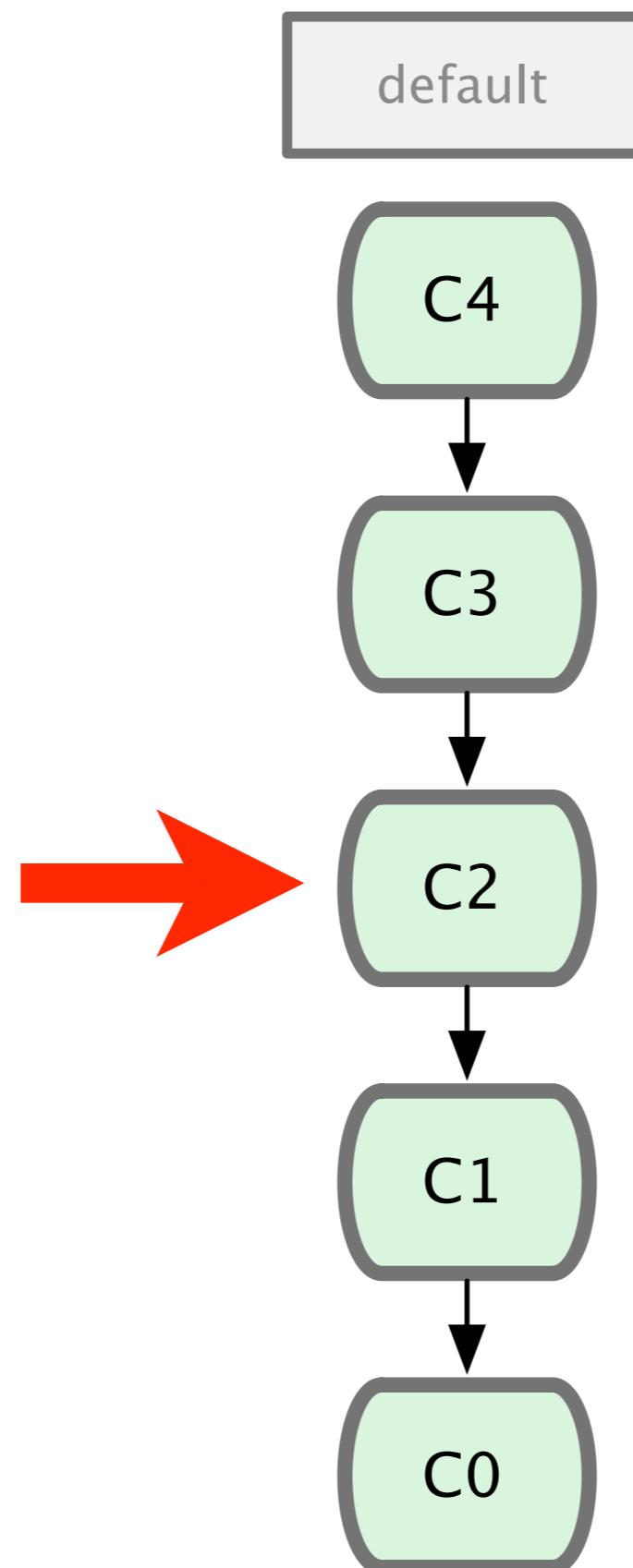


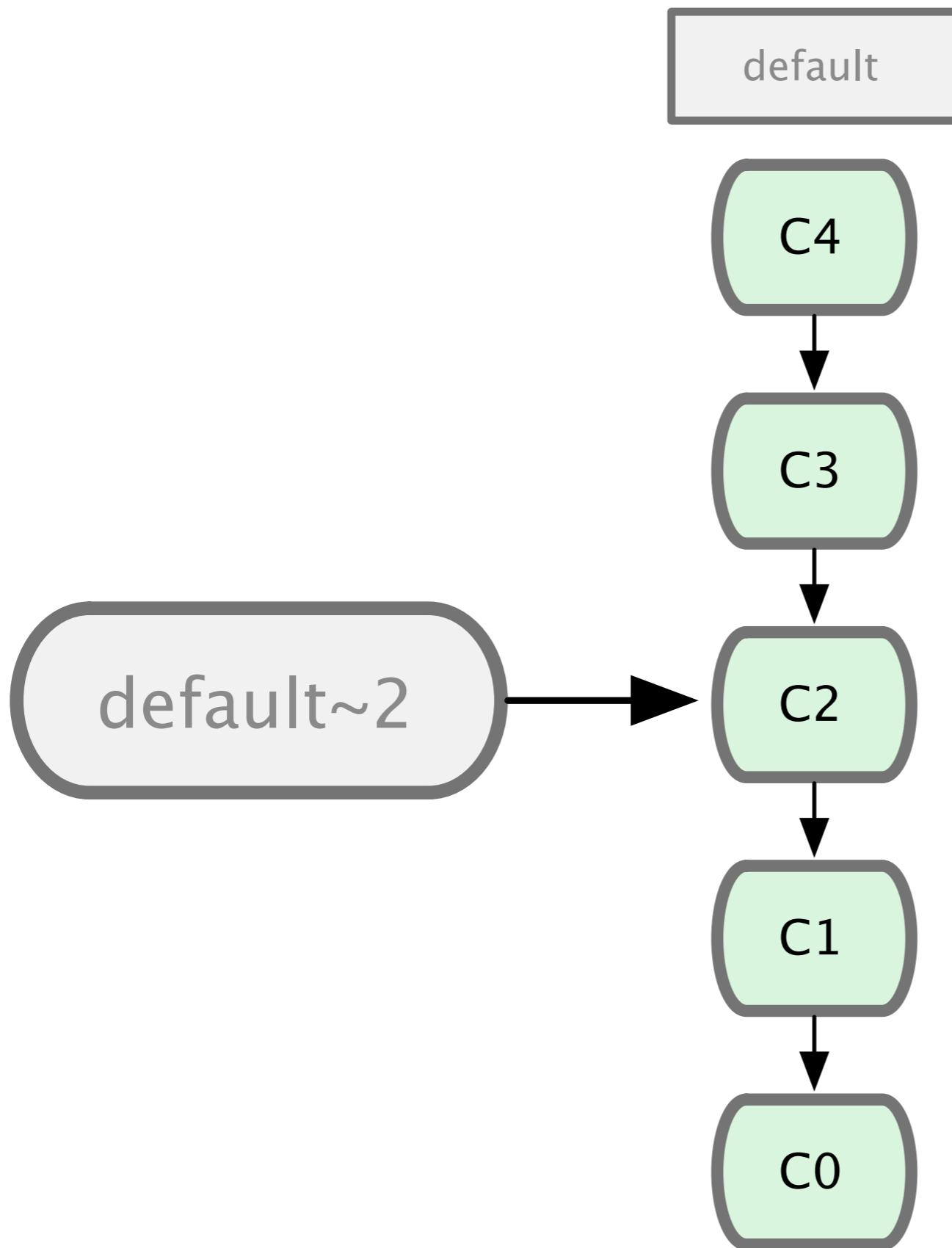
Fixing a commit
several back

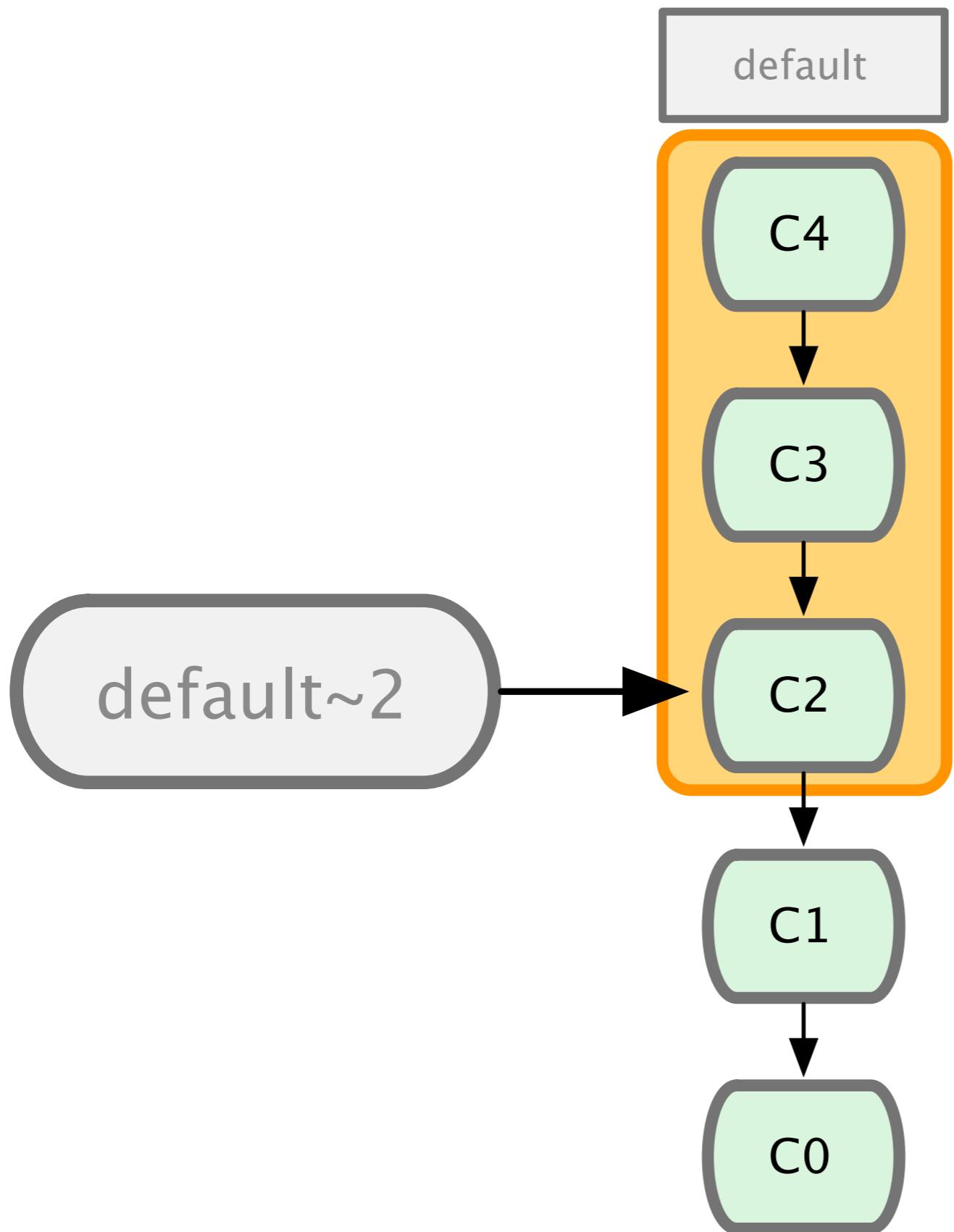
git rebase -i

git rebase --interactive









```
git rebase -i default~2^
```

```
git rebase -i default~2^
```

git rebase -i default~2[^]

```
pick 969c877 git apply --directory broken for new files
pick b75271d git diff <tree>{3,}: do not reverse order of args
pick 72d404d test-lib: fix broken printf
```

```
# Rebase f285a2d..5c283eb onto f285a2d
#
# Commands:
# p, pick = use commit
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
~ ~ ~ ~ ~
"~/projects/git/.git/rebase-merge/git-rebase-todo" 14L, 472C
```

C2

C3

C4

```
pick 969c877 git apply --directory broken for new files
pick b75271d git diff <tree>{3,}: do not reverse order of args
pick 72d404d test-lib: fix broken printf
```

```
# Rebase f285a2d..5c283eb onto f285a2d
#
# Commands:
# p, pick = use commit
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
~ ~ ~ ~ ~
"~/projects/git/.git/rebase-merge/git-rebase-todo" 14L, 472C
```

```
pick 969c877 git apply --directory broken for new files
pick b75271d git diff <tree>{3,}: do not reverse order of args
pick 72d404d test-lib: fix broken printf
```

```
# Rebase f285a2d..5c283eb onto f285a2d
#
# Commands:
# p, pick = use commit
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
~ ~ ~ ~ ~
"~/projects/git/.git/rebase-merge/git-rebase-todo" 14L, 472C
```

```
pick 969c877 git apply --directory broken for new files
pick b75271d git diff <tree>{3,}: do not reverse order of args
pick 72d404d test-lib: fix broken printf
```

```
# Rebase f285a2d..5c283eb onto f285a2d
#
# Commands:
# p, pick = use commit
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
~ ~ ~ ~ ~
"~/projects/git/.git/rebase-merge/git-rebase-todo" 14L, 472C
```

```
pick 969c877 git apply --directory broken for new files
pick b75271d git diff <tree>{3,}: do not reverse order of args
pick 72d404d test-lib: fix broken printf
```

```
# Rebase f285a2d..5c283eb onto f285a2d
#
# Commands:
# p, pick = use commit
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
~ ~ ~ ~ ~
"~/projects/git/.git/rebase-merge/git-rebase-todo" 14L, 472C
```

```
pick 969c877 git apply --directory broken for new files
pick b75271d git diff <tree>{3,}: do not reverse order of args
pick 72d404d test-lib: fix broken printf
```

```
# Rebase f285a2d..5c283eb onto f285a2d
#
# Commands:
# p, pick = use commit
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
~ ~ ~ ~ ~
"~/projects/git/.git/rebase-merge/git-rebase-todo" 14L, 472C
```

```
pick 969c877 git apply --directory broken for new files
pick b75271d git diff <tree>{3,}: do not reverse order of args
pick 72d404d test-lib: fix broken printf
```

```
# Rebase f285a2d..5c283eb onto f285a2d
#
# Commands:
# p, pick = use commit
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
~ ~ ~ ~ ~
"~/projects/git/.git/rebase-merge/git-rebase-todo" 14L, 472C
```

```
edit 969c877 git apply --directory broken for new files
pick b75271d git diff <tree>{3,}: do not reverse order of args
pick 72d404d test-lib: fix broken printf
```

```
# Rebase f285a2d..5c283eb onto f285a2d
#
# Commands:
# p, pick = use commit
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
~ ~ ~ ~ ~
"~/projects/git/.git/rebase-merge/git-rebase-todo" 14L, 472C
```

```
$ git rebase -i default~2^
Stopped at 969c877... git apply --directory broken for new files
You can amend the commit now, with
```

```
git commit --amend
```

Once you are satisfied with your changes, run

```
git rebase --continue
```

\$ _

```
$ git rebase -i default~2^  
Stopped at 969c877... git apply --directory broken for new files  
You can amend the commit now, with
```

```
git commit --amend
```

Once you are satisfied with your changes, run

```
git rebase --continue
```

\$ _

edit files

```
$ git rebase -i default~2^  
Stopped at 969c877... git apply --directory broken for new files  
You can amend the commit now, with
```

```
git commit --amend
```

Once you are satisfied with your changes, run

```
git rebase --continue
```

\$ _

edit files
git add

```
$ git rebase -i default~2^  
Stopped at 969c877... git apply --directory broken for new files  
You can amend the commit now, with
```

→ `git commit --amend`

Once you are satisfied with your changes, run

`git rebase --continue`

\$ _

edit files
git add
git commit --amend

```
$ git rebase -i default~2^  
Stopped at 969c877... git apply --directory broken for new files  
You can amend the commit now, with
```

git commit --amend

Once you are satisfied with your changes, run

→ git rebase --continue

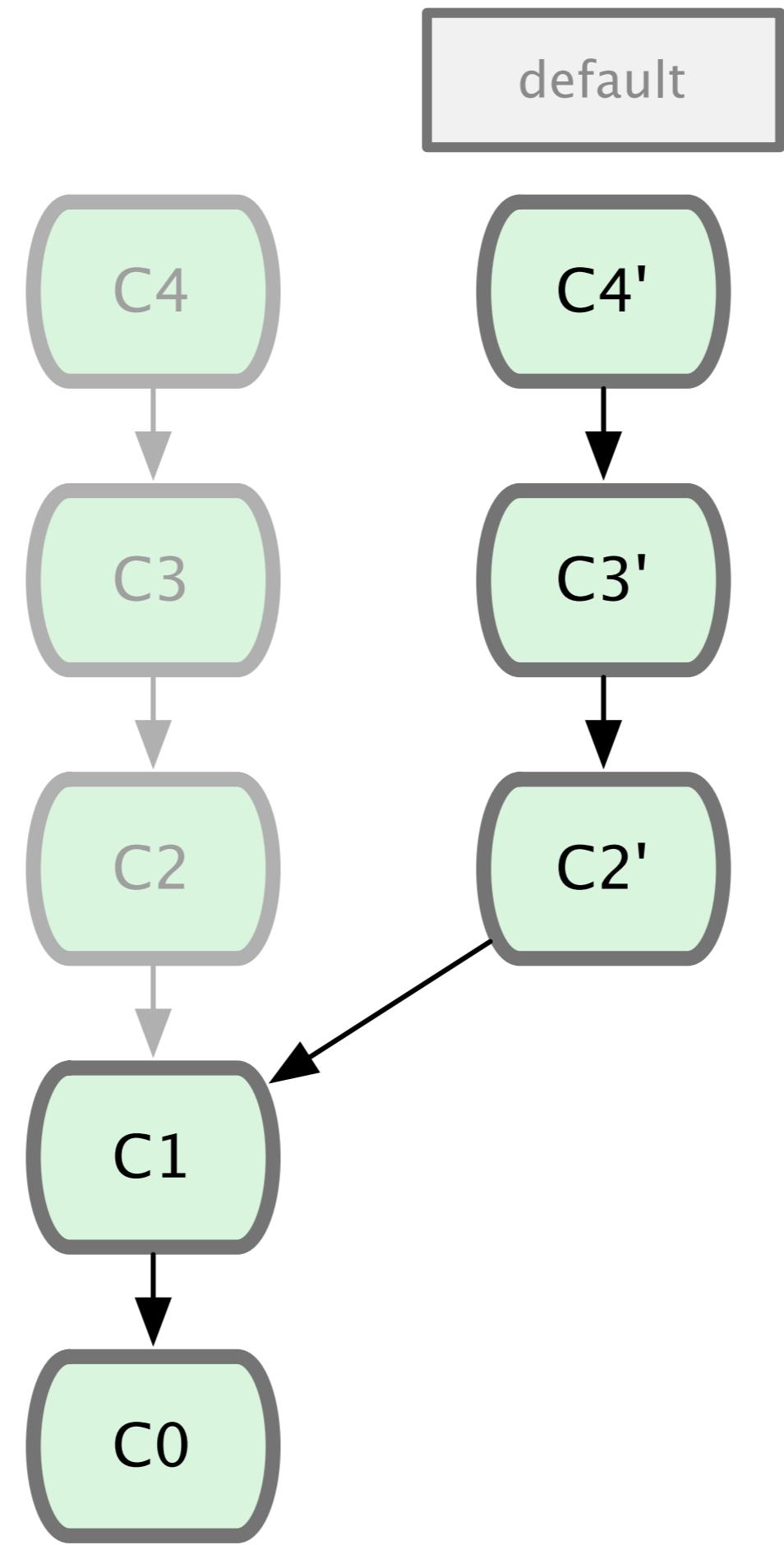
\$ _

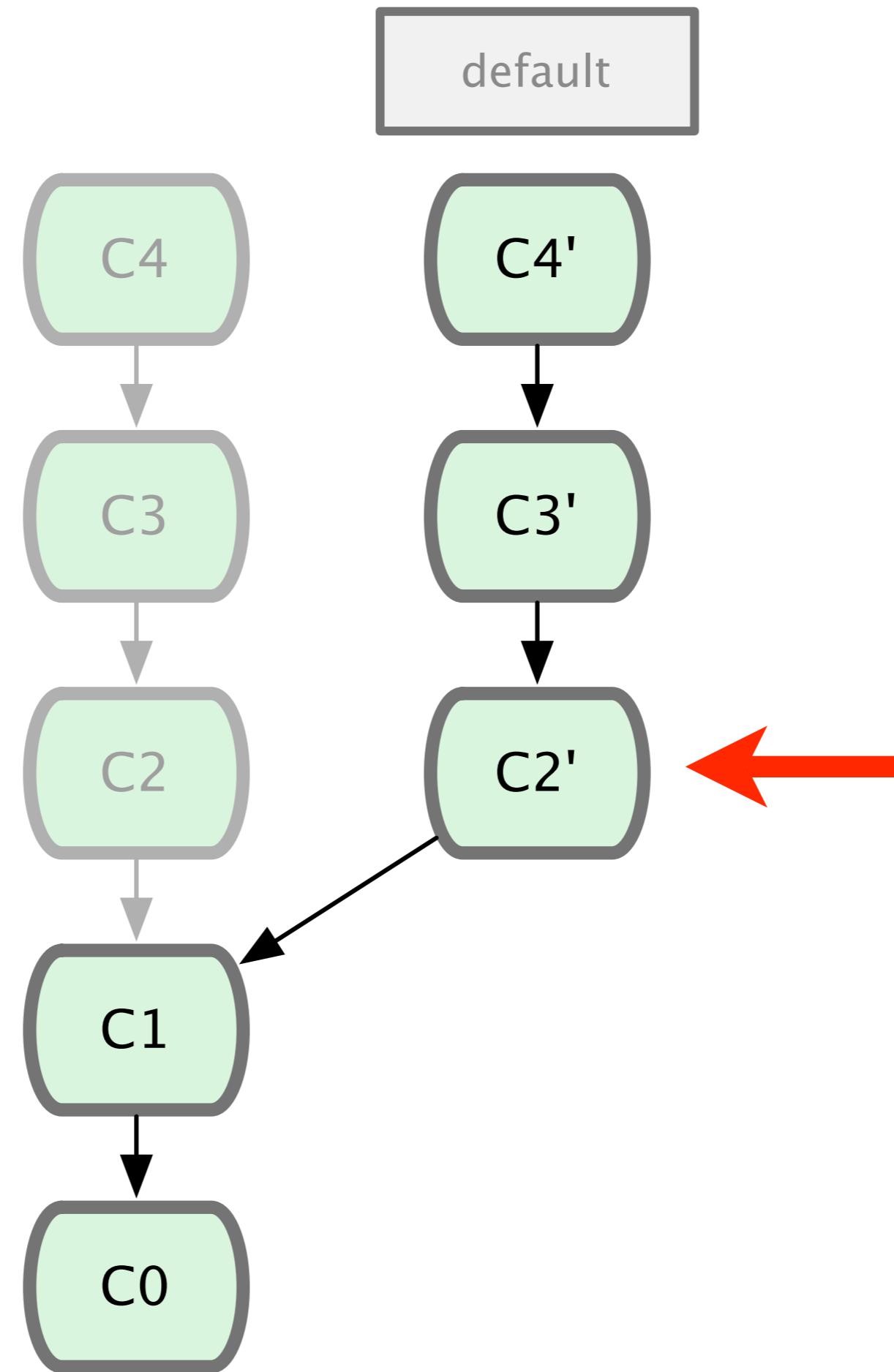
edit files

git add

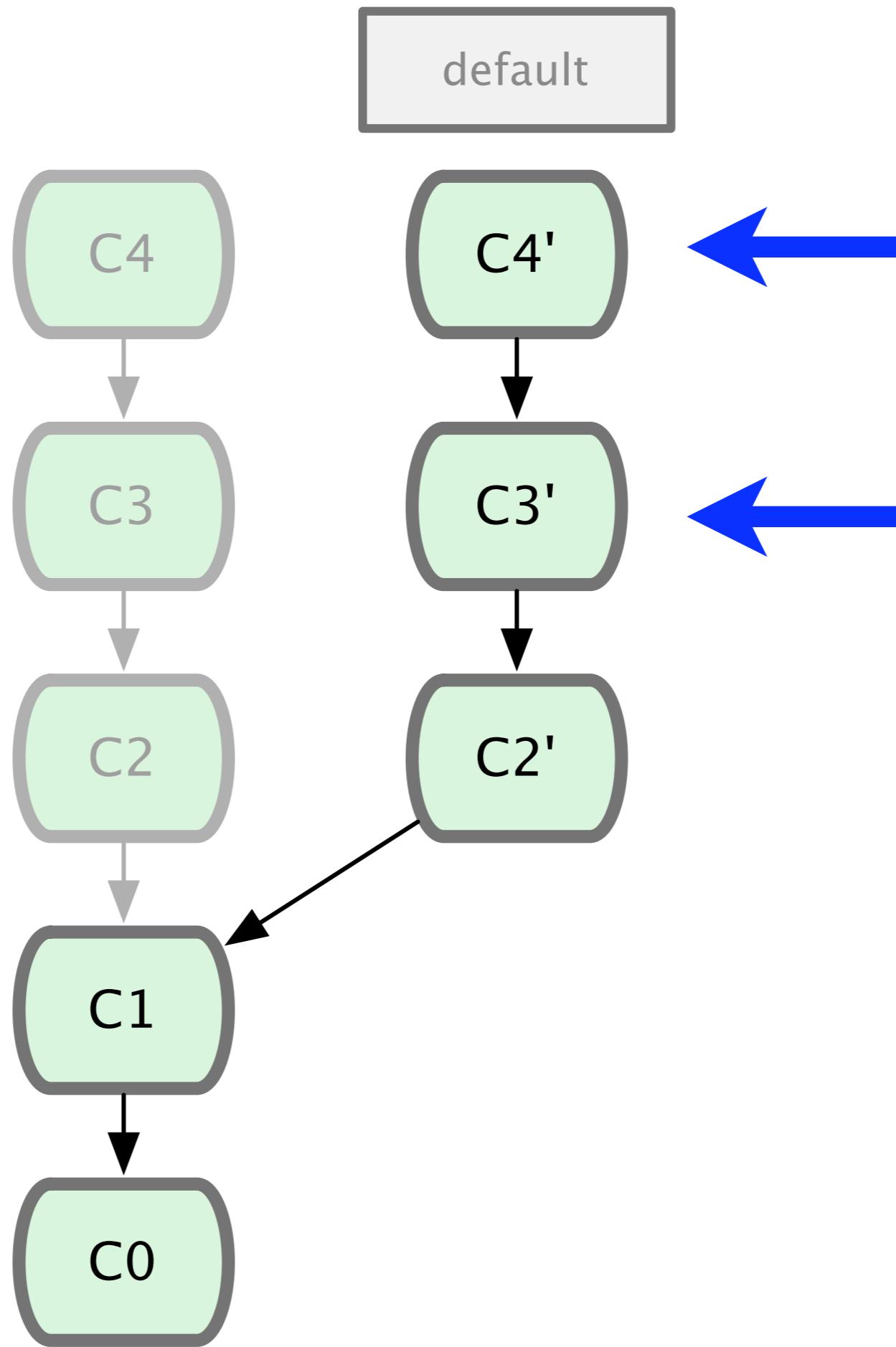
git commit --amend

git rebase --continue





default



Squashing commits
together

```
pick 969c877 git apply --directory broken for new files
pick b75271d git diff <tree>{3,}: do not reverse order of args
pick 72d404d test-lib: fix broken printf
```

```
# Rebase f285a2d..5c283eb onto f285a2d
#
# Commands:
# p, pick = use commit
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
~ ~ ~ ~ ~
"~/projects/git/.git/rebase-merge/git-rebase-todo" 14L, 472C
```

```
pick 969c877 git apply --directory broken for new files
squash b75271d git diff <tree>{3,}: do not reverse order of args
squash 72d404d test-lib: fix broken printf
```

```
# Rebase f285a2d..5c283eb onto f285a2d
#
# Commands:
# p, pick = use commit
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
~ ~ ~ ~ ~
"~/projects/git/.git/rebase-merge/git-rebase-todo" 14L, 472C
```

```
# This is a combination of 3 commits.
# The first commit's message is:
git apply --directory broken for new files

# This is the 2nd commit message:

git diff <tree>{3,}: do not reverse order of args

# This is the 3rd commit message:

test-lib: fix broken printf

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Author:    Jeff King <peff@peff.net>
#
# Not currently on any branch.
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:    builtin-apply.c
# modified:    builtin-diff.c
# modified:    t/t4013-diff-various.sh
# new file:    t/t4013/diff.diff_master_master^_side
# modified:    t/t4128-apply-root.sh
# modified:    t/test-lib.sh
#
~
~

".git/COMMIT_EDITMSG" 39L, 1454C
```

```
# This is a combination of 3 commits.
# The first commit's message is:
git apply --directory broken for new files

# This is the 2nd commit message:

git diff <tree>{3,}: do not reverse order of args

# This is the 3rd commit message:

test-lib: fix broken printf

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Author:    Jeff King <peff@peff.net>
#
# Not currently on any branch.
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:    builtin-apply.c
# modified:    builtin-diff.c
# modified:    t/t4013-diff-various.sh
# new file:    t/t4013/diff.diff_master_master^_side
# modified:    t/t4128-apply-root.sh
# modified:    t/test-lib.sh
#
~
~

".git/COMMIT_EDITMSG" 39L, 1454C
```

```
# This is a combination of 3 commits.
# The first commit's message is:
git apply --directory broken for new files

# This is the 2nd commit message:

git diff <tree>{3,}: do not reverse order of args

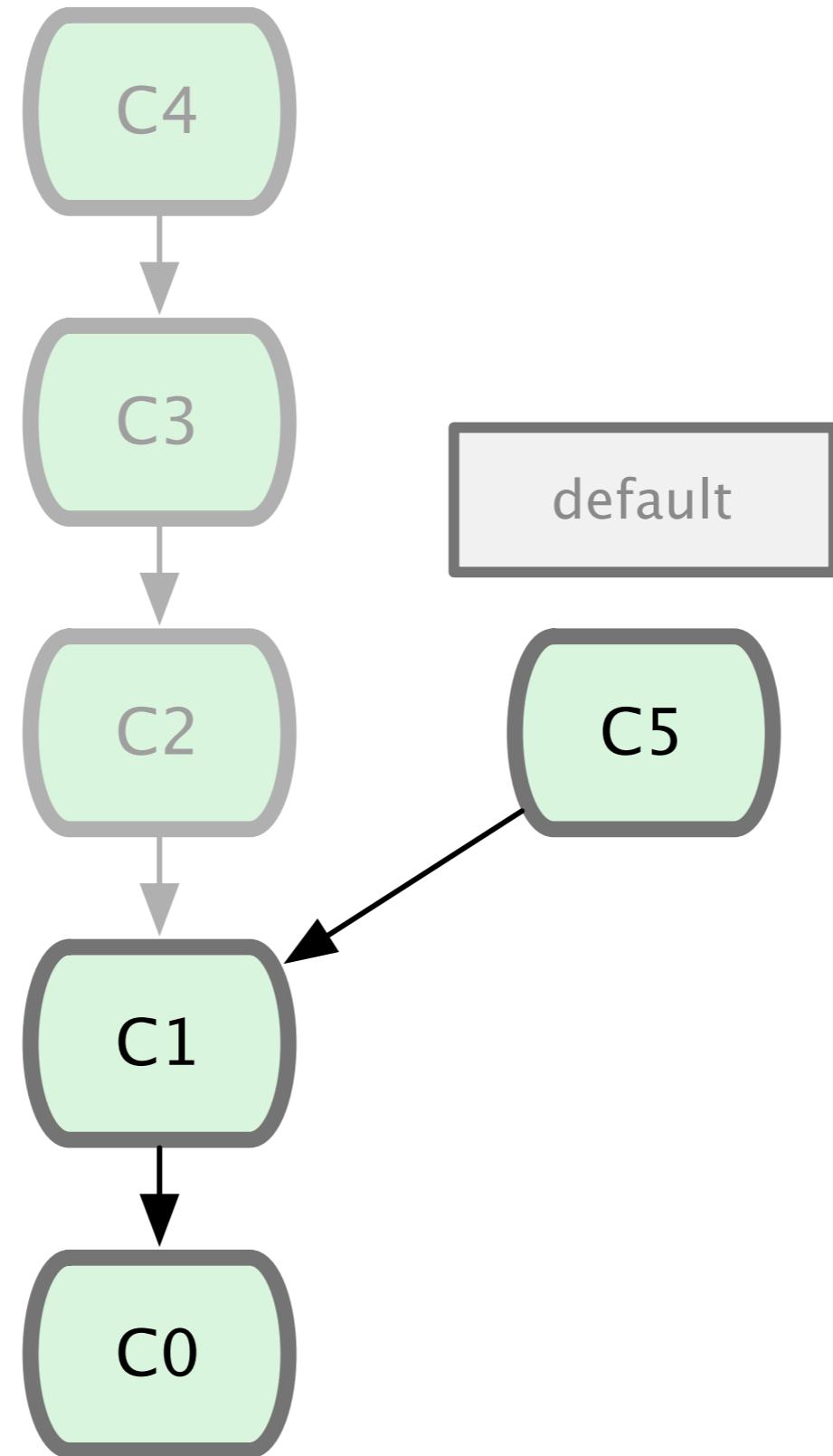
# This is the 3rd commit message:

test-lib: fix broken printf

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Author:    Jeff King <peff@peff.net>
#
# Not currently on any branch.
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:    builtin-apply.c
# modified:    builtin-diff.c
# modified:    t/t4013-diff-various.sh
# new file:   t/t4013/diff.diff_master_master^_side
# modified:    t/t4128-apply-root.sh
# modified:    t/test-lib.sh
#
~
~

".git/COMMIT_EDITMSG" 39L, 1454C
```

```
# This is a combination of 3 commits.  
# The first commit's message is:  
git apply --directory broken for new files  
  
# This is the 2nd commit message:  
  
git diff <tree>{3,}: do not reverse order of args  
  
# This is the 3rd commit message:  
  
test-lib: fix broken printf  
  
# Please enter the commit message for your changes. Lines starting  
# with '#' will be ignored, and an empty message aborts the commit.  
#  
# Author: Jeff King <peff@peff.net>  
#  
# Not currently on any branch.  
# Changes to be committed:  
#   (use "git reset HEAD <file>..." to unstage)  
#  
# modified: builtin-apply.c  
# modified: builtin-diff.c  
# modified: t/t4013-diff-various.sh  
# new file: t/t4013/diff.diff_master_master^_side  
# modified: t/t4128-apply-root.sh  
# modified: t/test-lib.sh  
#  
~  
~  
".git/COMMIT_EDITMSG" 39L, 1454C
```

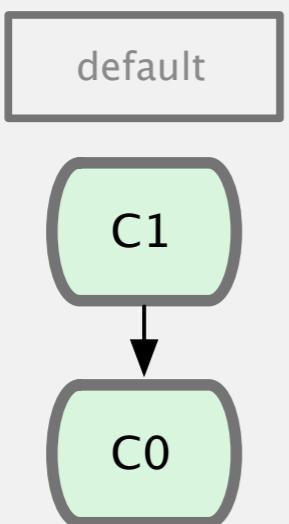


default

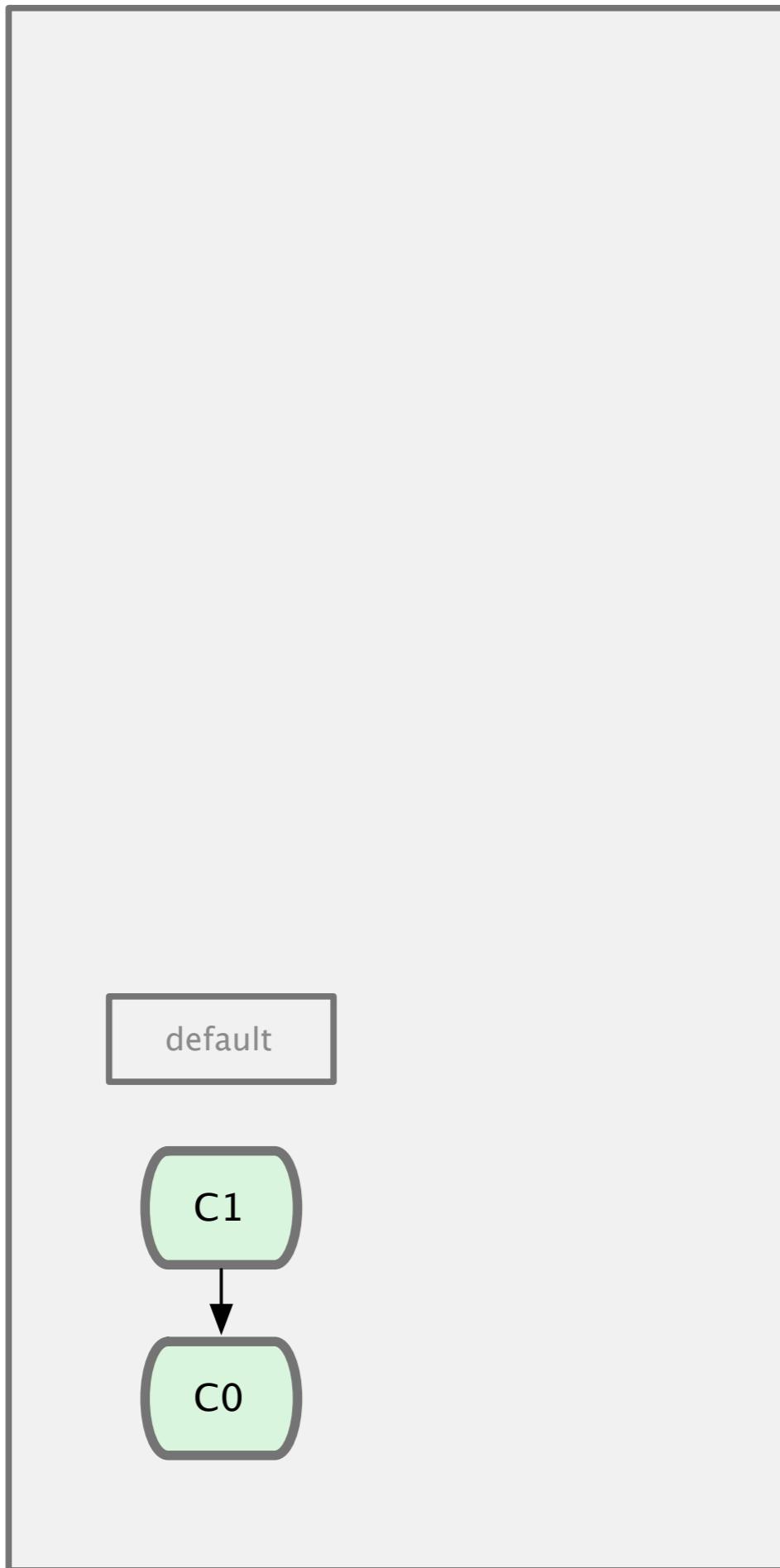
The Perils

scott

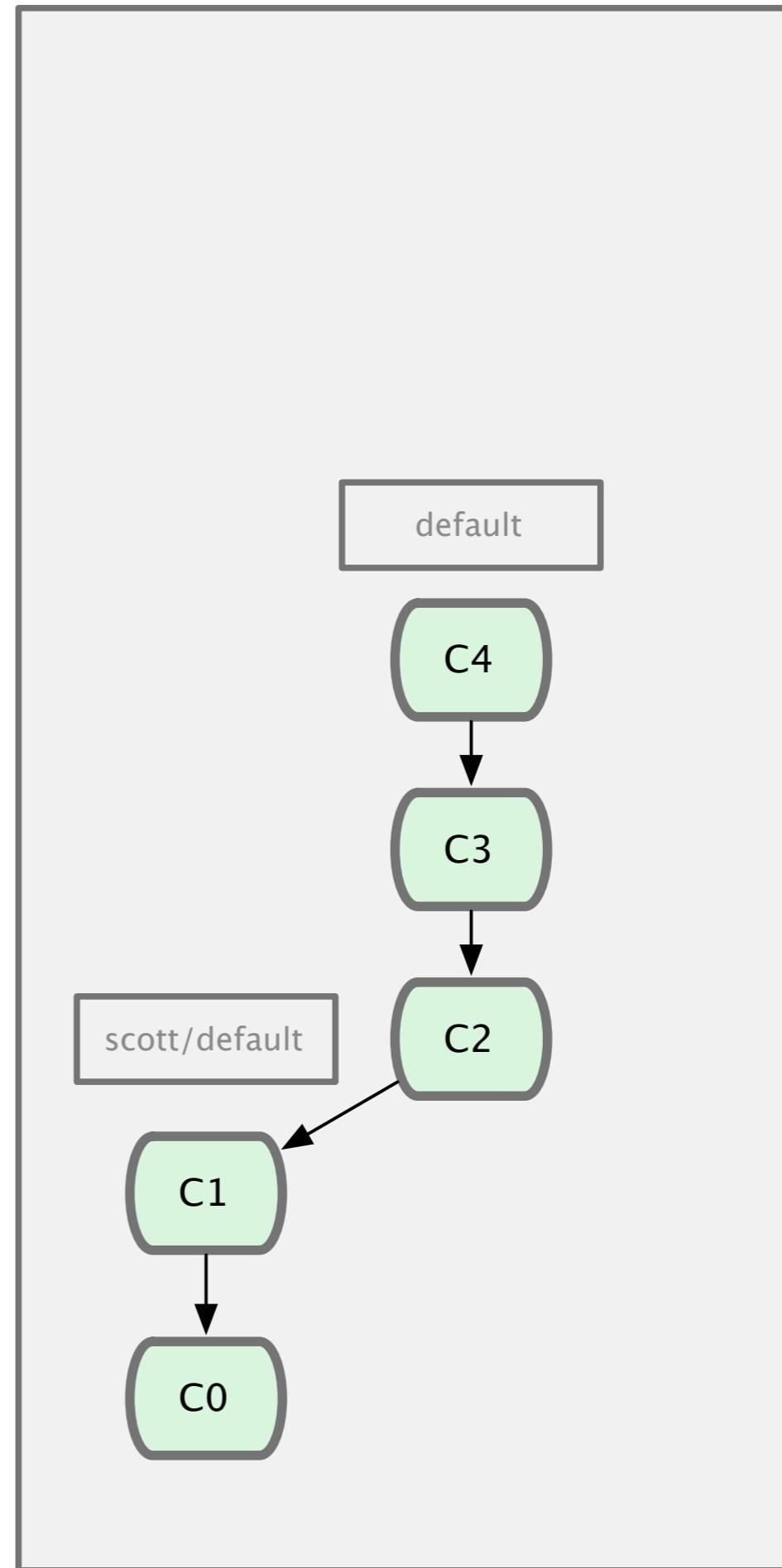
jessica



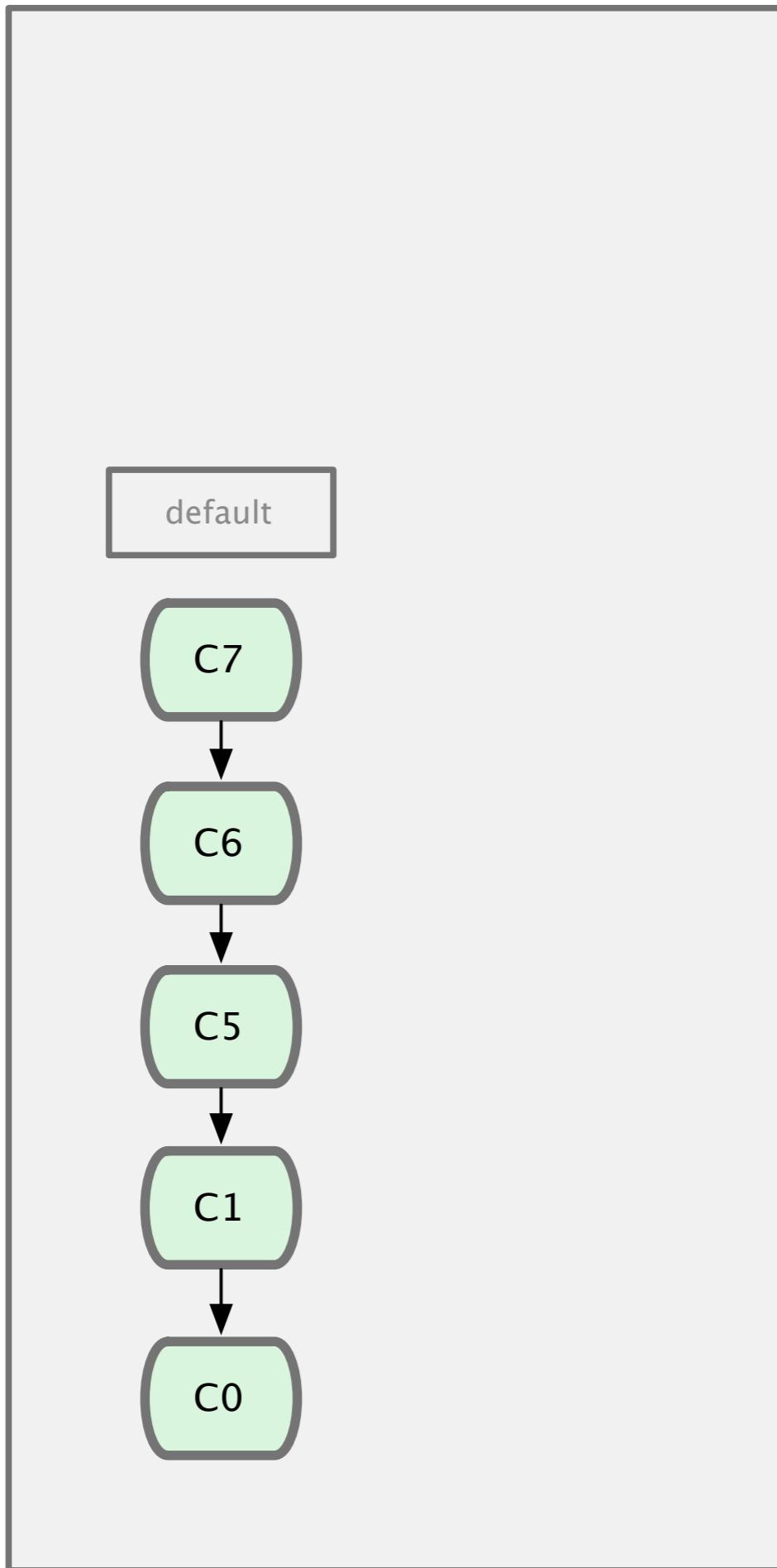
scott



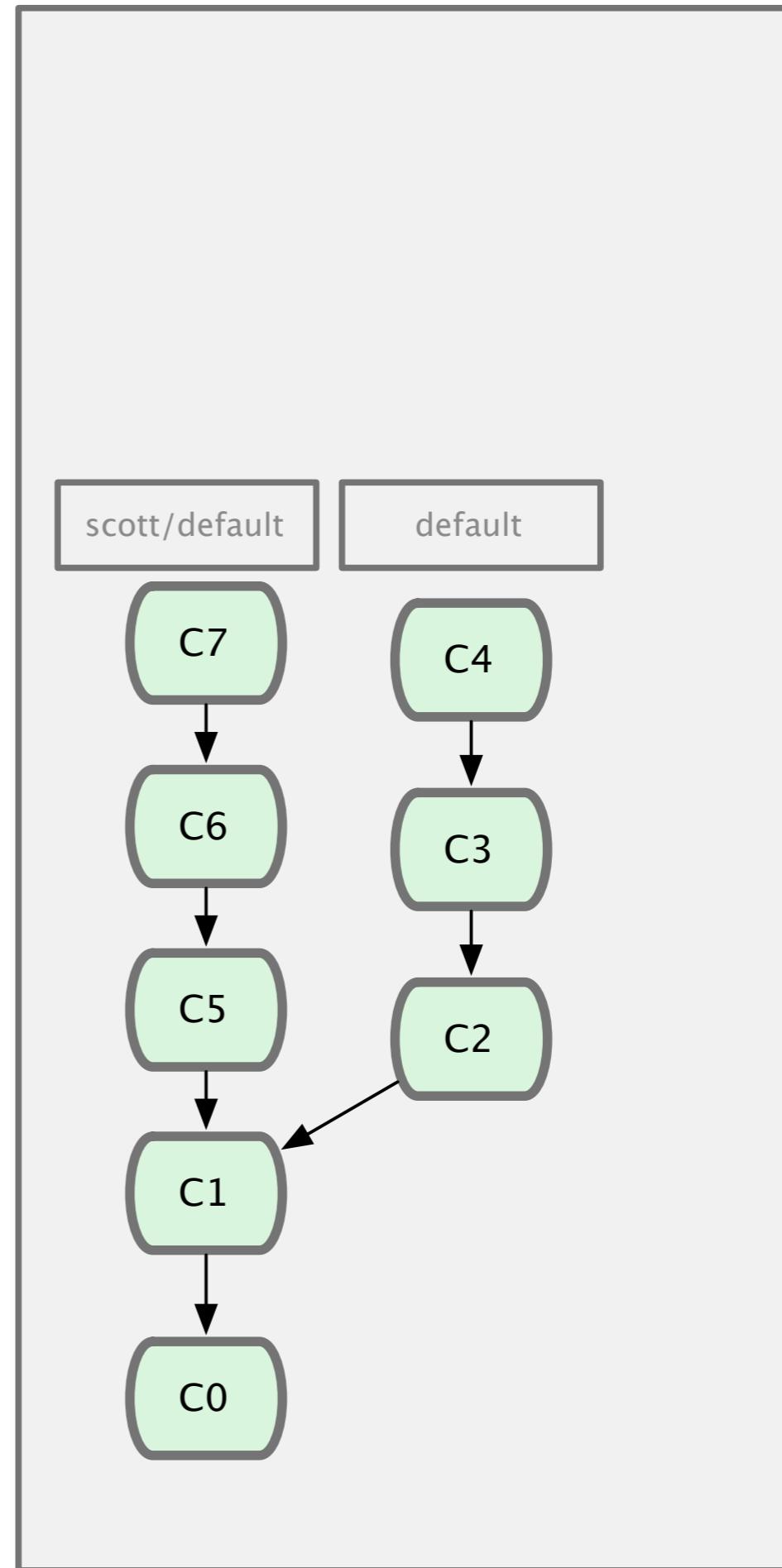
jessica



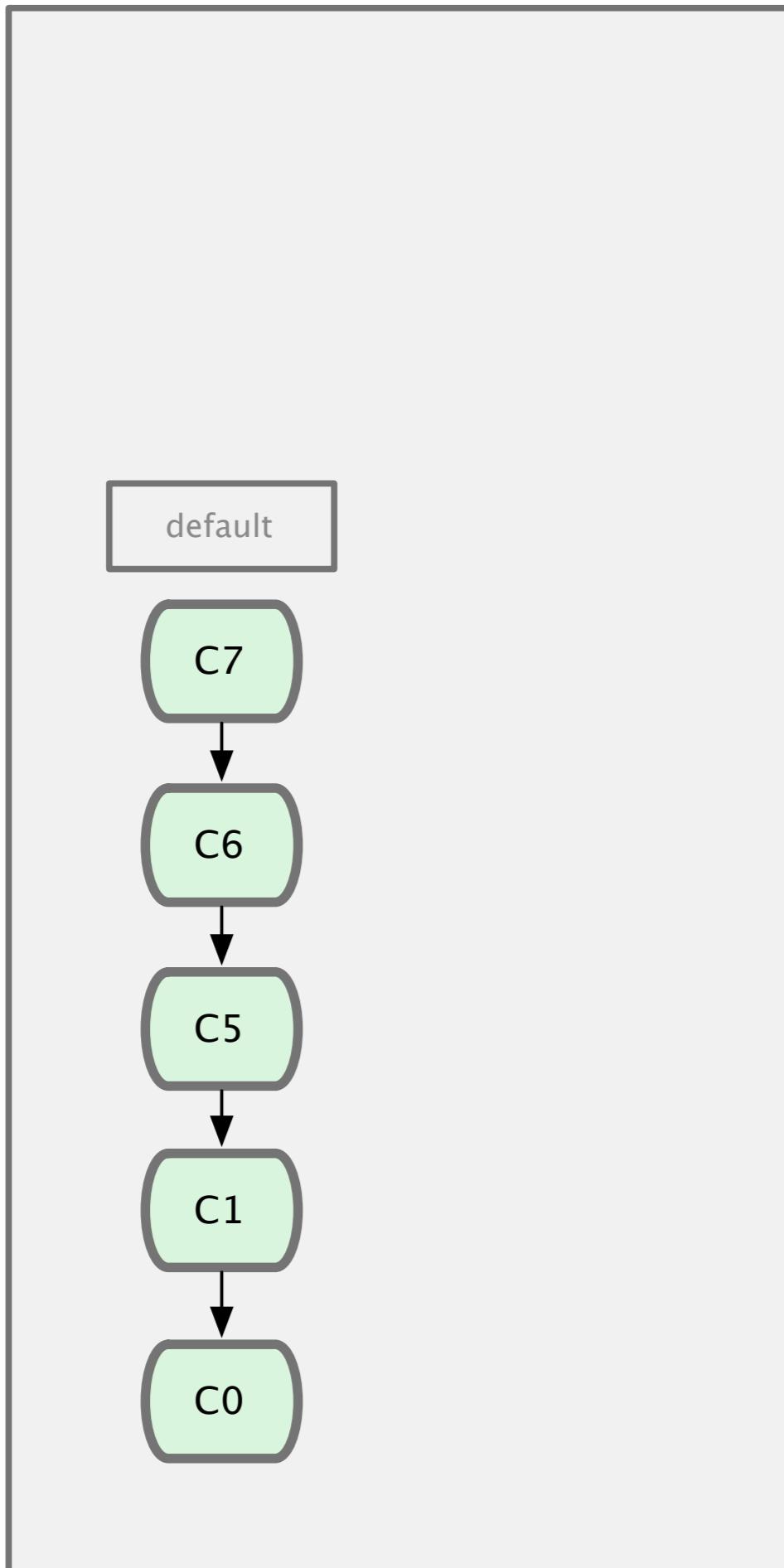
scott



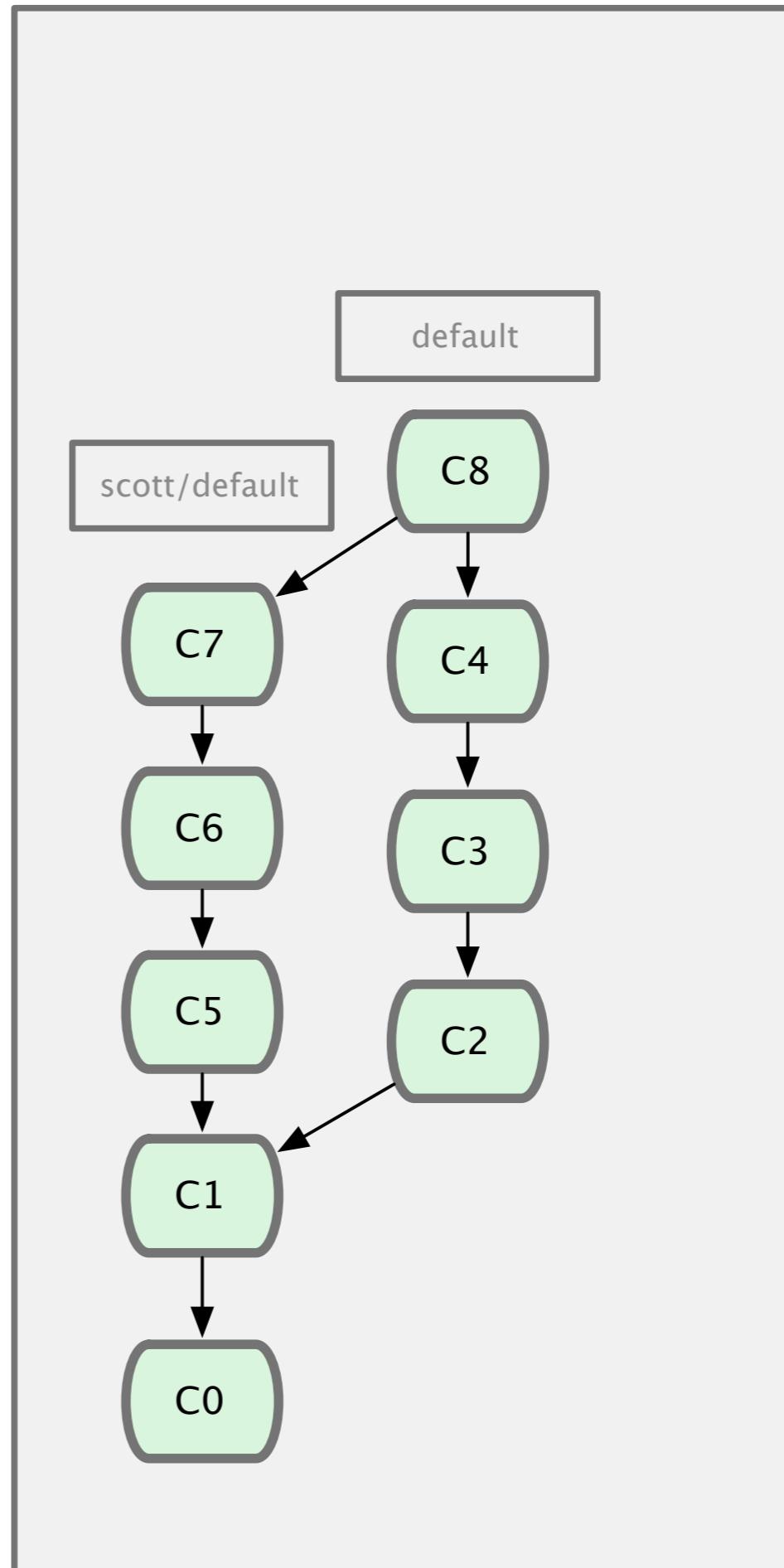
jessica



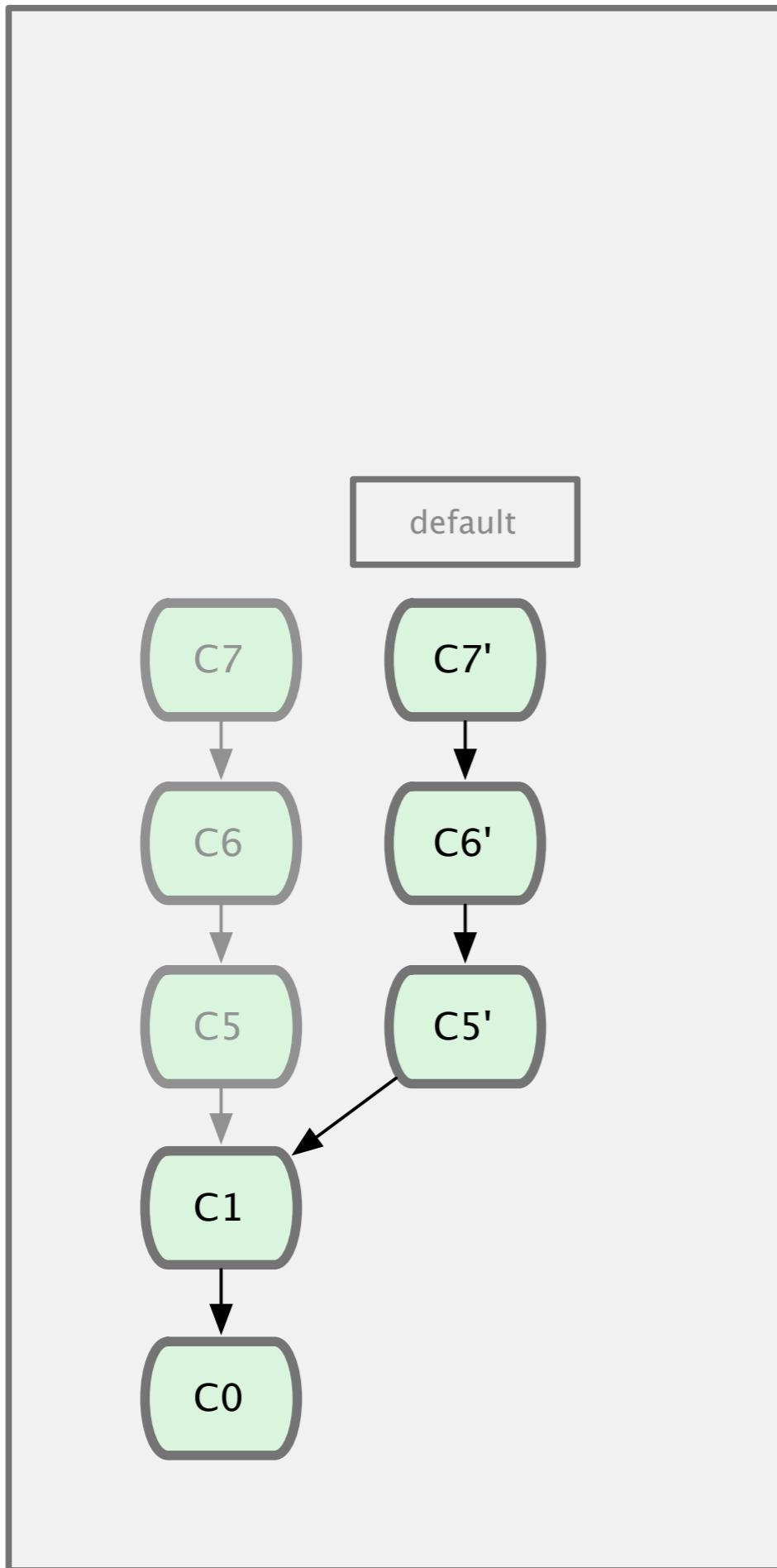
scott



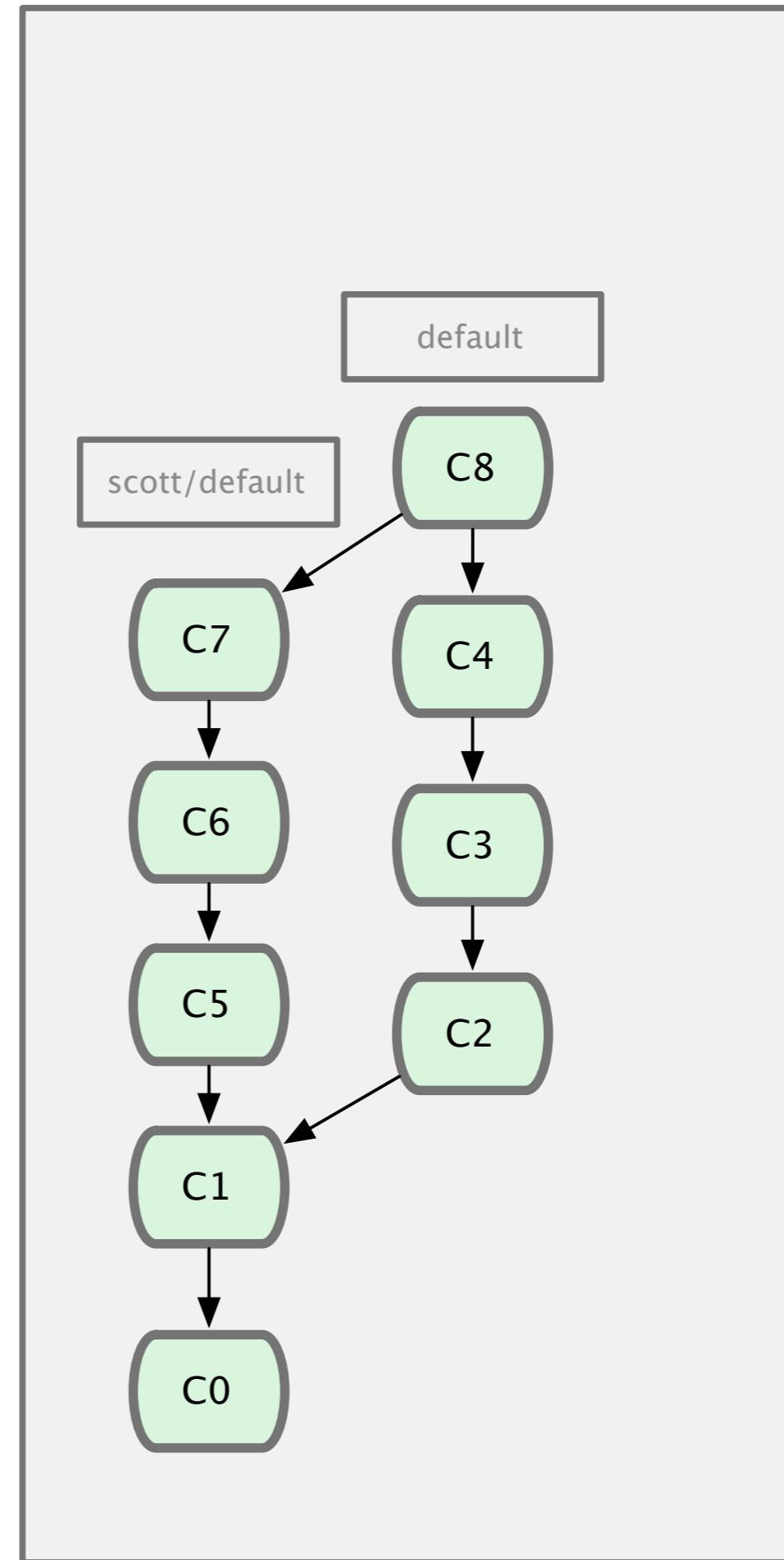
jessica



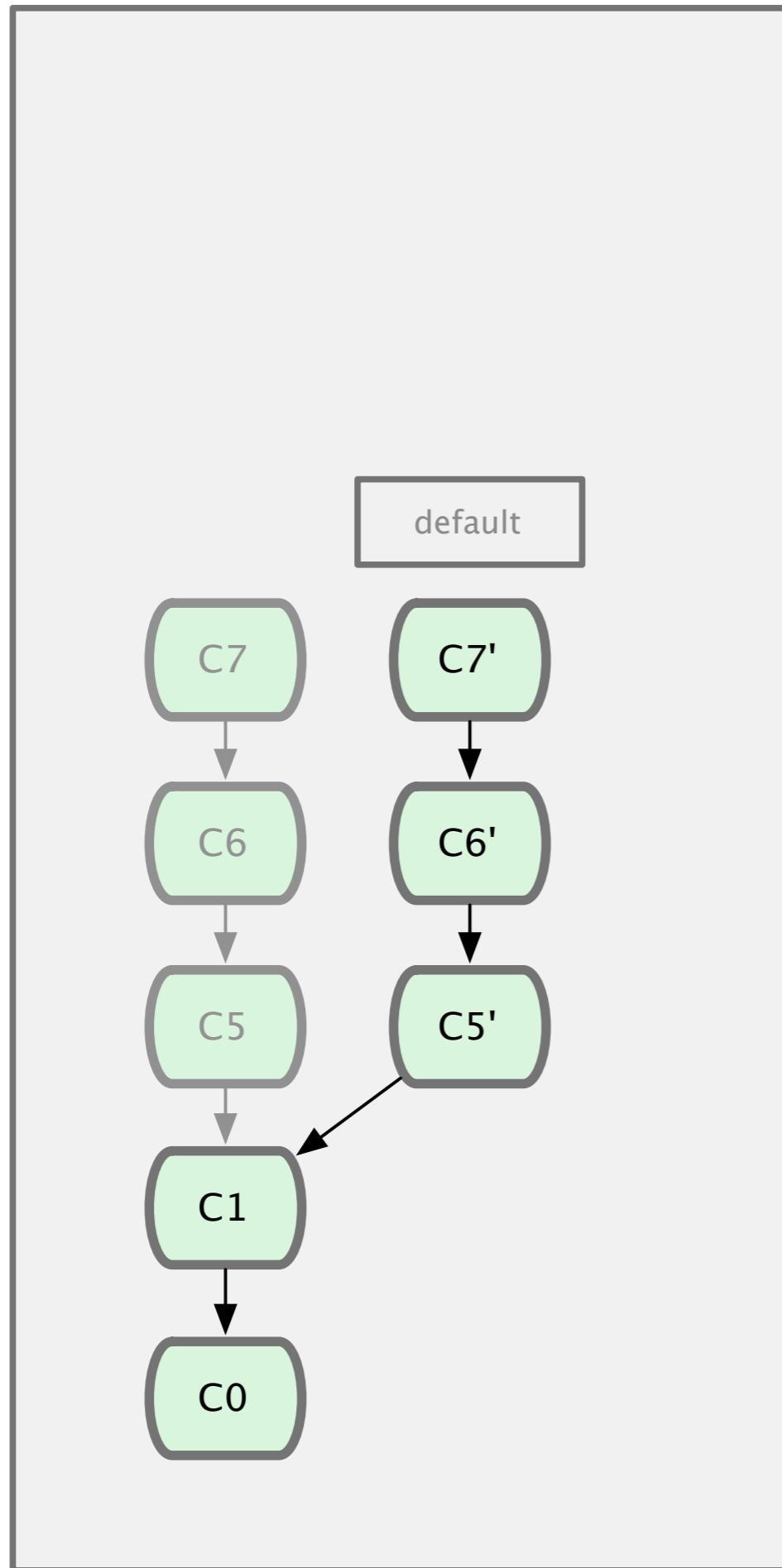
scott



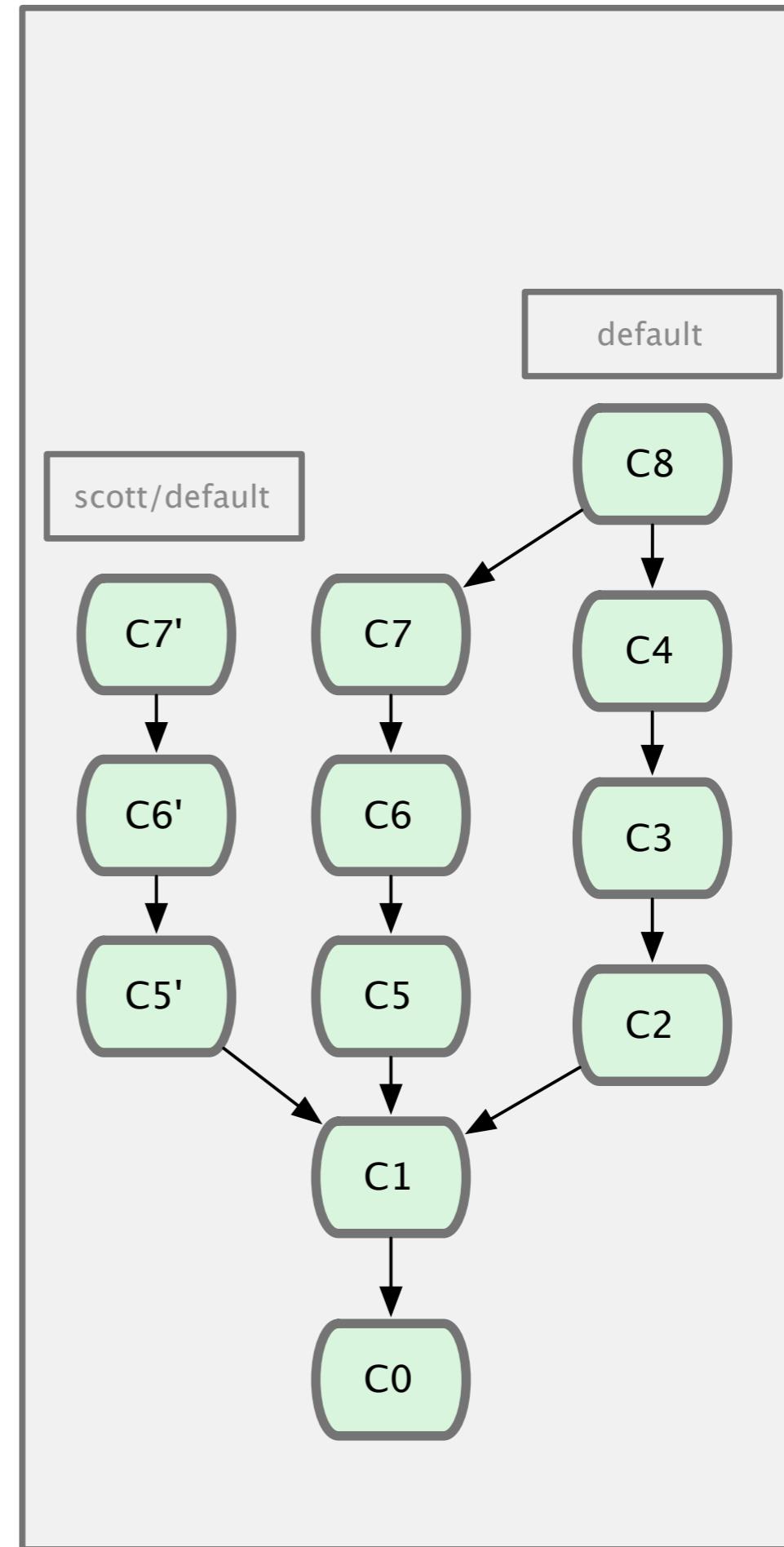
jessica



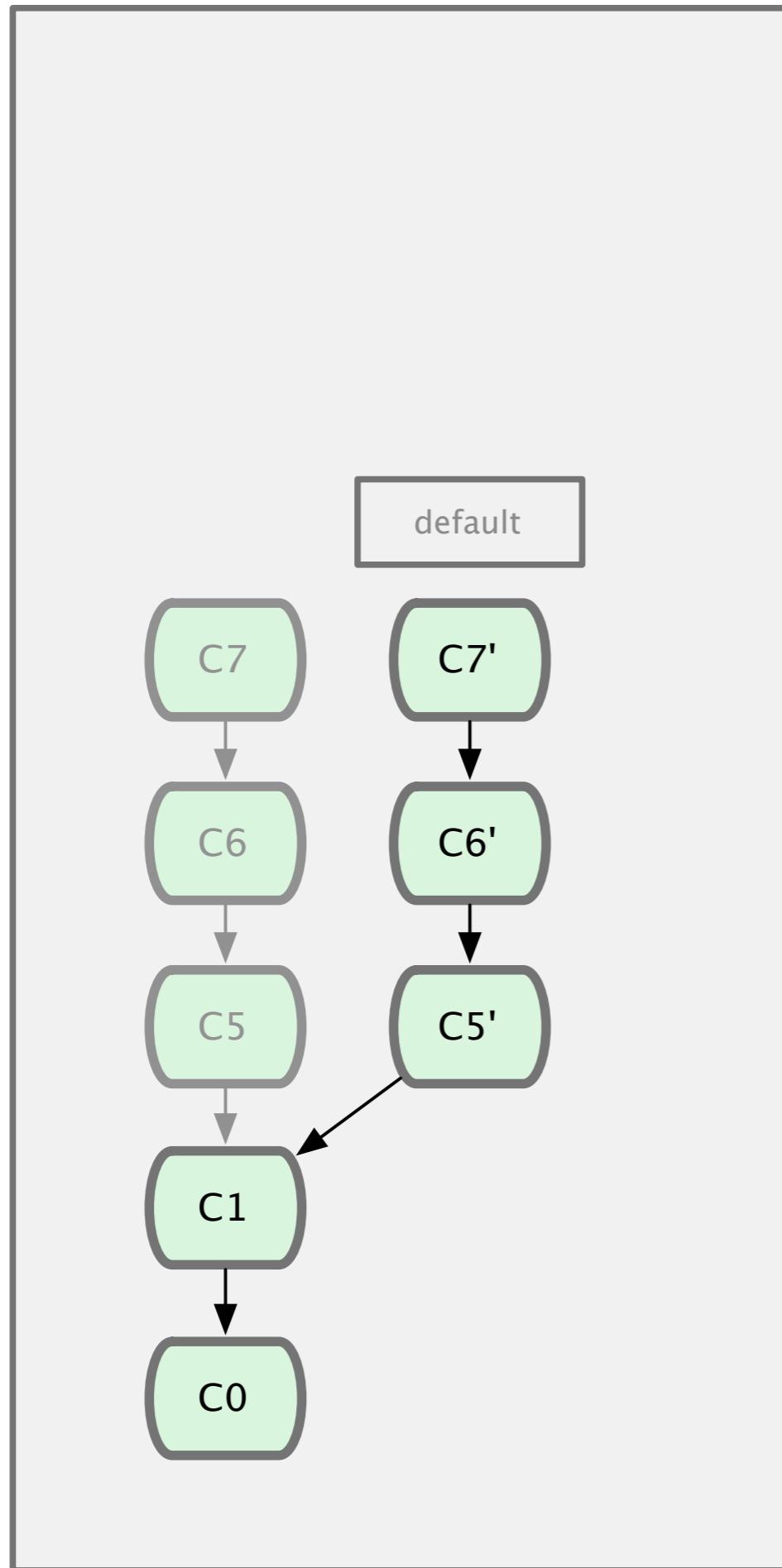
scott



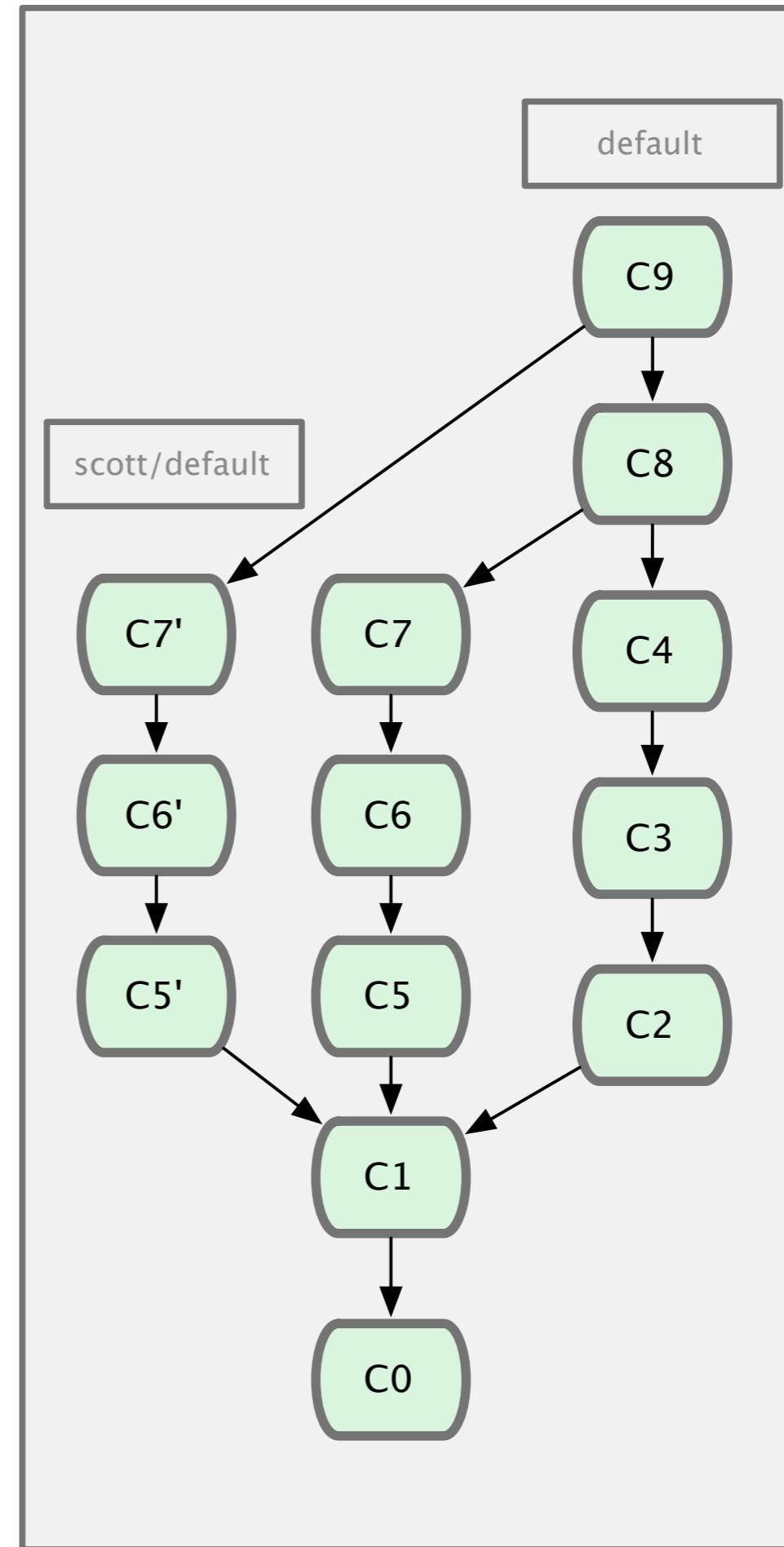
jessica



scott



jessica



Filter Branch

History Revision on Steroids

remove all instances of a
file from every commit

```
git filter-branch  
--tree-filter 'rm -f filename'  
HEAD
```

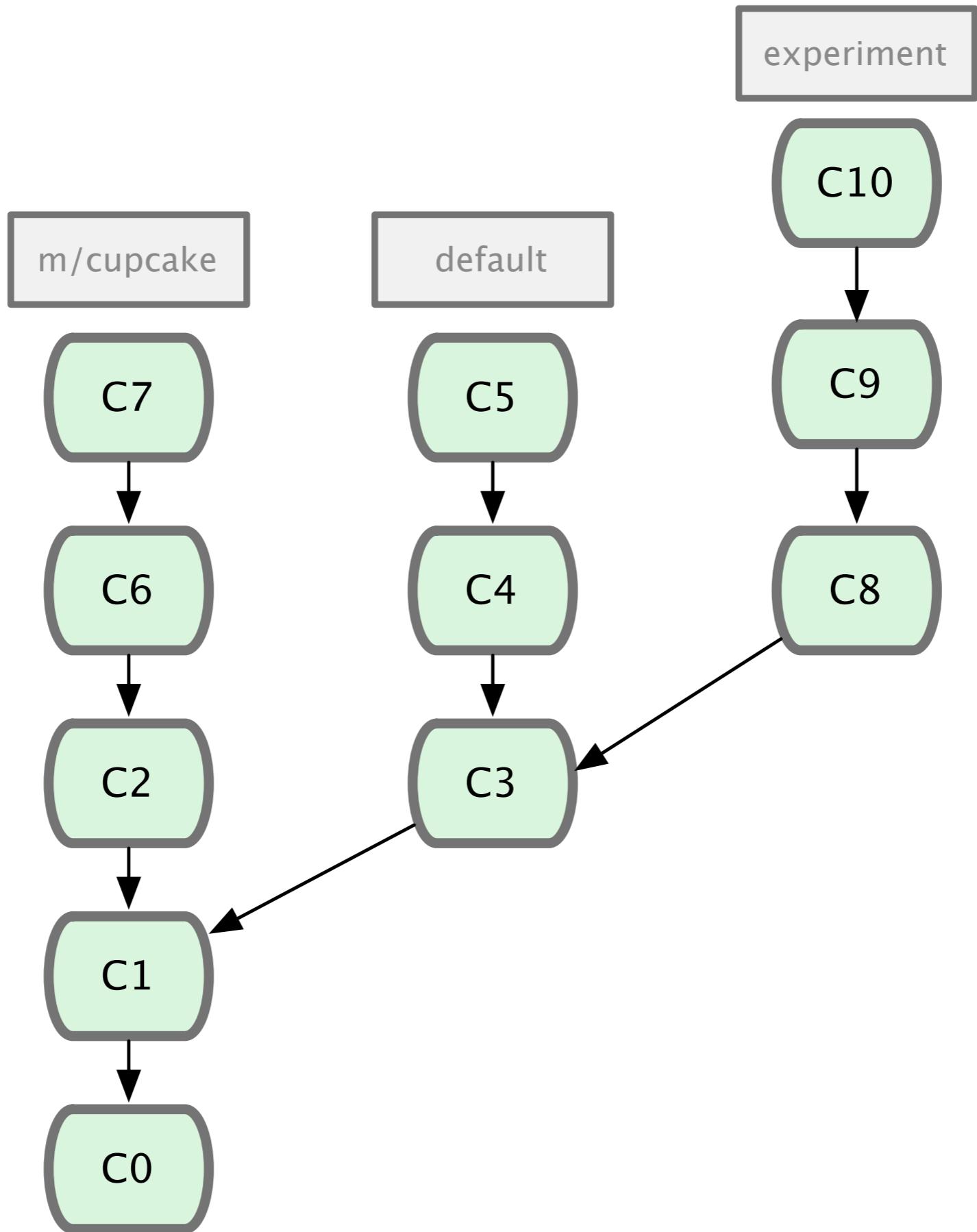
change your email
in all commits

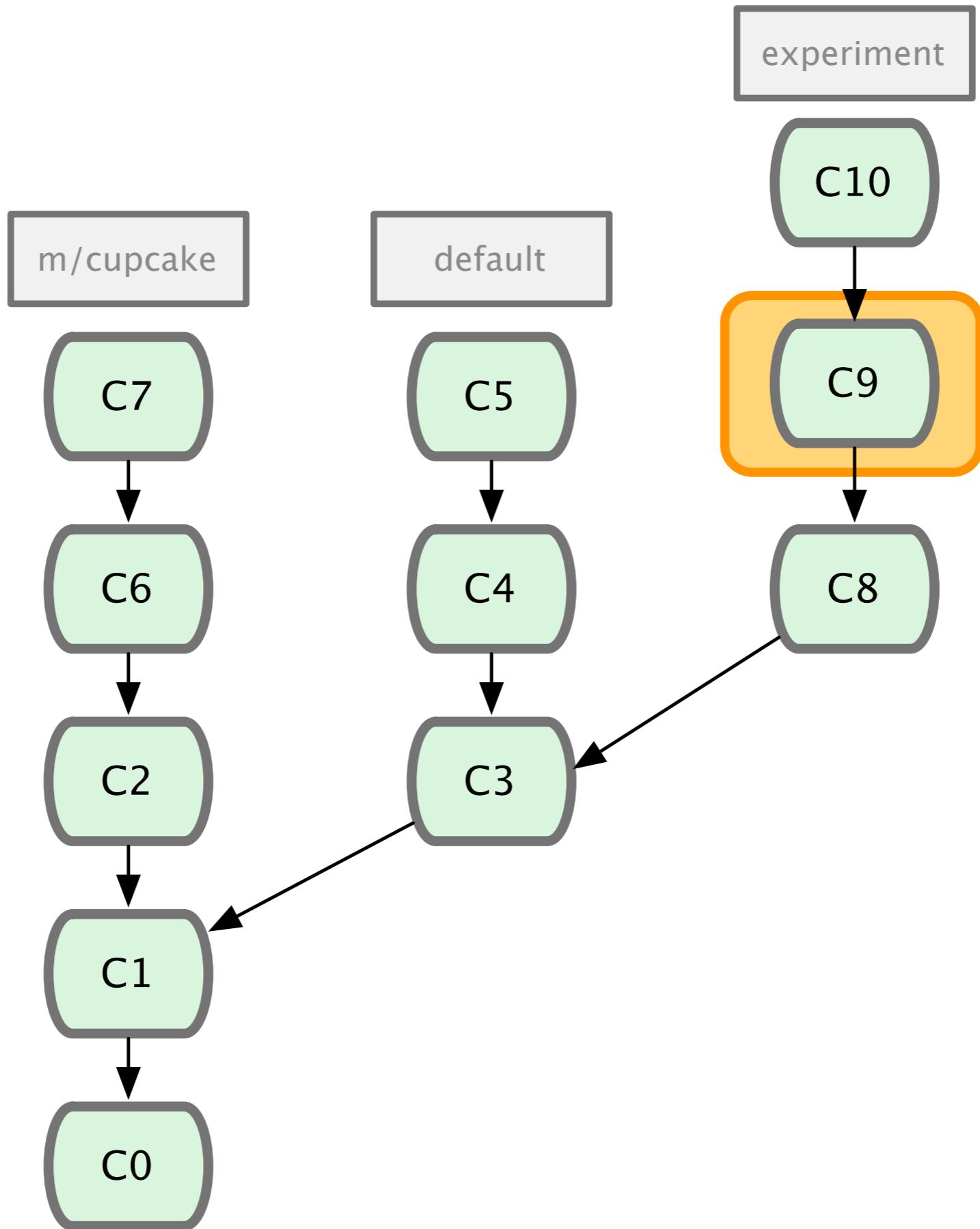
```
git filter-branch --env-filter  
"export GIT_AUTHOR_EMAIL=you@email.com"  
HEAD
```

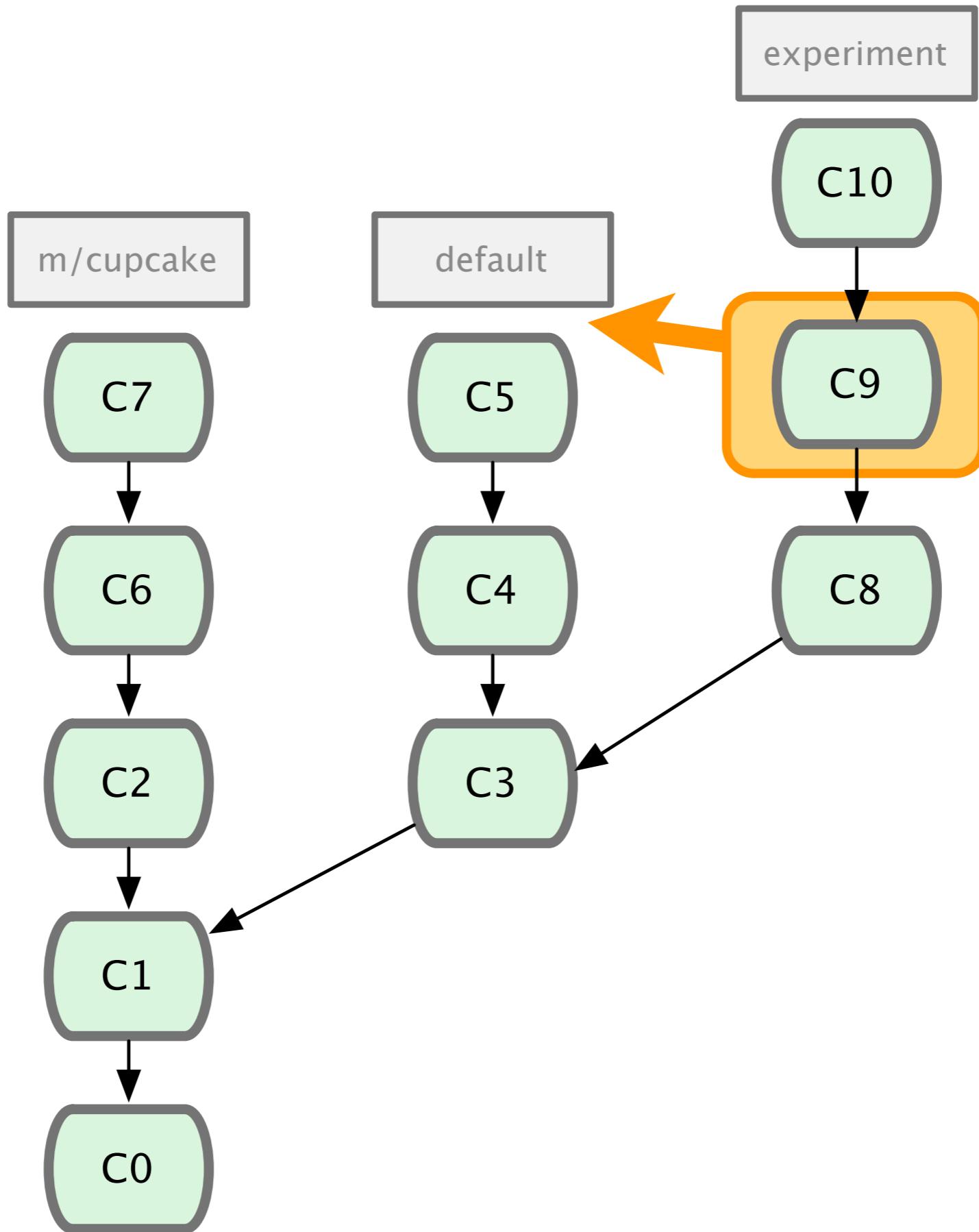
```
git filter-branch --env-filter  
"export GIT_AUTHOR_EMAIL=you@email.com"  
origin/master..HEAD
```

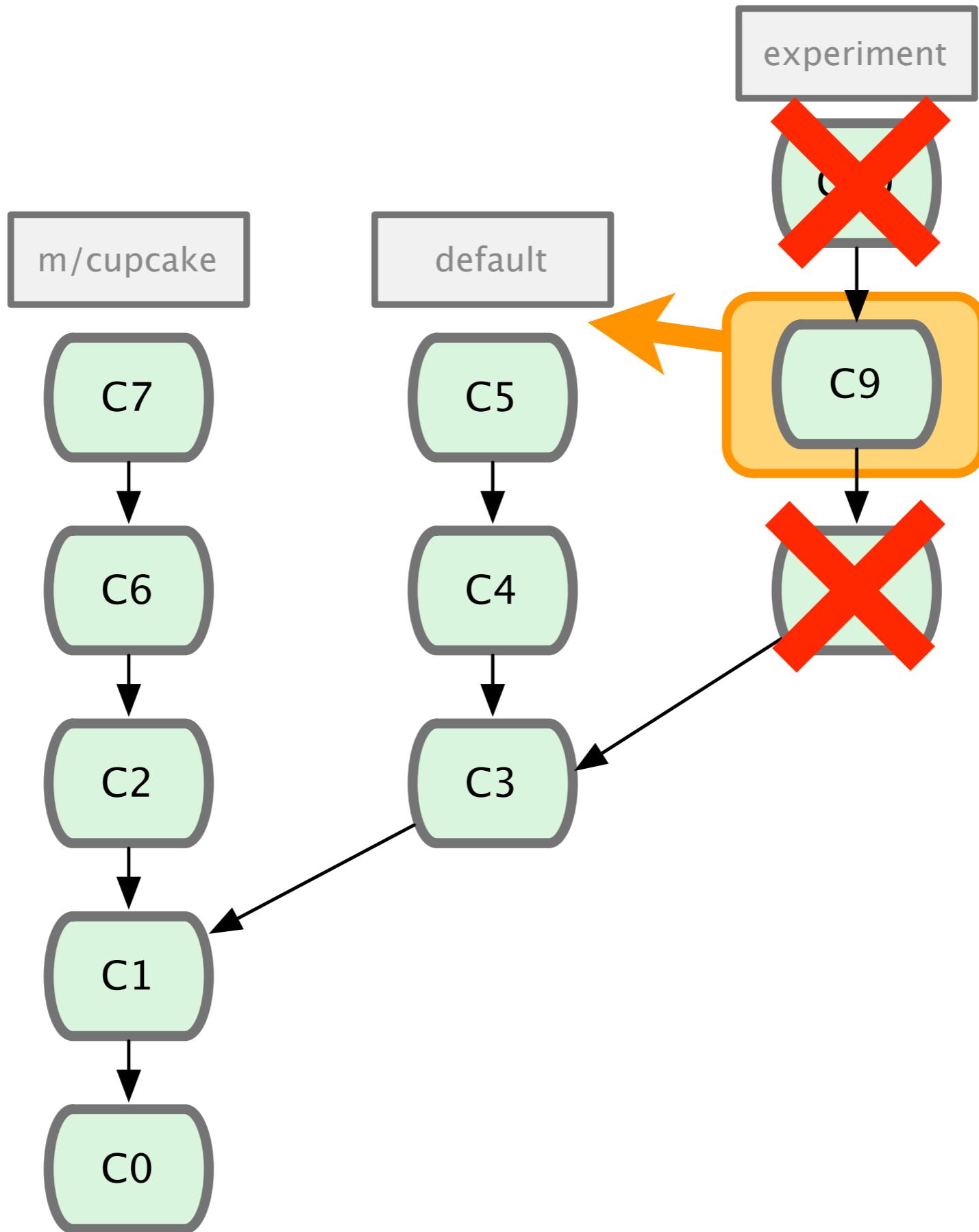
Cherry Picking

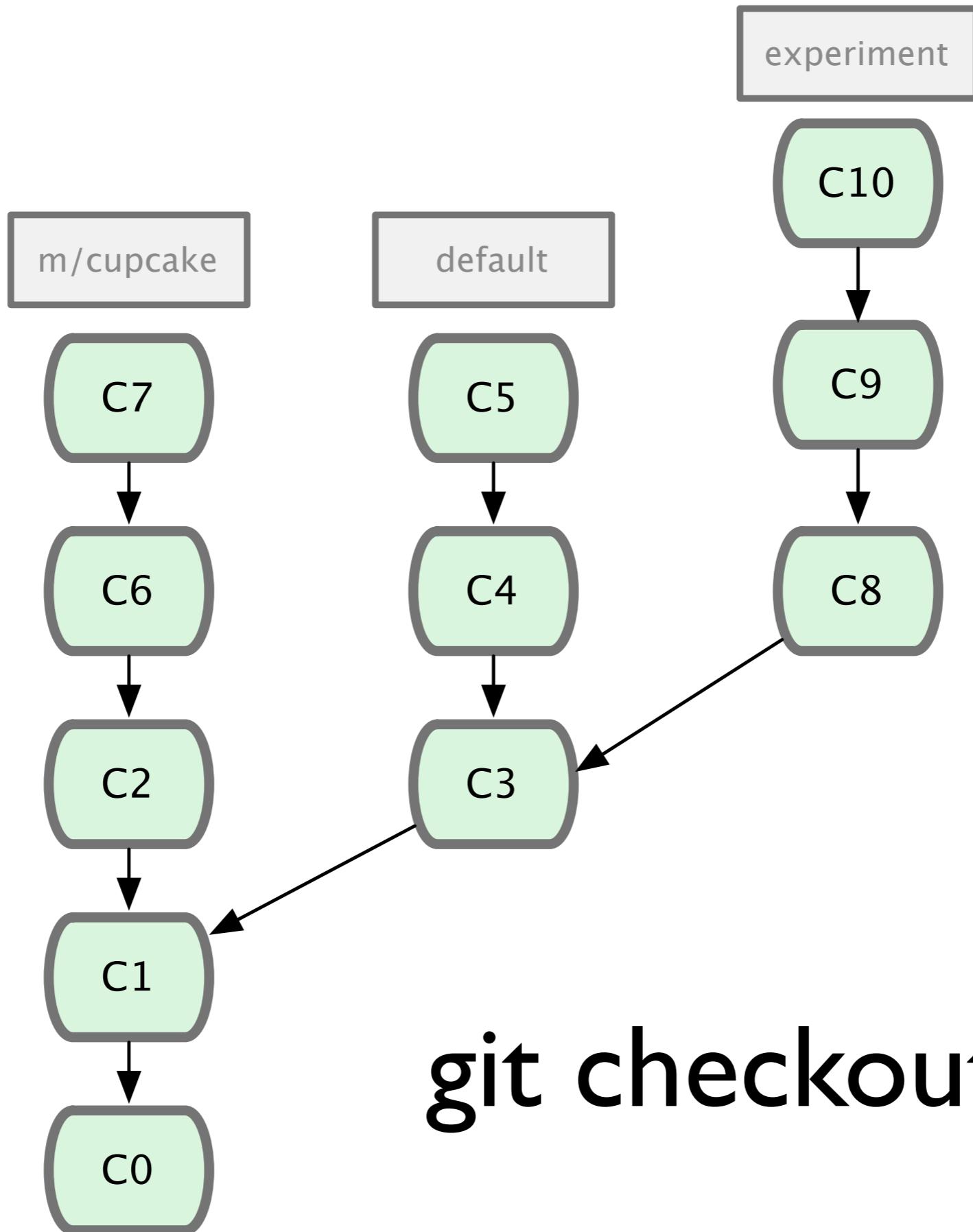
git cherry-pick 3f40a



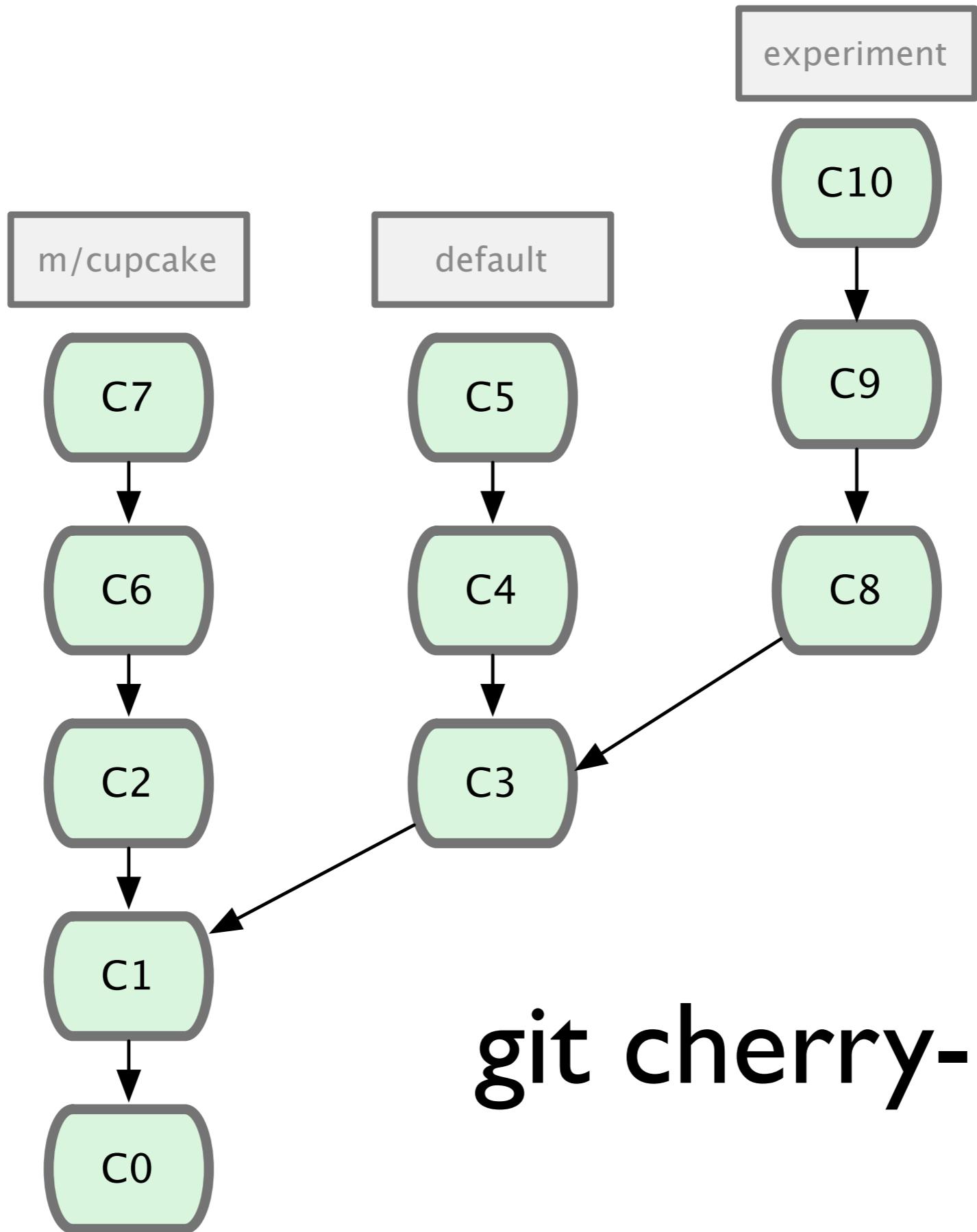




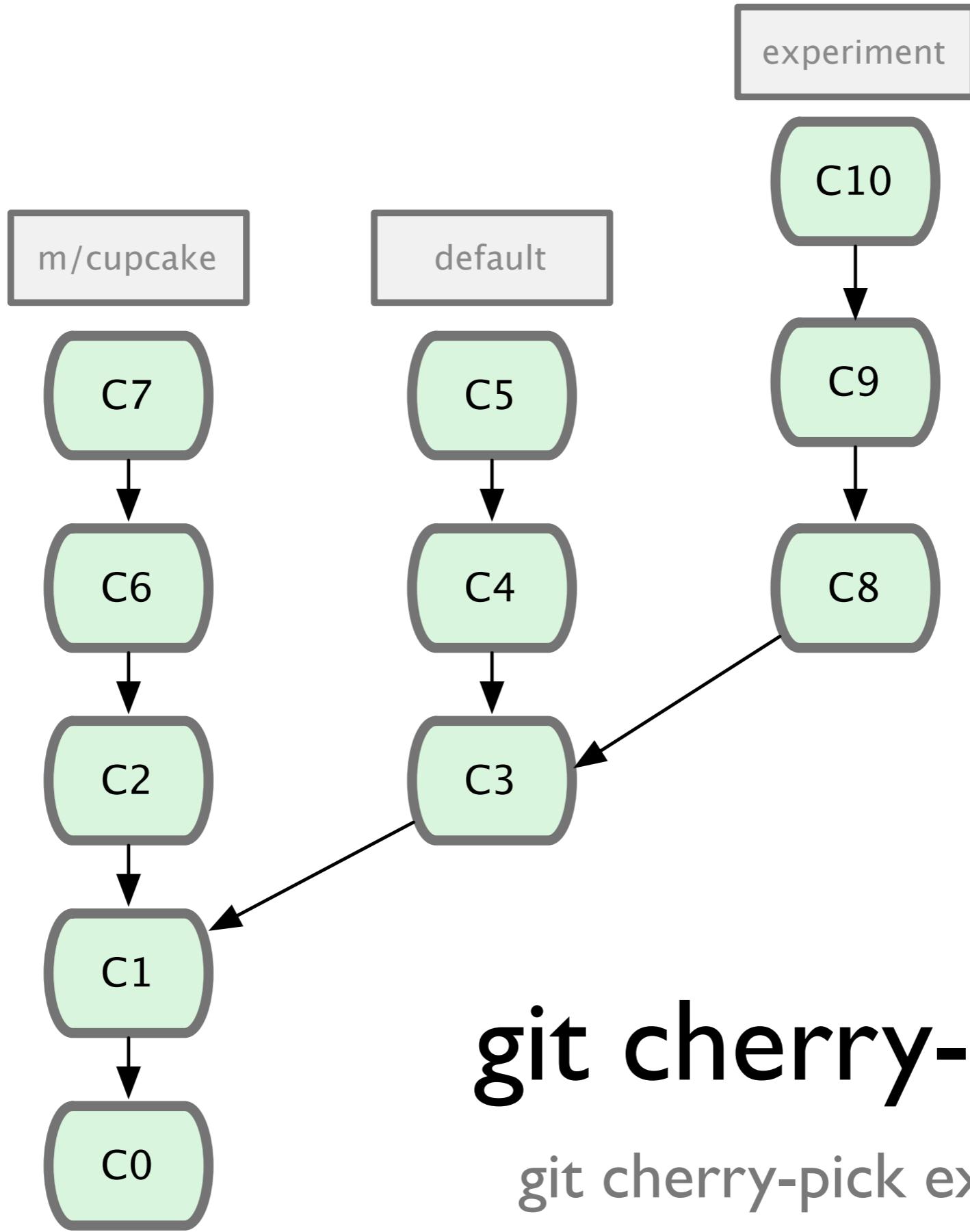




git checkout default

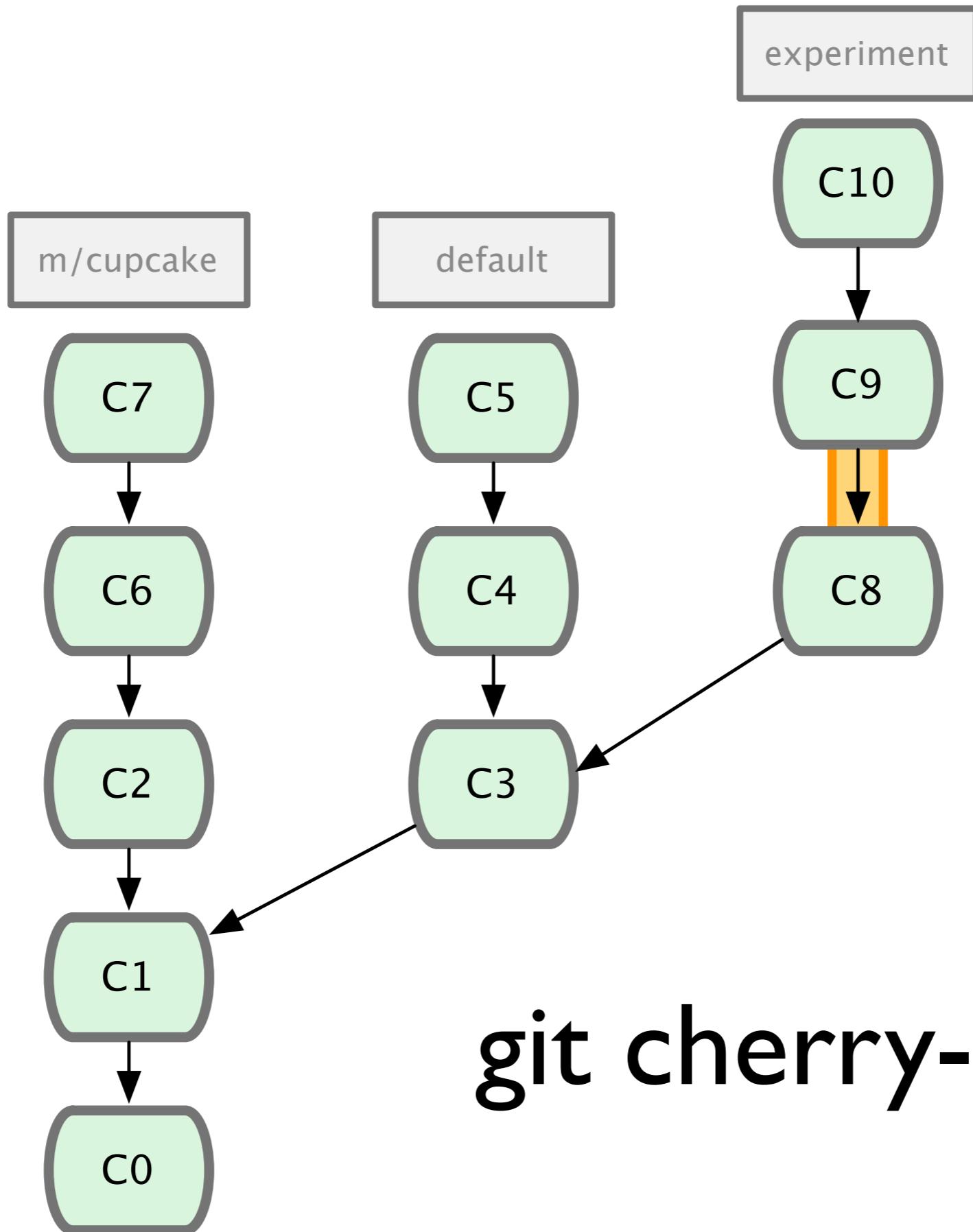


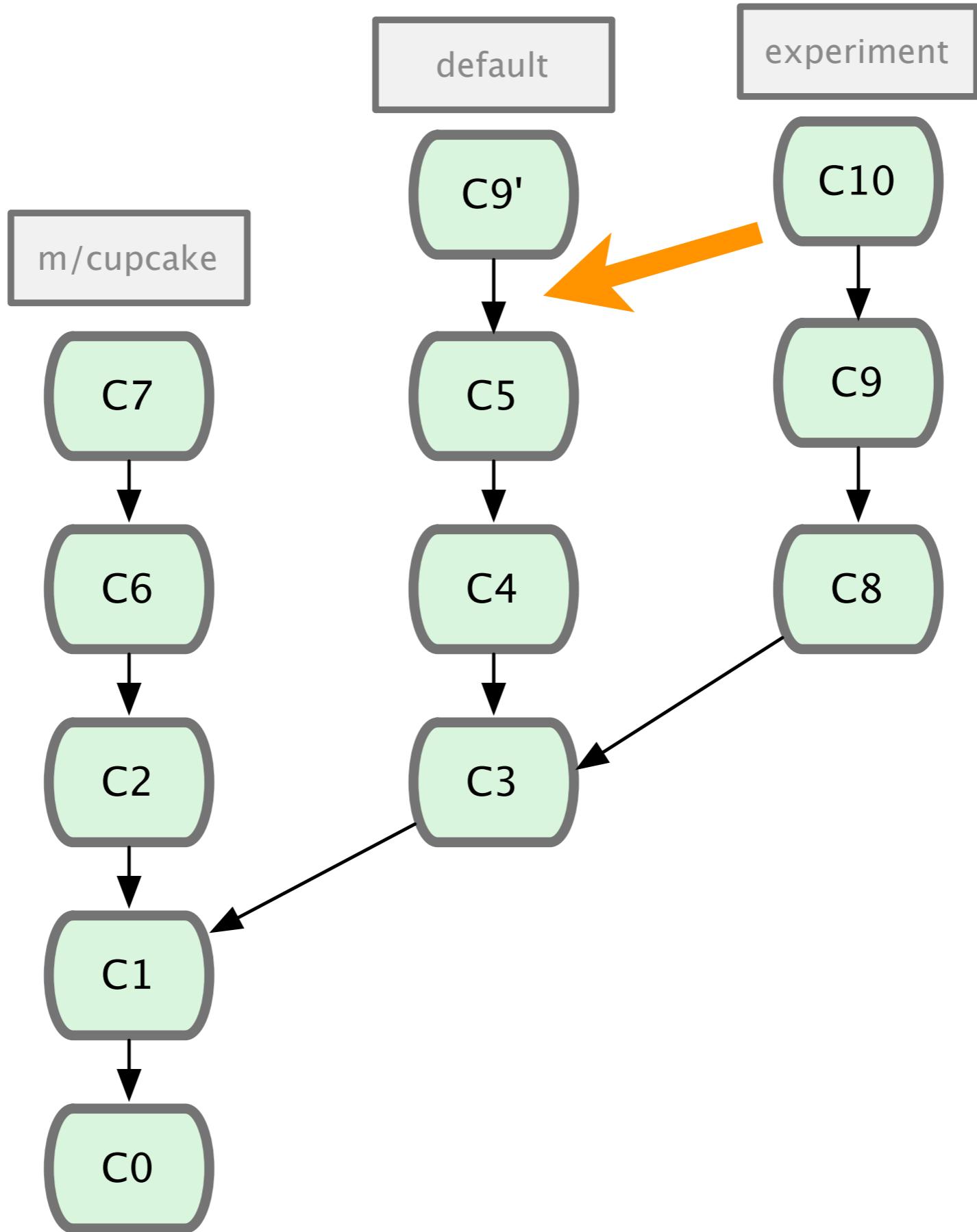
git cherry-pick c9

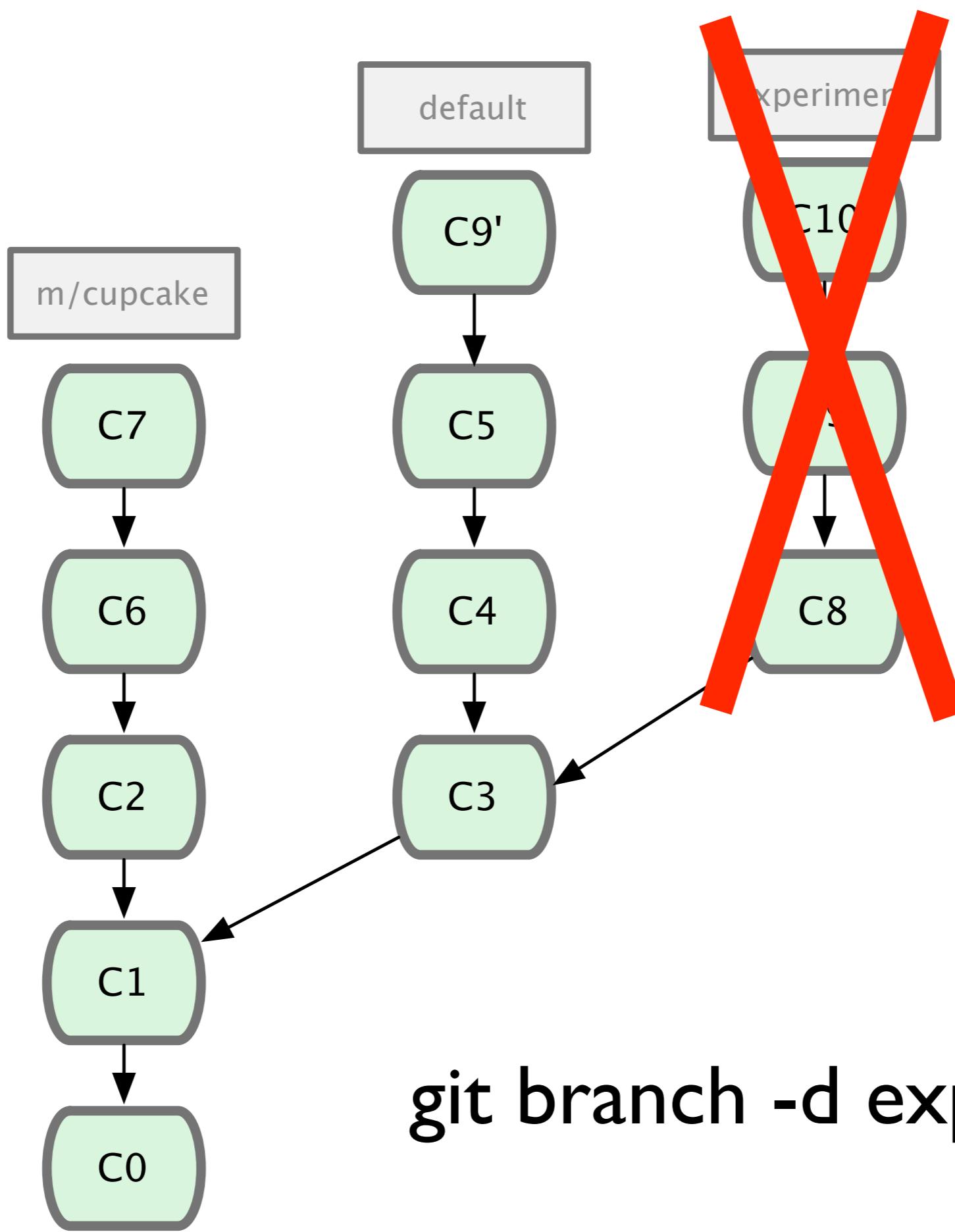


git cherry-pick c9

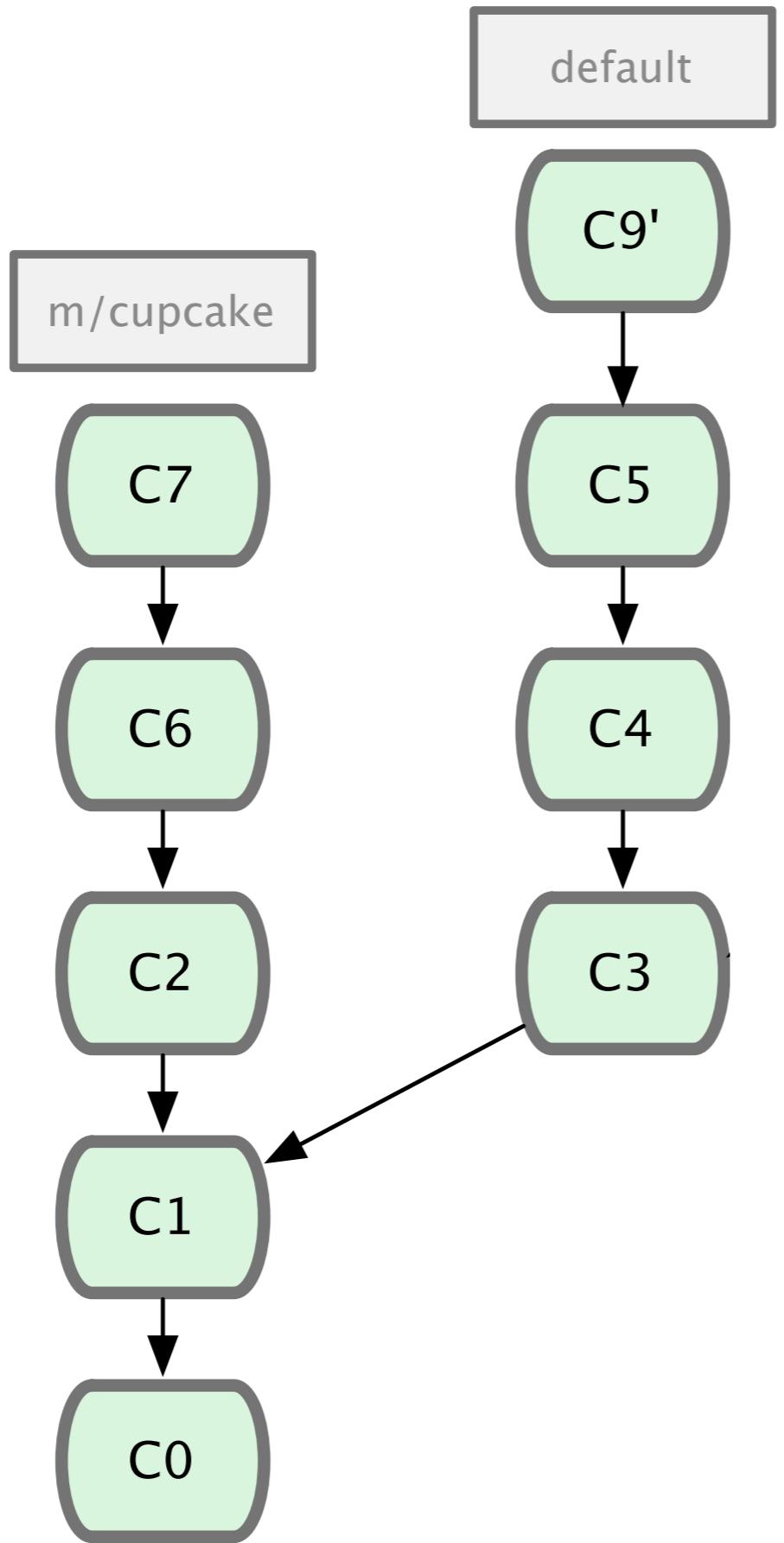
git cherry-pick experiment^







git branch -d experiment



Undo

Single File

git checkout [commit] file

```
$ git checkout master@{yesterday}  
system/core/netcfg/netcfg.c
```

Entire Project

git reset

```
$ git status
```

```
$ git status
# Not currently on any branch.
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:   src/com/android/launcher/Workspace.java
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in
working directory)
#
# modified:   src/com/android/launcher/LiveFolder.java
#
```

```
$ git status
# Not currently on any branch.
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:   src/com/android/launcher/Workspace.java
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in
working directory)
#
# modified:   src/com/android/launcher/LiveFolder.java
#
```

```
$ git status
# Not currently on any branch.
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:   src/com/android/launcher/Workspace.java
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in
working directory)
#
# modified:   src/com/android/launcher/LiveFolder.java
#
```

by default, reset changes
the staging area

```
$ git status
# Not currently on any branch.
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:   src/com/android/launcher/Workspace.java
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in
working directory)
#
# modified:   src/com/android/launcher/LiveFolder.java
#
```

```
$ git reset --hard
```

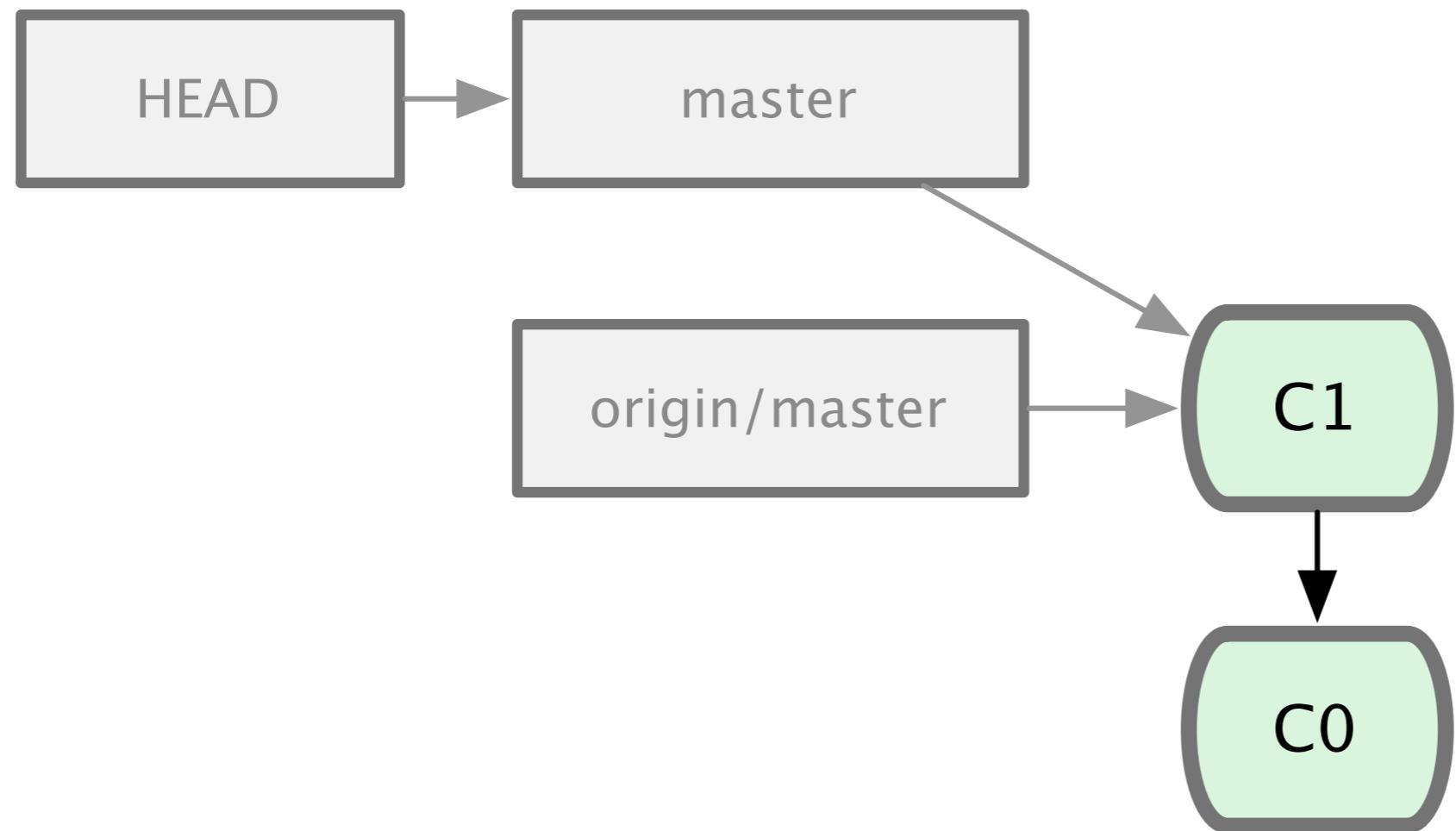
--hard changes stage
and working directory

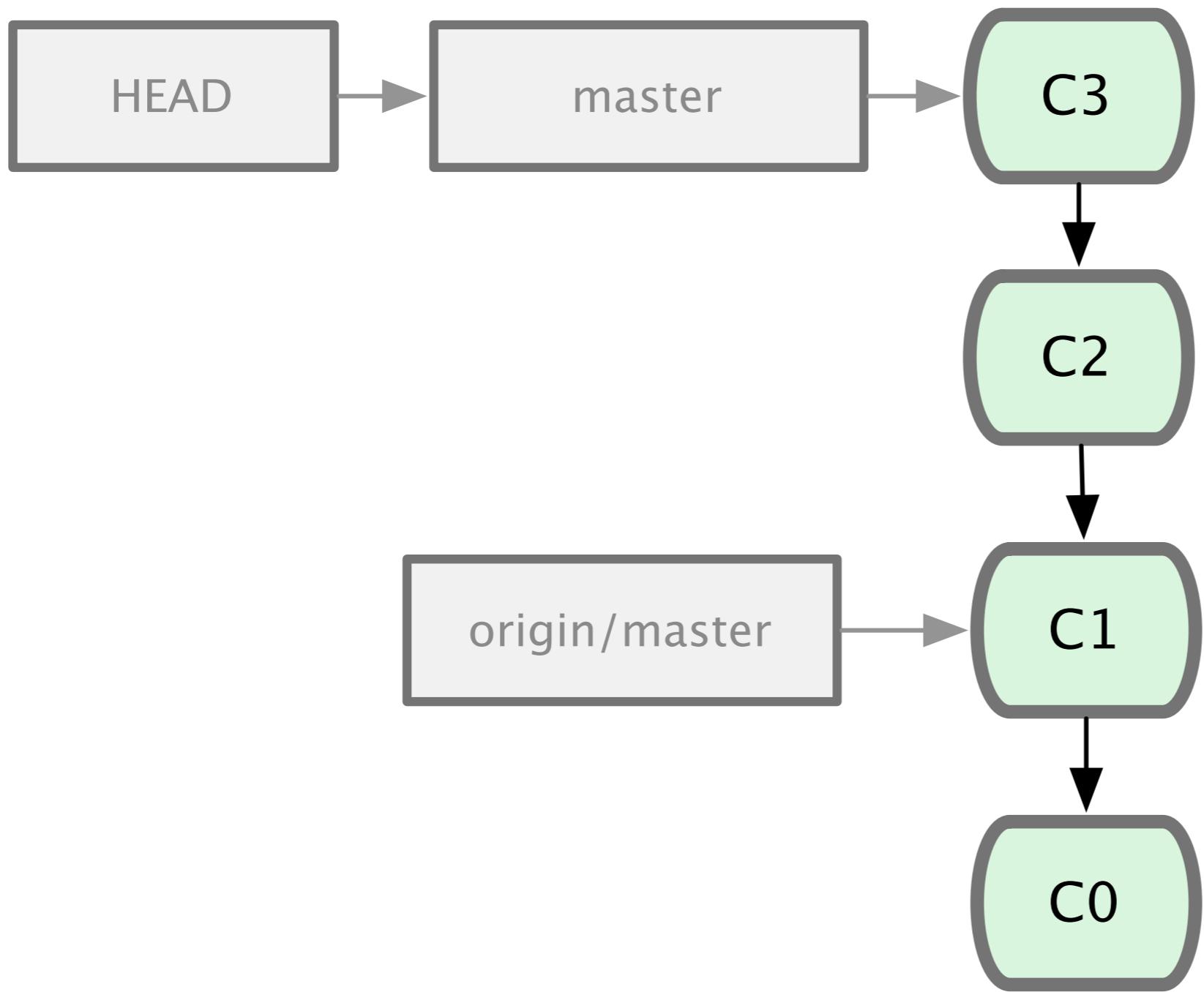
you've made commits,
but want to move them
to a topic branch

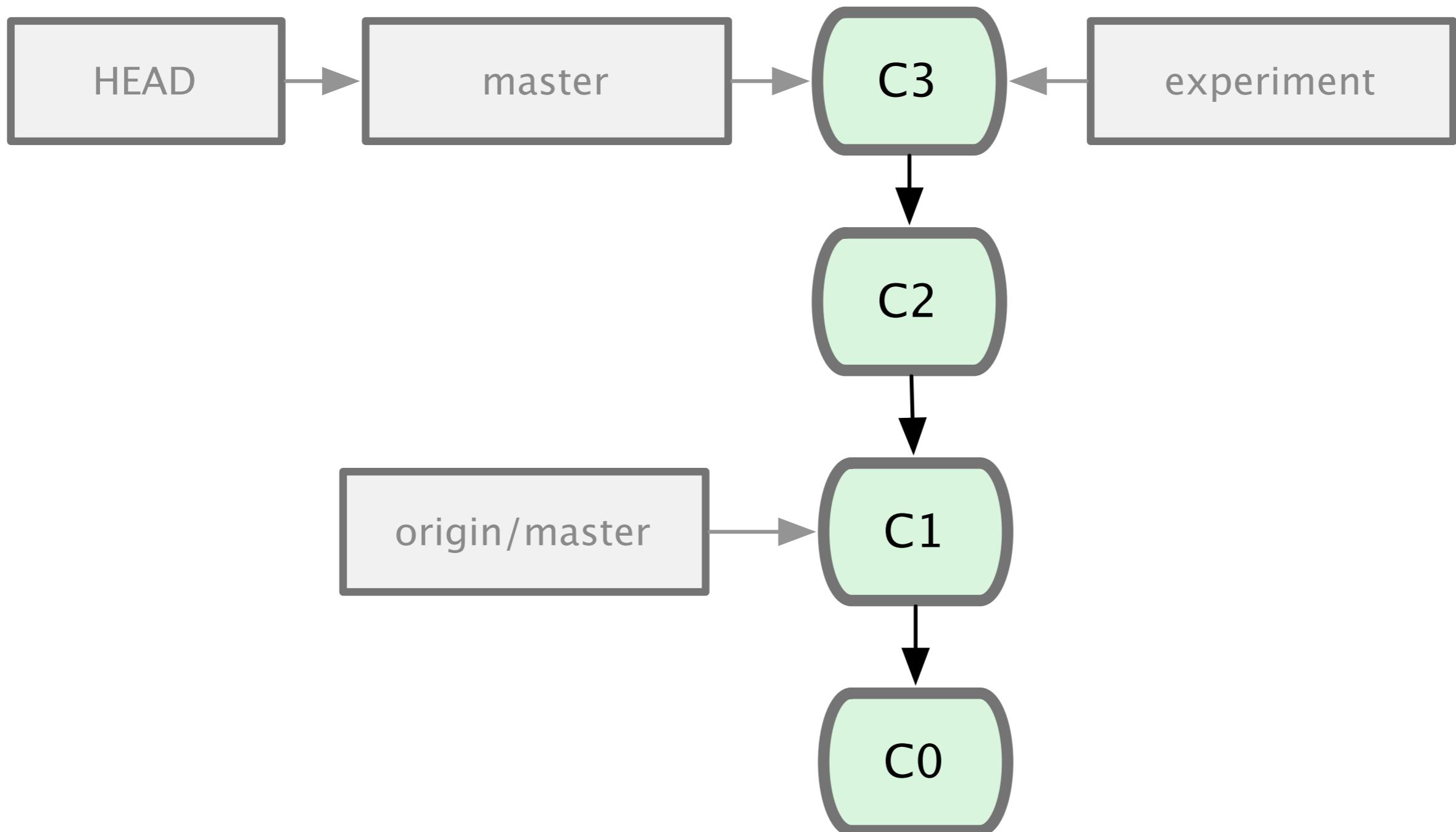
```
$ git branch experiment
```

```
$ git branch experiment  
$ git reset --hard origin/master
```

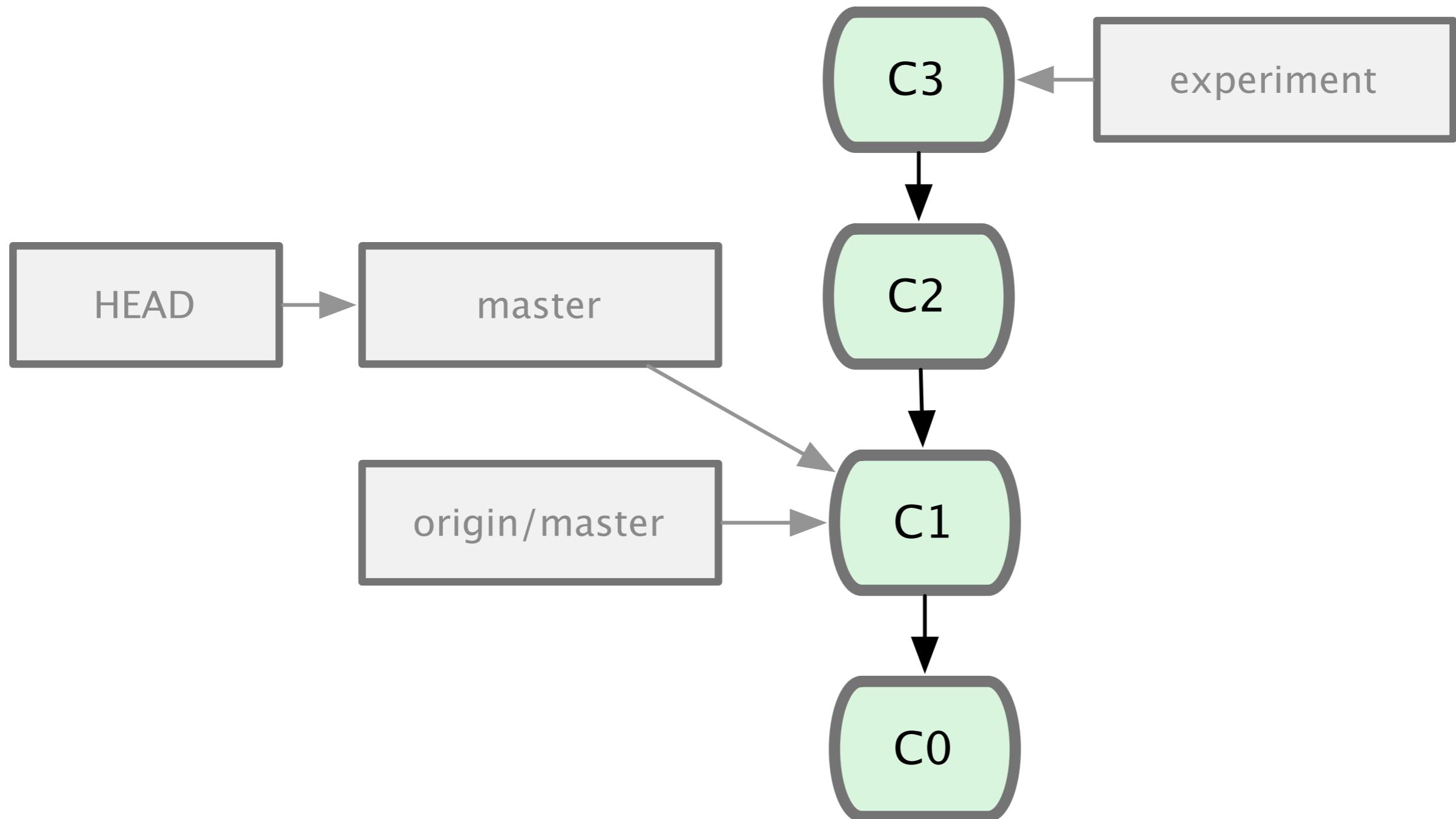
```
$ git branch experiment  
$ git reset --hard origin/master  
$ git checkout experiment
```



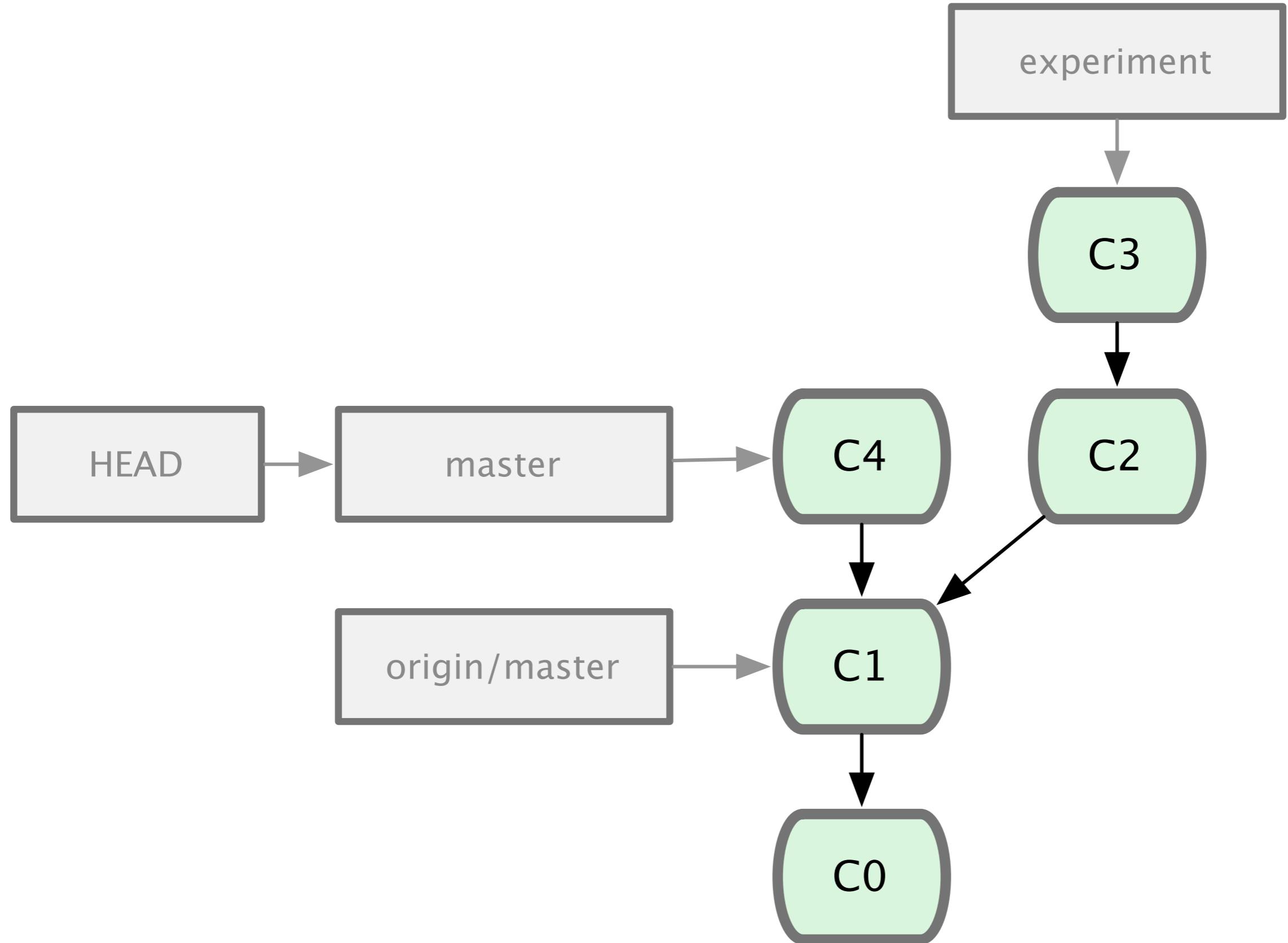




git checkout experiment



git reset --hard origin/master



git commit

Bisect

**binary search for where
a bug was introduced**

```
$ git bisect start
```

```
$ git bisect start  
$ git bisect bad
```

```
$ git bisect start  
$ git bisect bad  
$ git bisect good 3acb4c2c6666ed6cb91cb0b983efe9d50cf8cfbe
```

```
$ git bisect start
$ git bisect bad
$ git bisect good 3acb4c2c6666ed6cb91cb0b983efe9d50cf8cfbe
Bisecting: 6 revisions left to test after this
[ecb6e1bc347ccecc5f9350d878ce677feb13d3b2] error handling on repo
```

```
$ git bisect start
$ git bisect bad
$ git bisect good 3acb4c2c6666ed6cb91cb0b983efe9d50cf8cfbe
Bisecting: 6 revisions left to test after this
[ecb6e1bc347ccecc5f9350d878ce677feb13d3b2] error handling on repo
$ git bisect good
```

```
$ git bisect start
$ git bisect bad
$ git bisect good 3acb4c2c6666ed6cb91cb0b983efe9d50cf8cfbe
Bisecting: 6 revisions left to test after this
[ecb6e1bc347ccecc5f9350d878ce677feb13d3b2] error handling on repo
$ git bisect good
Bisecting: 3 revisions left to test after this
[b047b02ea83310a70fd603dc8cd7a6cd13d15c04] secure this thing
```

```
$ git bisect start
$ git bisect bad
$ git bisect good 3acb4c2c6666ed6cb91cb0b983efe9d50cf8cfbe
Bisecting: 6 revisions left to test after this
[ecb6e1bc347ccecc5f9350d878ce677feb13d3b2] error handling on repo
$ git bisect good
Bisecting: 3 revisions left to test after this
[b047b02ea83310a70fd603dc8cd7a6cd13d15c04] secure this thing
$ git bisect bad
```

```
$ git bisect start
$ git bisect bad
$ git bisect good 3acb4c2c6666ed6cb91cb0b983efe9d50cf8cfbe
Bisecting: 6 revisions left to test after this
[ecb6e1bc347ccecc5f9350d878ce677feb13d3b2] error handling on repo
$ git bisect good
Bisecting: 3 revisions left to test after this
[b047b02ea83310a70fd603dc8cd7a6cd13d15c04] secure this thing
$ git bisect bad
Bisecting: 1 revisions left to test after this
[f71ce38690acf49c1f3c9bea38e09d82a5ce6014] drop exceptions table
```

```
$ git bisect start
$ git bisect bad
$ git bisect good 3acb4c2c6666ed6cb91cb0b983efe9d50cf8cfbe
Bisecting: 6 revisions left to test after this
[ecb6e1bc347ccecc5f9350d878ce677feb13d3b2] error handling on repo
$ git bisect good
Bisecting: 3 revisions left to test after this
[b047b02ea83310a70fd603dc8cd7a6cd13d15c04] secure this thing
$ git bisect bad
Bisecting: 1 revisions left to test after this
[f71ce38690acf49c1f3c9bea38e09d82a5ce6014] drop exceptions table
$ git bisect good
```

```
$ git bisect start
$ git bisect bad
$ git bisect good 3acb4c2c6666ed6cb91cb0b983efe9d50cf8cfbe
Bisecting: 6 revisions left to test after this
[ecb6e1bc347ccecc5f9350d878ce677feb13d3b2] error handling on repo
$ git bisect good
Bisecting: 3 revisions left to test after this
[b047b02ea83310a70fd603dc8cd7a6cd13d15c04] secure this thing
$ git bisect bad
Bisecting: 1 revisions left to test after this
[f71ce38690acf49c1f3c9bea38e09d82a5ce6014] drop exceptions table
$ git bisect good
b047b02ea83310a70fd603dc8cd7a6cd13d15c04 is first bad commit
commit b047b02ea83310a70fd603dc8cd7a6cd13d15c04
Author: PJ Hyett <pjhyett@gmail.com>
Date:   Tue Jan 27 14:48:32 2009 -0800
```

secure this thing

```
:040000 040000 40ee3e7821b895e52c1695092db9bdc4c61d1730
f24d3c6ebcf639b1a3814550e62d60b8e68a8e4 M config
```

```
$ git bisect start
$ git bisect bad
$ git bisect good 3acb4c2c6666ed6cb91cb0b983efe9d50cf8cfbe
Bisecting: 6 revisions left to test after this
[ecb6e1bc347ccecc5f9350d878ce677feb13d3b2] error handling on repo
$ git bisect good
Bisecting: 3 revisions left to test after this
[b047b02ea83310a70fd603dc8cd7a6cd13d15c04] secure this thing
$ git bisect bad
Bisecting: 1 revisions left to test after this
[f71ce38690acf49c1f3c9bea38e09d82a5ce6014] drop exceptions table
$ git bisect good
```

b047b02ea83310a70fd603dc8cd7a6cd13d15c04 is first bad commit

commit b047b02ea83310a70fd603dc8cd7a6cd13d15c04

Author: PJ Hyett <pjhyett@gmail.com>

Date: Tue Jan 27 14:48:32 2009 -0800

secure this thing

```
:040000 040000 40ee3e7821b895e52c1695092db9bdc4c61d1730
f24d3c6ebcf639b1a3814550e62d60b8e68a8e4 M config
```

```
$ git bisect reset
```

now...

go use Git to write an amazing program that helps create an inexpensive and unending food supply that will make sure than nobody goes hungry ever again

boom.

that's it!

don't forget your stickers!

Resources

Resources

git-scm.com

Resources

git-scm.com

gitcasts.com

Resources

git-scm.com

gitcasts.com

learn.github.com

Resources

git-scm.com

gitcasts.com

learn.github.com

#git / #github on IRC

Resources

git-scm.com

gitcasts.com

learn.github.com

#git / #github on IRC

peepcode - git book and screencast

Resources

git-scm.com

gitcasts.com

learn.github.com

#git / #github on IRC

peepcode - git book and screencast

schacon@gmail.com

Resources

git-scm.com

gitcasts.com

learn.github.com

#git / #github on IRC

peepcode - git book and screencast

schacon@gmail.com

me, in person, tonight, someplace that serves whiskey