

Git 101

Scott Chacon

Introduction

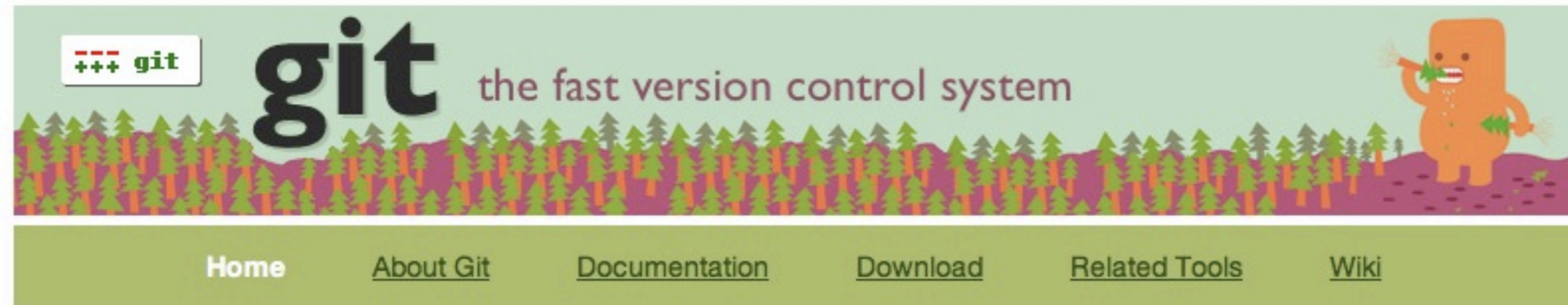
Me

Scott Chacon



github
SOCIAL CODE HOSTING

git-scm.com



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?

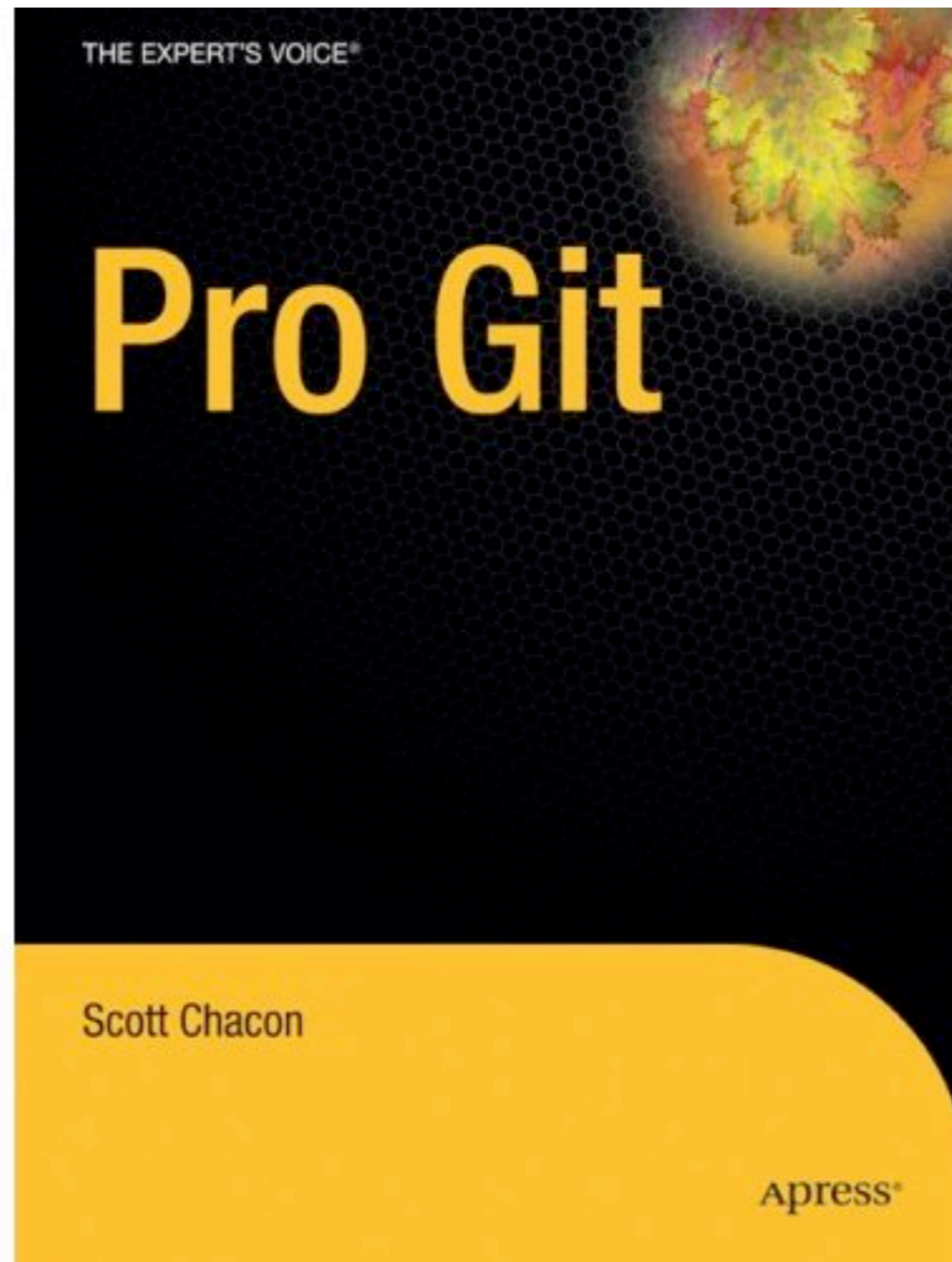
If you are curious about something, feel

THE EXPERT'S VOICE®

Pro Git

Scott Chacon

Apress®



<http://progit.org>

schacon@gmail.com

@chacon

twitter

</me>

What is Git?

**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

mercurial (hg)

bazaar

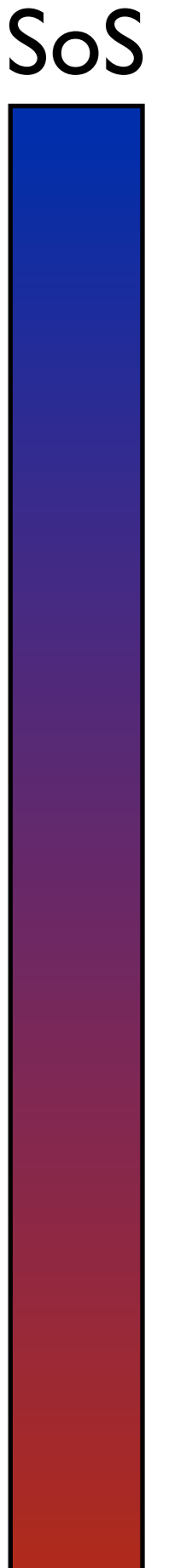
subversion (svn)

version control

concurrent version system (cvs)

perforce

visual source safe





mercurial (hg)

“not bad”

SoS

bazaar

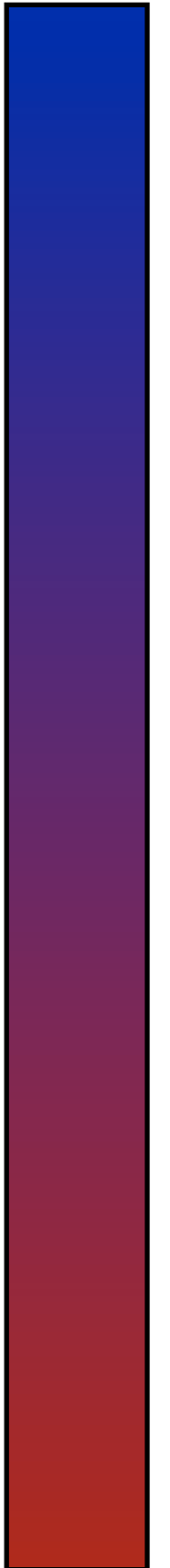
subversion (svn)

version control

concurrent version system (cvs)

perforce

visual source safe



mercurial (hg)

bazaar

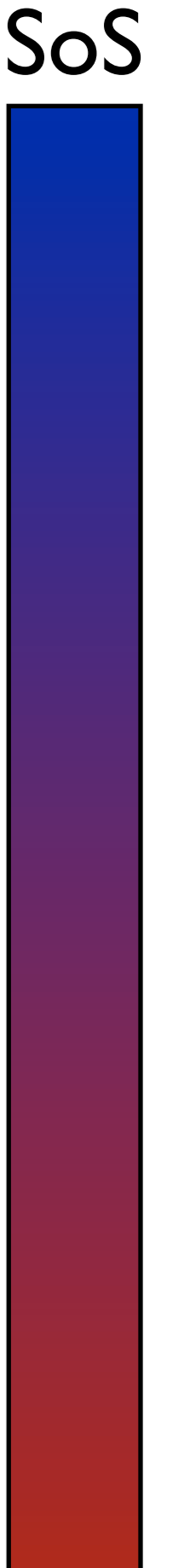
subversion (svn)

version control

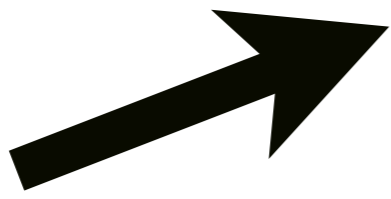
concurrent version system (cvs)

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



+++ git **git** the fast version control system

Home About Git Documentation Download Tools & Hosting Wiki

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

everything is fast

every clone is a backup

work offline

No Network Needed

Performing a diff

Viewing file history

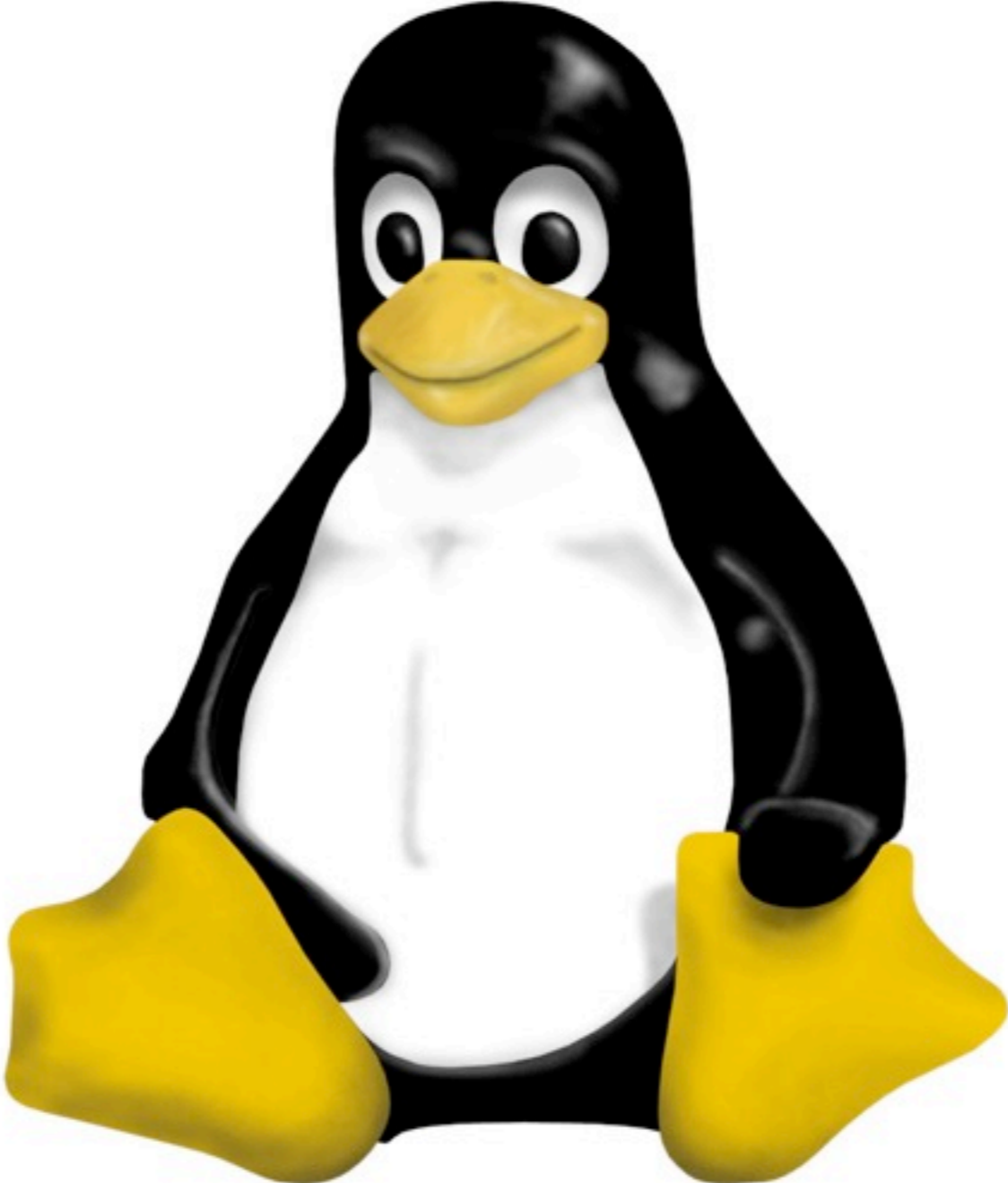
Committing changes

Merging branches

Obtaining other revisions of a file

Switching branches


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



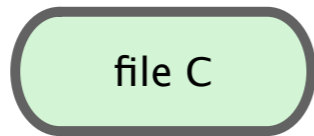
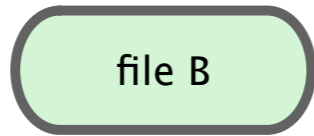
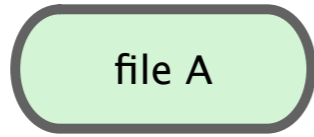
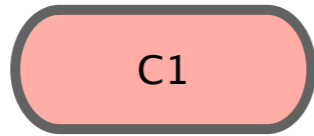
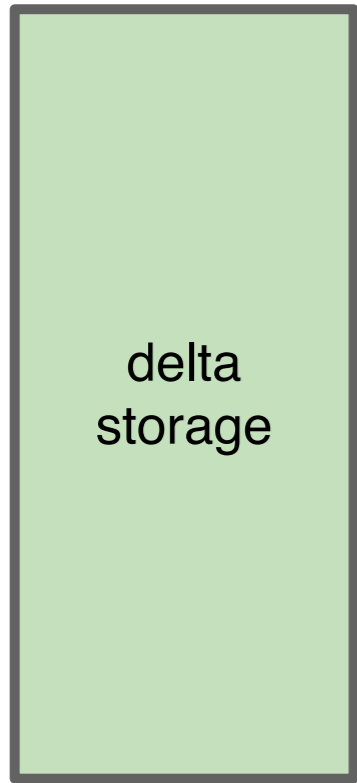
Immutable

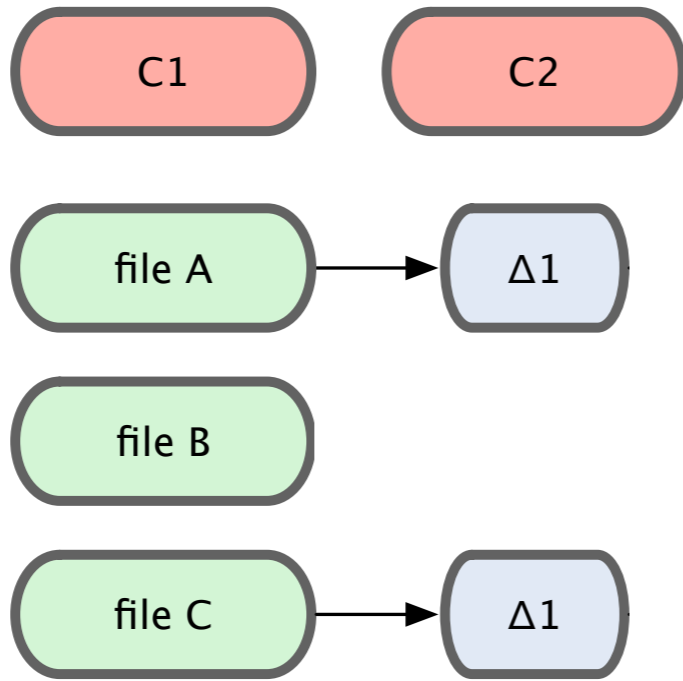
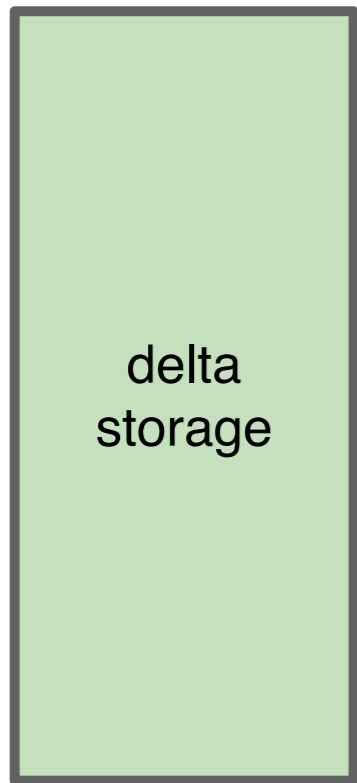
(almost) never removes data

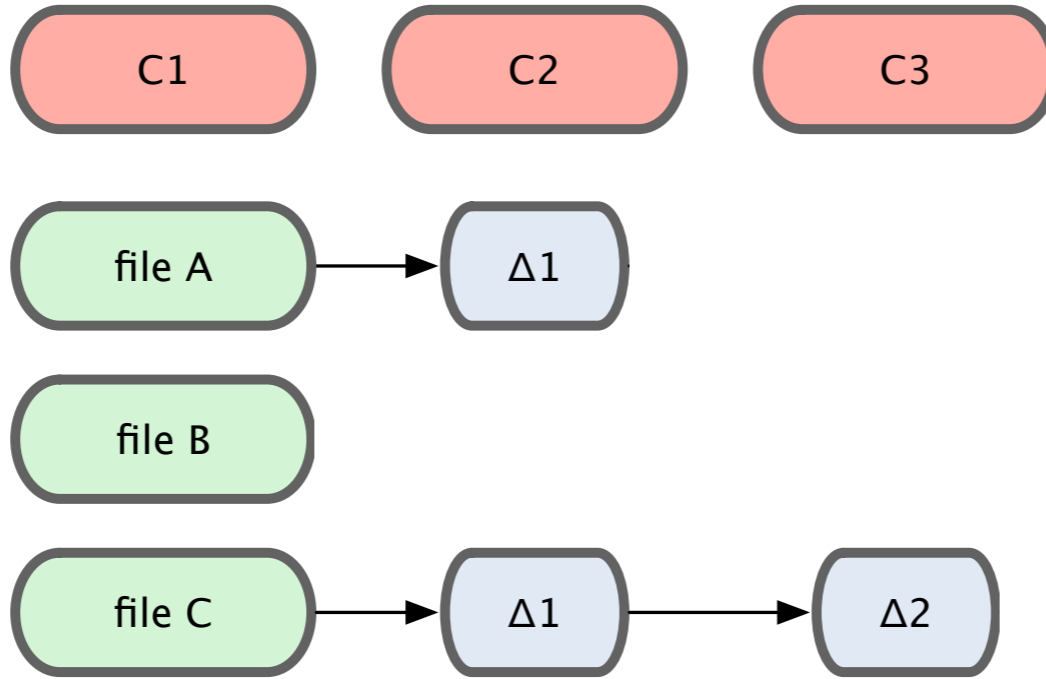
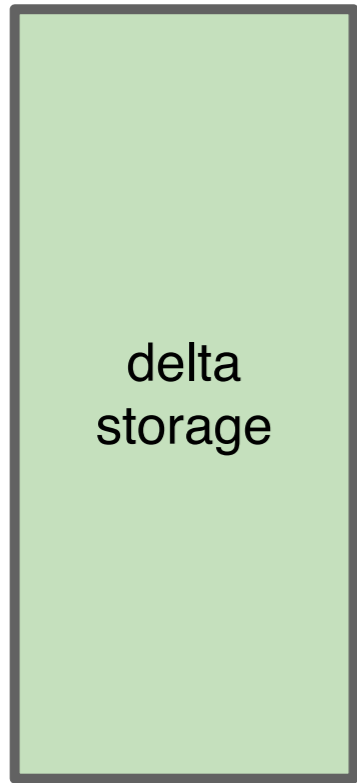
Snapshots, not Patches

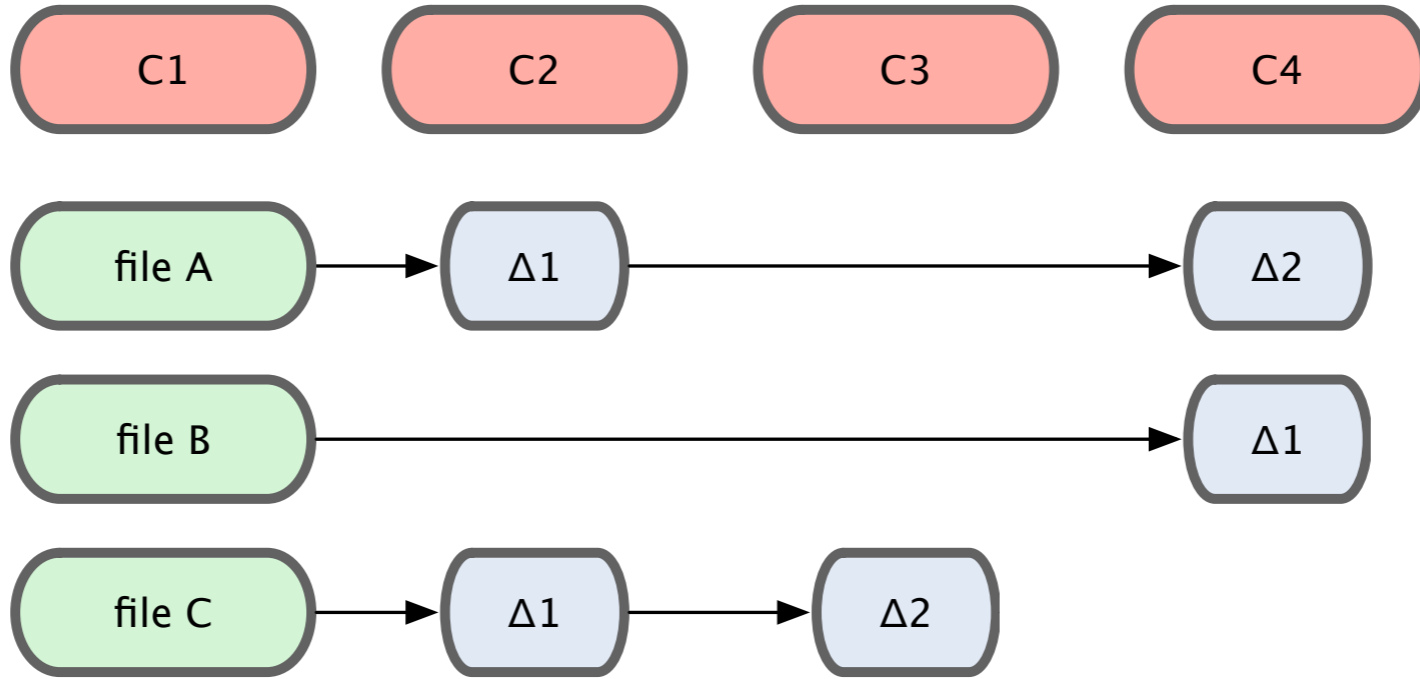
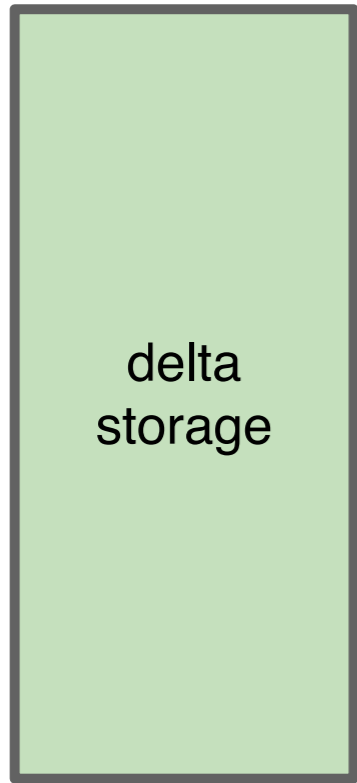


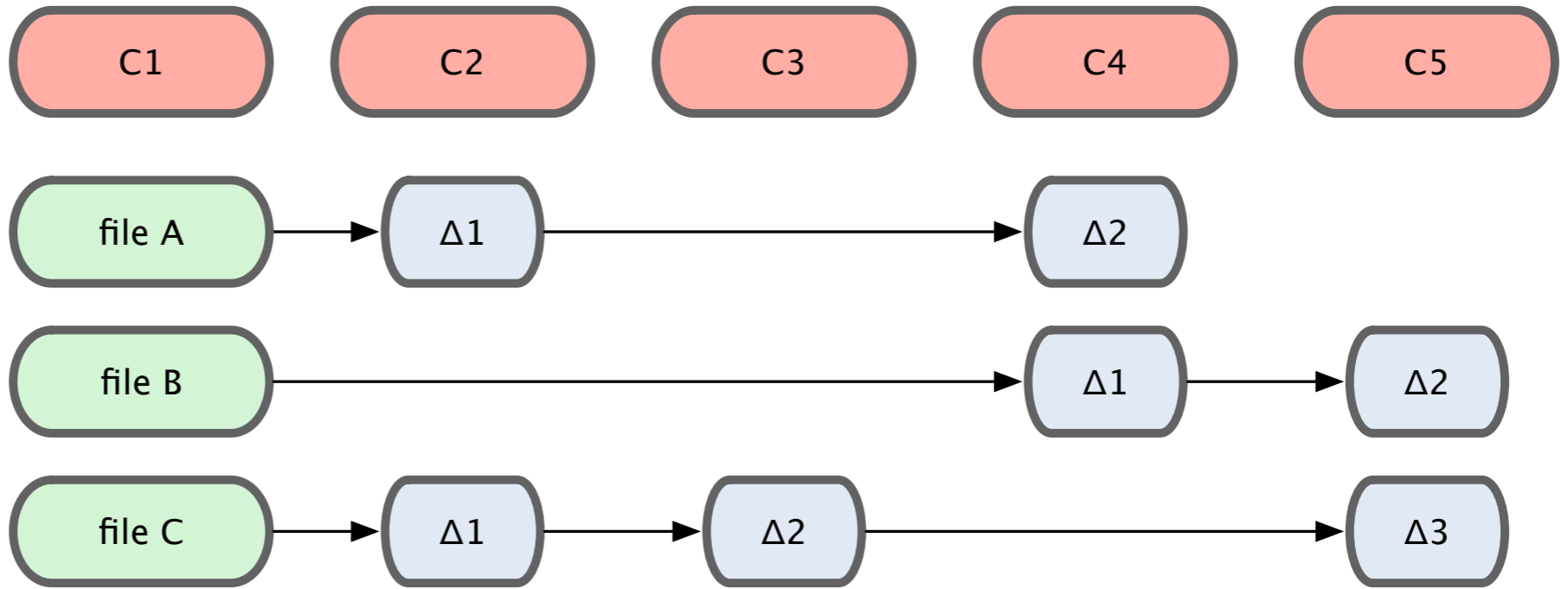
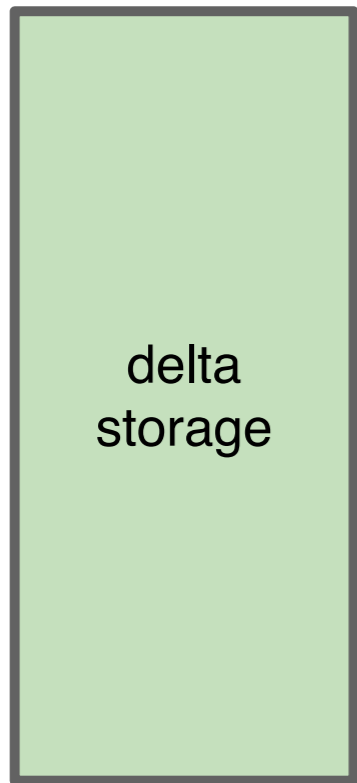
delta
storage

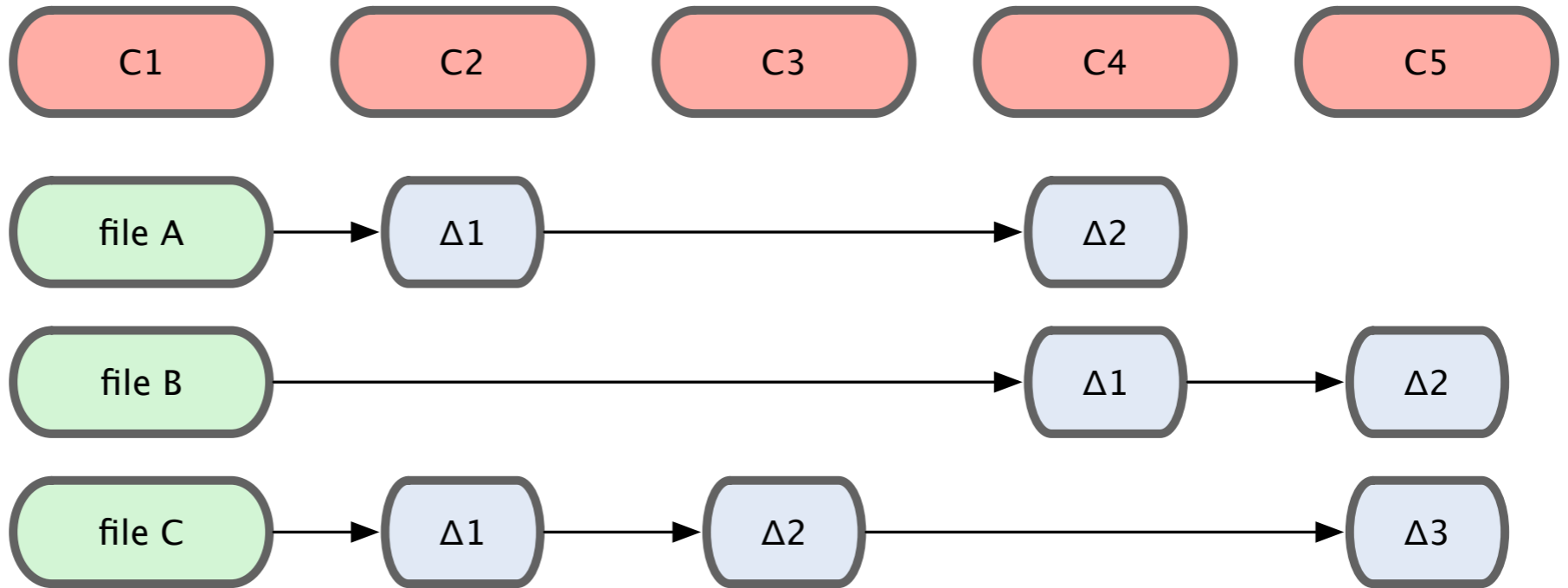
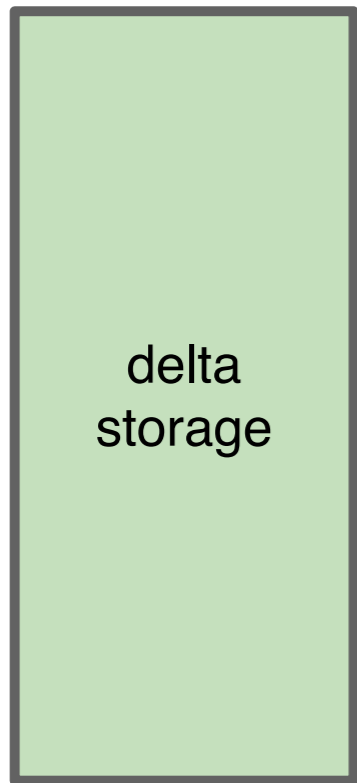


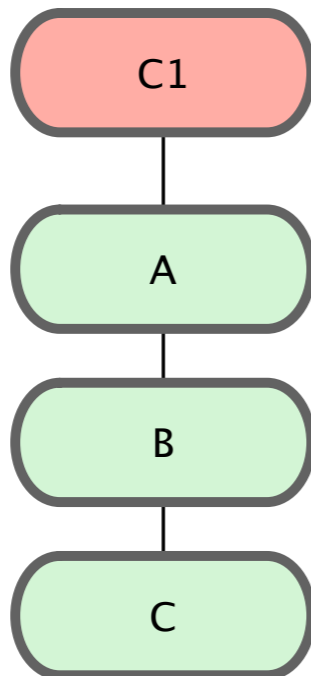
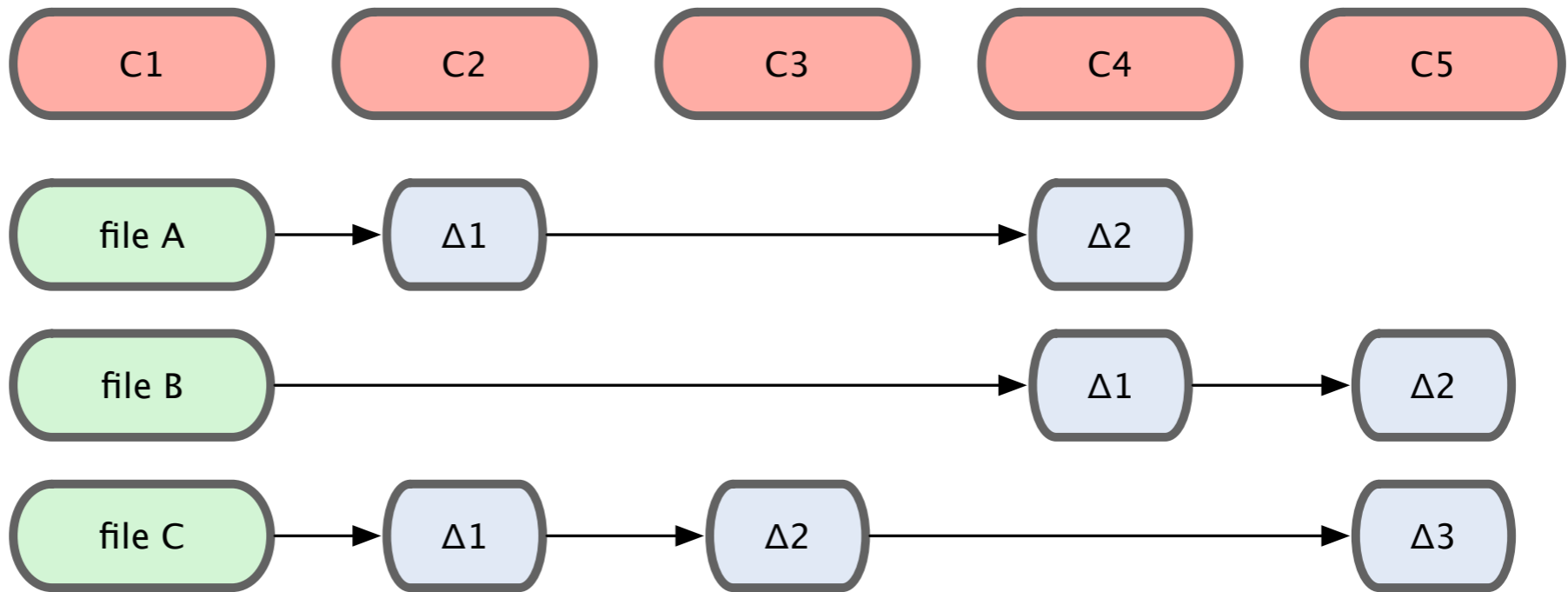
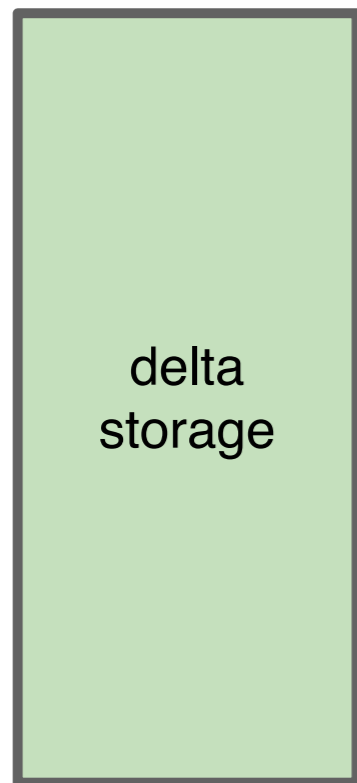


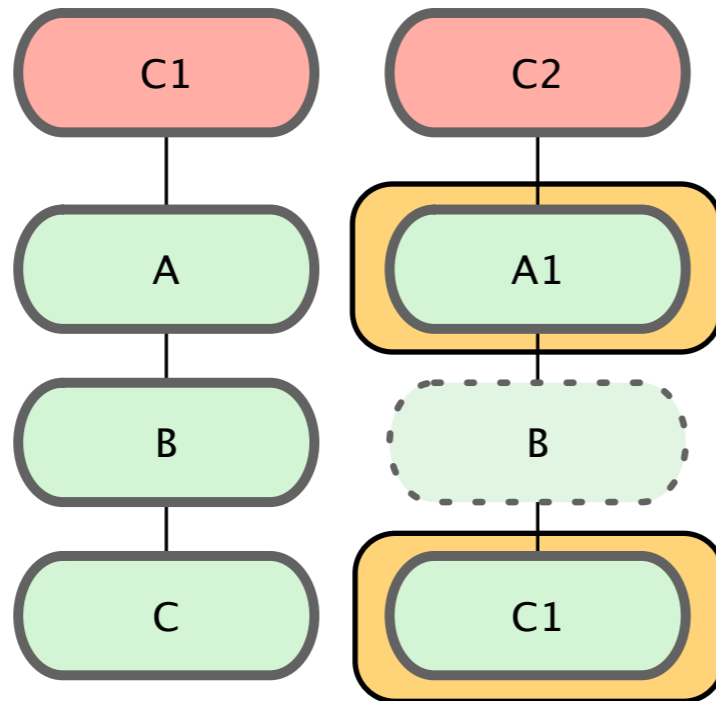
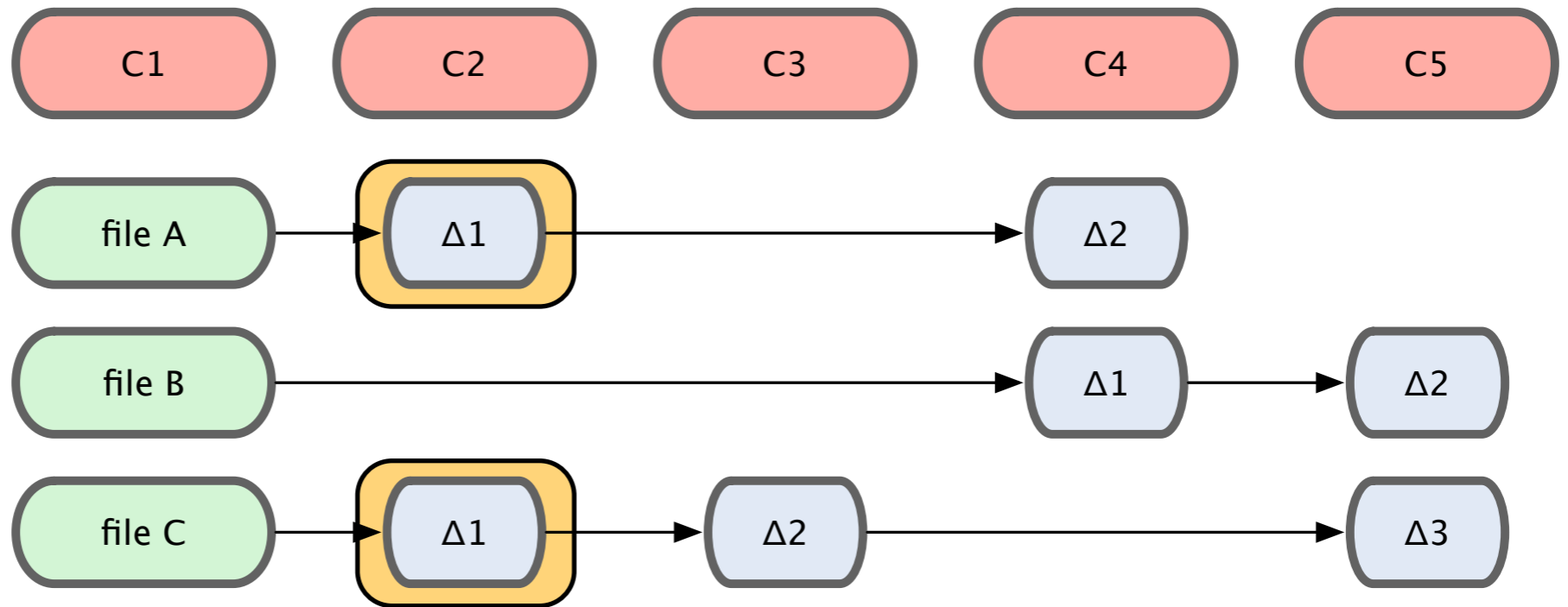
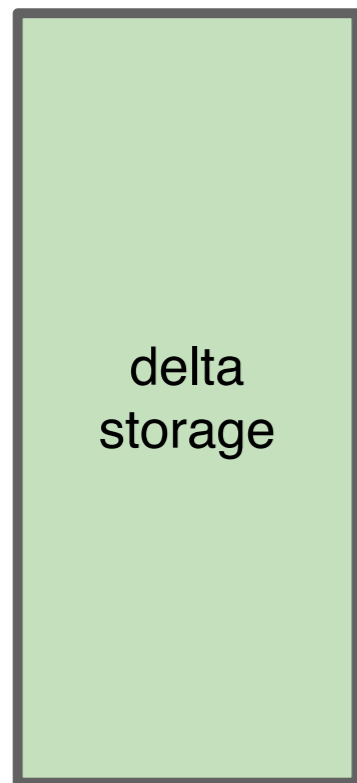


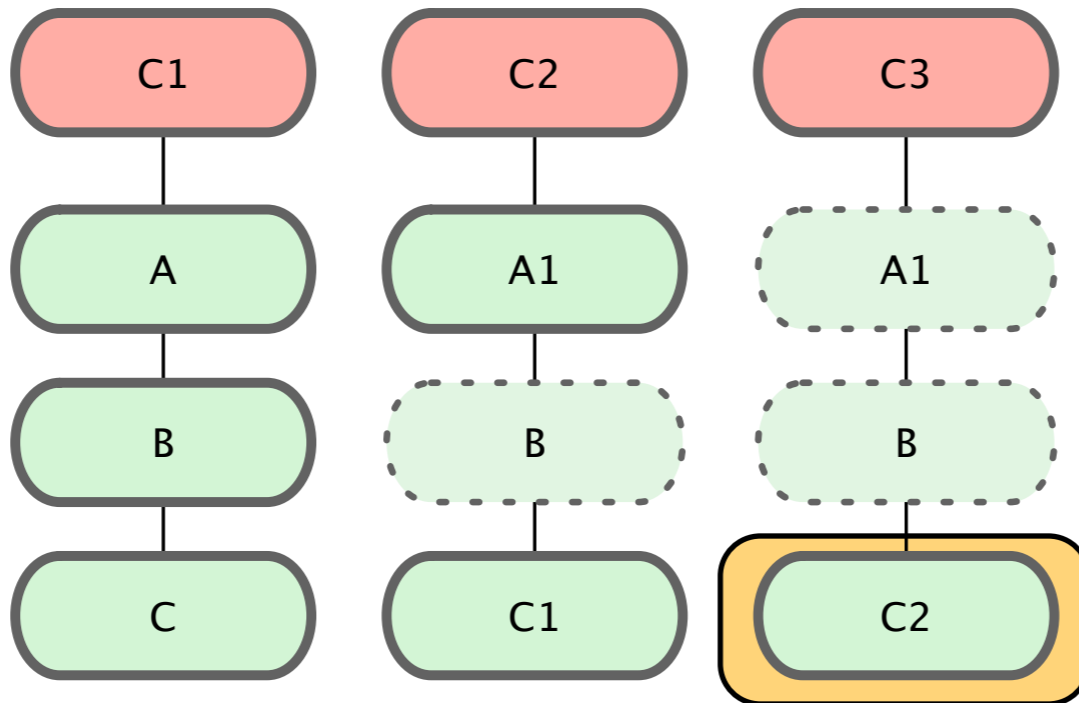
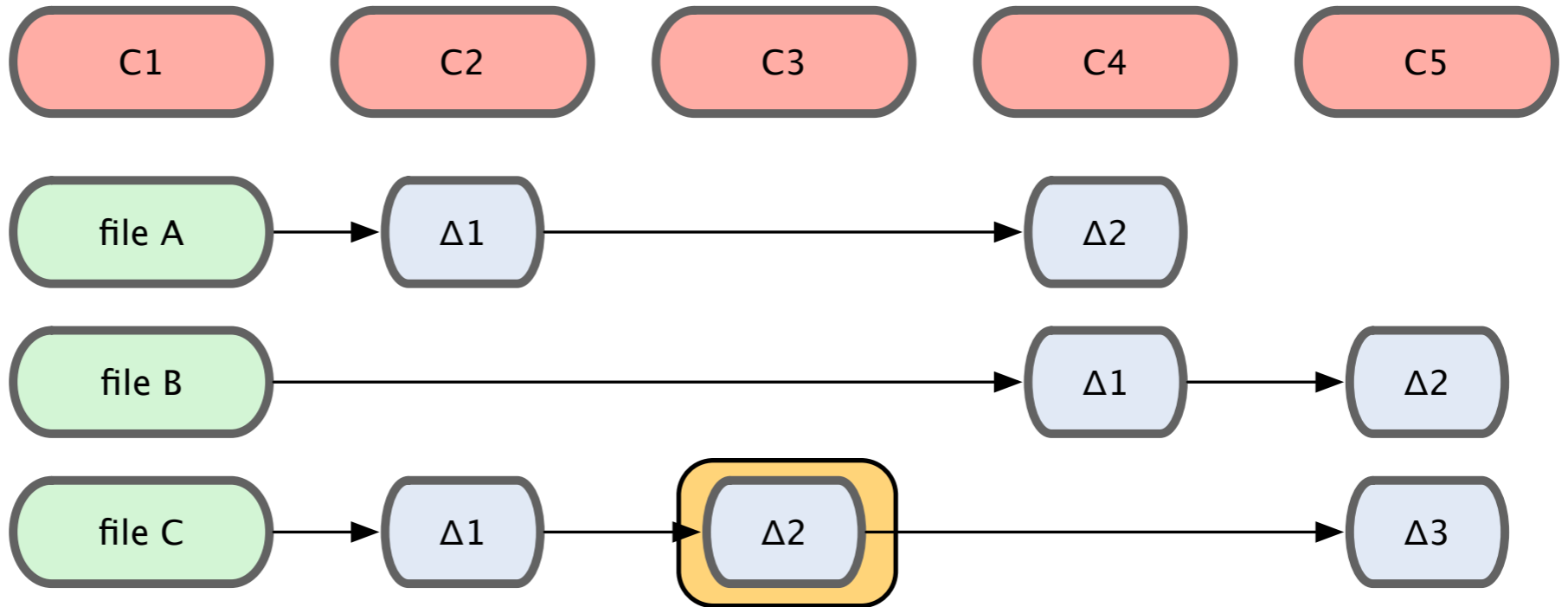
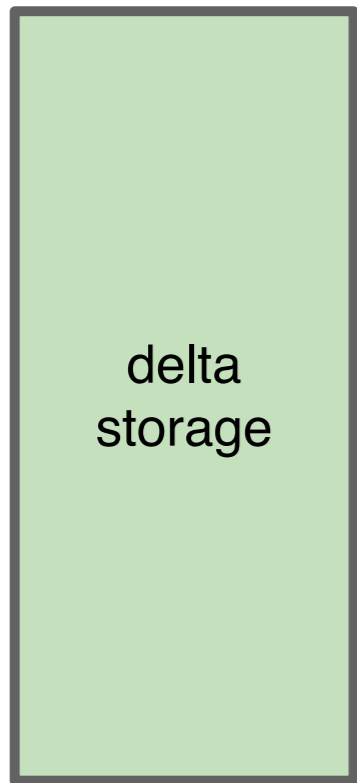


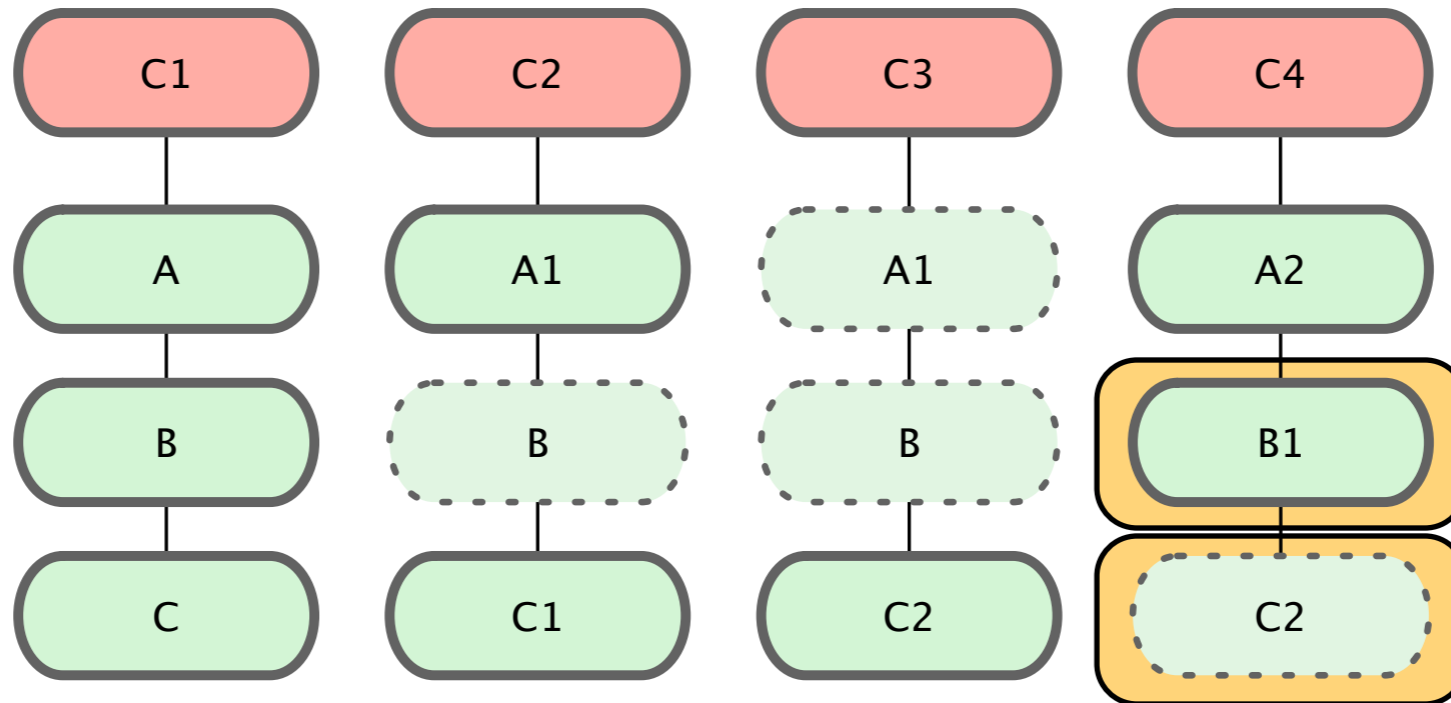
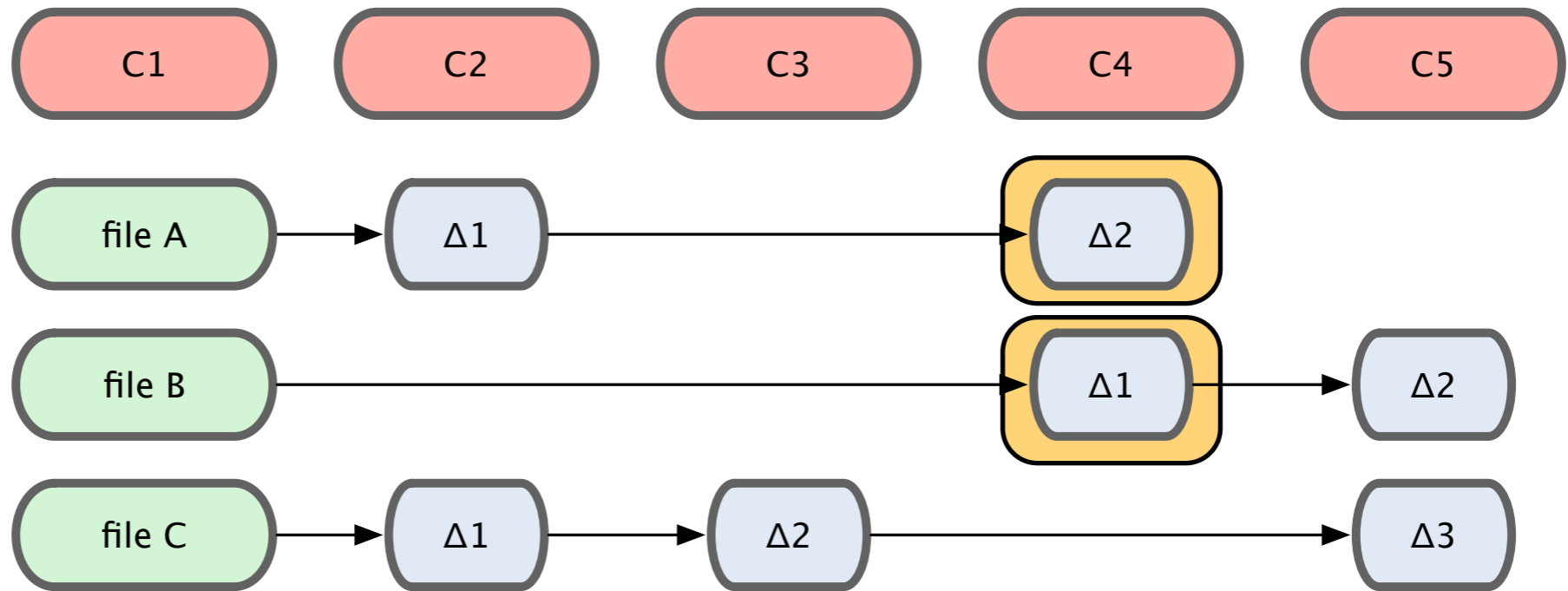
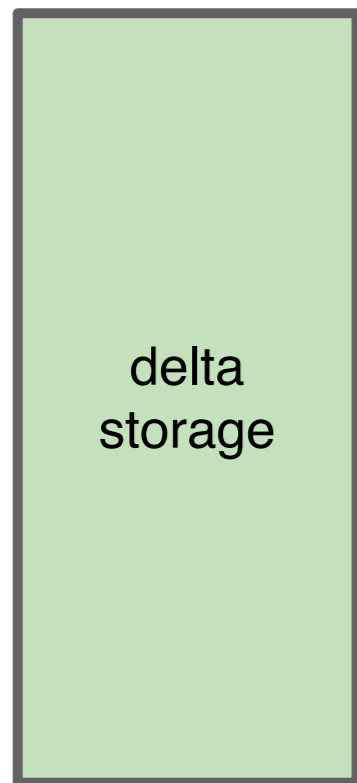


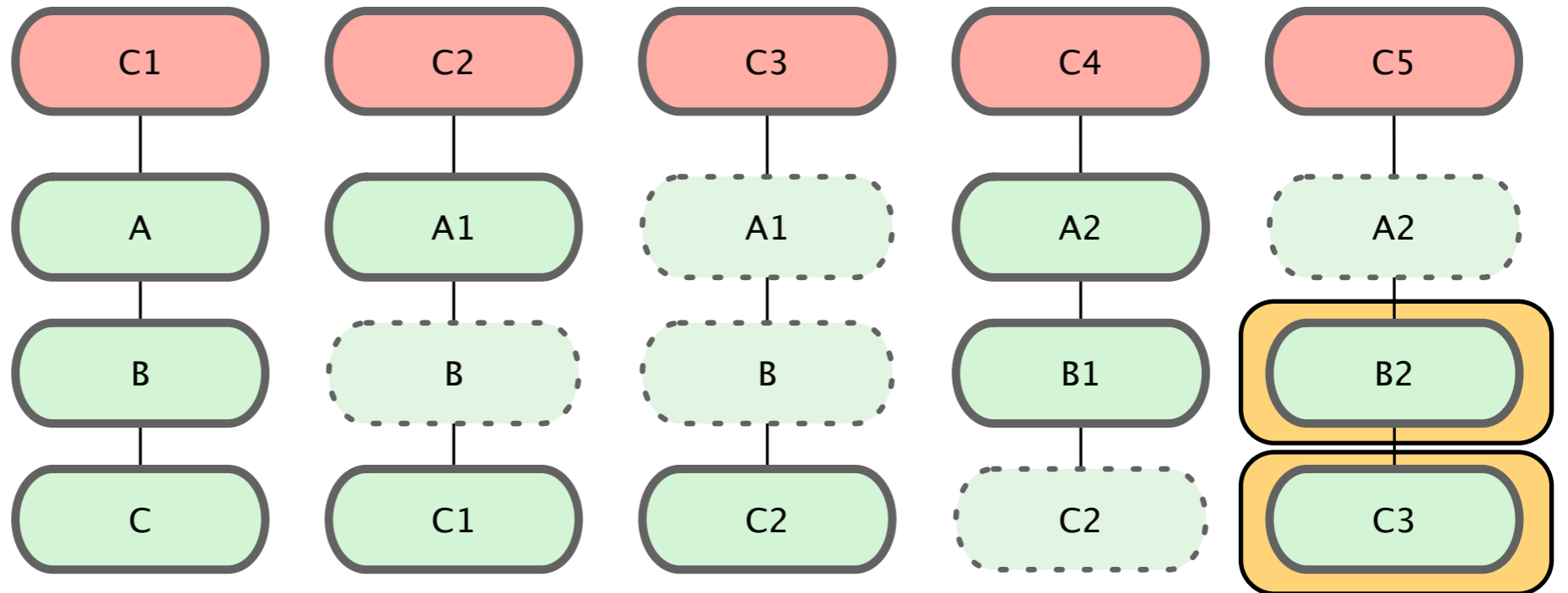
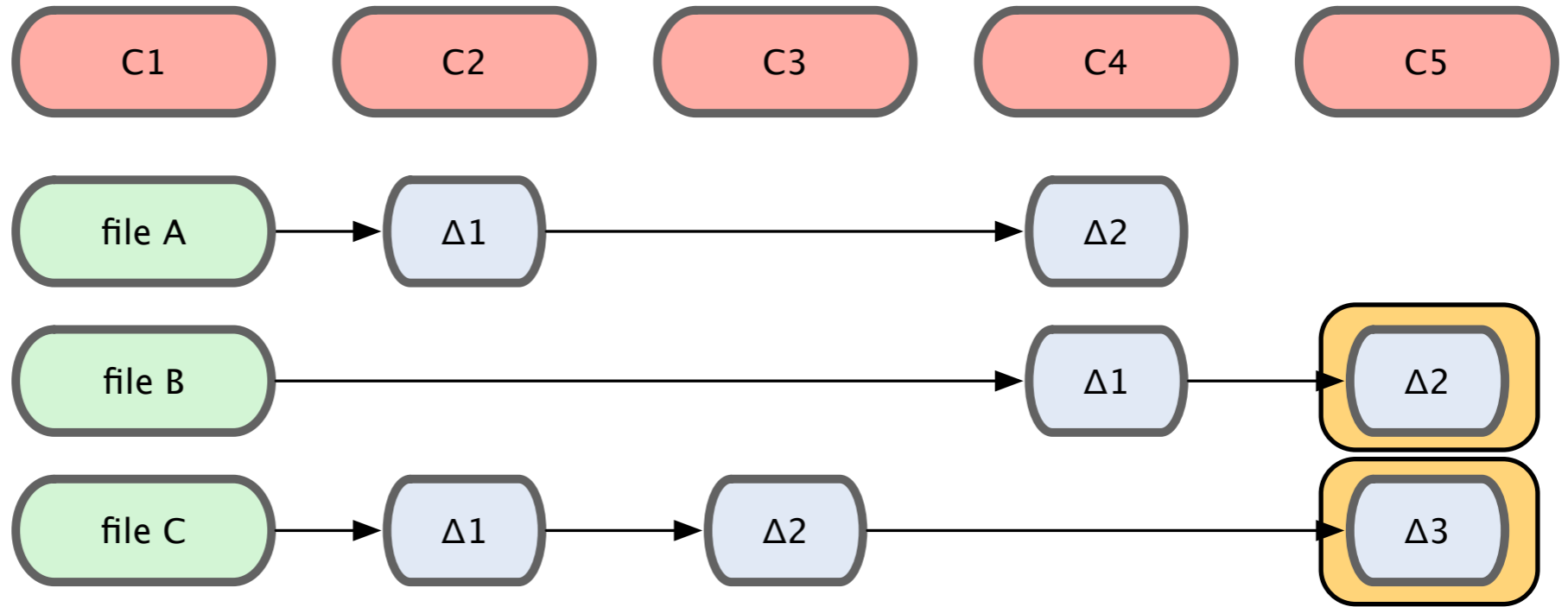
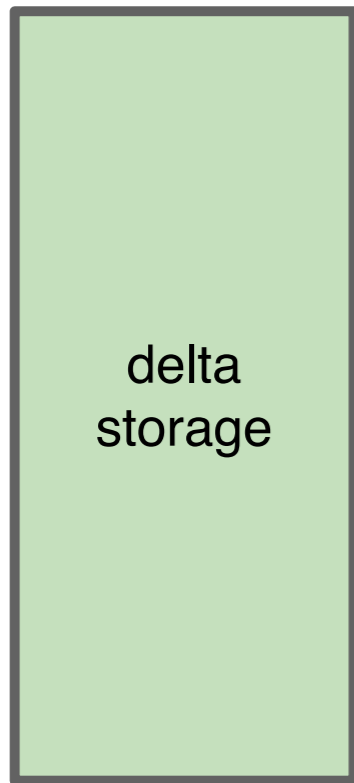


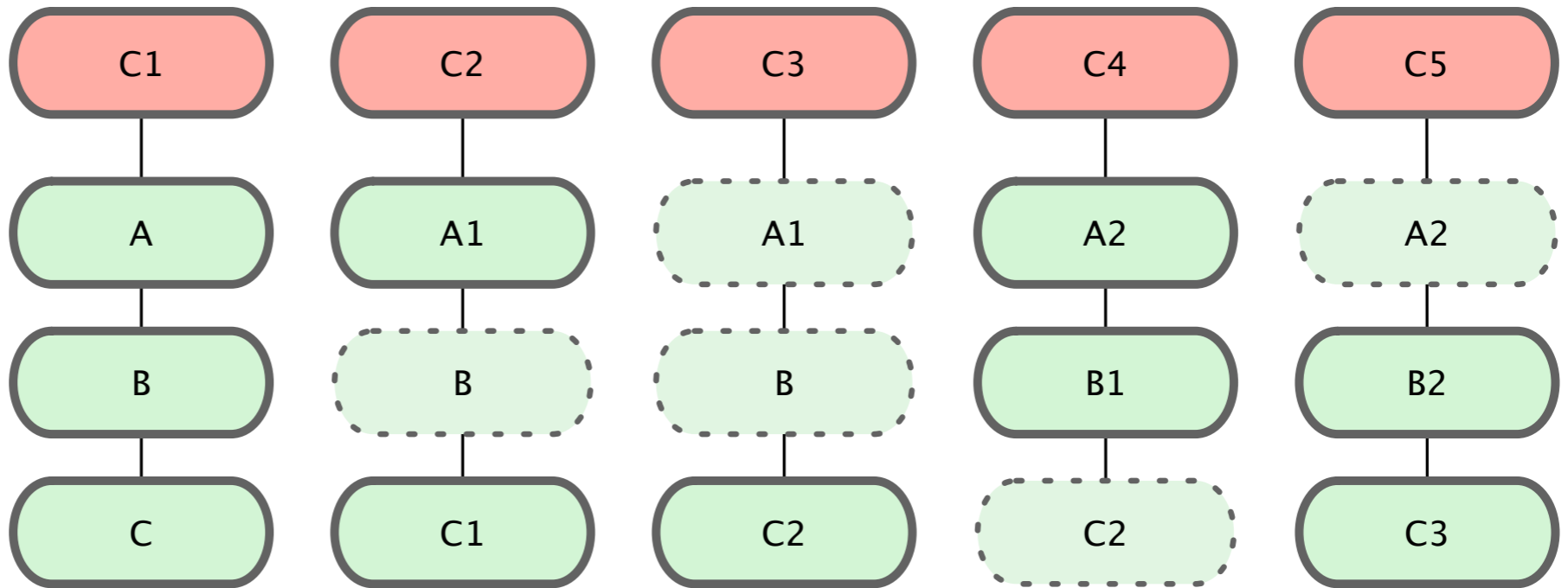
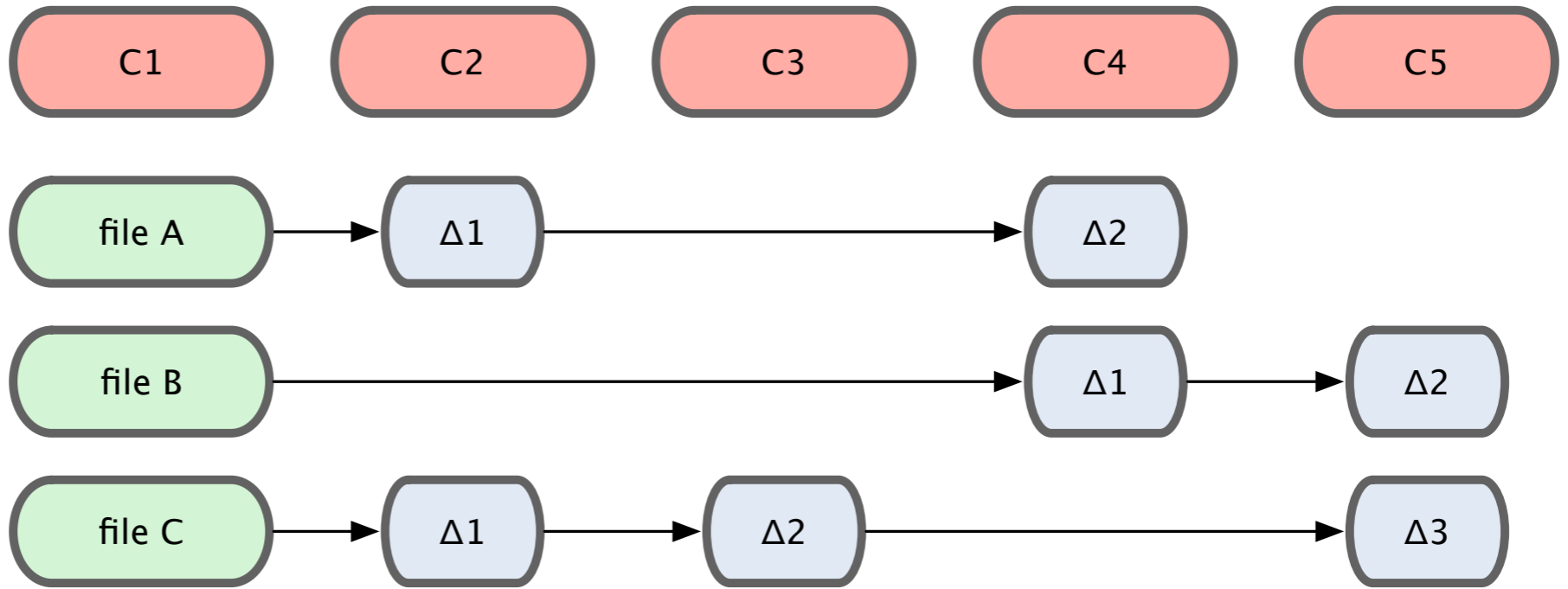
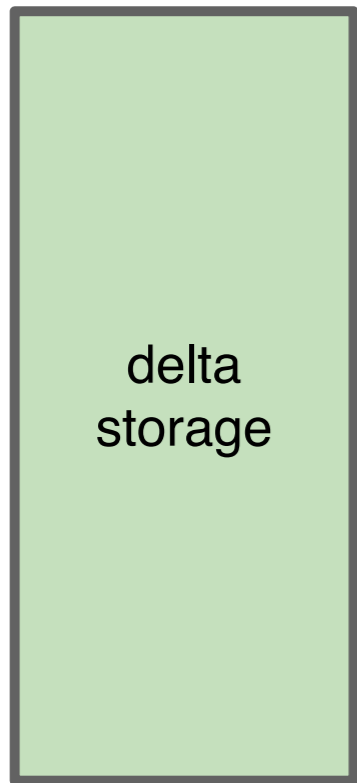


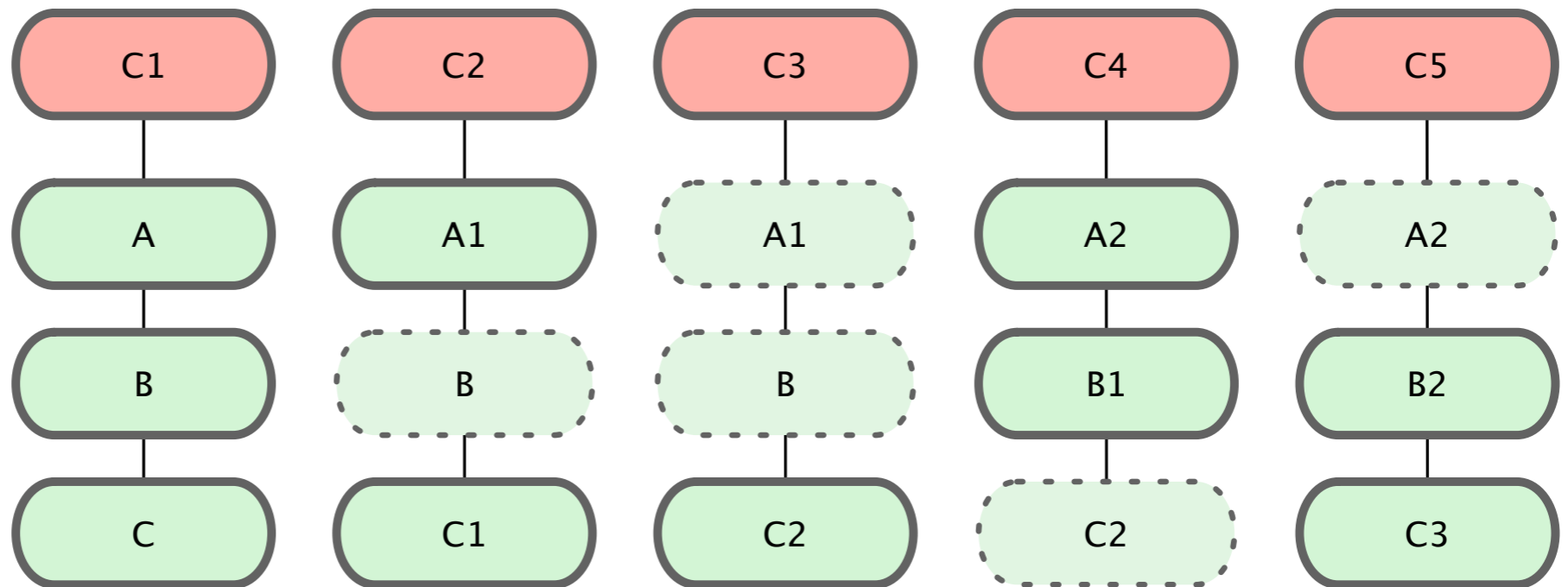
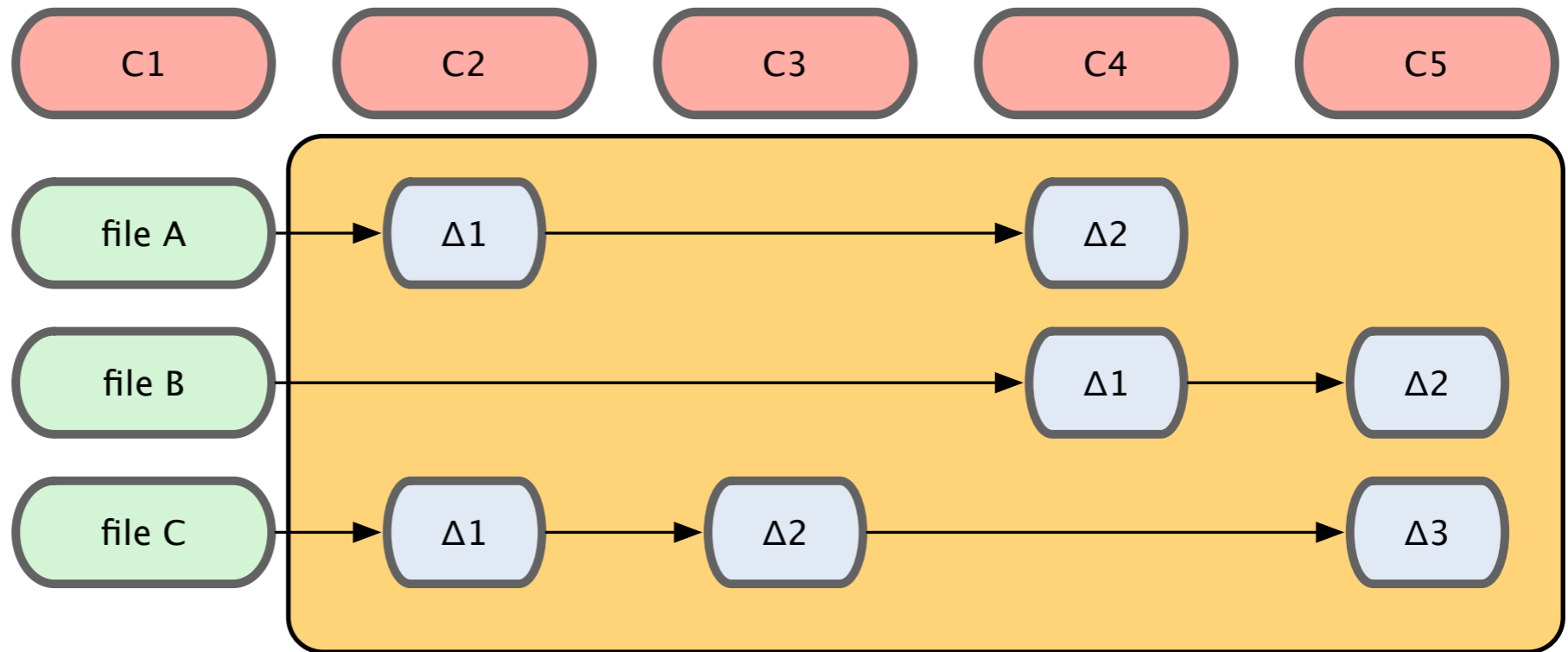
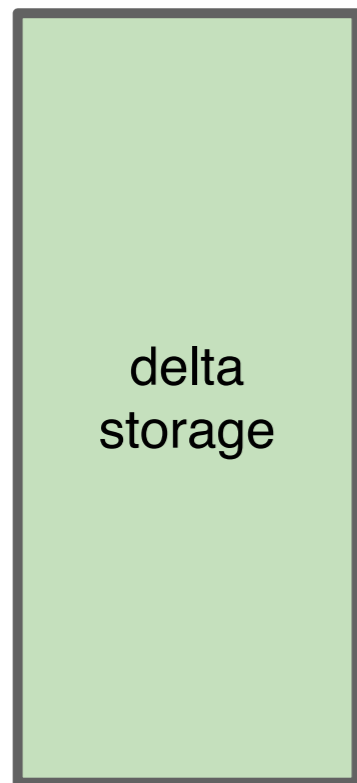


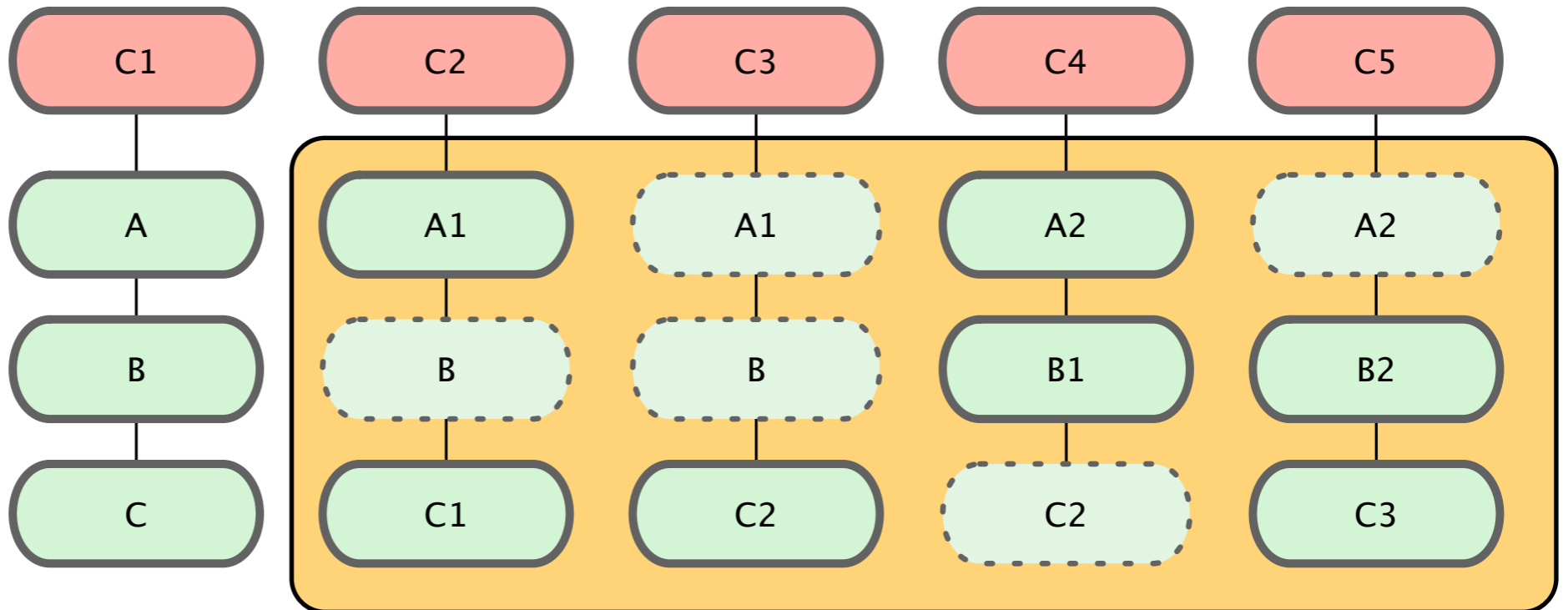
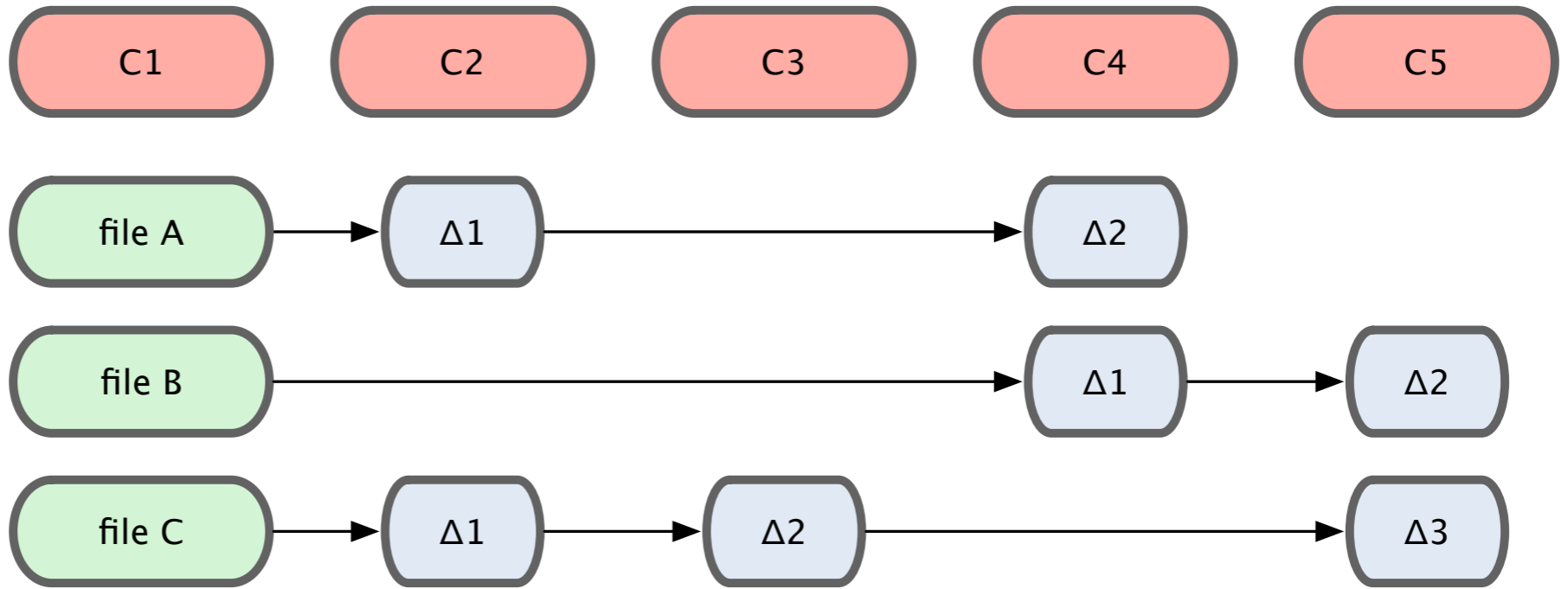
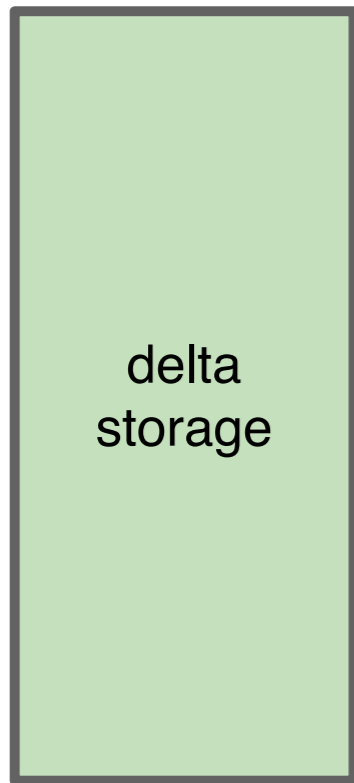












Basic Git

First Steps


```
$ git config --global user.name "Scott Chacon"
```

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

Getting a Repo

Create One

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

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

```
$ touch hello_world.rb
$ git init
$ git add .
$ git commit -m 'first commit'
```

```
$ tree -a
```

```
·
|-- .git
|   |-- COMMIT_EDITMSG
|   |-- HEAD
|   |-- branches
|   |-- config
|   |-- description
|   |-- hooks
|   |   |-- applypatch-msg.sample
|   |   `-- update.sample
|   |-- index
|   |-- info
|   |   `-- exclude
|   |-- logs
|   |   |-- HEAD
|   |   `-- refs
|   |       |-- heads
|   |       `-- master
|   |-- objects
|   |   |-- 32/09658ac8d80bc9726d3a33d77e3dfc5fe6035e
|   |   |-- 53/9cd7886a627841d525a78d45cbc6396be20b41
|   |   |-- e6/9de29bb2d1d6434b8b29ae775ad8c2e48c5391
|   |   |-- info
|   |   `-- pack
|   |-- refs
|   |   |-- heads
|   |   |   `-- master
|   |   `-- tags
|   `-- remotes
`-- hello_world.rb
```

```
17 directories, 33 files
```

```
$ tree -a
```

```
.  
|-- .git  
|   |-- COMMIT_EDITMSG  
|   |-- HEAD  
|   |-- branches  
|   |-- config  
|   |-- description  
|   |-- hooks  
|       |-- applypatch-msg.sample  
|       `-- update.sample  
|-- index  
|-- info  
|   `-- exclude  
|-- logs  
|   |-- HEAD  
|   `-- refs  
|       |-- heads  
|       `-- master  
|-- objects  
|   |-- 32/09658ac8d80bc9726d3a33d77e3dfc5fe6035e  
|   |-- 53/9cd7886a627841d525a78d45cbc6396be20b41  
|   |-- e6/9de29bb2d1d6434b8b29ae775ad8c2e48c5391  
|   |-- info  
|   `-- pack  
|-- refs  
|   |-- heads  
|   |   `-- master  
|   `-- tags  
|   `-- remotes  
`-- hello_world.rb
```

```
17 directories, 33 files
```

```
$ tree -a
```

```
.  
|-- .git  
|   |-- COMMIT_EDITMSG  
|   |-- HEAD  
|   |-- branches  
|   |-- config  
|   |-- description  
|   |-- hooks  
|   |   |-- applypatch-msg.sample  
|   |   `-- update.sample  
|   |-- index  
|   |-- info  
|   |   `-- exclude  
|   |-- logs  
|   |   |-- HEAD  
|   |   `-- refs  
|   |       |-- heads  
|   |       `-- master  
|   |-- objects  
|   |   |-- 32/09658ac8d80bc9726d3a33d77e3dfc5fe6035e  
|   |   |-- 53/9cd7886a627841d525a78d45cbc6396be20b41  
|   |   |-- e6/9de29bb2d1d6434b8b29ae775ad8c2e48c5391  
|   |   |-- info  
|   |   `-- pack  
|   |-- refs  
|   |   |-- heads  
|   |   |   |-- master  
|   |   |   `-- tags  
|   |   `-- remotes  
|   `-- hello_world.rb
```

```
17 directories, 33 files
```

Clone One

```
git clone
```



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

```
$ git clone git://github.com/schacon/ticgit.git
Initialized empty Git repository in /private/tmp/ticgit/.git/
remote: Counting objects: 591, done.
remote: Compressing objects: 100% (267/267), done.
remote: Total 591 (delta 253), reused 587 (delta 252)
Receiving objects: 100% (591/591), 73.05 KiB, done.
Resolving deltas: 100% (253/253), done.
$
```

```
$ git clone git://github.com/schacon/ticgit.git
Initialized empty Git repository in /private/tmp/ticgit/.git/
remote: Counting objects: 591, done.
remote: Compressing objects: 100% (267/267), done.
remote: Total 591 (delta 253), reused 587 (delta 252)
Receiving objects: 100% (591/591), 73.05 KiB, done.
Resolving deltas: 100% (253/253), done.
$ cd ticgit/
$
```

```
$ git clone git://github.com/schacon/ticgit.git
Initialized empty Git repository in /private/tmp/ticgit/.git/
remote: Counting objects: 591, done.
remote: Compressing objects: 100% (267/267), done.
remote: Total 591 (delta 253), reused 587 (delta 252)
Receiving objects: 100% (591/591), 73.05 KiB, done.
Resolving deltas: 100% (253/253), done.
$ cd ticgit/
$ ls
.git  LICENSE      Rakefile  examples  note
ticgit.gemspec  README      bin       lib       spec
$
```

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

repository

“staging”

A Basic Workflow

Edit files

Stage the changes

Review your changes

Commit the changes

\$

\$

\$ 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

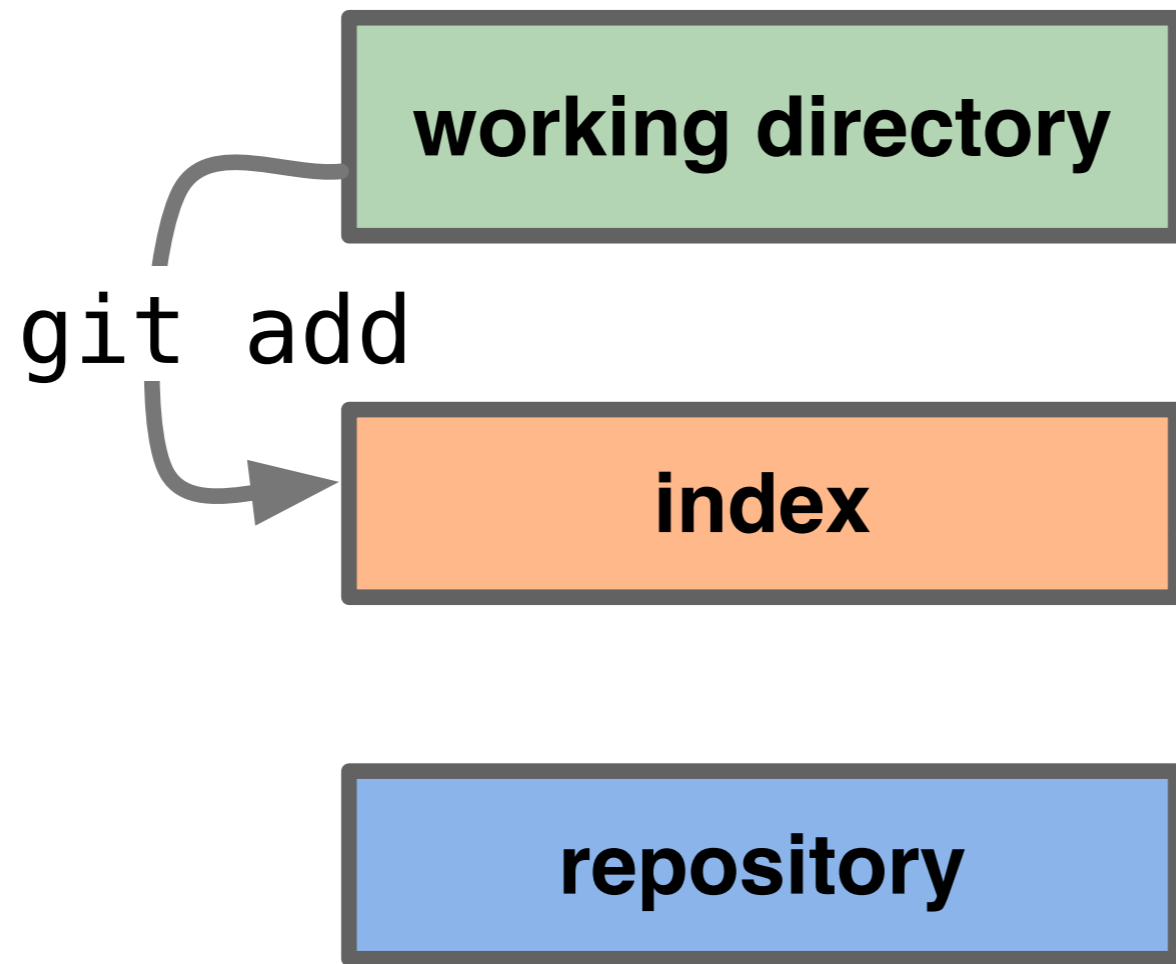
Commit the changes

`git add`

working directory

index

repository



\$


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

```
$ git 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
```

```
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
```

\$

\$

```
$ 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 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('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
```

```
# 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 committed)
```

```
#
```

```
# 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
    (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()
```

**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

\$

A Basic Workflow

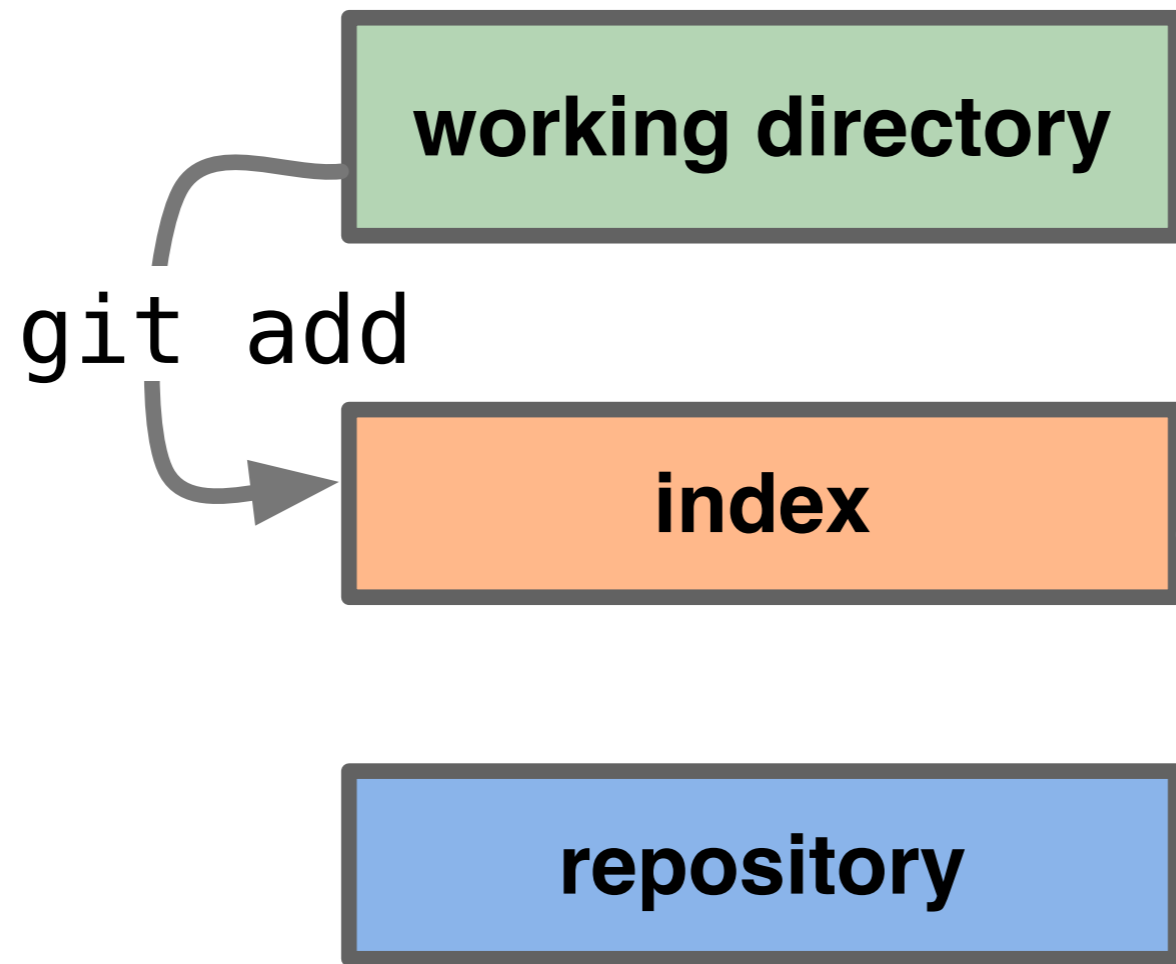
Edit files

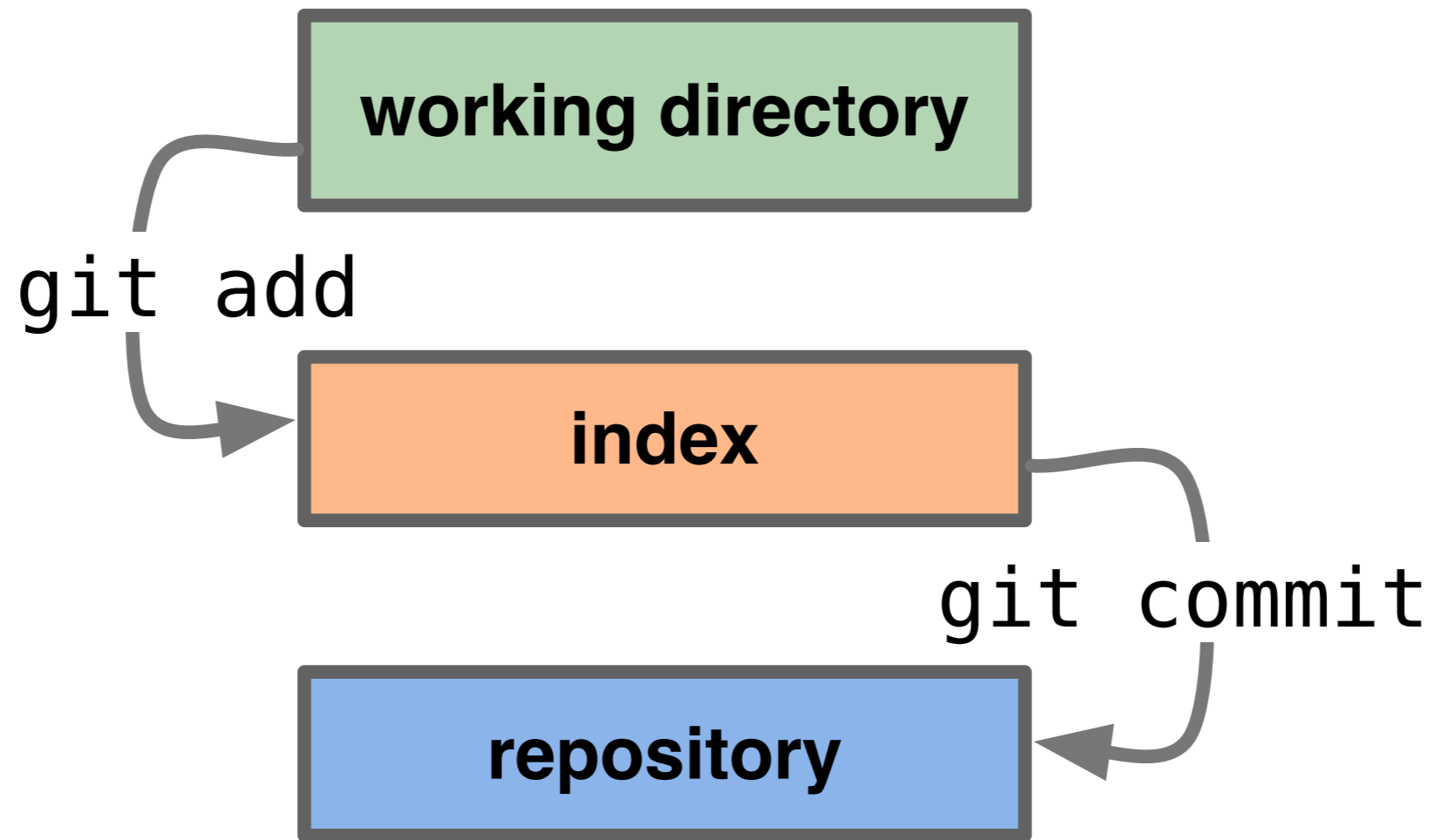
Stage the changes

Review your changes

Commit the changes

`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

```
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

A Basicer Workflow

Edit files

`vim / emacs / etc`

Stage and Commit

`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

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

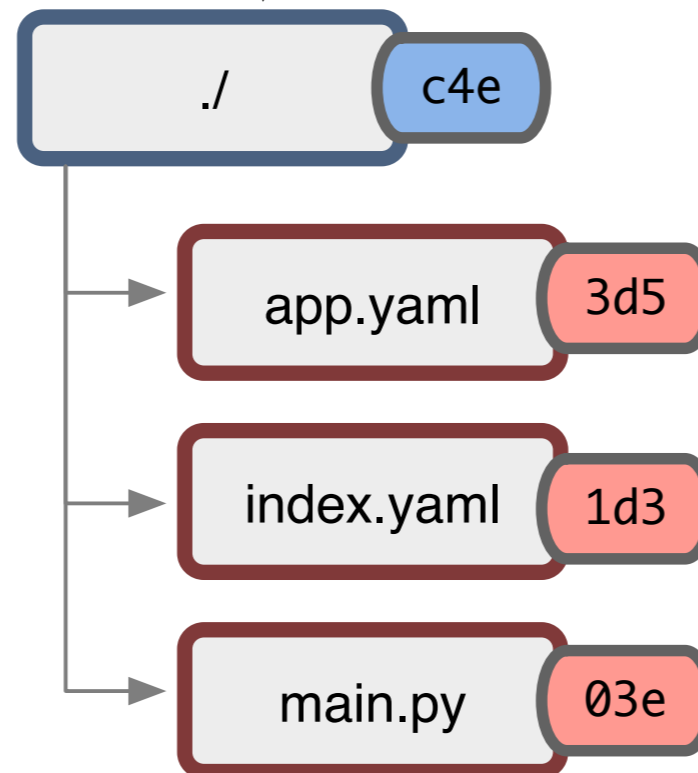
descriptive commit message
```

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

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

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

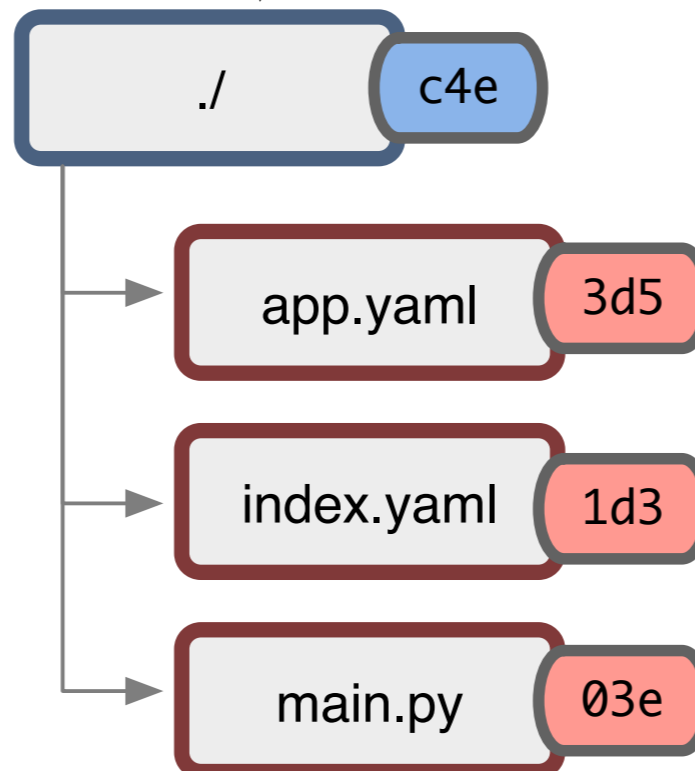
commit	size
tree	c4ec5
parent	a149e
author	Scott
committer	Scott
my commit message goes here and it is really, really cool	



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

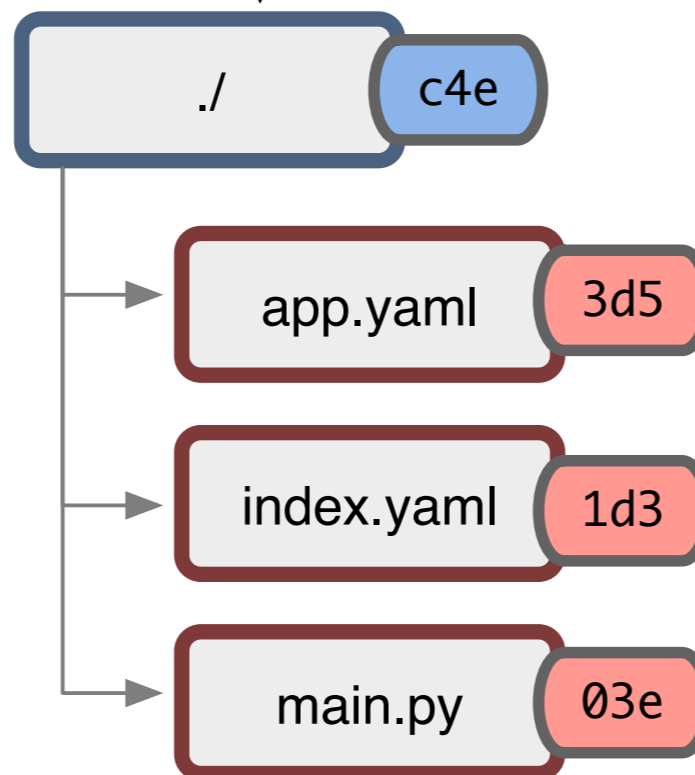
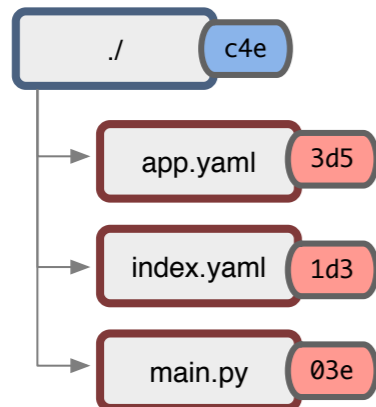


commit	size
tree	c4ec5
parent	a149e
author	Scott
committer	Scott
my commit message goes here and it is really, really cool	



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

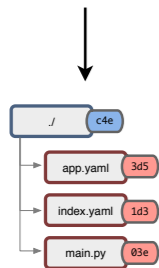
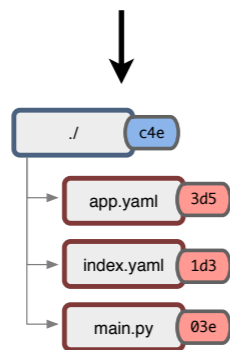
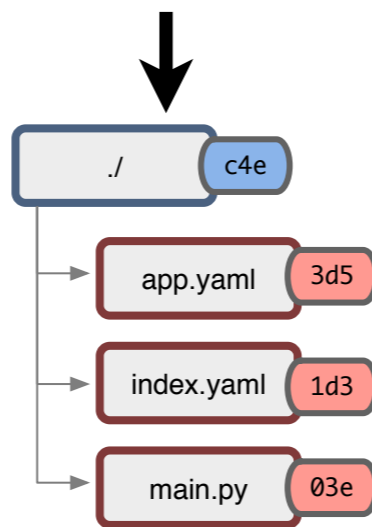
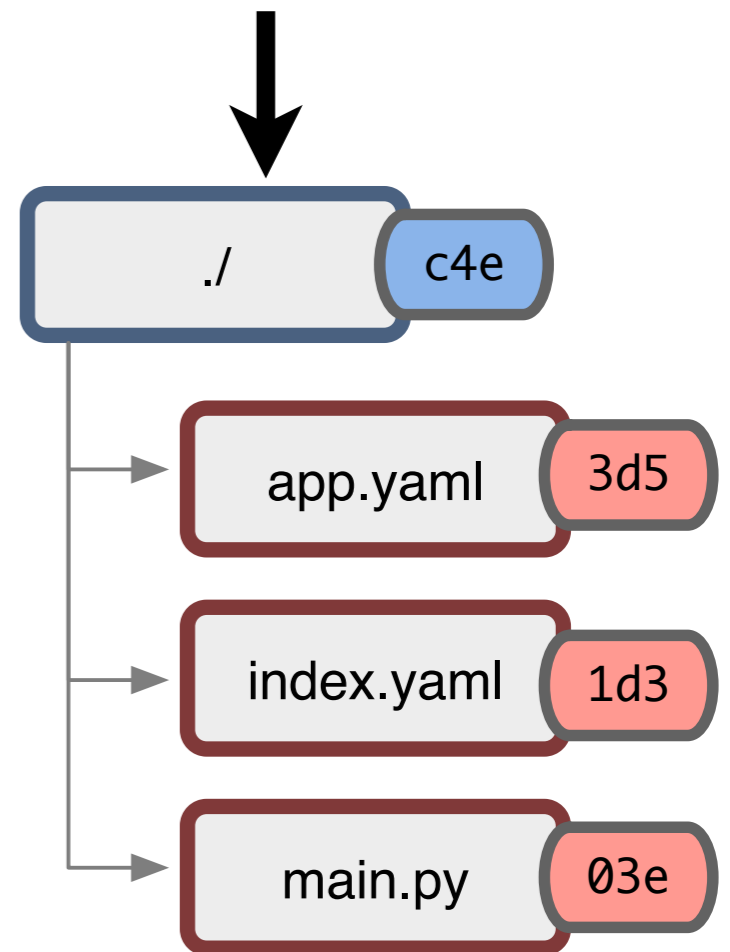


commit	size
tree	c4ec5
parent	a149e
author	Scott
committer	Scott
my commit message goes here and it is really, really cool	

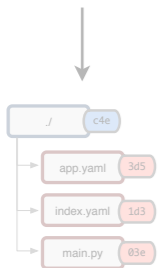
commit	size
tree	2de54
parent	38def
author	Scott
committer	Scott
this is the previous commit and I am very proud of it	

commit	size
tree	2fe65
parent	90ecd
author	Scott
committer	Scott
this is the commit before that and I'm not sure why	

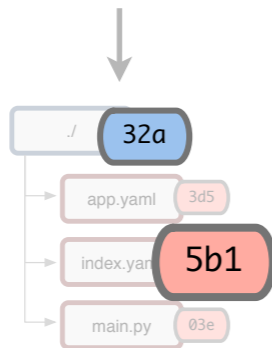
commit	size
tree	2fe65
parent	90ecd
author	Scott
committer	Scott
this is the commit before that and I'm not sure why	



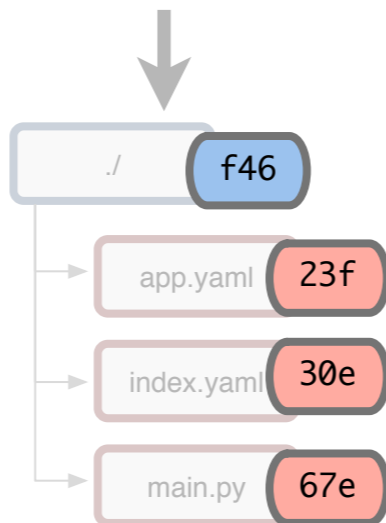
commit	size
tree	
parent	
author	
committer	Scott
this is the commit before that and I'm not sure why	



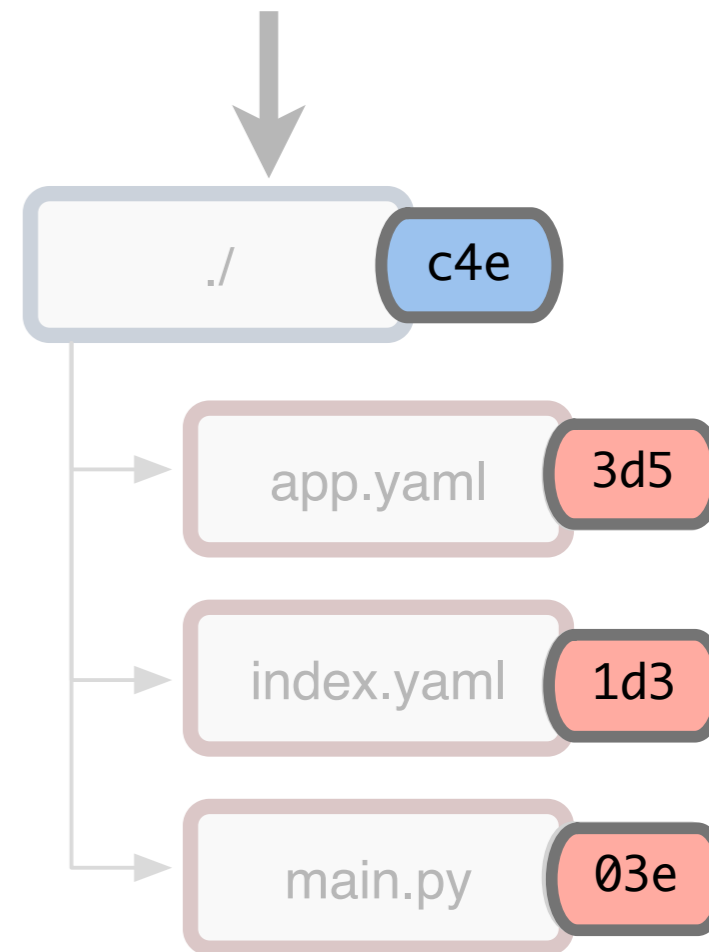
commit	size
tree	2fe65
parent	38d
author	Scott
committer	Scott
this is the commit before that and I'm not sure why	



commit	size
tree	2de54
parent	a14
author	Scott
committer	Scott
this is the previous commit and I am very proud of it	



commit	size
tree	c4ec5
parent	77d
author	Scott
committer	Scott
my commit message goes here and it is really, really cool	



Repository

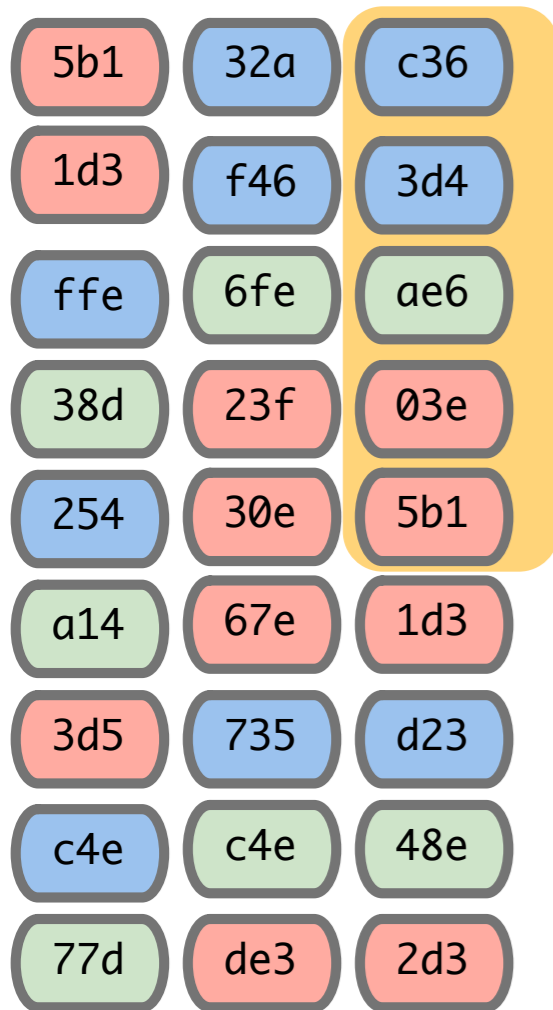
5b1	32a	c36
1d3	f46	3d4
ffe	6fe	ae6
38d	23f	03e
254	30e	5b1
a14	67e	1d3
3d5	735	d23
c4e	c4e	48e
77d	de3	2d3

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

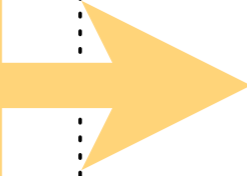


```
git checkout ae635
```

Repository

Index

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



./	3d4
./Rakefile	1d3
./README	03e
./lib/	c36
./lib/simplegit.rb	5b1

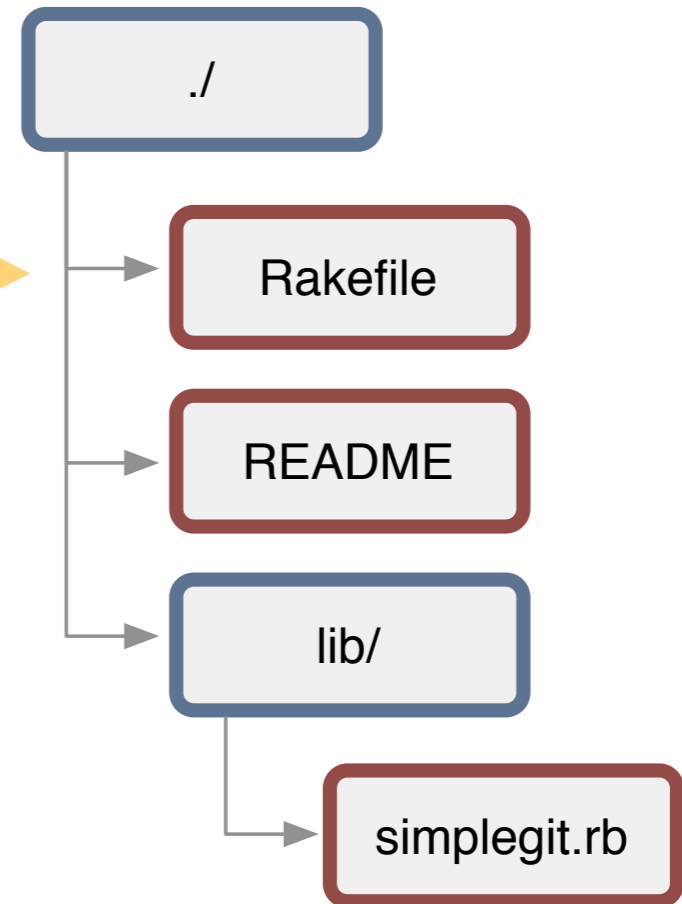
Repository

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

Index

./	3d4
./Rakefile	1d3
./README	03e
./lib/	c36
./lib/simplegit.rb	5b1

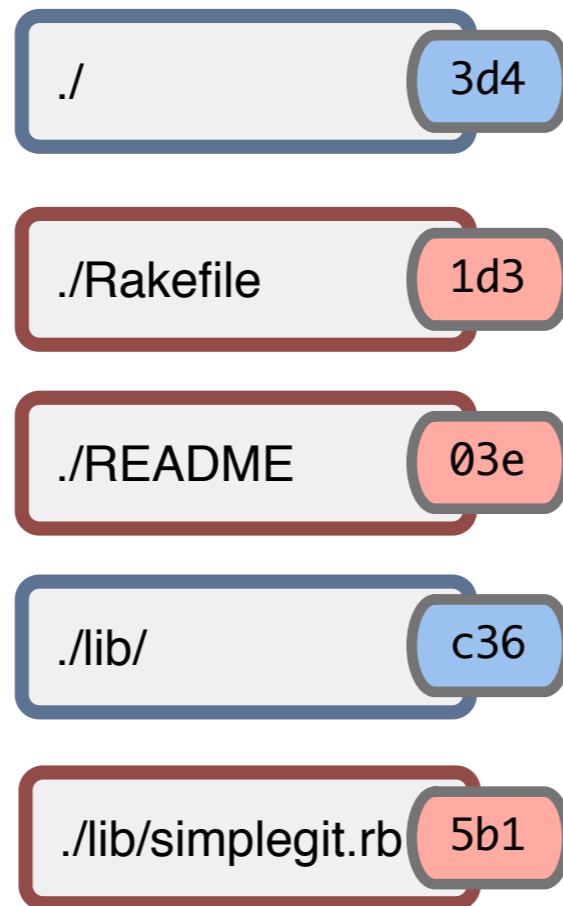
Working Directory



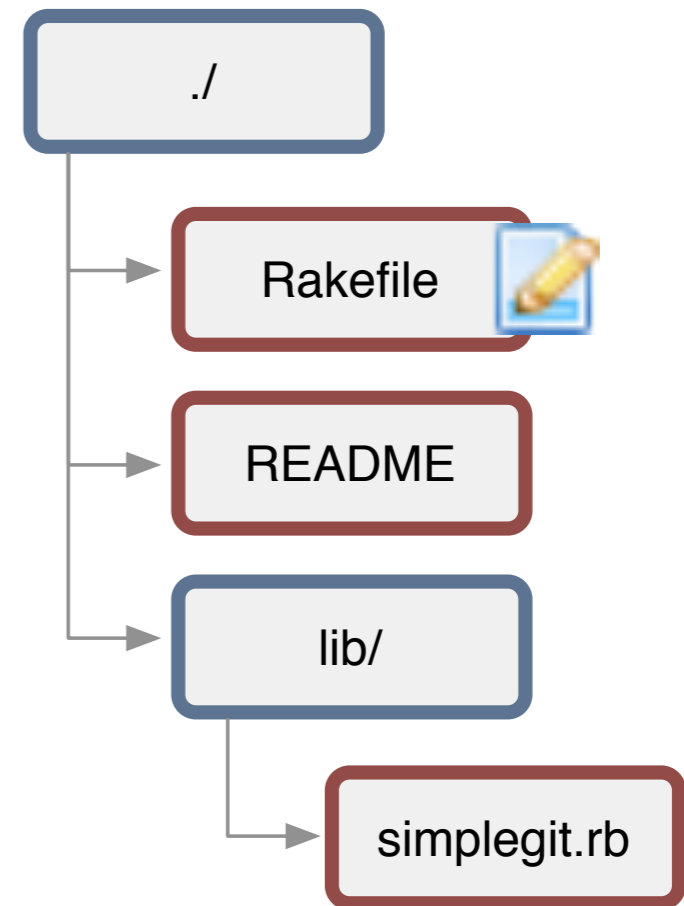
Repository



Index



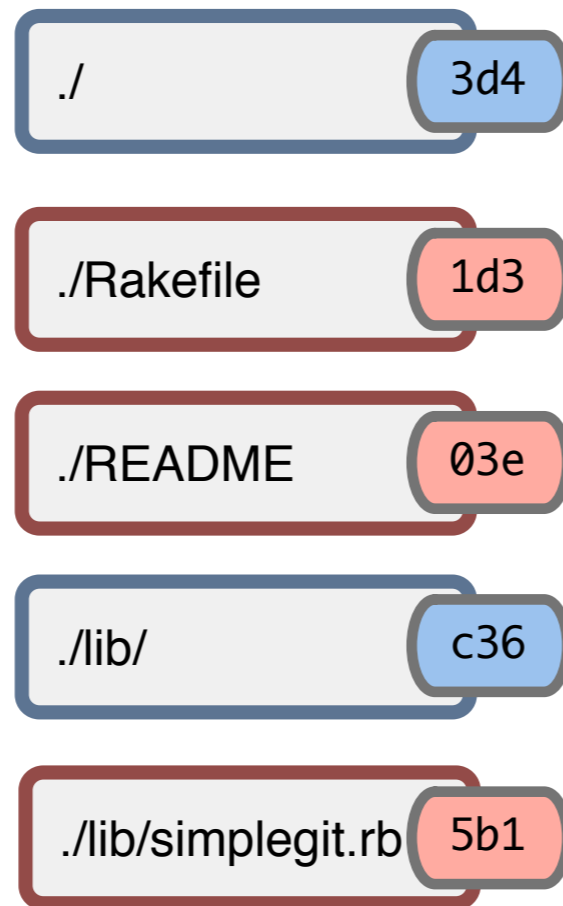
Working Directory



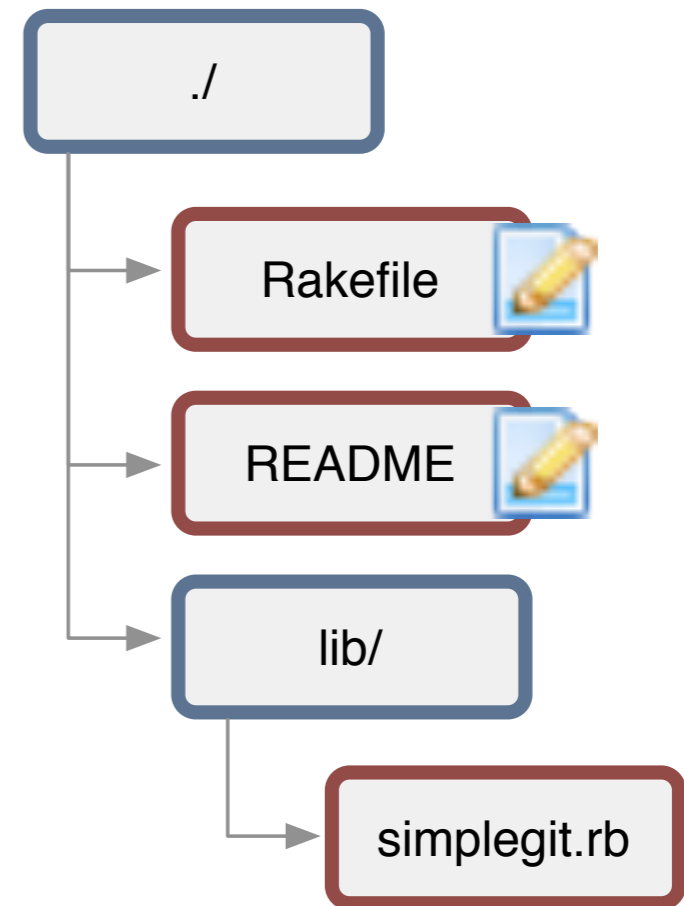
Repository



Index



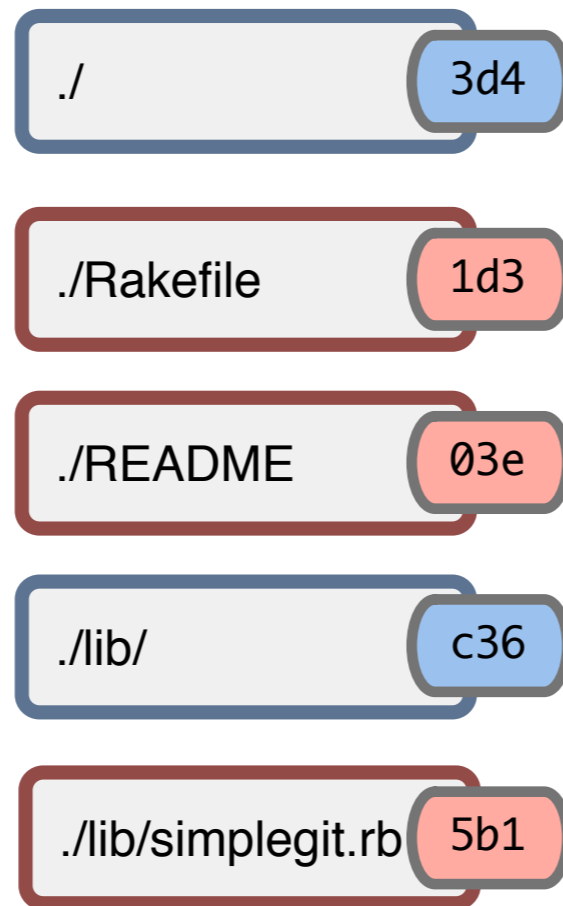
Working Directory



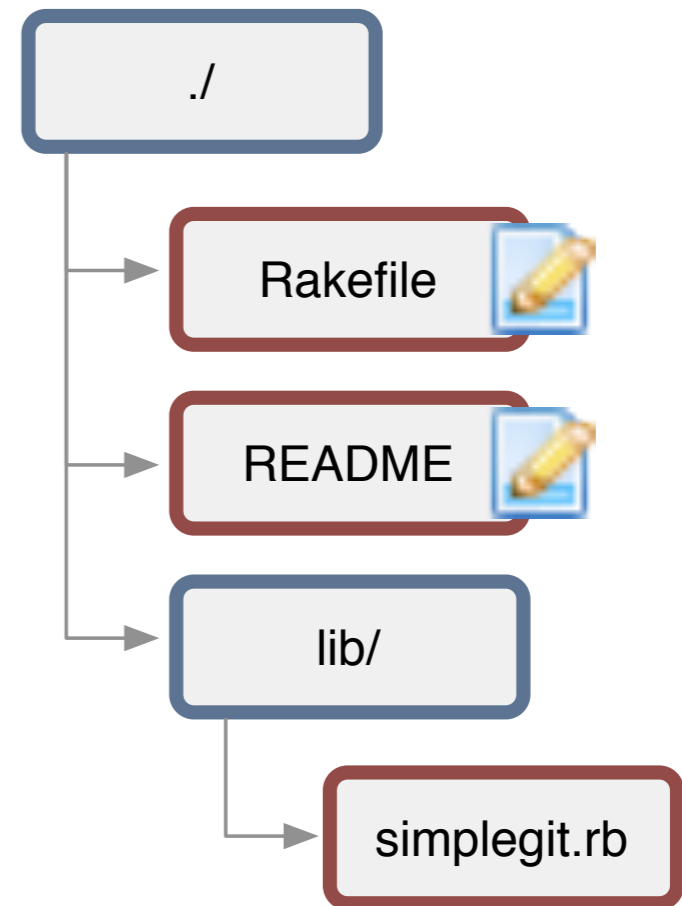
Repository



Index



Working Directory

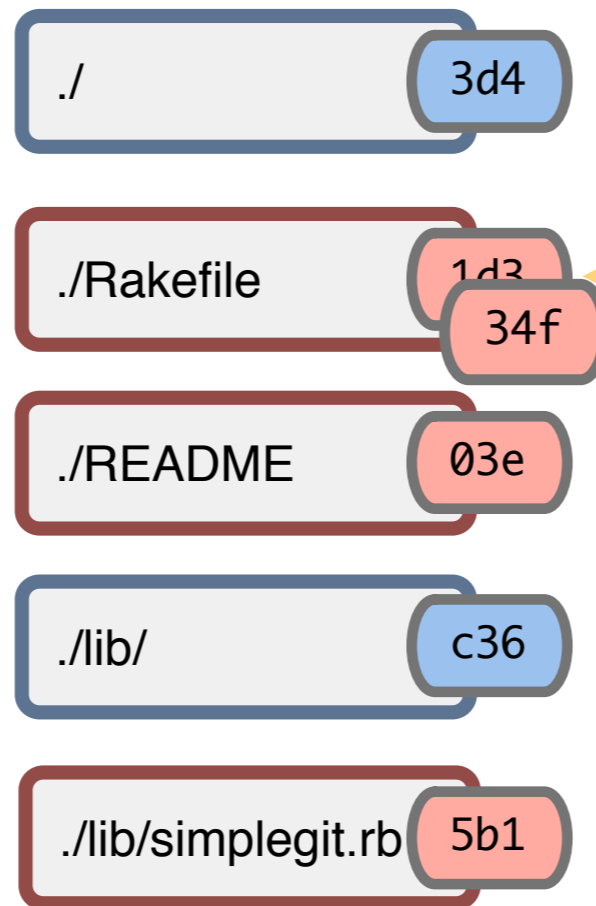


git add

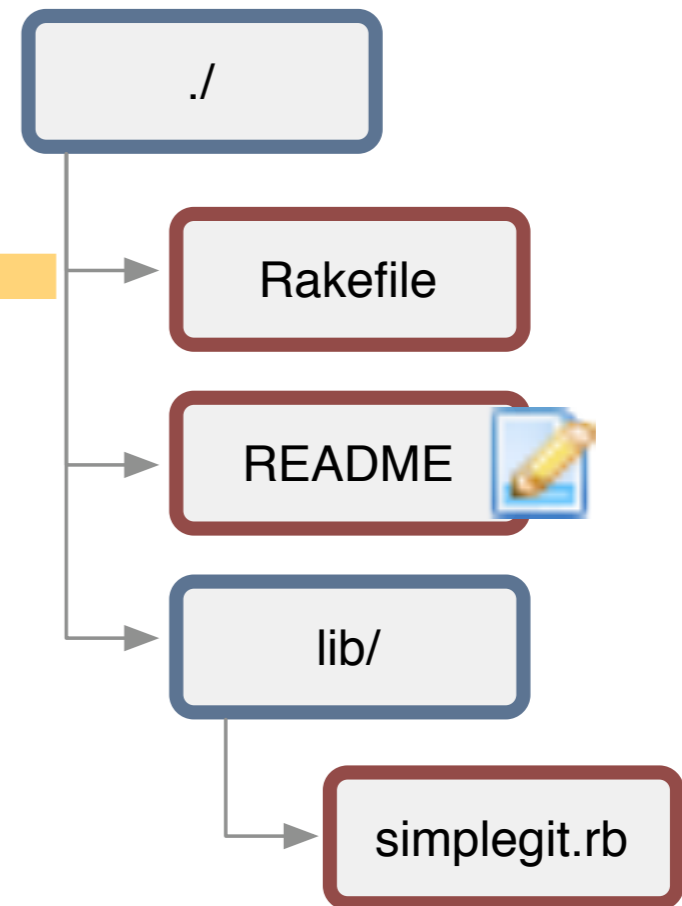
Repository



Index



Working Directory

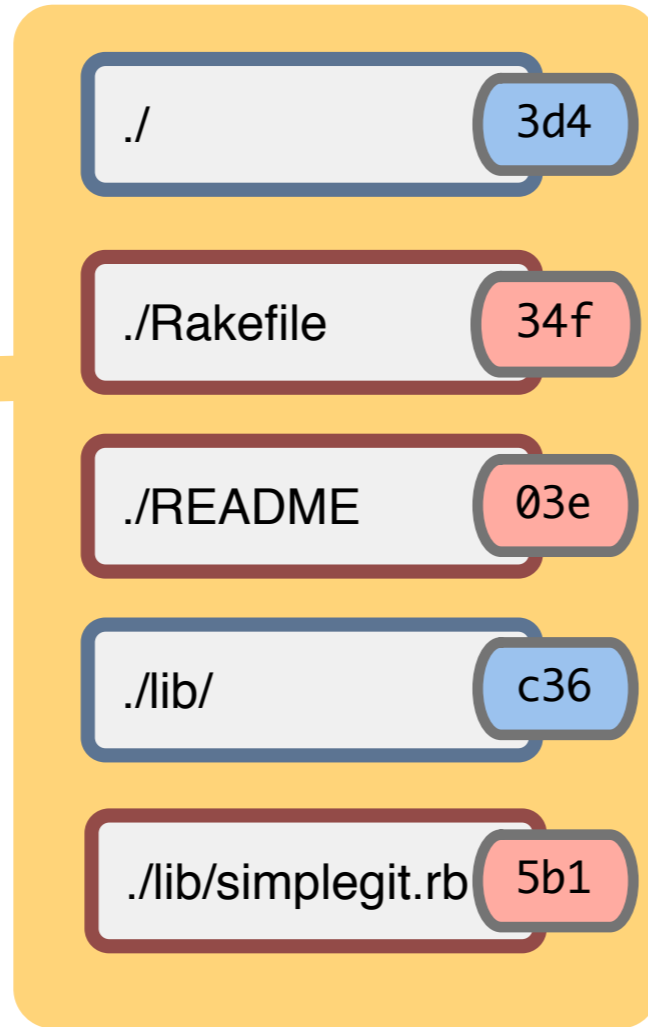


git add

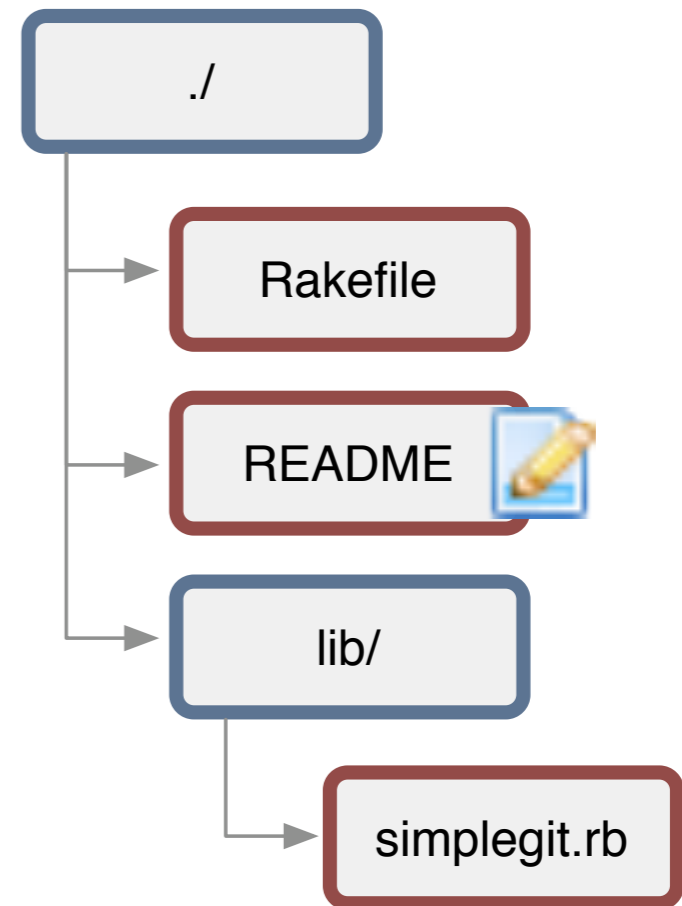
Repository



Index



Working Directory

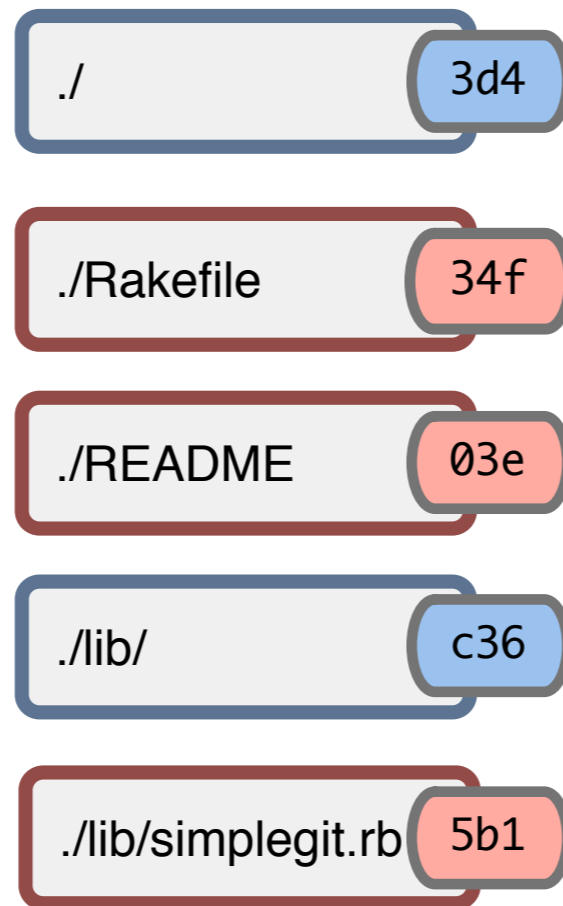


git commit

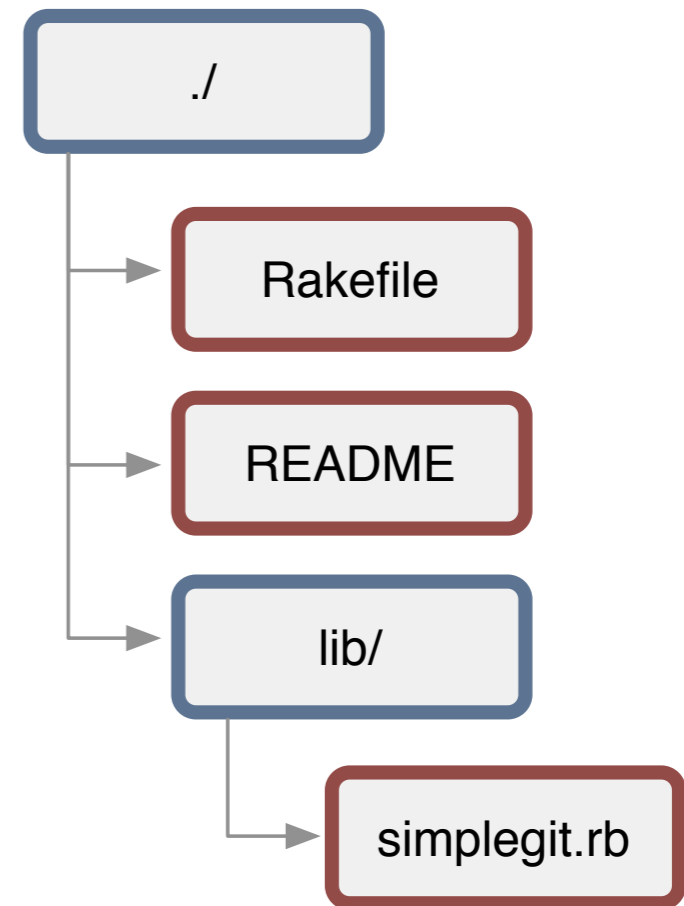
Repository



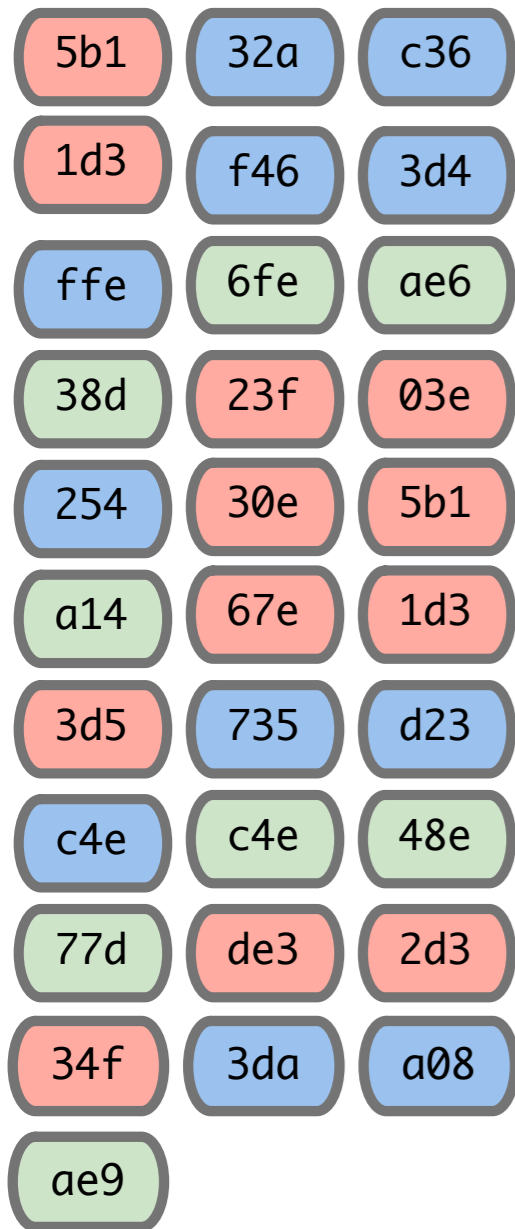
Index



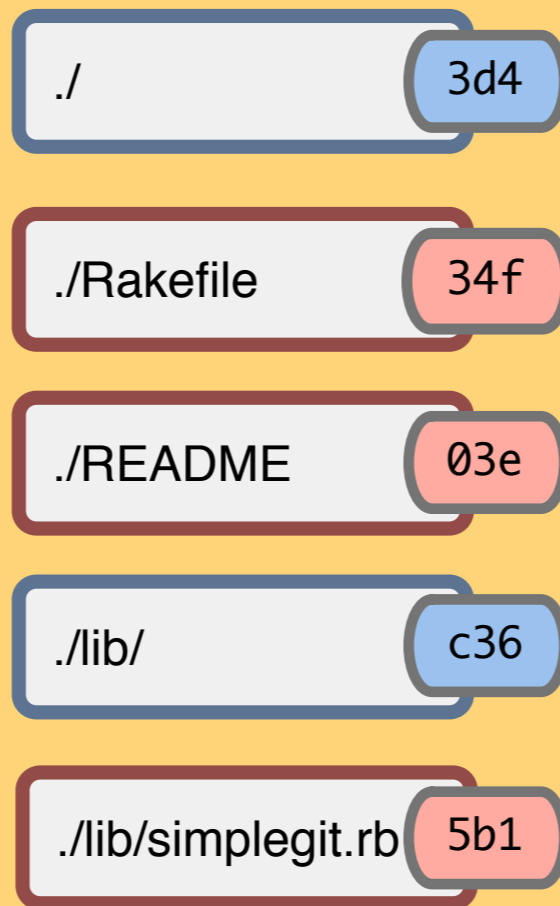
Working Directory



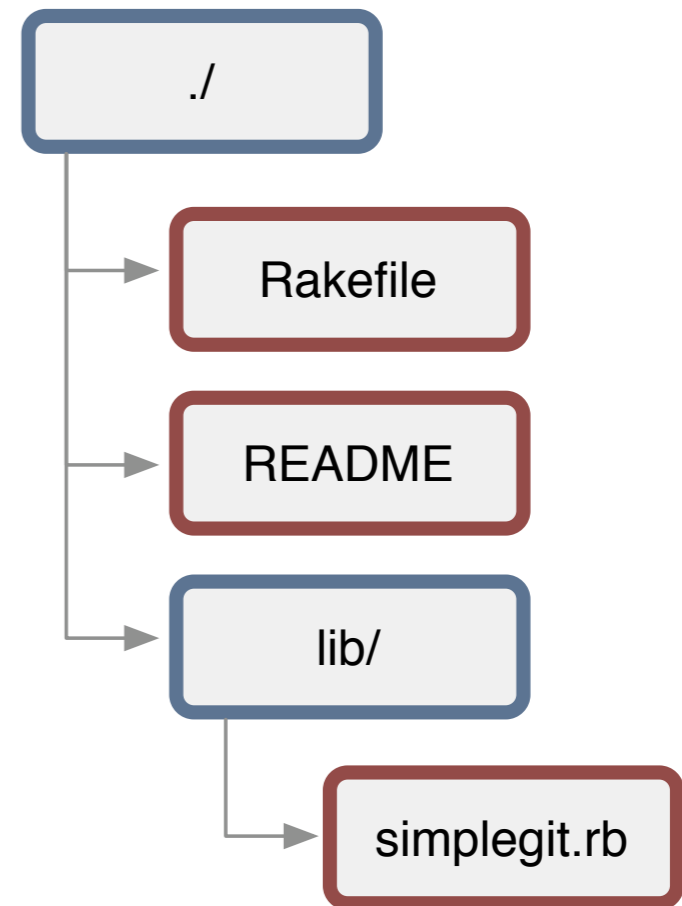
Repository



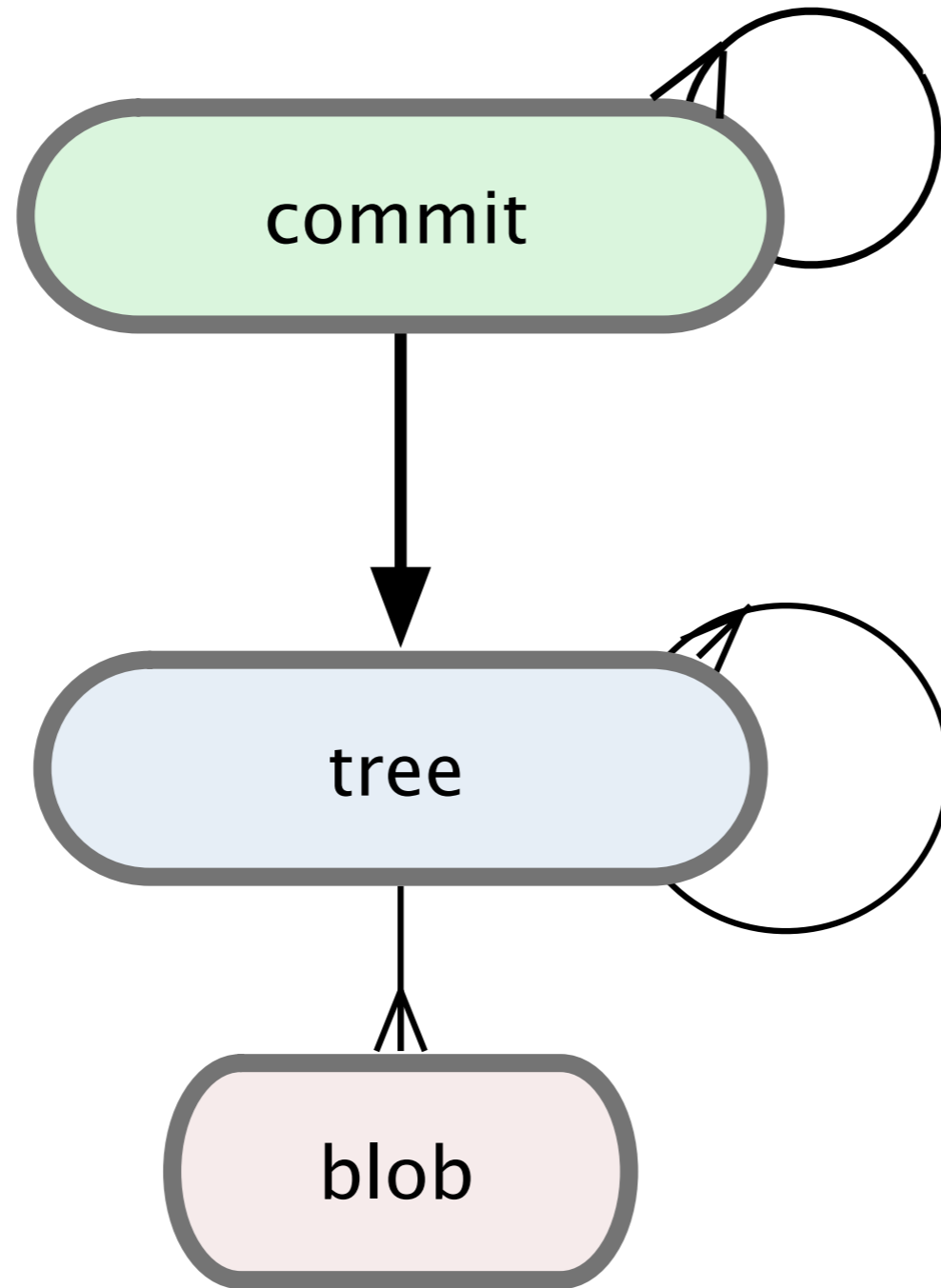
Index



Working Directory



object model

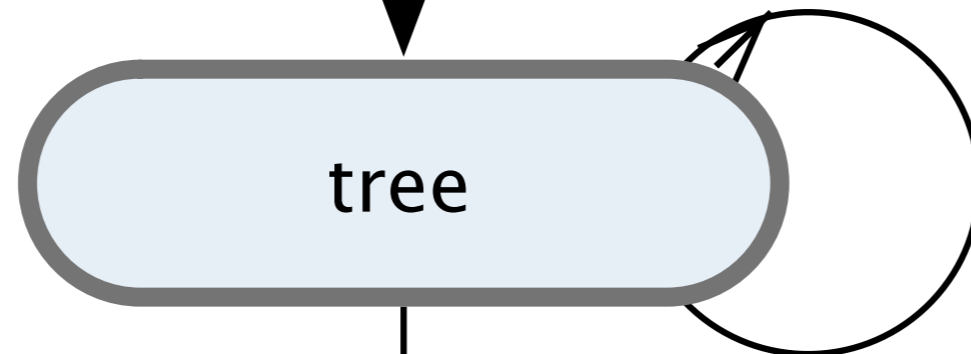


object model

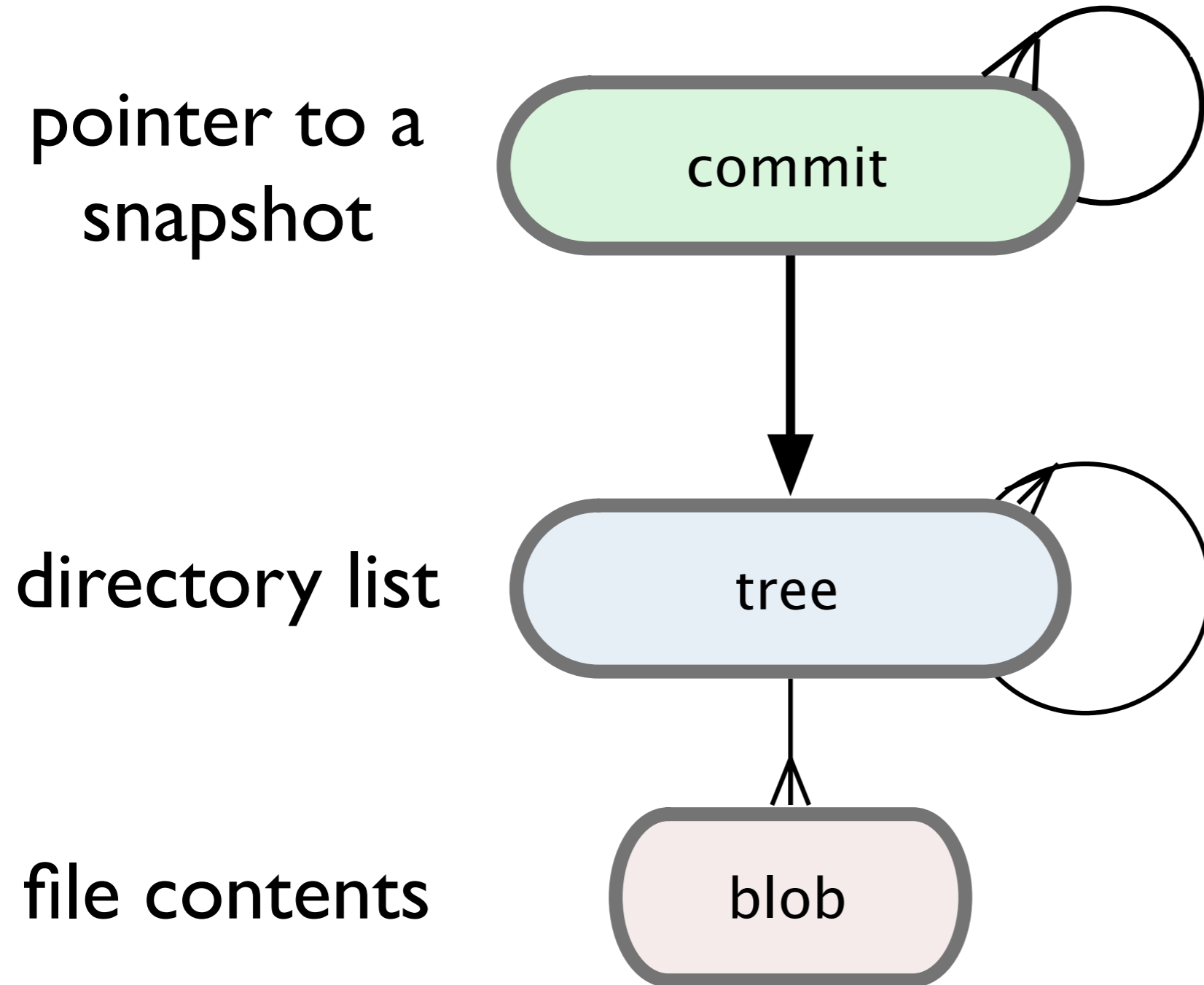
pointer to a
snapshot

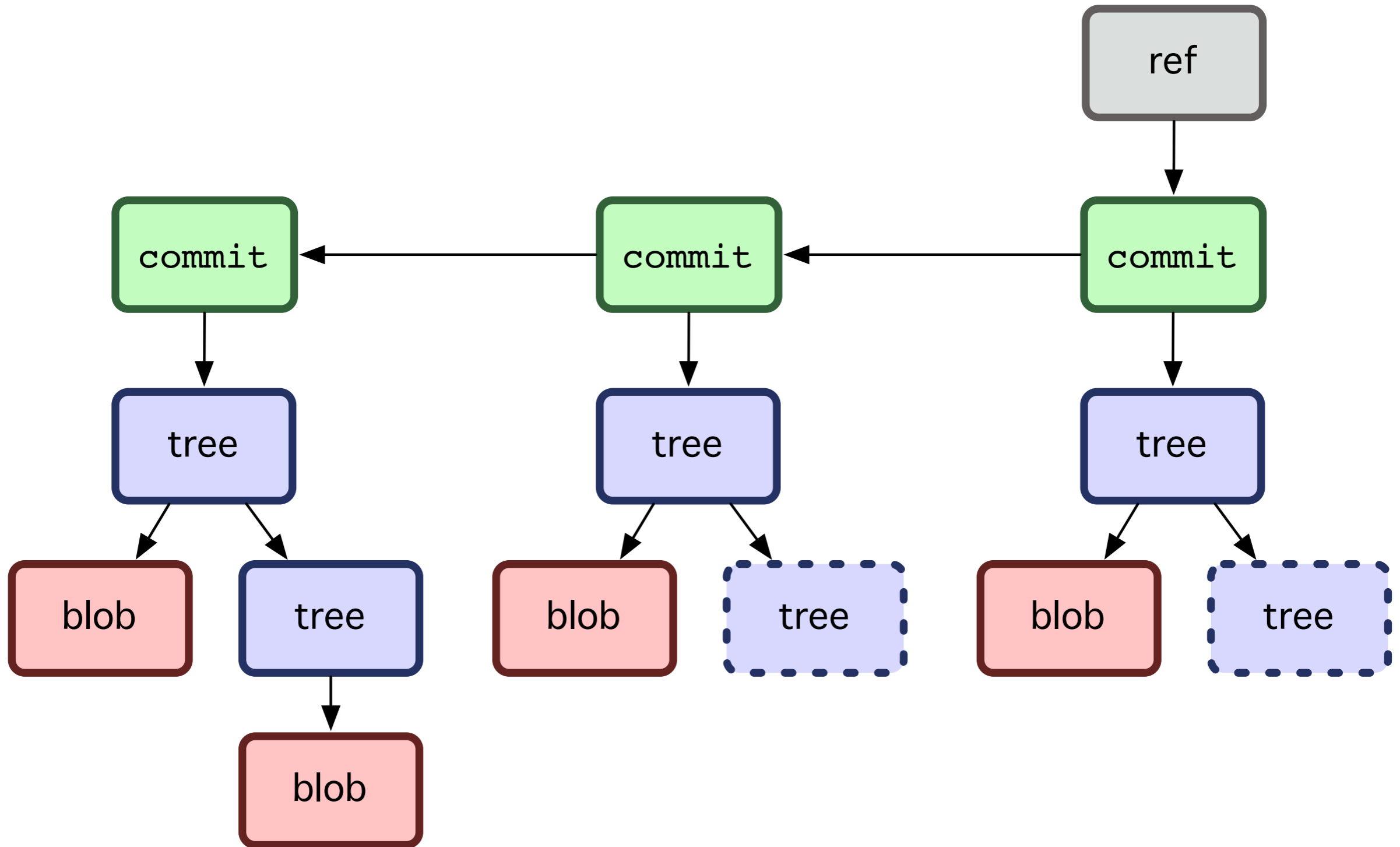


directory list



file contents



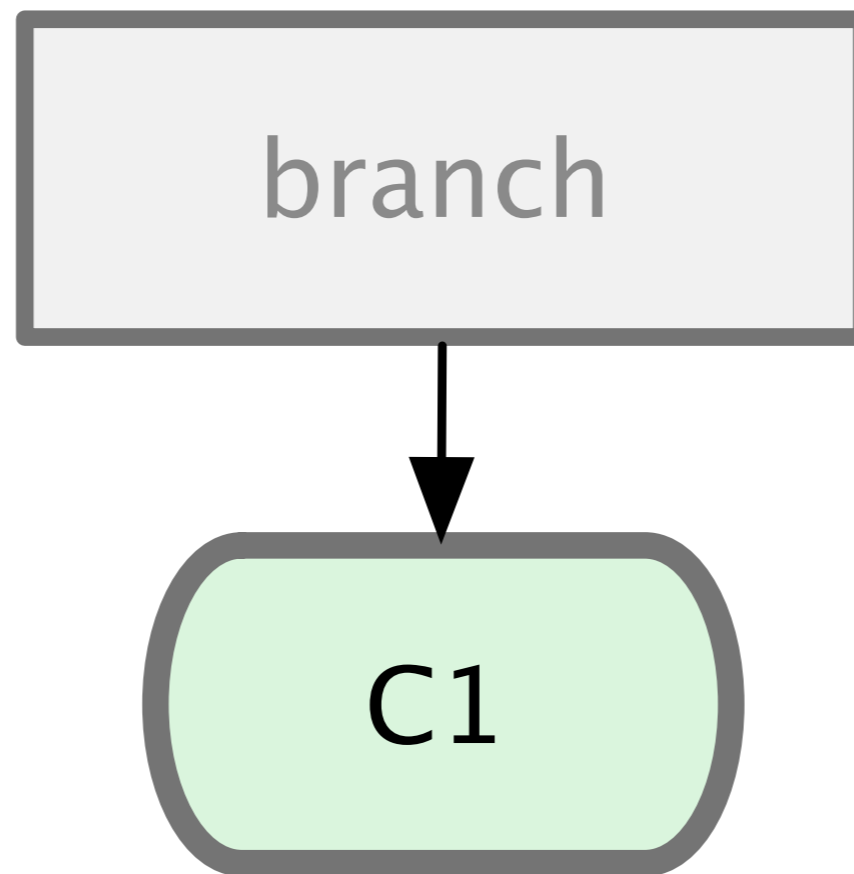


Branching and Merging

branches

branches

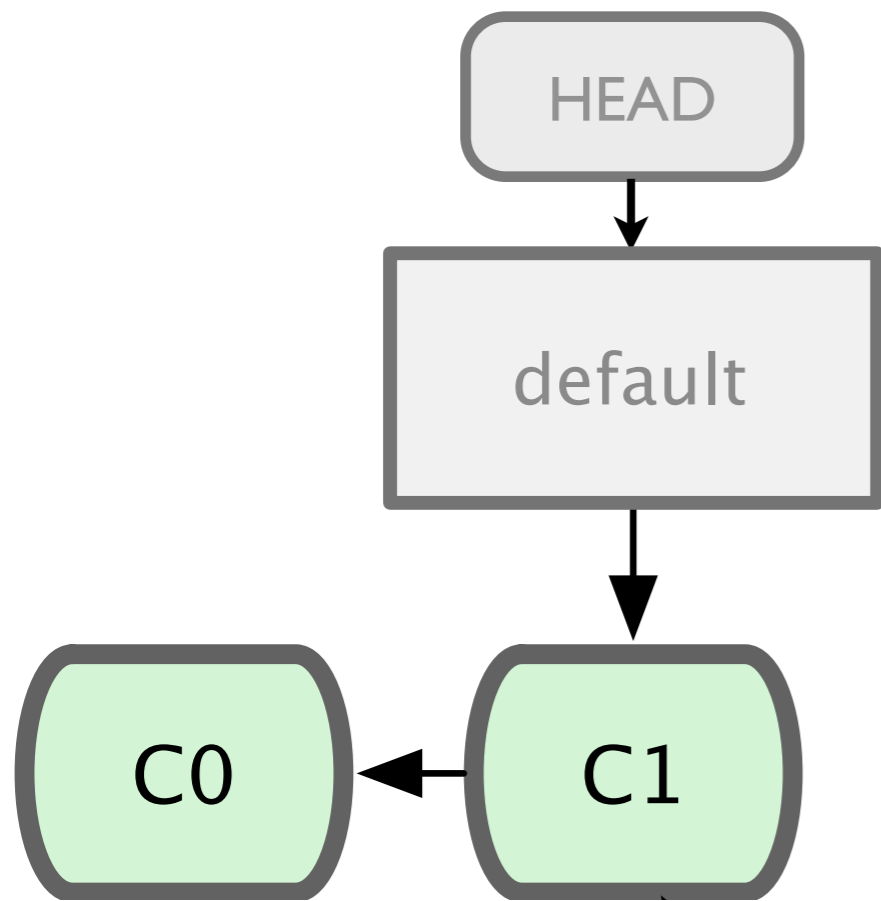
lightweight, movable
pointers to a commit

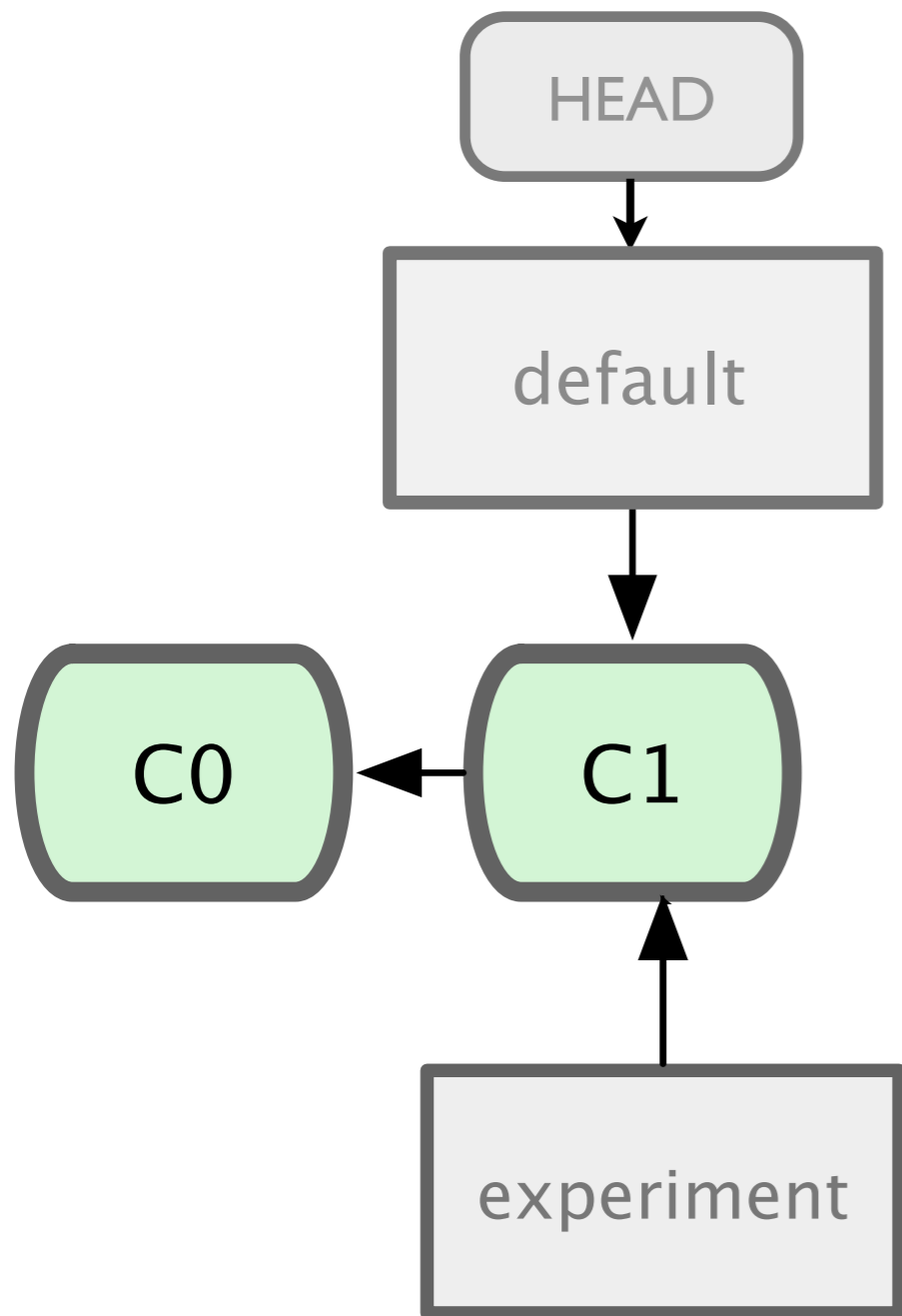


branching

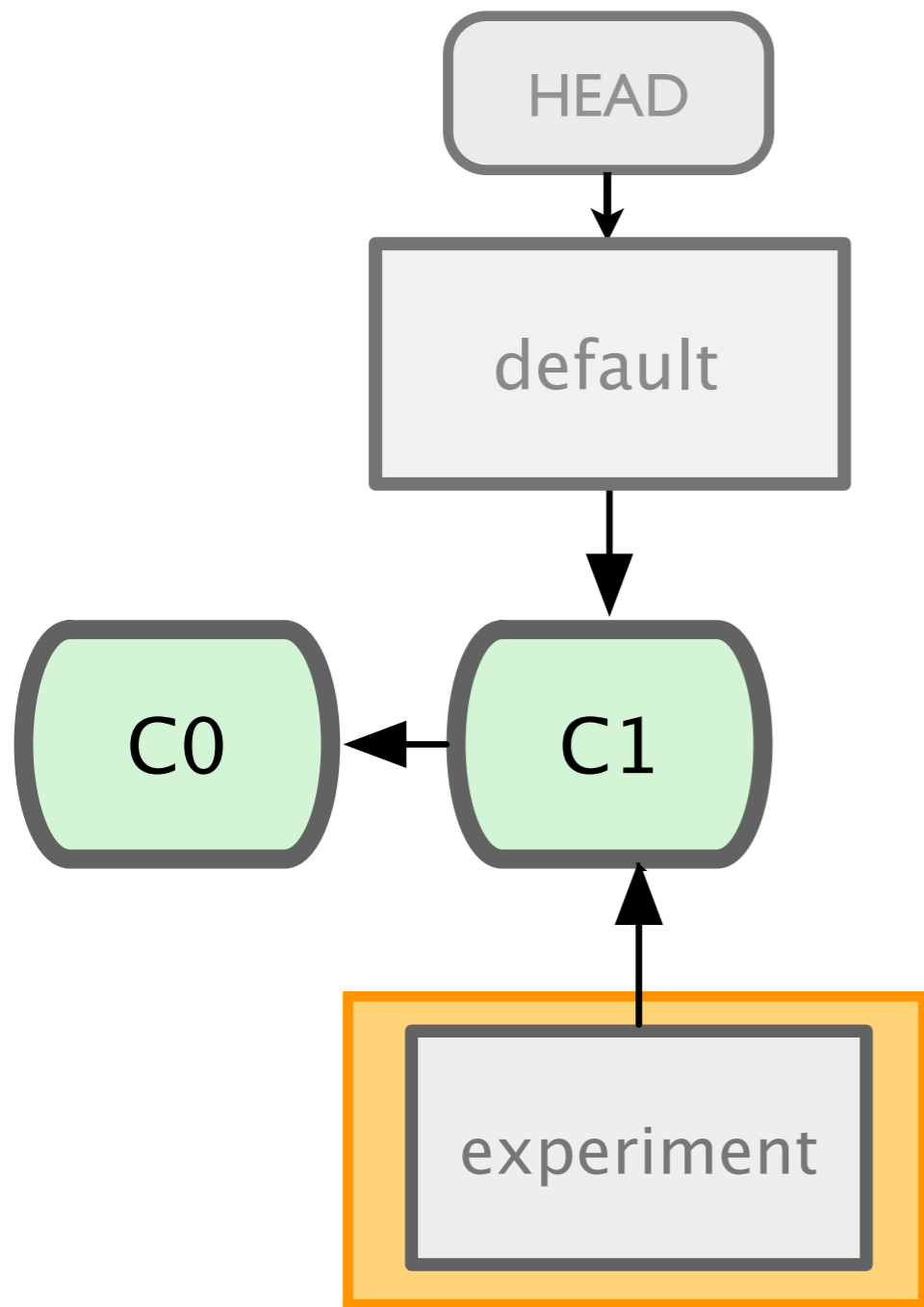
git branch

`git checkout`

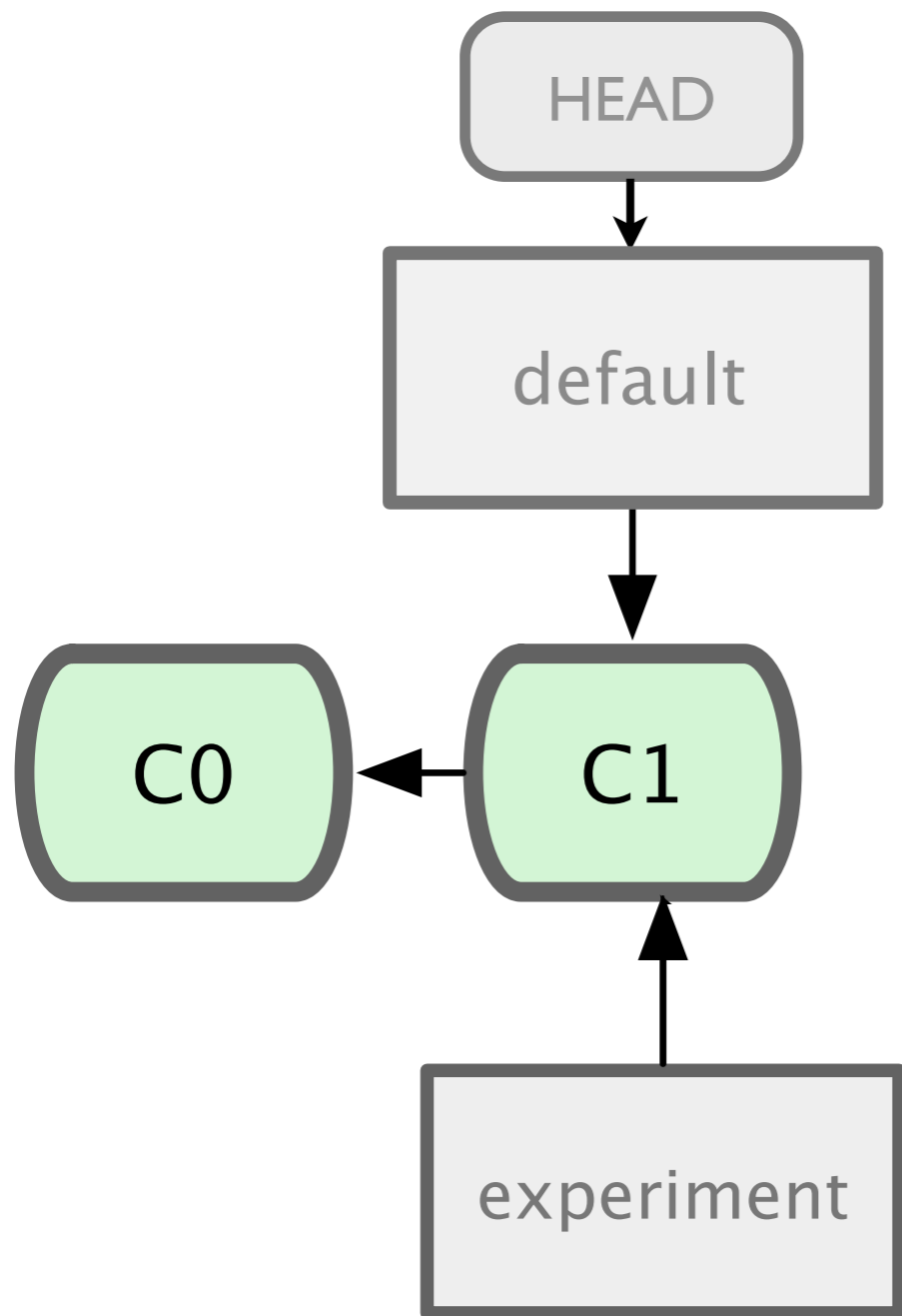




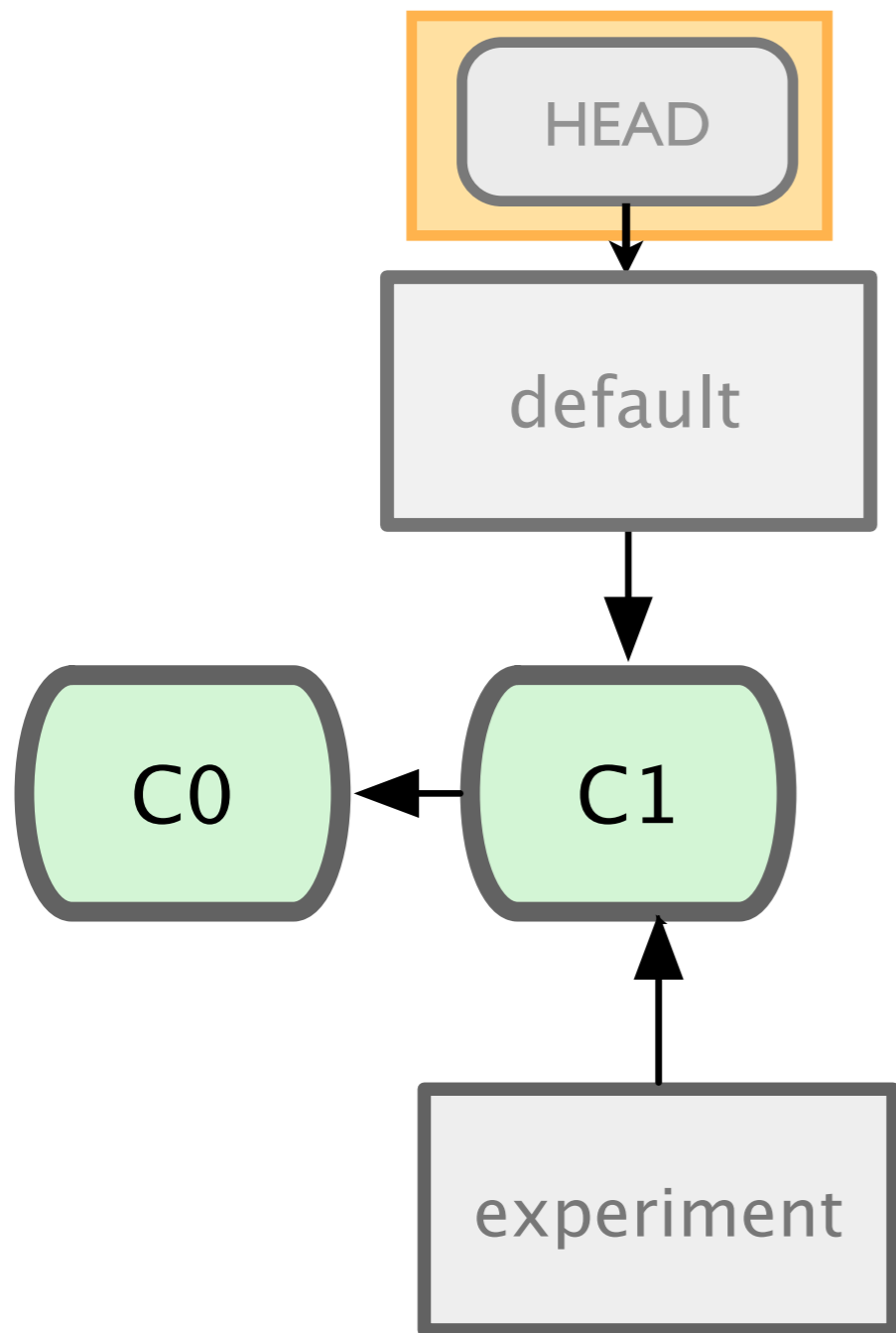
git branch experiment



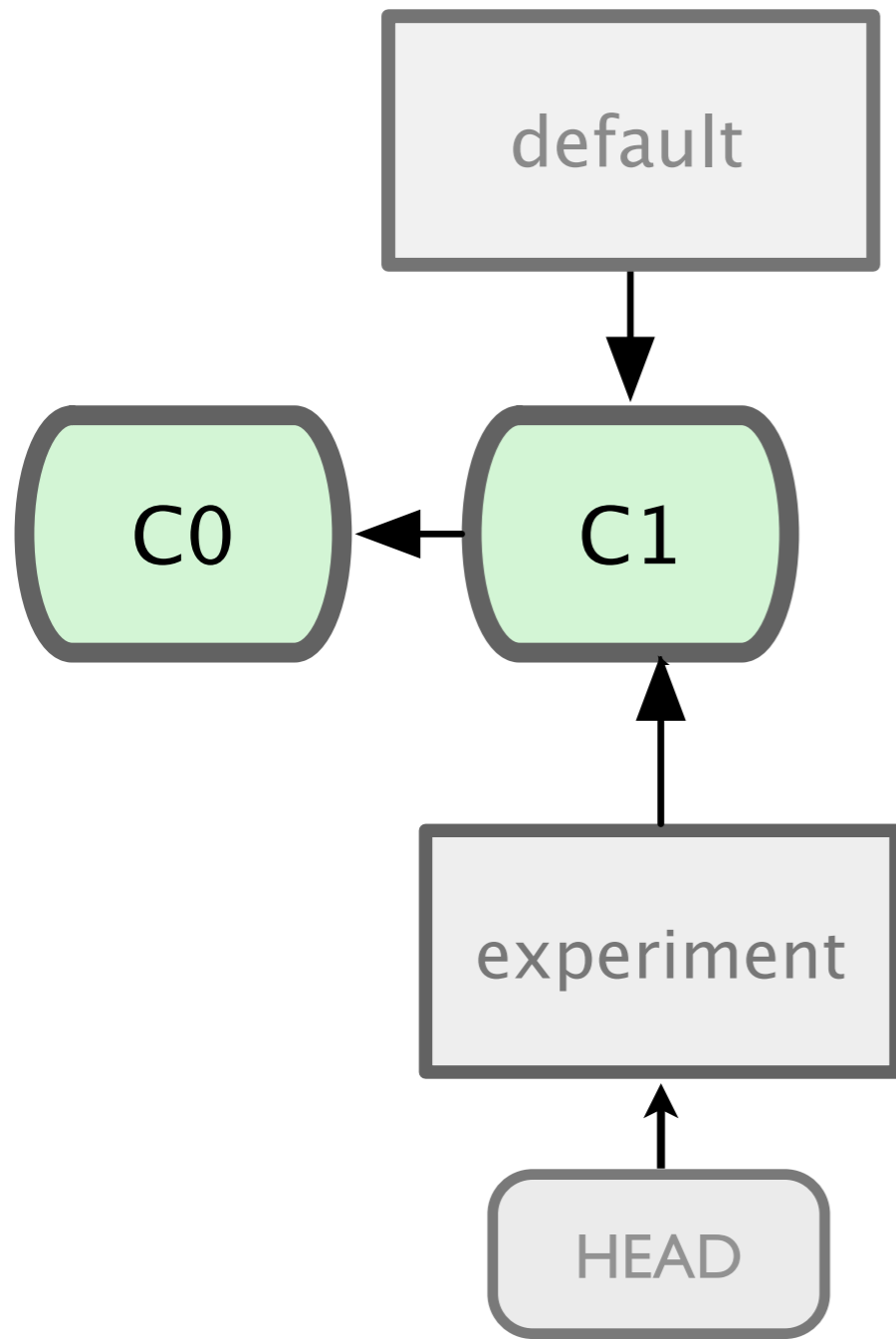
git branch experiment



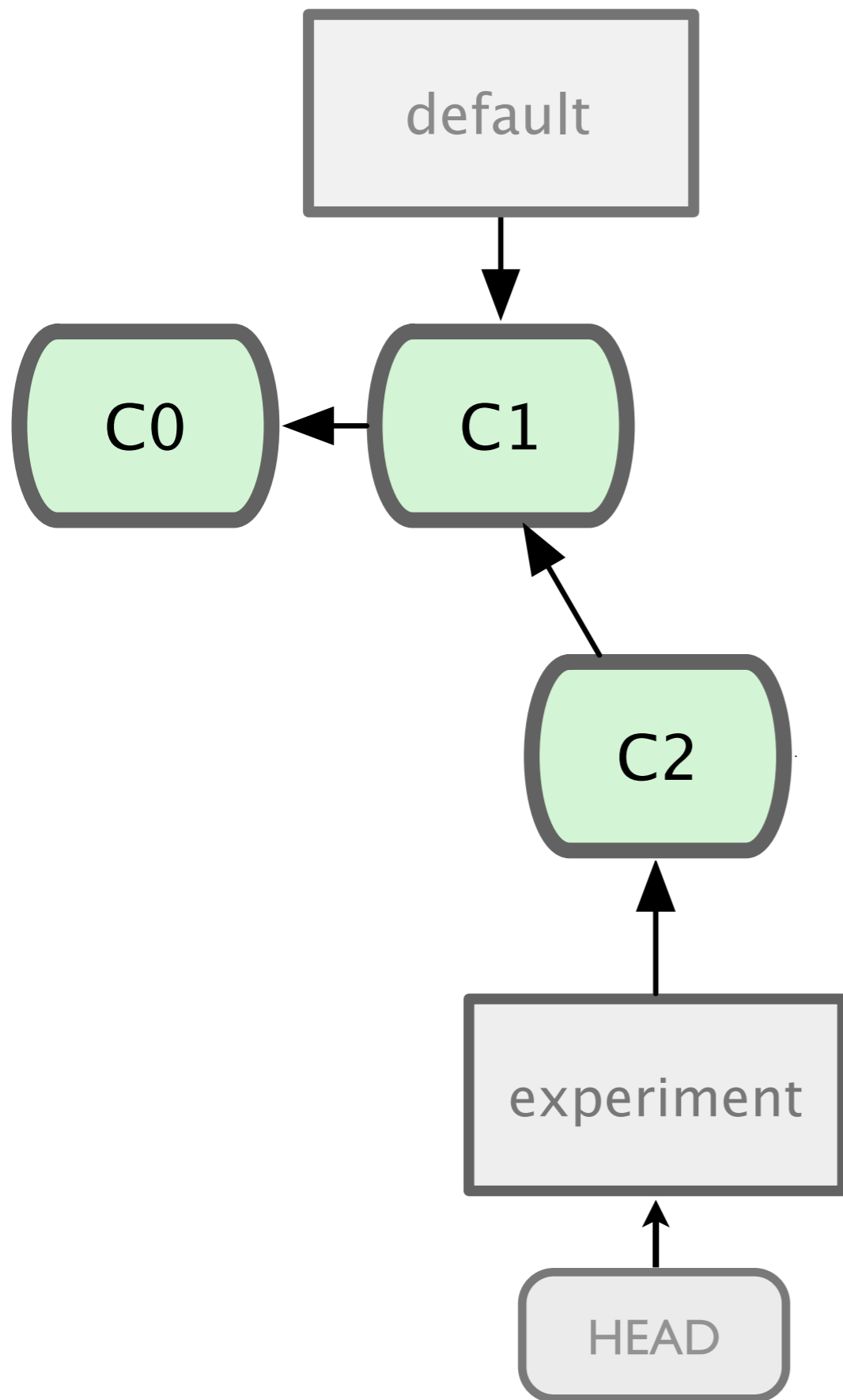
\$ git branch
* default
experiment



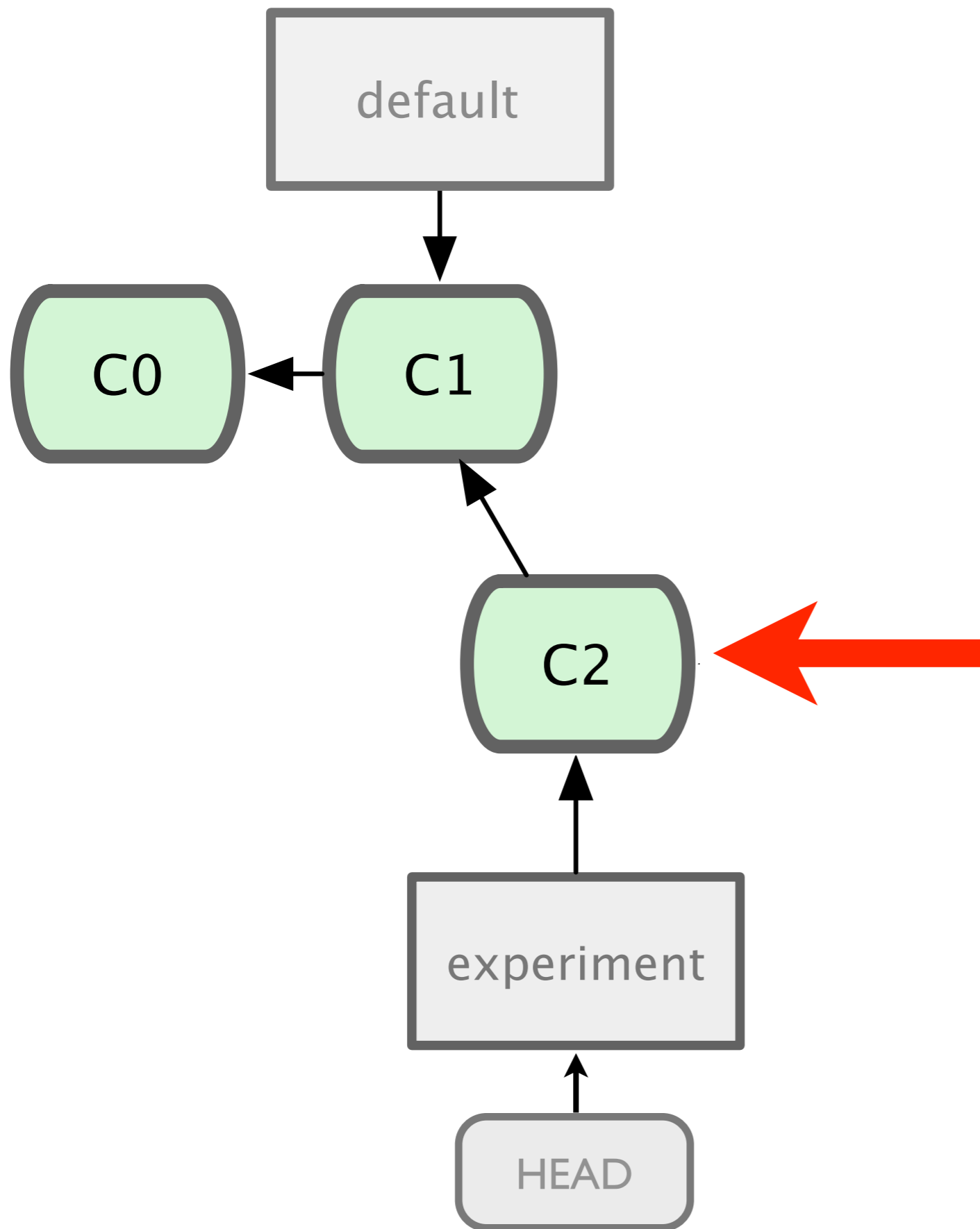
\$ git branch
* default
experiment



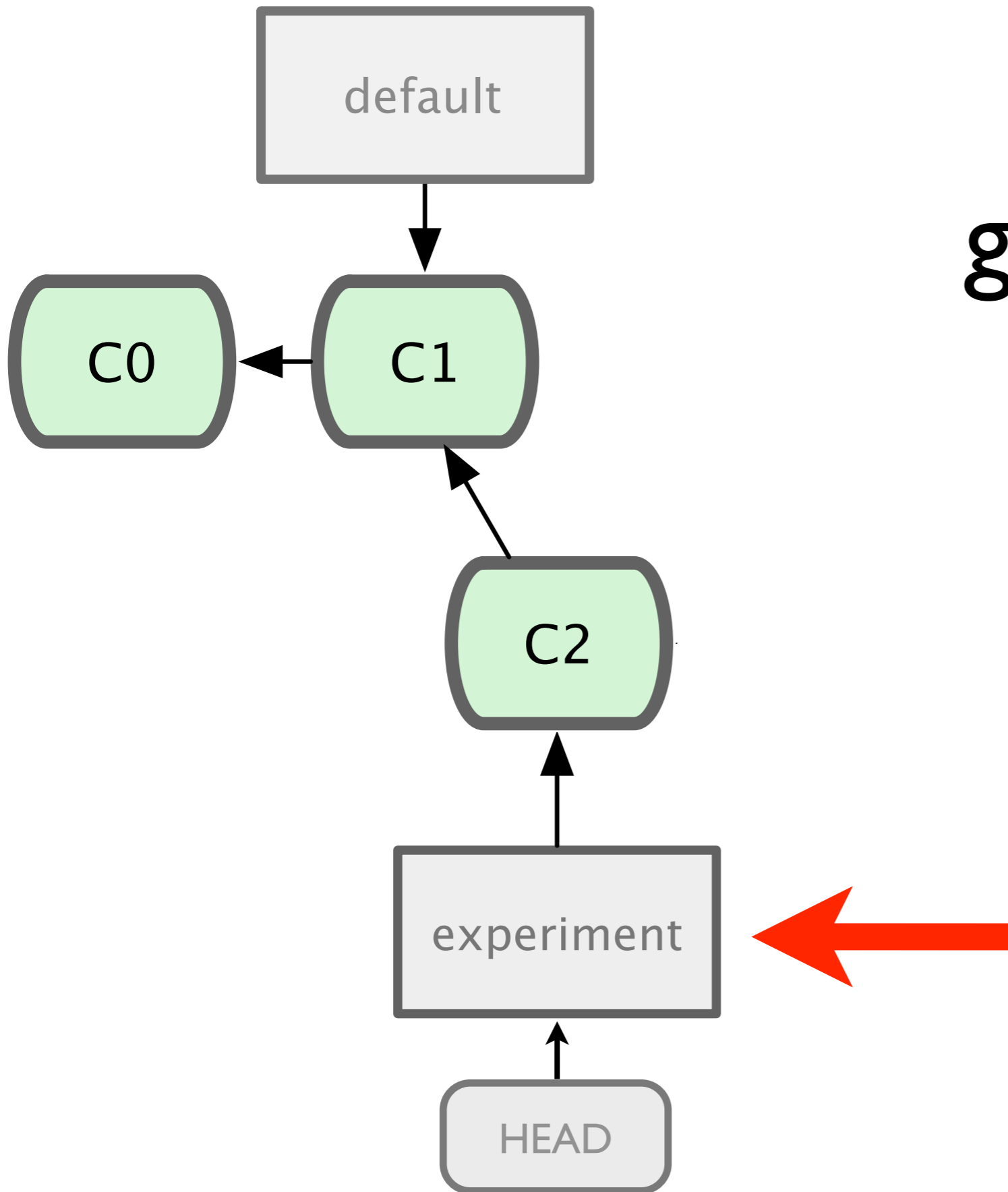
git checkout experiment



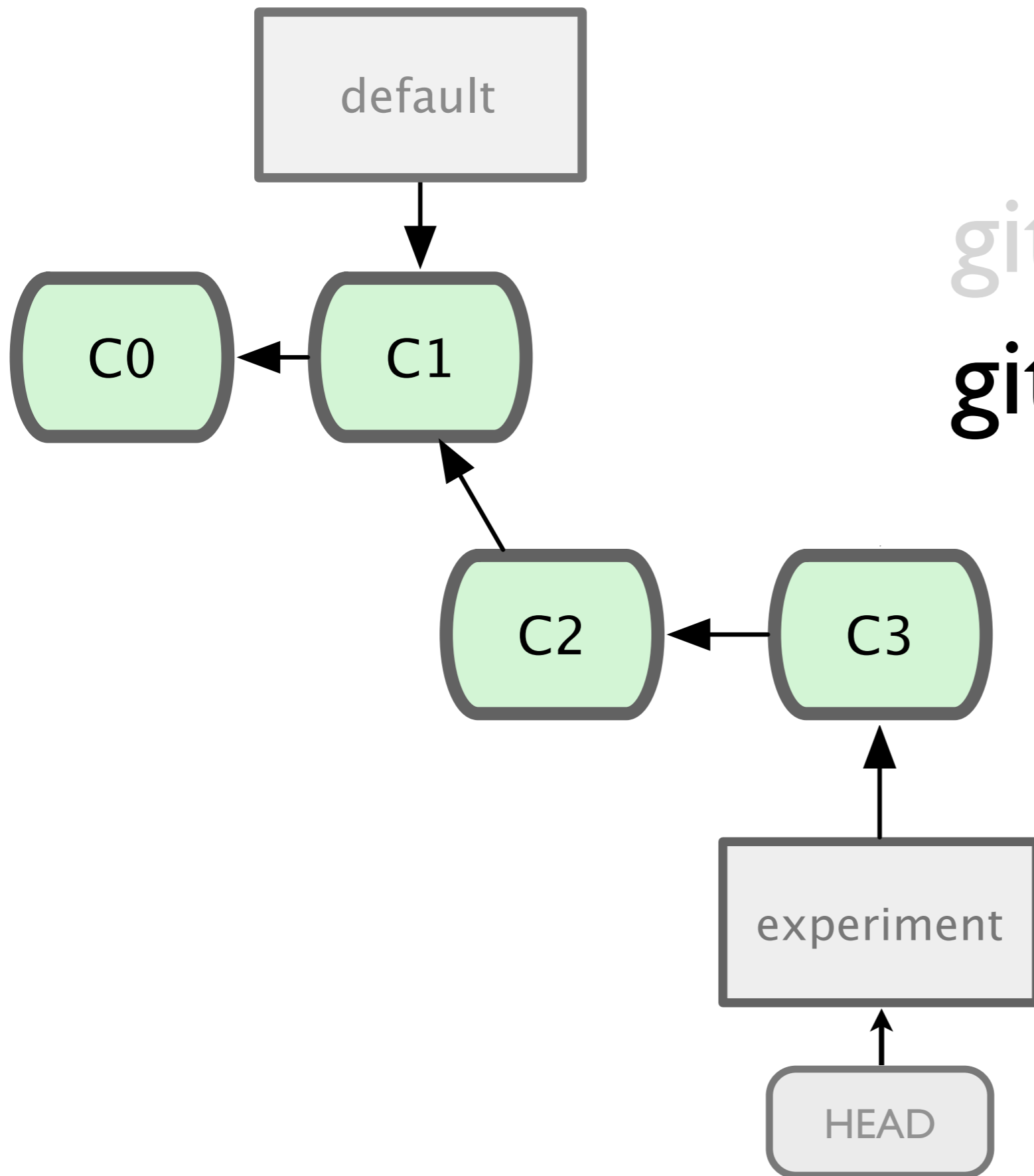
git commit



git commit

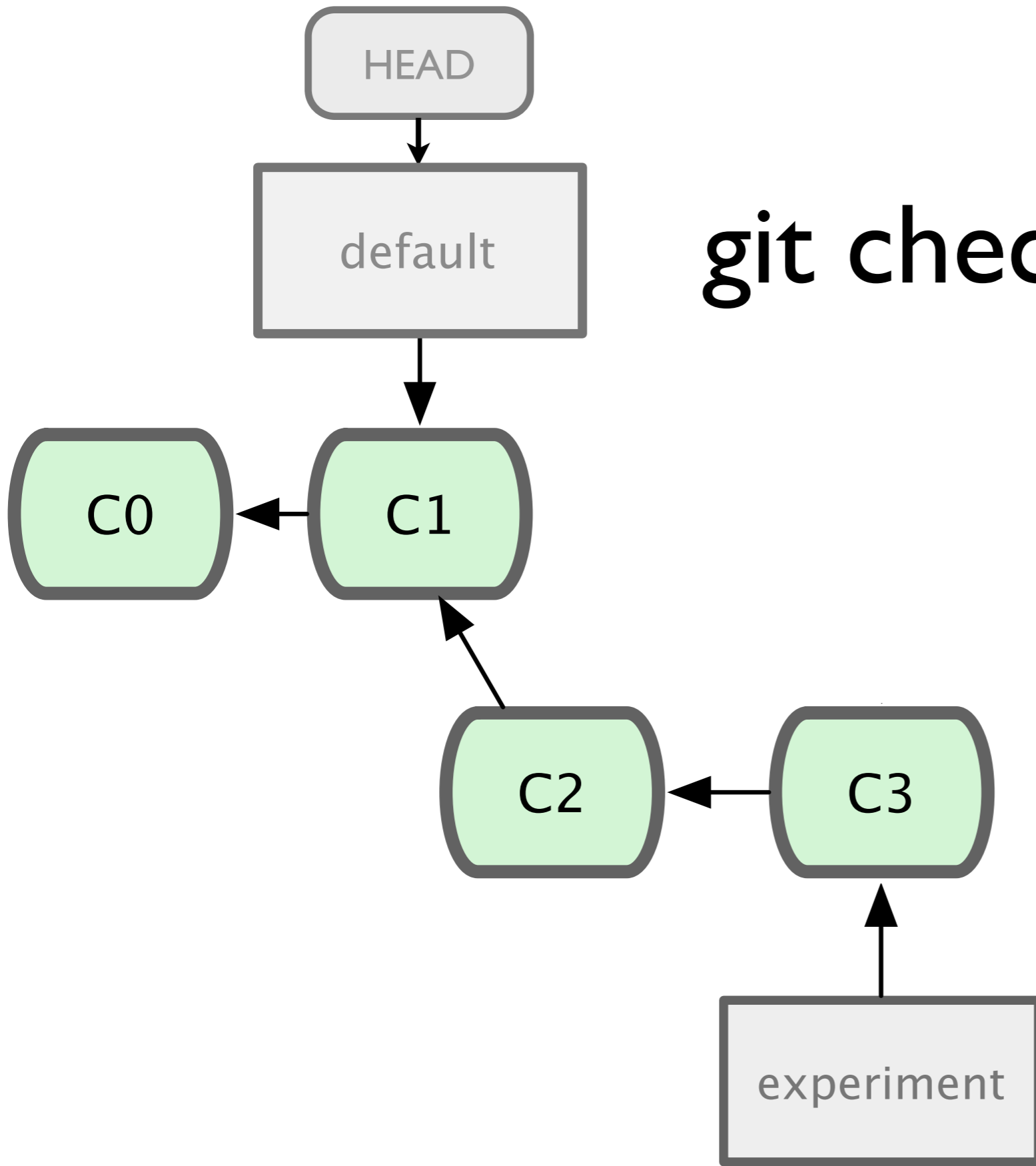


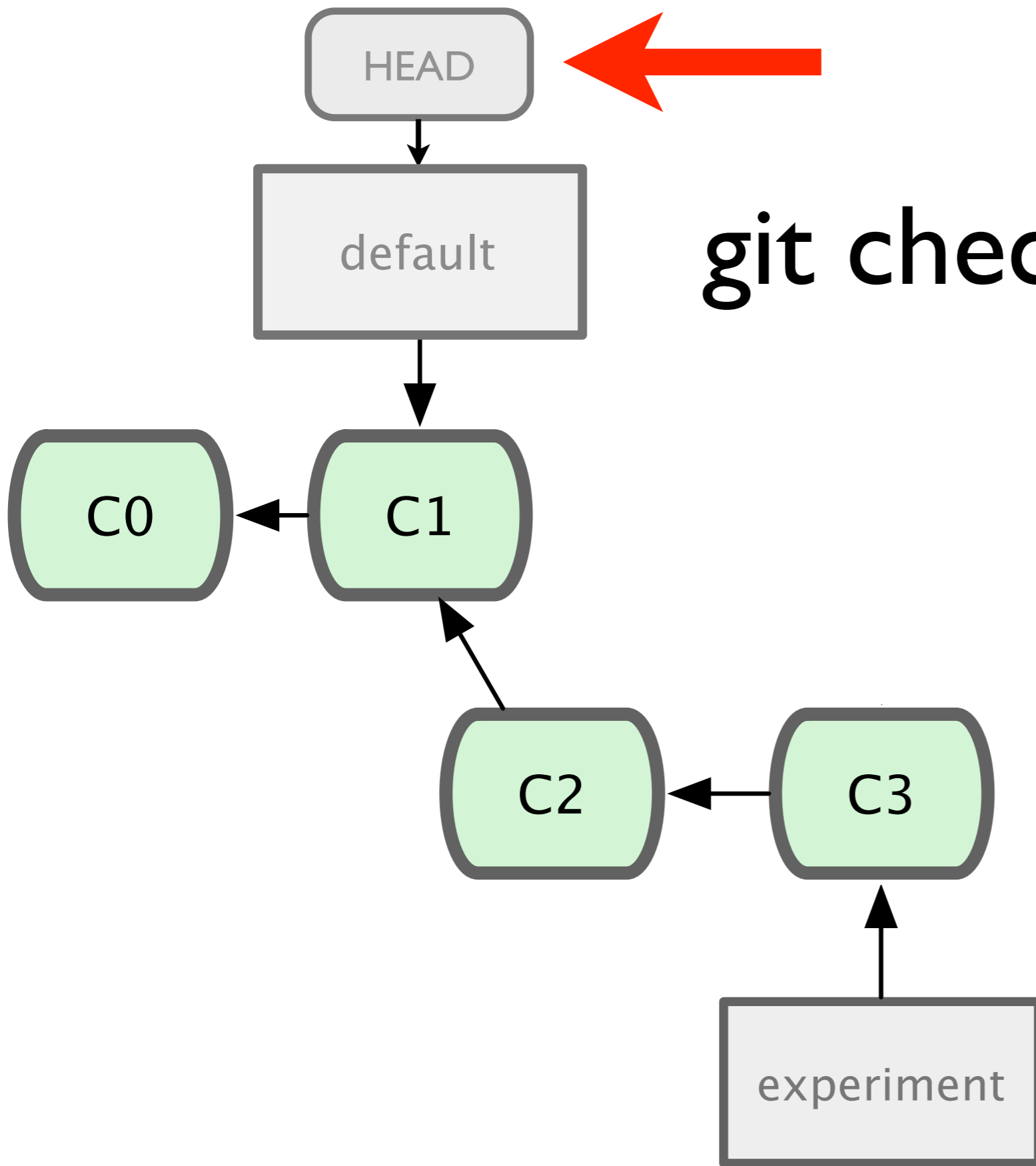
git commit



git commit
git commit

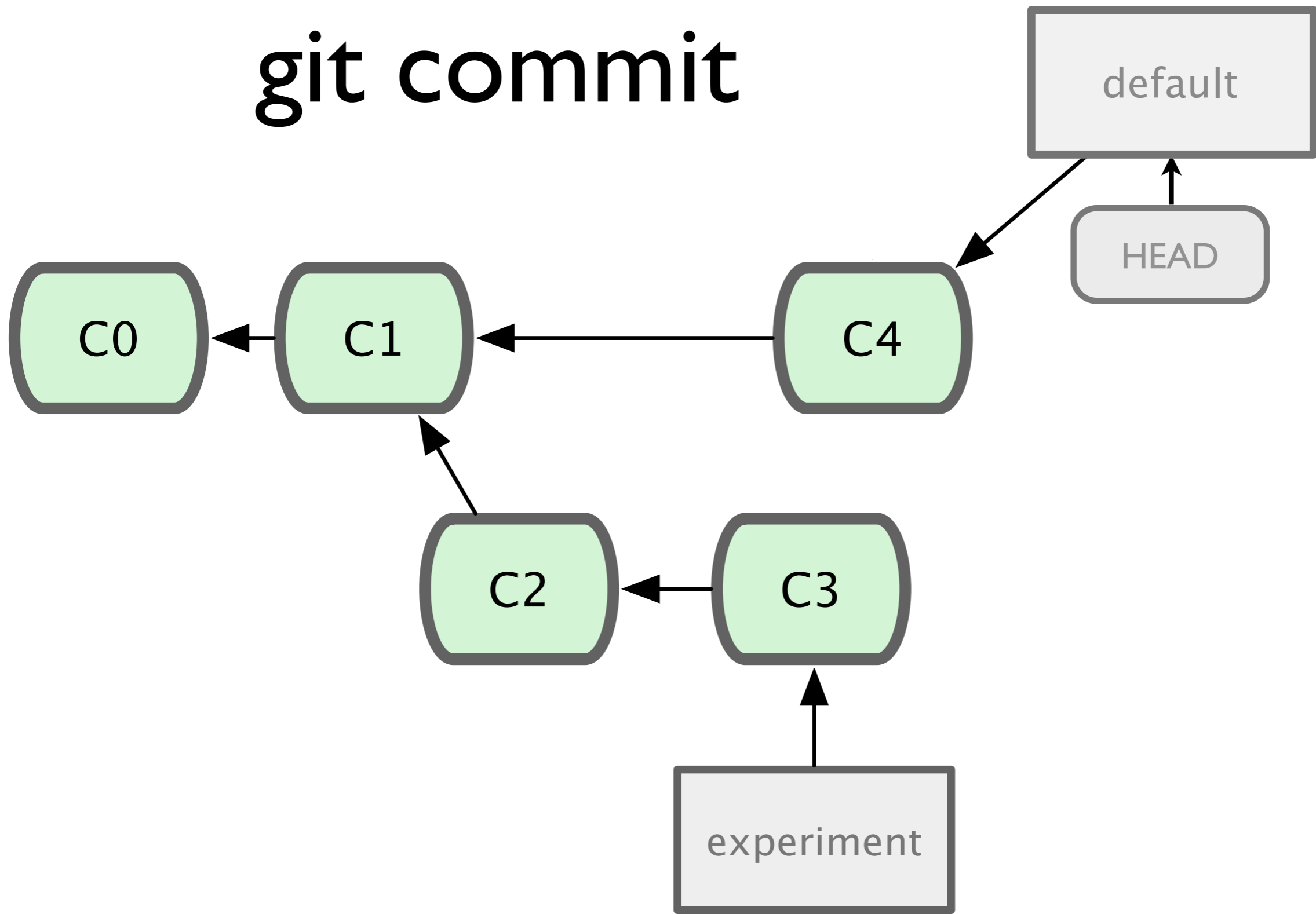
git checkout default



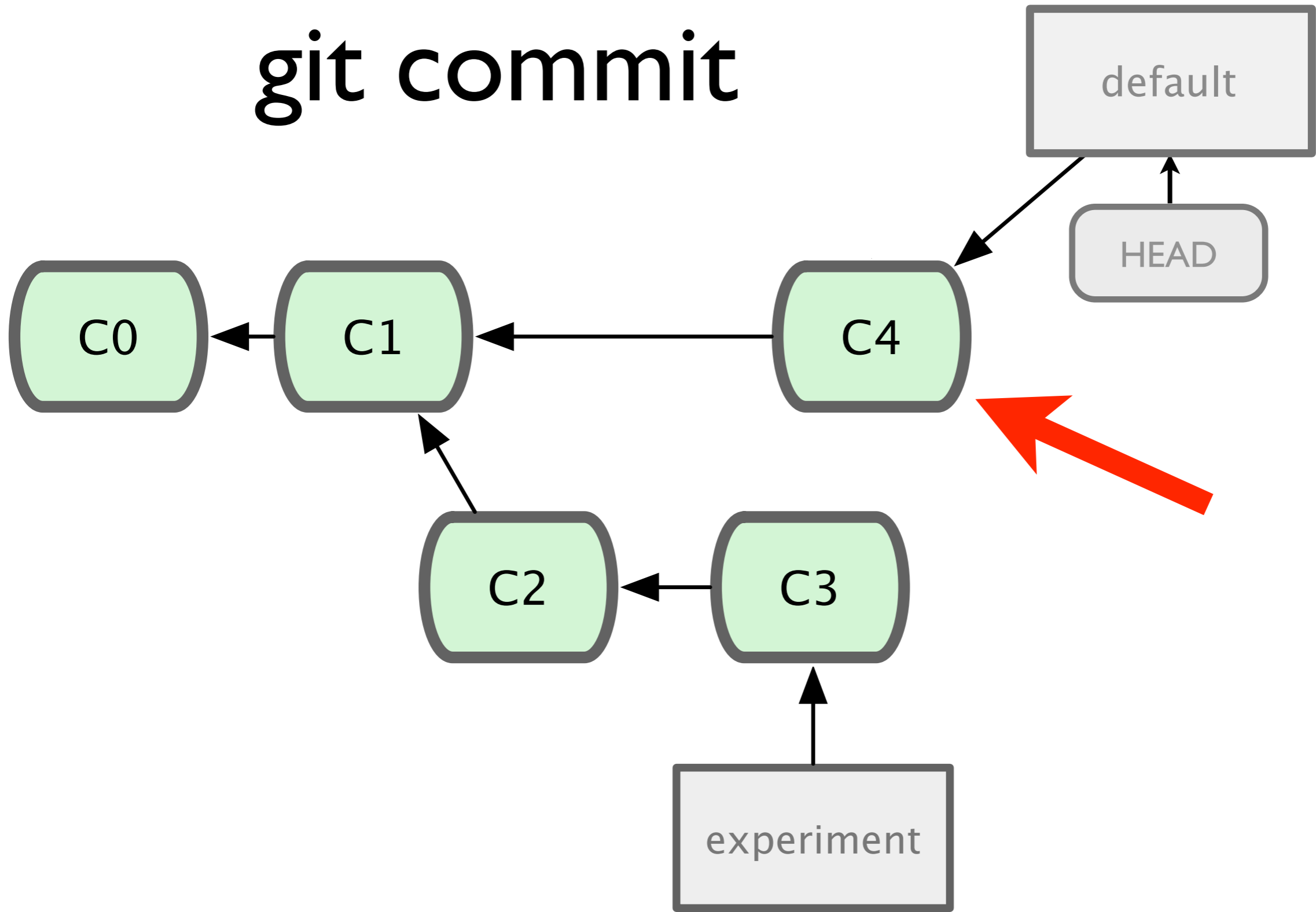


git checkout default

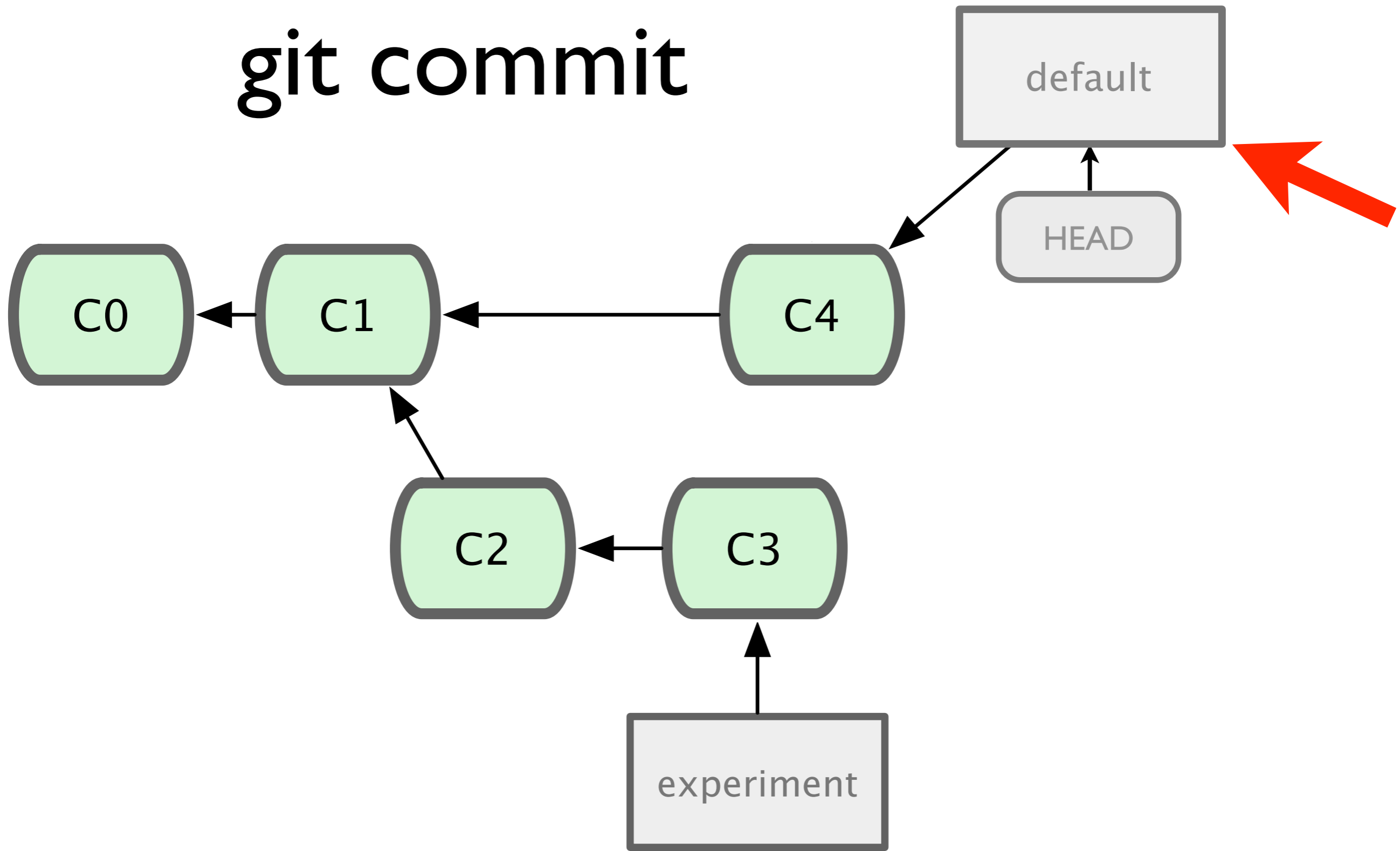
git commit

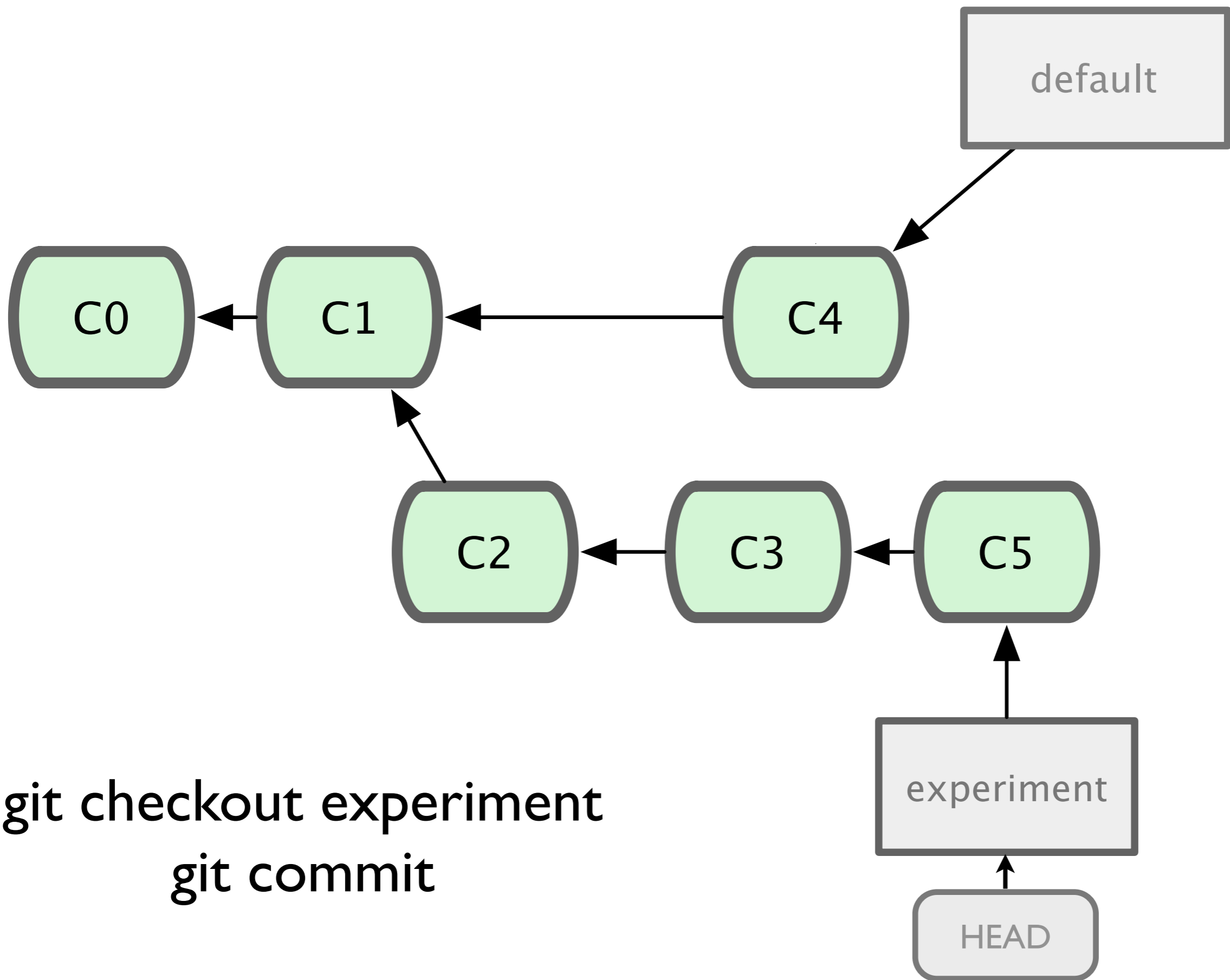


git commit



git commit





what is a branch, really?

```
$ find .git/refs
```

```
$ find .git/refs
.git/refs
.git/refs/heads
.git/refs/heads/default
.git/refs/heads/experiment
```

```
$ find .git/refs
```

```
.git/refs
```

```
.git/refs/heads
```

```
.git/refs/heads/default
```

```
.git/refs/heads/experiment
```



```
$ find .git/refs
```

```
.git/refs
```

```
.git/refs/heads
```

```
.git/refs/heads/default
```

```
.git/refs/heads/experiment
```

```
$ cat .git/refs/heads/default
```

```
6370409dc9e38af91565082bdf93577ff555489e
```

```
$ find .git/refs
```

```
.git/refs
```

```
.git/refs/heads
```

```
.git/refs/heads/default
```

```
.git/refs/heads/experiment
```

```
$ cat .git/refs/heads/default
```

```
6370409dc9e38af91565082bdf93577ff555489e
```

```
$ cat .git/refs/heads/experiment
```

```
33ce13c2f0ed35775956f81191b01a644448dcfc
```

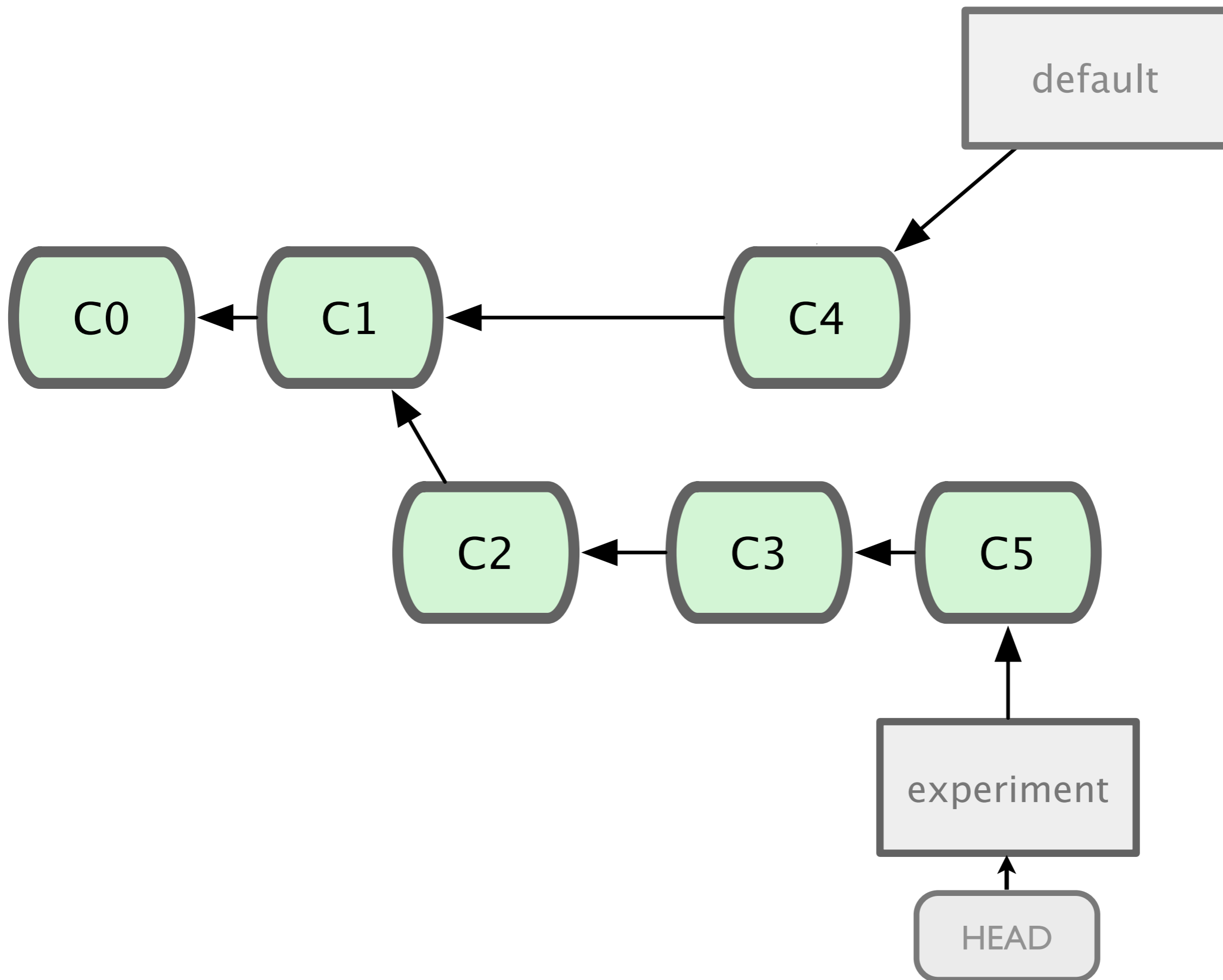
```
$ cat .git/HEAD
```

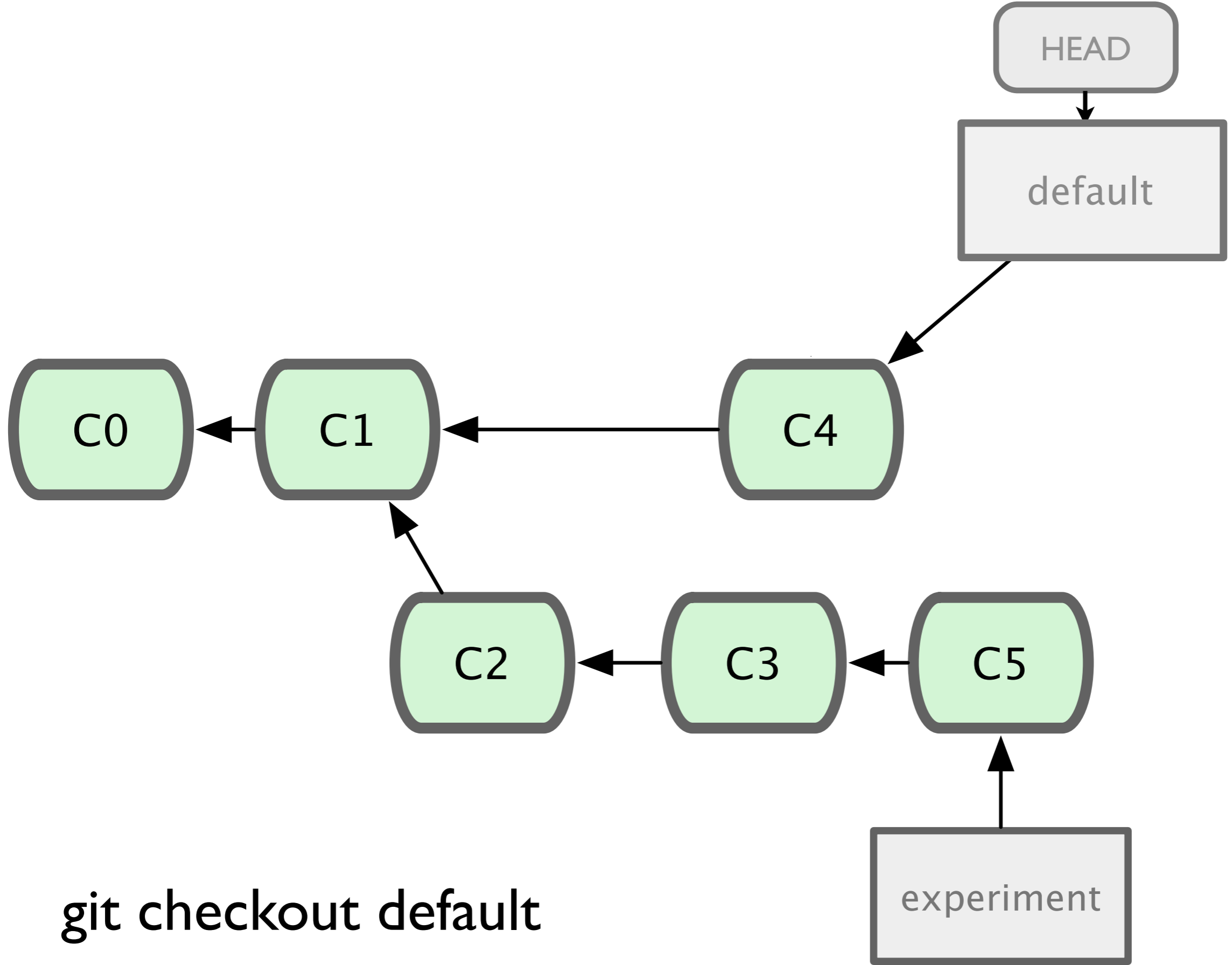
```
$ cat .git/HEAD  
ref: refs/heads/experiment
```

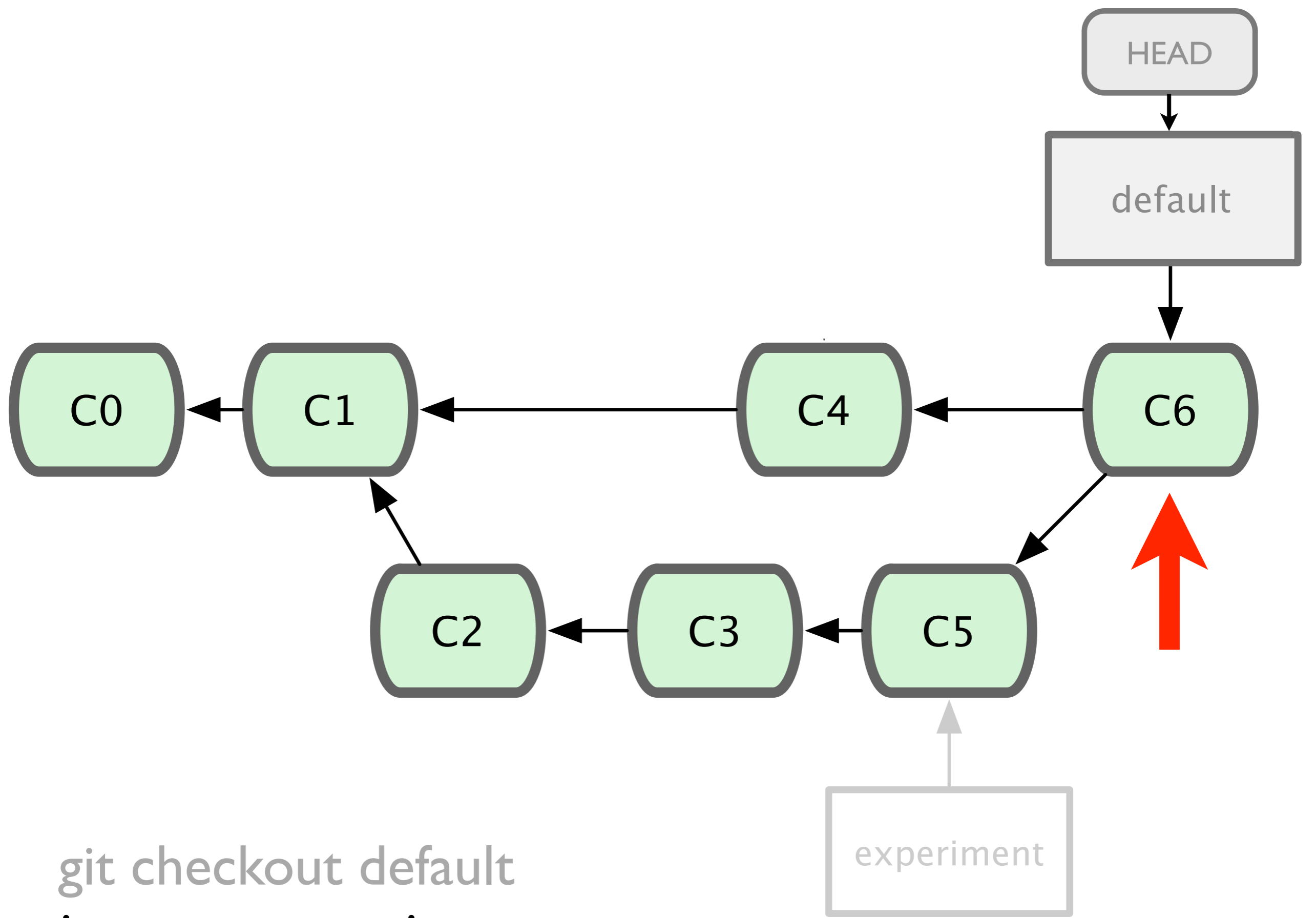
```
$ cat .git/HEAD  
ref: refs/heads/experiment
```

merging

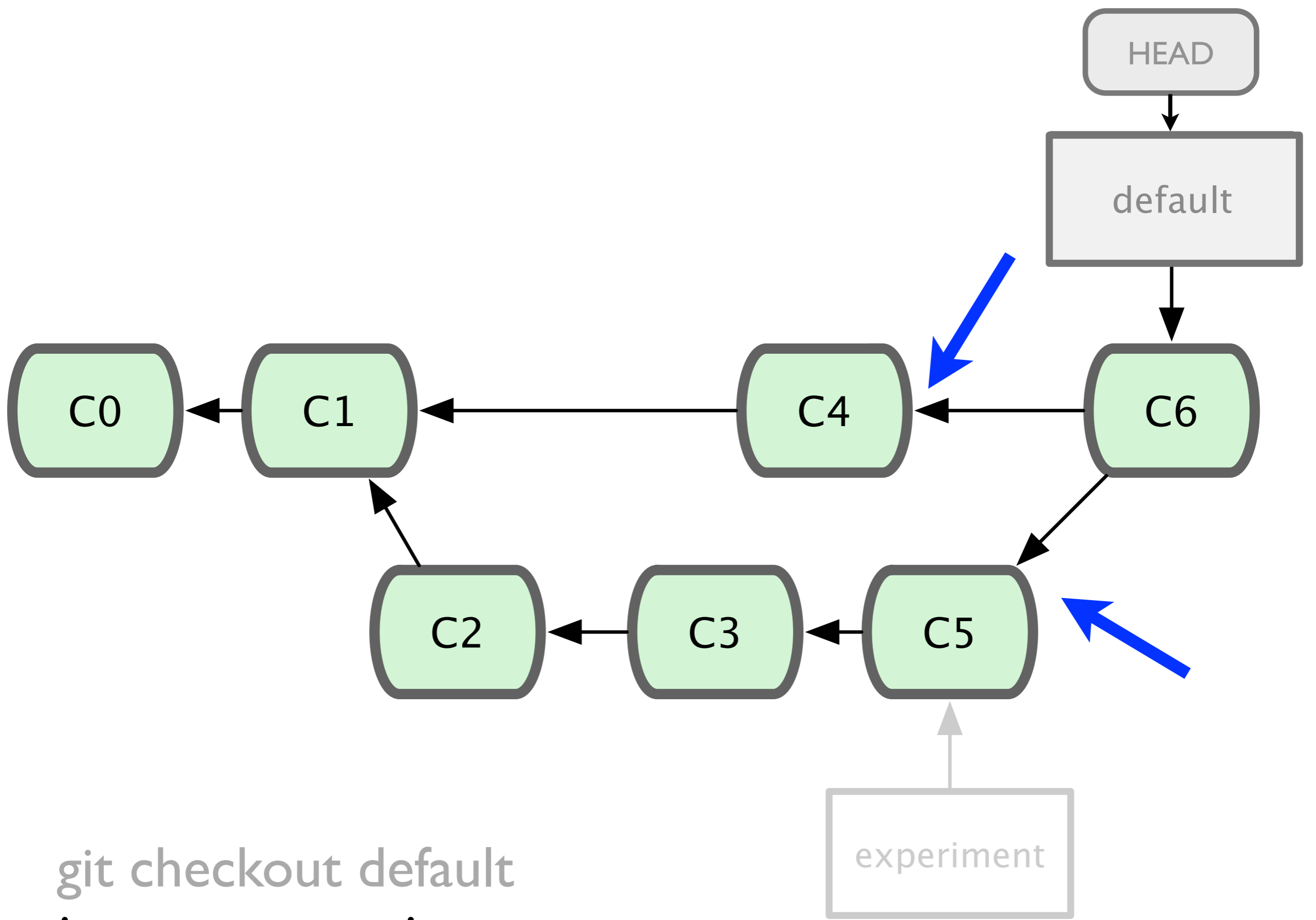
git merge



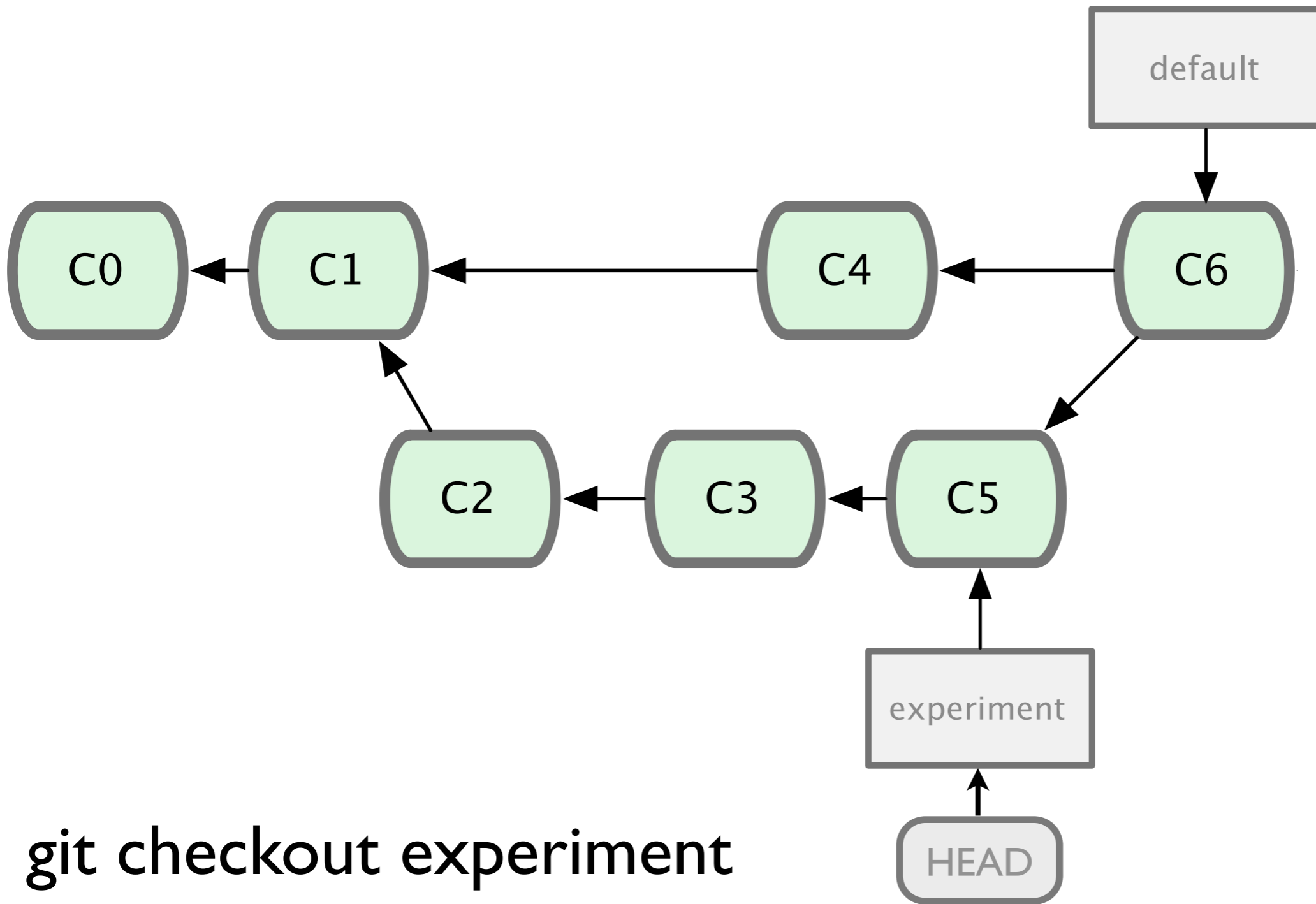




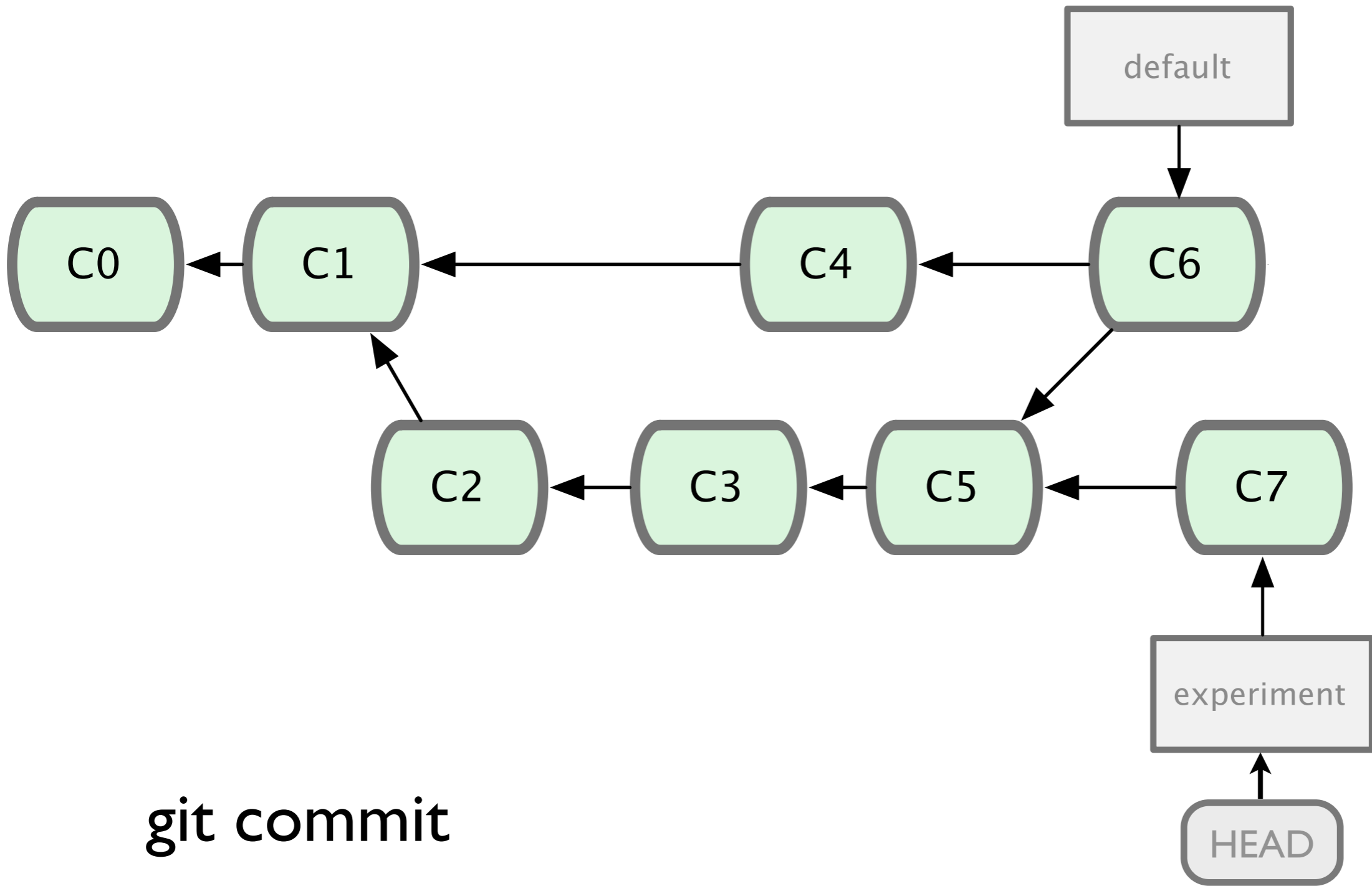
git checkout default
git merge experiment



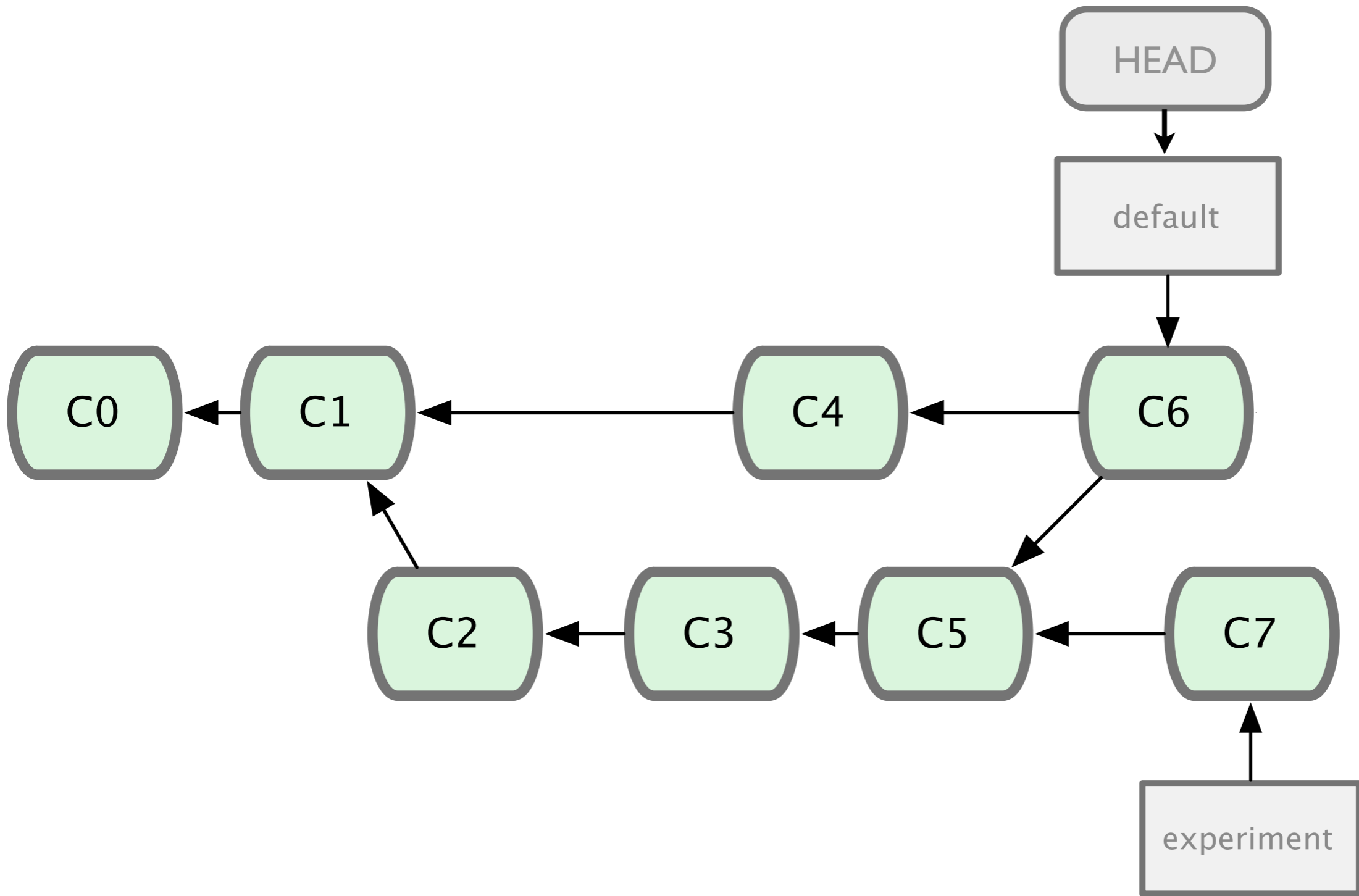
git checkout default
git merge experiment



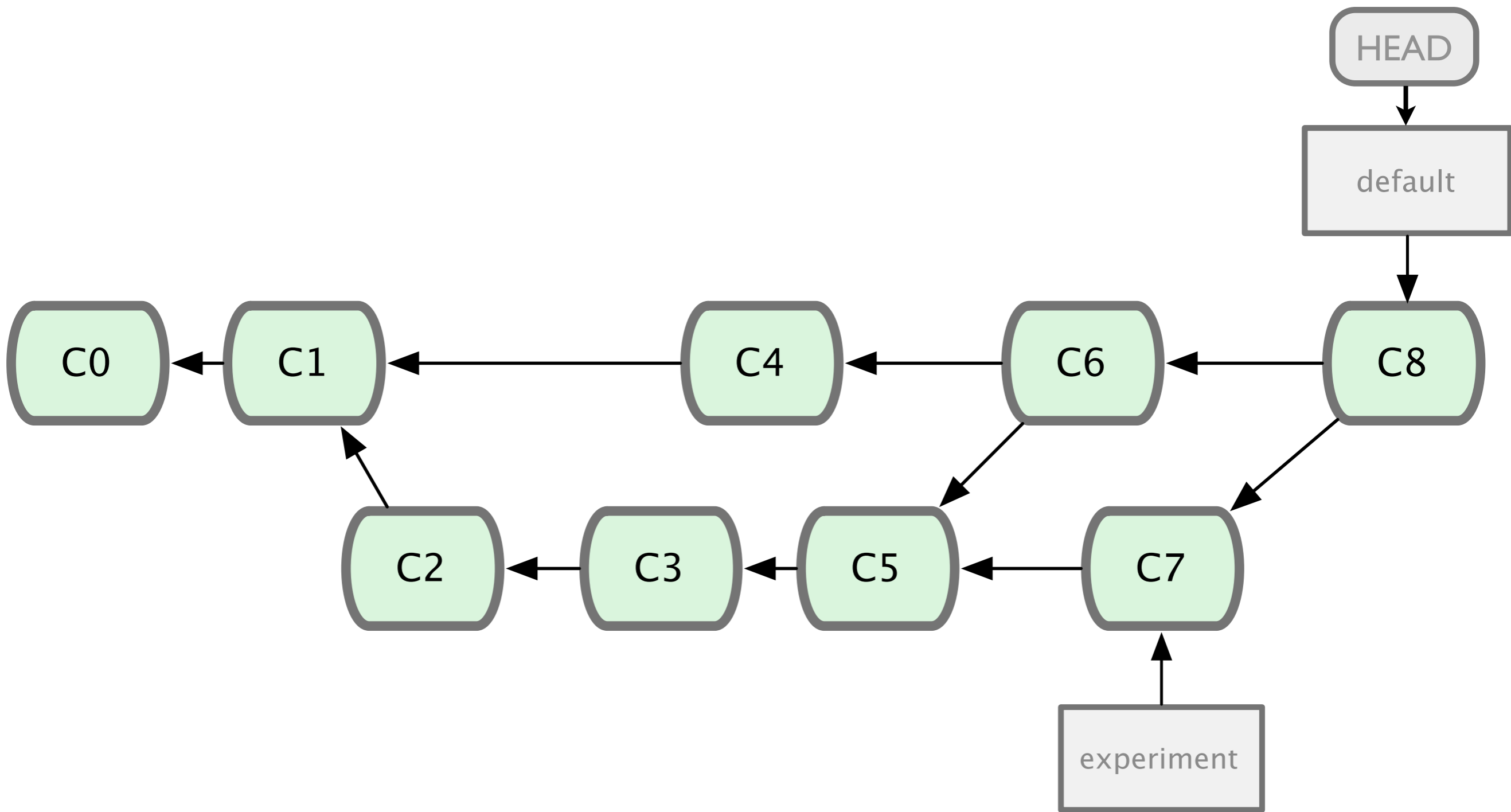
git checkout experiment



git commit



git checkout default

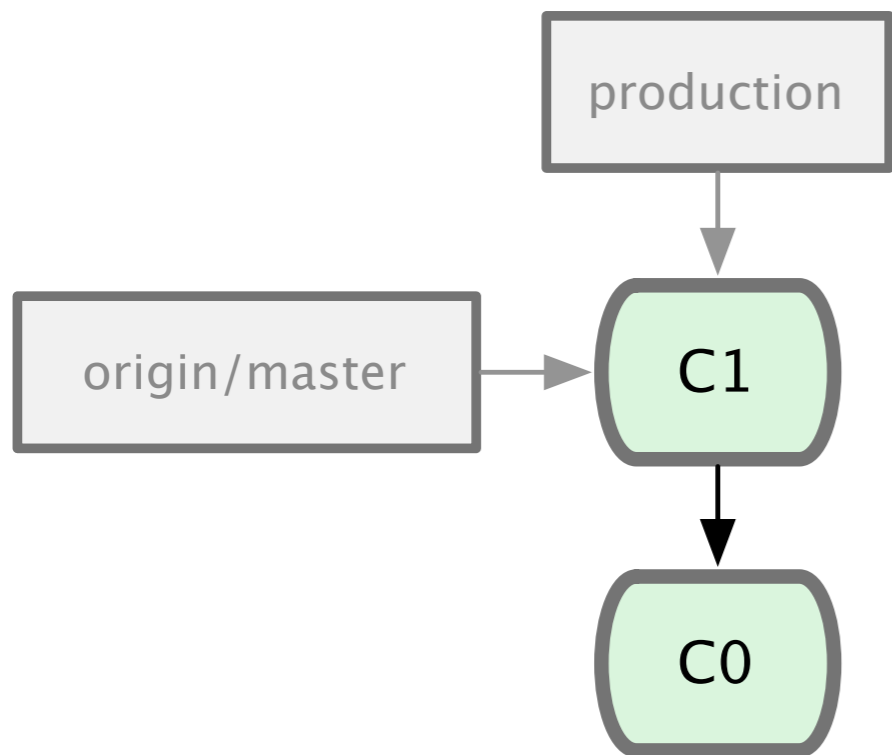


git merge experiment

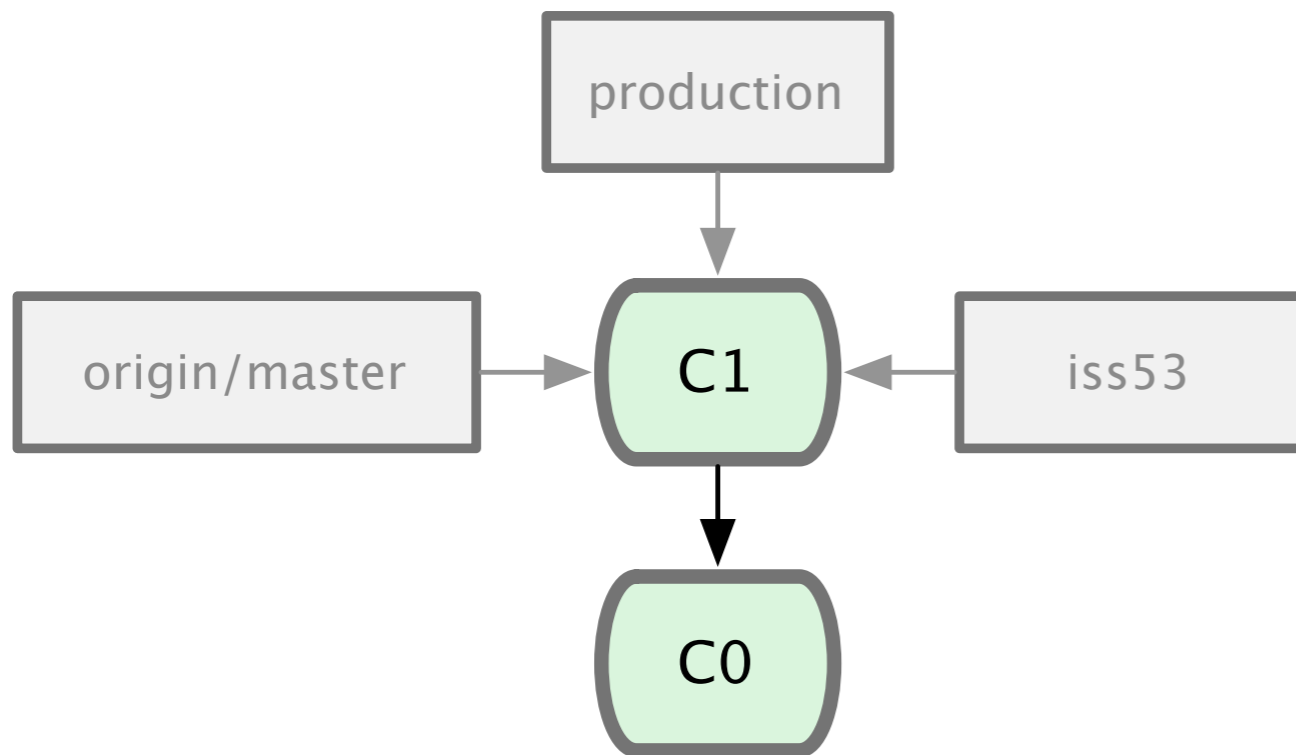
Why is this cool?

non-linear development

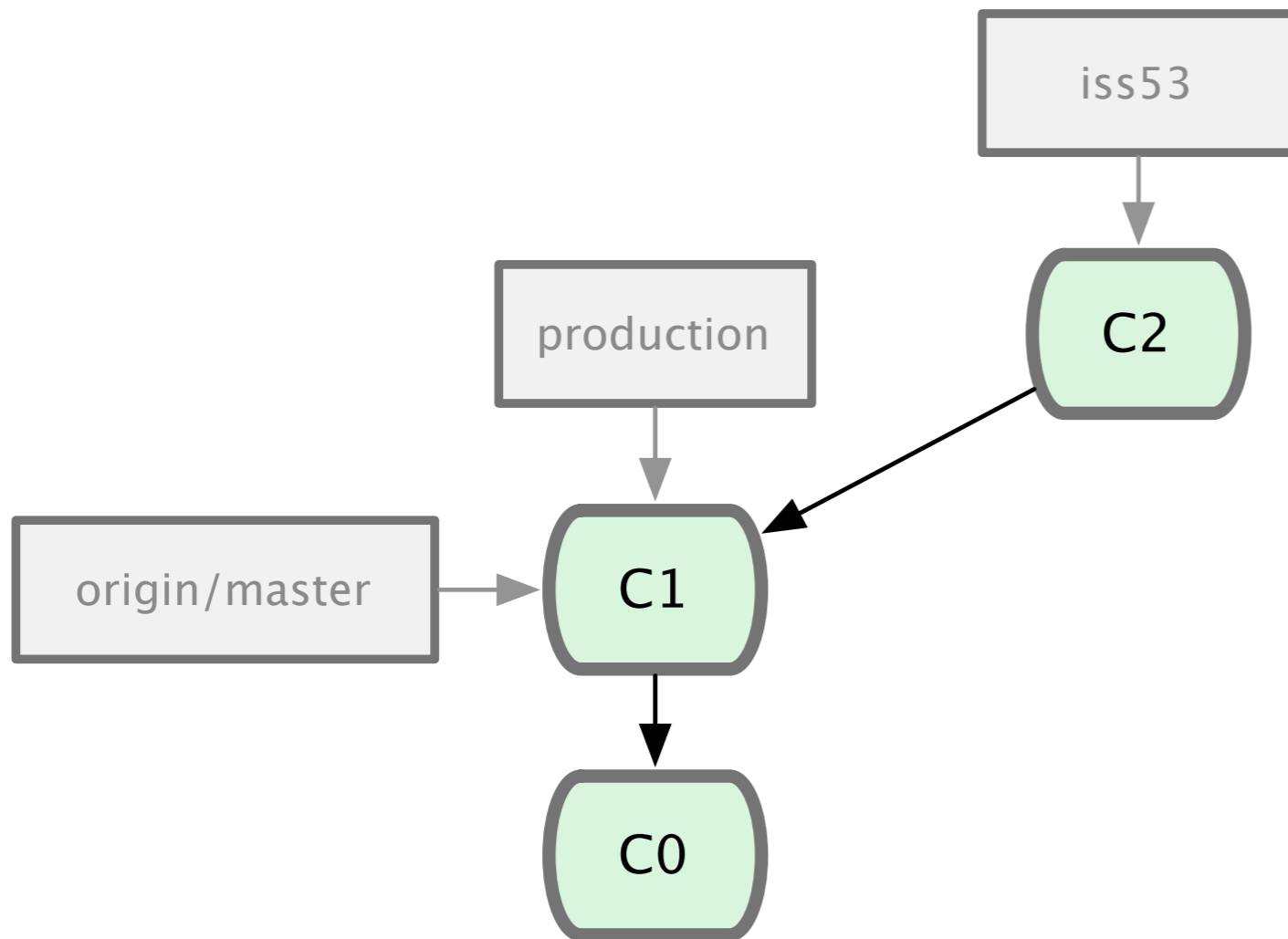
- clone the code that is in production
- create a branch for issue #53 (iss53)
- work for 10 minutes
- someone asks for a hotfix for issue #102
- checkout 'production'
- create a branch (iss102)
- fix the issue
- checkout 'production', merge 'iss102'
- push 'production'
- checkout 'iss53' and keep working



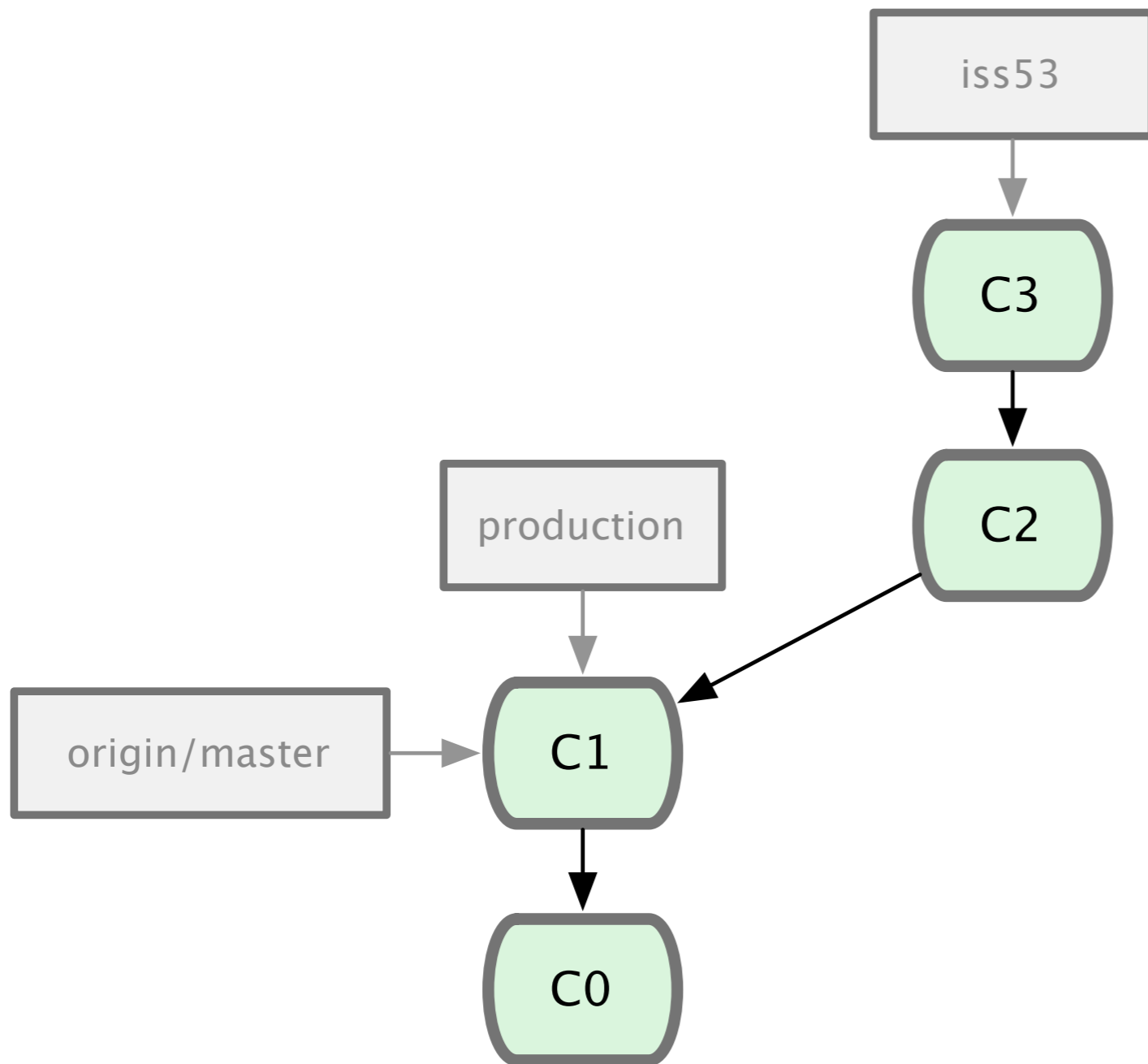
git clone



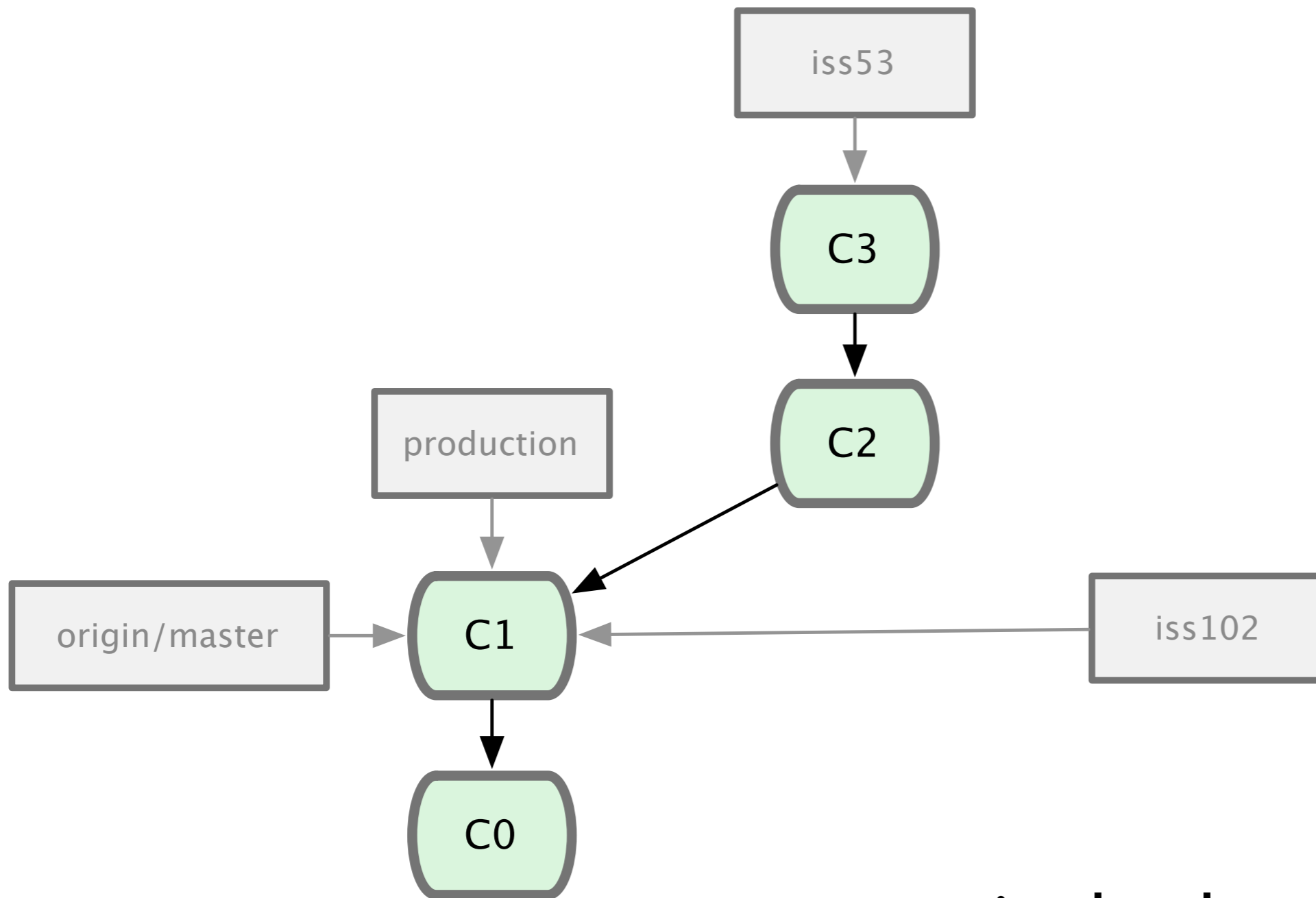
git checkout -b iss53



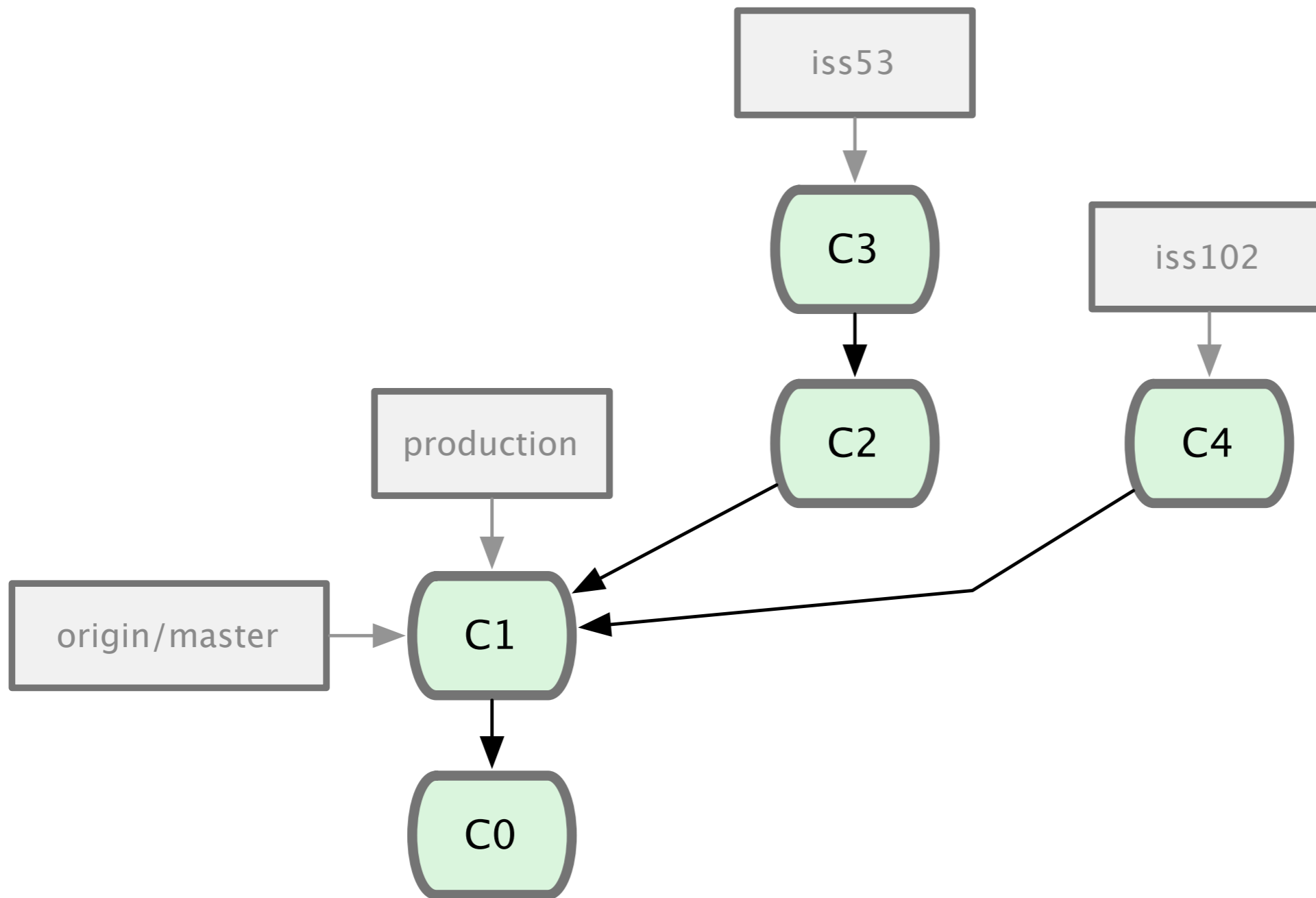
git commit



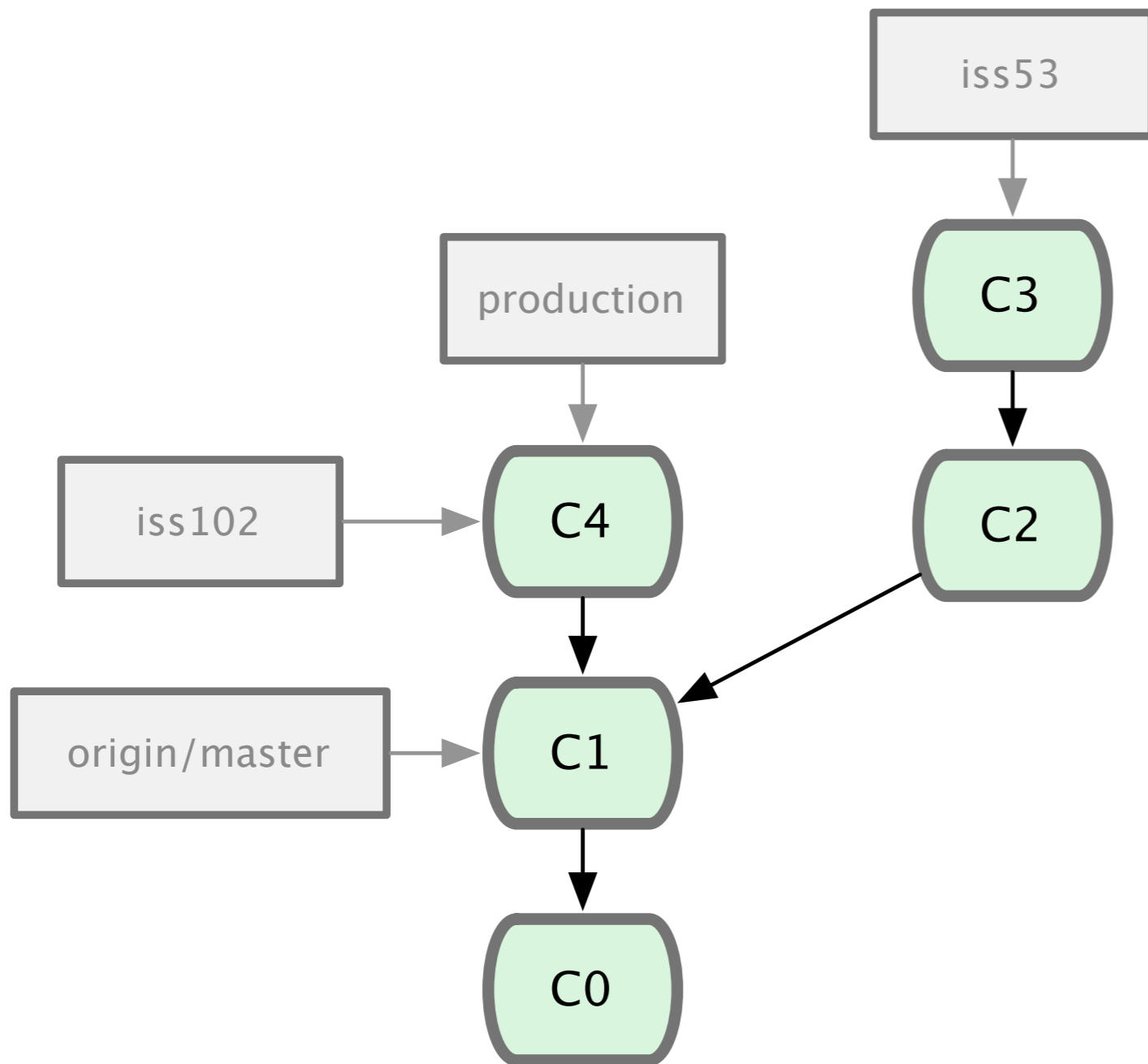
git commit



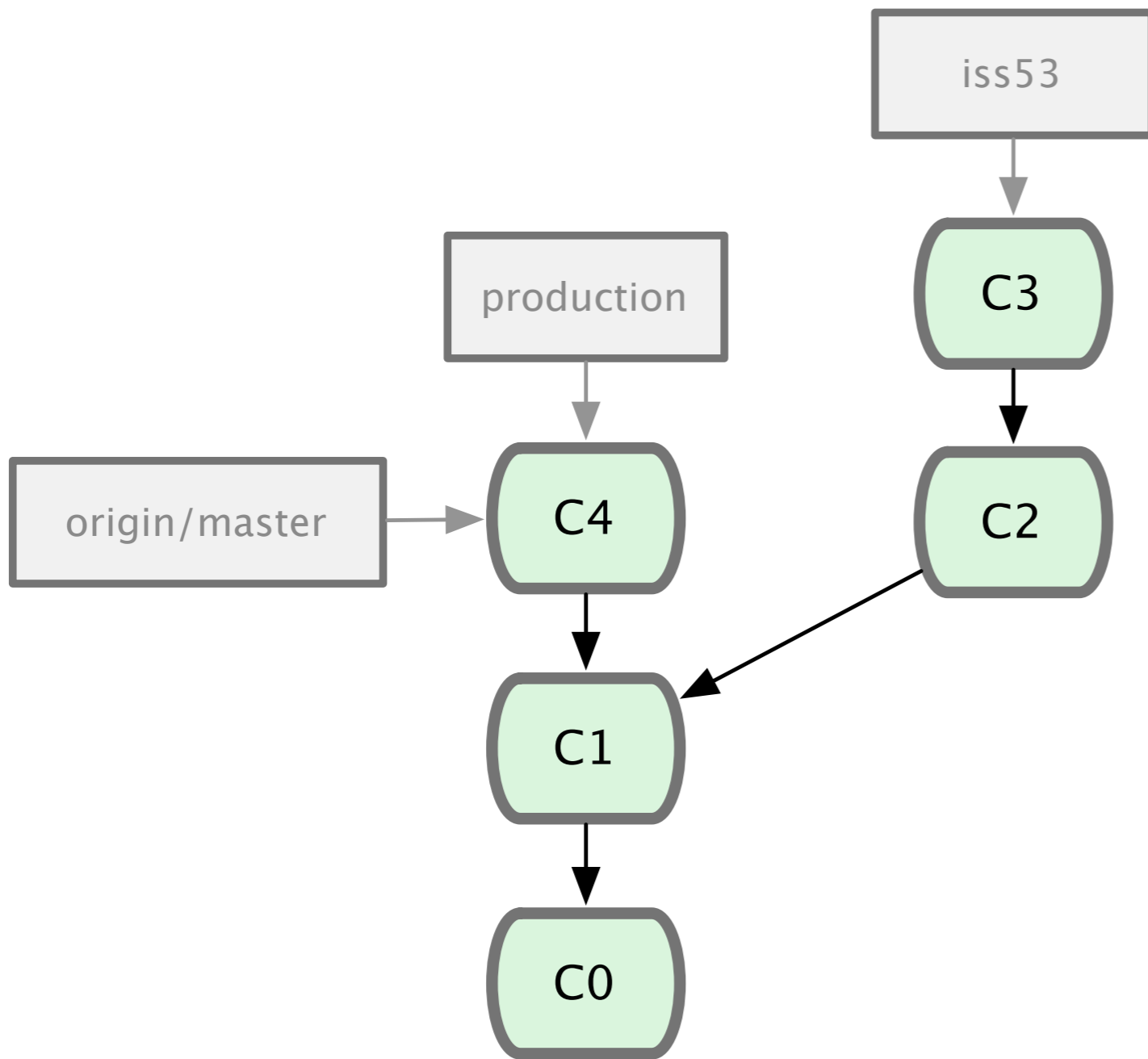
git checkout production
git checkout -b iss102



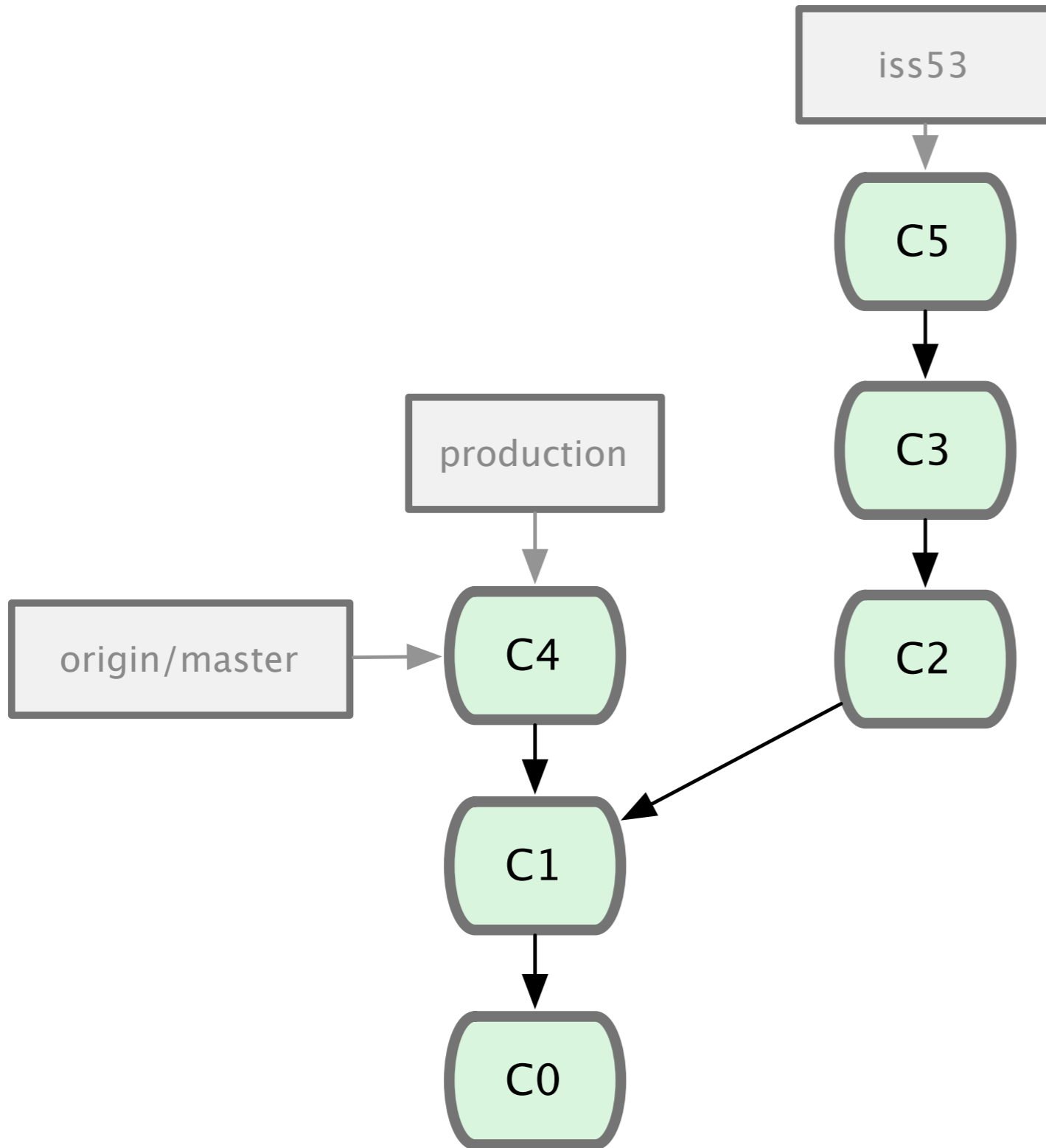
git commit



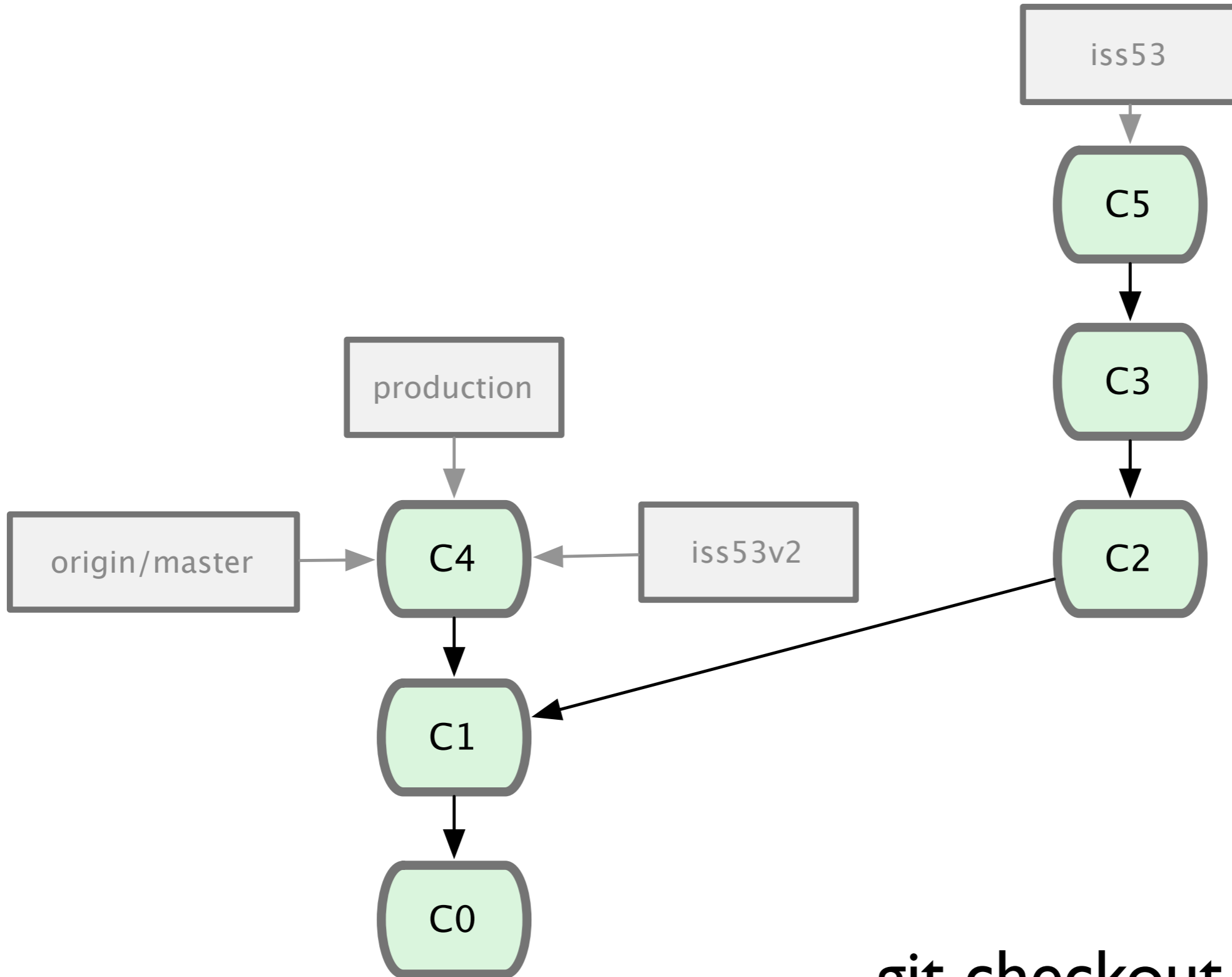
git checkout production
git merge iss102



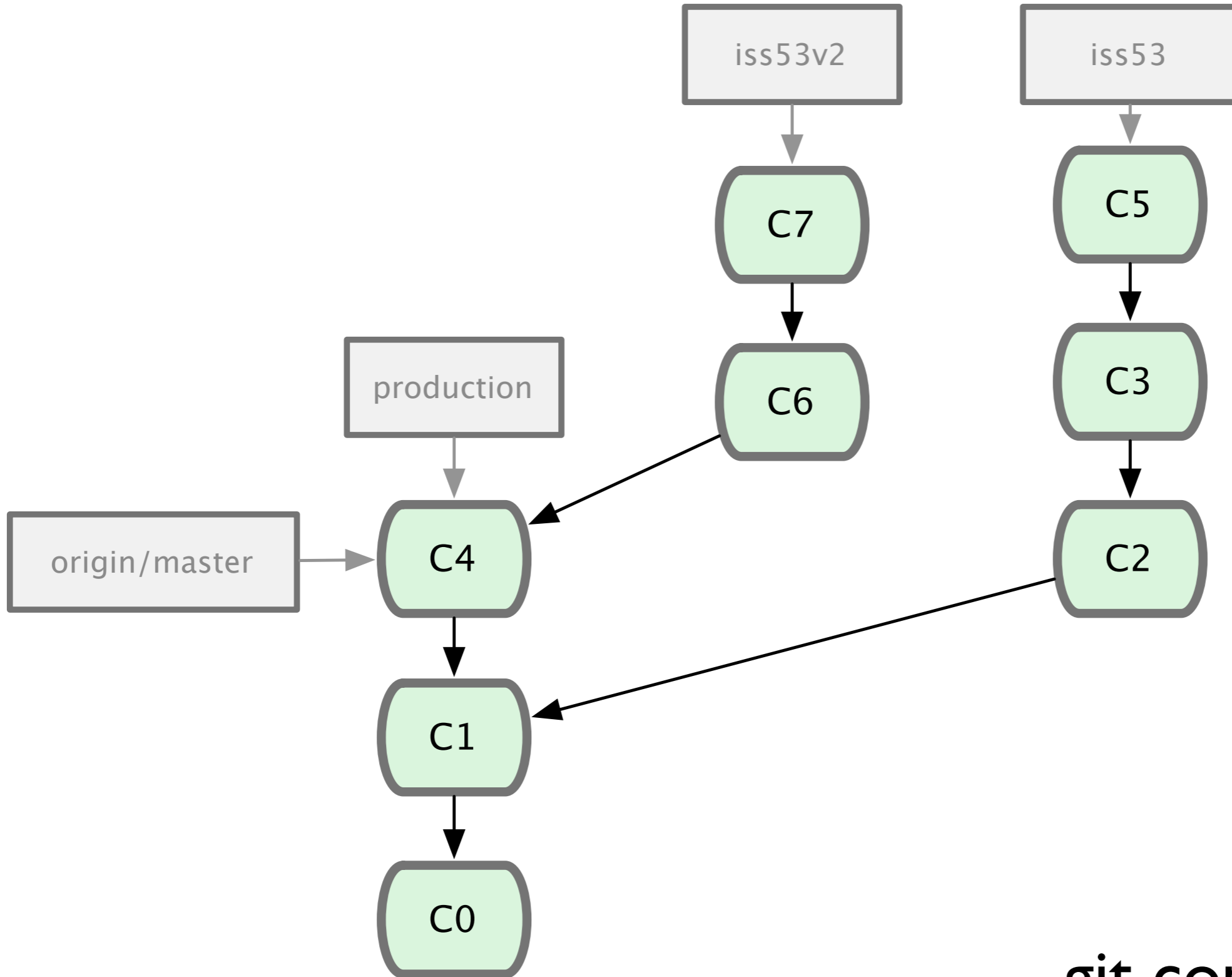
git push



`git checkout iss53`
`git commit`



`git checkout production`
`git checkout -b iss53v2`



git commit
git commit

try out an idea

isolate work units

long running topics

all local, so far

Remotes

sharing git

```
$>git remote
```

```
$>
```



News Feed

[for you](#) | [from you](#)

Magnus Chacon committed to [simplegit](#) 4 minutes ago



[fc62e5543b195f18391886b9f663d5a7eca38e84](#)
added file_size

Magnus Chacon committed to [simplegit](#) 27 minutes ago



[f40831985e265eb8cca2d0e66c11ac48ccfe170d](#)
changes

Magnus Chacon committed to [simplegit](#) 27 minutes ago



[25ce6394a94d8c3c30240960b386f79f4ebdd1da](#)
changes

Magnus Chacon committed to [simplegit](#) 27 minutes ago



[3e625e2139a71bfbaa85b055c139bceafab7e2c0](#)
updated rakefile

Magnus Chacon committed to [simplegit](#) 27 minutes ago



[4fefbaafd51ff3e79b84e9ee1a0dac2848e21a98](#)
commit the todo

Your Repositories [\(create a new one\)](#)

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

[capistrano](#)

[dst](#)

[facebook](#)

[git-ruby](#)

[git-source](#)

[git-wiki](#)

[grit](#)

[munger](#)

[rest-client](#)

[ruby-git](#)

[simplegit](#)

[ticgit](#)

Watched Repositories



News Feed

[for you](#) | [from you](#)

Magnus Chacon committed to [simplegit](#) 4 minutes ago

[fc62e5543b195f18391886b9f663d5a7eca38e84](#)
added file_size

Magnus Chacon committed to [simplegit](#) 27 minutes ago

[f40831985e265eb8cca2d0e66c11ac48ccfe170d](#)
changes

Magnus Chacon committed to [simplegit](#) 27 minutes ago

[25ce6394a94d8c3c30240960b386f79f4ebdd1da](#)
changes

Magnus Chacon committed to [simplegit](#) 27 minutes ago

[3e625e2139a71bfbaa85b055c139bceafab7e2c0](#)
updated rakefile

Magnus Chacon committed to [simplegit](#) 27 minutes ago

[4fefbaafd51ff3e79b84e9ee1a0dac2848e21a98](#)
commit the todo

Your Repositories [\(create a new one\)](#)

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

[capistrano](#)

[dst](#)

[facebook](#)

[git-ruby](#)

[git-source](#)

[git-wiki](#)

[grit](#)

[munger](#)

[rest-client](#)

[ruby-git](#)

[simplegit](#)

[ticgit](#)

Watched Repositories

Create a New Repository

Create a new empty repository into which you can push your local git repo.

NOTE: If you intend to push a copy of a repository that is already hosted on GitHub, then you should [fork](#) it instead.

Project Name

Description

Homepage URL

Who has access to this repository? (You can change this later)

- Anyone ([learn more about public repos](#))
- Only the people I specify ([learn more about private repos](#))



schacon

[account](#) | [profile](#) | [guides](#) | [log out](#)

✉ 0

repositories: [all](#) | [search](#)

schacon / simplegit2  [edit](#)

 **this repo is viewable by everyone**

Description: another repo for simplegit for demo puposes [edit](#)
Homepage: [Click to edit](#) [edit](#)
Public Clone URL: [git://github.com/schacon/simplegit2.git](https://github.com/schacon/simplegit2.git)
Your Clone URL: [git@github.com:schacon/simplegit2.git](https://github.com/schacon/simplegit2.git)
Push URL: [git@github.com:schacon/simplegit2.git](https://github.com/schacon/simplegit2.git)



schacon

[account](#) | [profile](#) | [guides](#) | [log out](#)

0

repositories: [all](#) | [search](#)

schacon / simplegit2 edit

this repo is viewable by everyone

Description: another repo for simplegit for demo puposes [edit](#)
Homepage: [Click to edit](#) [edit](#)
Public Clone URL: [git://github.com/schacon/simplegit2.git](https://github.com/schacon/simplegit2.git)
Your Clone URL: [git@github.com:schacon/simplegit2.git](https://github.com/schacon/simplegit2.git)
Push URL: [git@github.com:schacon/simplegit2.git](https://github.com/schacon/simplegit2.git)

```
$>git remote
```

```
$>git remote add public git@github.com:schacon/simplegit2.git
```

```
$>
```

```
$>git remote
```

```
$>git remote add public git@github.com:schacon/simplegit2.git
```

```
$>
```

```
$>git remote
```

```
$>git remote add public git@github.com:schacon/simplegit2.git
```

```
$>
```

```
$>git remote
$>git remote add public git@github.com:schacon/simplegit2.git
$>git push public master
Counting objects: 115, done.
Compressing objects: 100% (73/73), done.
Writing objects: 100% (115/115), 11.20 KiB, done.
Total 115 (delta 42), reused 65 (delta 24)
refs/heads/master: 0000000000000000000000000000000000000000000000000000000 -> fc6
5543b195f18391886b9f663d5a7eca38e84
To git@github.com:schacon/simplegit2.git
 * [new branch]      master -> master
$>
```

```
$>git remote
$>git remote add public git@github.com:schacon/simplegit2.git
$>git push public master
Counting objects: 115, done.
Compressing objects: 100% (73/73), done.
Writing objects: 100% (115/115), 11.20 KiB, done.
Total 115 (delta 42), reused 65 (delta 24)
refs/heads/master: 0000000000000000000000000000000000000000000000000000000 -> fc6
5543b195f18391886b9f663d5a7eca38e84
To git@github.com:schacon/simplegit2.git
 * [new branch]      master -> master
$>
```

```
$>git remote
$>git remote add public git@github.com:schacon/simplegit2.git
$>git push public master
Counting objects: 115, done.
Compressing objects: 100% (73/73), done.
Writing objects: 100% (115/115), 11.20 KiB, done.
Total 115 (delta 42), reused 65 (delta 24)
refs/heads/master: 0000000000000000000000000000000000000000000000000000000 -> fc6
5543b195f18391886b9f663d5a7eca38e84
To git@github.com:schacon/simplegit2.git
 * [new branch]      master -> master
$>
```

```
$>git remote
$>git remote add public git@github.com:schacon/simplegit2.git
$>git push public master
Counting objects: 115, done.
Compressing objects: 100% (73/73), done.
Writing objects: 100% (115/115), 11.20 KiB, done.
Total 115 (delta 42), reused 65 (delta 24)
refs/heads/master: 0000000000000000000000000000000000000000000000000000000 -> fc6
5543b195f18391886b9f663d5a7eca38e84
To git@github.com:schacon/simplegit2.git
 * [new branch]      master -> master
$>
```

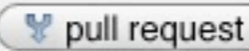


```
$>git remote
$>git remote add public git@github.com:schacon/simplegit2.git
$>git push public master
Counting objects: 115, done.
Compressing objects: 100% (73/73), done.
Writing objects: 100% (115/115), 11.20 KiB, done.
Total 115 (delta 42), reused 65 (delta 24)
refs/heads/master: 0000000000000000000000000000000000000000000000000000000 -> fc6
5543b195f18391886b9f663d5a7eca38e84
To git@github.com:schacon/simplegit2.git
 * [new branch]      master -> master
$>git remote
public
$>
```

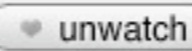
```
$>git remote
$>git remote add public git@github.com:schacon/simplegit2.git
$>git push public master
Counting objects: 115, done.
Compressing objects: 100% (73/73), done.
Writing objects: 100% (115/115), 11.20 KiB, done.
Total 115 (delta 42), reused 65 (delta 24)
refs/heads/master: 0000000000000000000000000000000000000000000000000000000 -> fc6
5543b195f18391886b9f663d5a7eca38e84
To git@github.com:schacon/simplegit2.git
 * [new branch]      master -> master
$>git remote
public
$>git remote show public
* remote public
  URL: git@github.com:schacon/simplegit2.git
  Tracked remote branches
    master
$>_
```



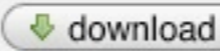
edit



pull request



unwatch



download

this repo is viewable by everyone

Description: another repo for simplegit for demo puposes [edit](#)

Homepage: [Click to edit](#) [edit](#)

Public Clone URL: [git://github.com/schacon/simplegit2.git](https://github.com/schacon/simplegit2.git)

Your Clone URL: [git@github.com:schacon/simplegit2.git](https://github.com:schacon/simplegit2.git)

Push URL: [git@github.com:schacon/simplegit2.git](https://github.com:schacon/simplegit2.git)

added file_size



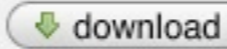
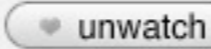
Magnus Chacon (author)

17 minutes ago

commit fc62e5543b195f18391886b9f663d5a7eca38e84
tree 3ec275327fe6f8474fd4d044ebb1adfe04ca38db
parent f40831985e265eb8cca2d0e66c11ac48ccfe170d

simplegit2 /

name	age	message	history
README	about 14 hours ago	updated README [Magnus Chacon]	
Rakefile	39 minutes ago	changes [Magnus Chacon]	
TODO	39 minutes ago	changes [Magnus Chacon]	
lib/	16 minutes ago	added file_size [Magnus Chacon]	



Description: another repo for simplegit for demo puposes [edit](#)

Homepage: [Click to edit](#) [edit](#)

Public Clone URL: [git://github.com/schacon/simplegit2.git](https://github.com/schacon/simplegit2.git)

Your Clone URL: [git@github.com:schacon/simplegit2.git](https://github.com:schacon/simplegit2.git)

Push URL: [git@github.com:schacon/simplegit2.git](https://github.com:schacon/simplegit2.git)

added file_size



Magnus Chacon (author)

17 minutes ago

commit fc62e5543b195f18391886b9f663d5a7eca38e84
tree 3ec275327fe6f8474fd4d044ebb1adfe04ca38db
parent f40831985e265eb8cca2d0e66c11ac48ccfe170d

simplegit2 /

name	age	message	history
README	about 14 hours ago	updated README [Magnus Chacon]	
Rakefile	39 minutes ago	changes [Magnus Chacon]	
TODO	39 minutes ago	changes [Magnus Chacon]	
lib/	16 minutes ago	added file_size [Magnus Chacon]	





schacon / simplegit2 [edit](#) [pull request](#) [unwatch](#) [download](#)  this repo is viewable by everyone

Description: another repo for simplegit for demo puposes [edit](#)
Homepage: [Click to edit](#) [edit](#)
Public Clone URL: [git://github.com/schacon/simplegit2.git](https://github.com/schacon/simplegit2.git)
Your Clone URL: [git@github.com:schacon/simplegit2.git](https://github.com/schacon/simplegit2.git)
Push

git clone git://github.com/schacon/simplegit2.git

 Magnus Chacon (author) 17 minutes ago | parent [f40831985e265eb8cca2d0e66c11ac48ccfe170d](#)

simplegit2 /

name	age	message	history
 README	about 14 hours ago	updated README [Magnus Chacon]	
 Rakefile	39 minutes ago	changes [Magnus Chacon]	
 TODO	39 minutes ago	changes [Magnus Chacon]	
 lib/	16 minutes ago	added file_size [Magnus Chacon]	

distributed workflow

distributed workflow

fetch, pull and push

fetch


```
$>git remote  
github  
$>
```

```
$>git remote
github
$>git remote show github
* remote github
  URL: git://github.com/schacon/simplegit.git
  New remote branches (next fetch will store in remotes/github)
    master
$>
```

```
$>git remote
github
$>git remote show github
* remote github
  URL: git://github.com/schacon/simplegit.git
  New remote branches (next fetch will store in remotes/github)
    master
$>
```

```
$>git remote
github
$>git remote show github
* remote github
  URL: git://github.com/schacon/simplegit.git
  New remote branches (next fetch will store in remotes/github)
    master
$>git fetch github
remote: Generating pack...
remote: Done counting 106 objects.
remote: Result has 101 objects.
remote: Deltifying 101 objects...
remote: 100% (101/101) done
remote: Total 101 (delta 35), reused 101 (delta 35)
Receiving objects: 100% (101/101), 9.49 KiB, done.
Resolving deltas: 100% (35/35), done.
From git://github.com/schacon/simplegit
 * [new branch]      master      -> github/master
$>
```

```
$>git remote
github
$>git remote show github
* remote github
  URL: git://github.com/schacon/simplegit.git
  New remote branches (next fetch will store in remotes/github)
    master
$>git fetch github
remote: Generating pack...
remote: Done counting 106 objects.
remote: Result has 101 objects.
remote: Deltifying 101 objects...
remote: 100% (101/101) done
remote: Total 101 (delta 35), reused 101 (delta 35)
Receiving objects: 100% (101/101), 9.49 KiB, done.
Resolving deltas: 100% (35/35), done.
From git://github.com/schacon/simplegit
 * [new branch]      master      -> github/master
$>git branch -a
  develop
  master
 * story95
  github/master
$>_
```

```
$>git remote
github
$>git remote show github
* remote github
  URL: git://github.com/schacon/simplegit.git
  New remote branches (next fetch will store in remotes/github)
    master
$>git fetch github
remote: Generating pack...
remote: Done counting 106 objects.
remote: Result has 101 objects.
remote: Deltifying 101 objects...
remote: 100% (101/101) done
remote: Total 101 (delta 35), reused 101 (delta 35)
Receiving objects: 100% (101/101), 9.49 KiB, done.
Resolving deltas: 100% (35/35), done.
From git://github.com/schacon/simplegit
 * [new branch]      master      -> github/master
$>git branch -a
  develop
  master
 * story95
  github/master
$>_
```

pull

pull = fetch + merge

push

```
$>git remote  
public  
$>
```

```
$>git remote
public
$>git remote show public
* remote public
  URL: git@github.com:schacon/simplegit.git
  Tracked remote branches
    master
$>
```

```
$>git remote
public
$>git remote show public
* remote public
  URL: git@github.com:schacon/simplegit.git
  Tracked remote branches
    master
$>
```

```
$>git remote
public
$>git remote show public
* remote public
  URL: git@github.com:schacon/simplegit.git
  Tracked remote branches
    master
```

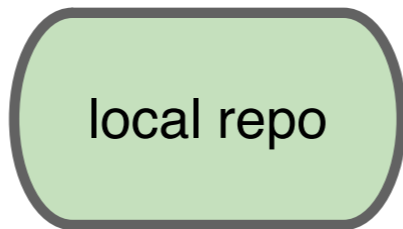
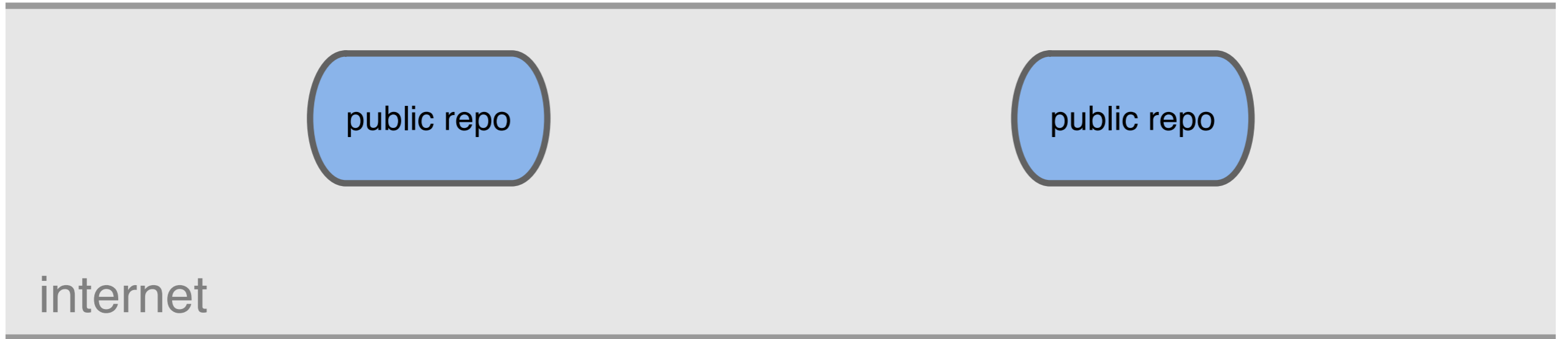
```
$>git push public master
```

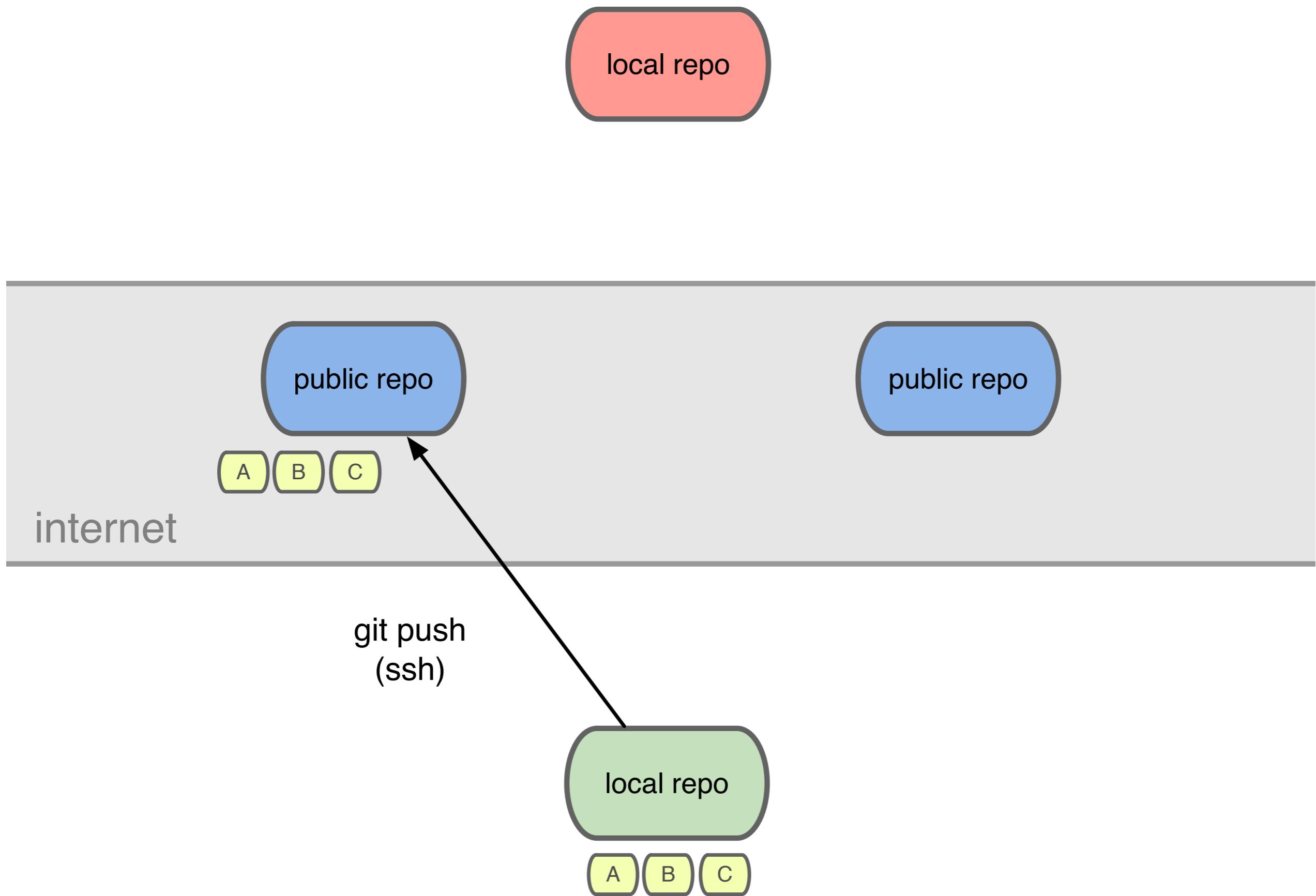
```
Counting objects: 7, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 365 bytes, done.
Total 4 (delta 2), reused 0 (delta 0)
refs/heads/master: f40831985e265eb8cca2d0e66c11ac48ccfe170d -> fc62
5543b195f18391886b9f663d5a7eca38e84
To git@github.com:schacon/simplegit.git
  f408319..fc62e55  master -> master
$>_
```

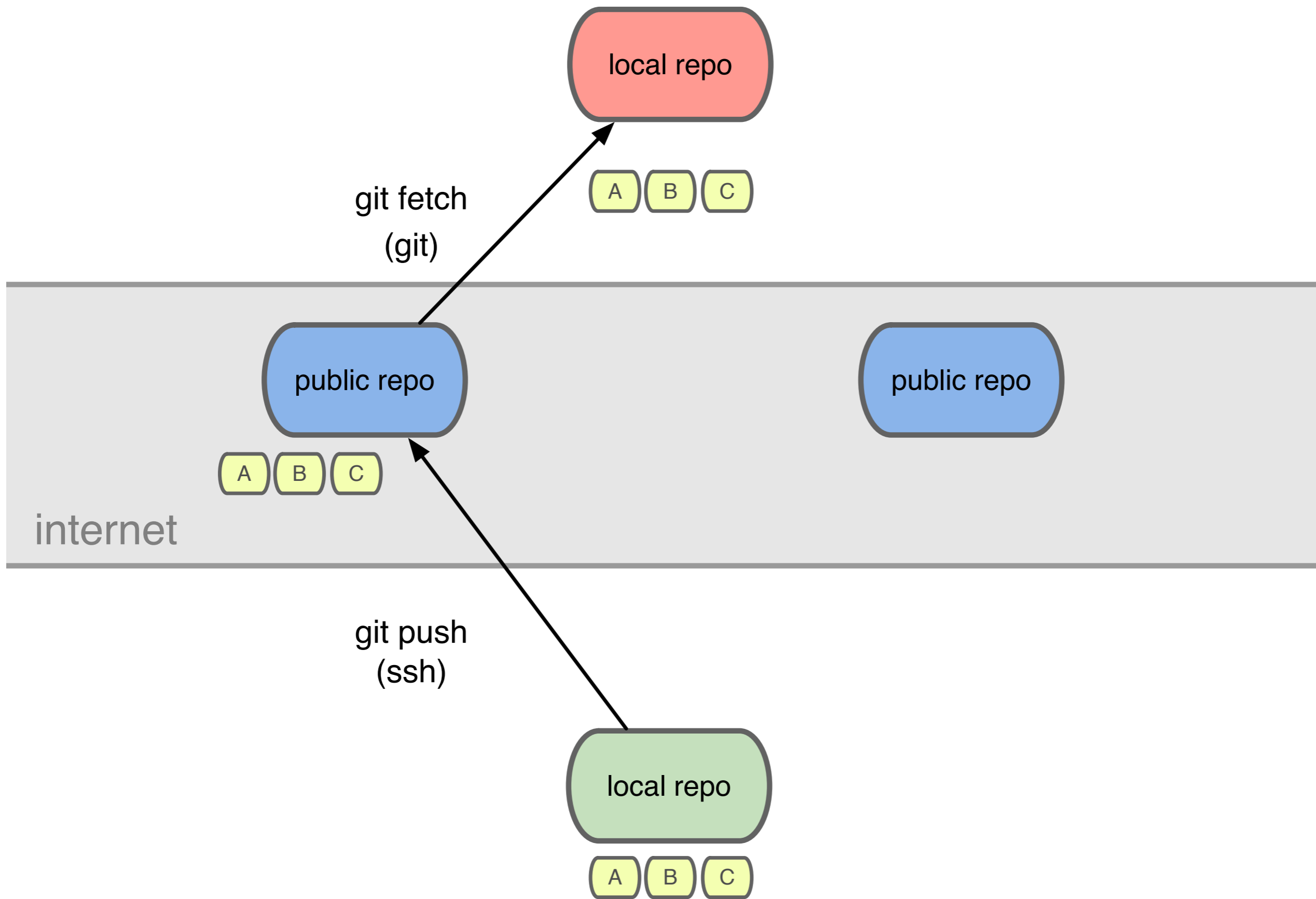
git push

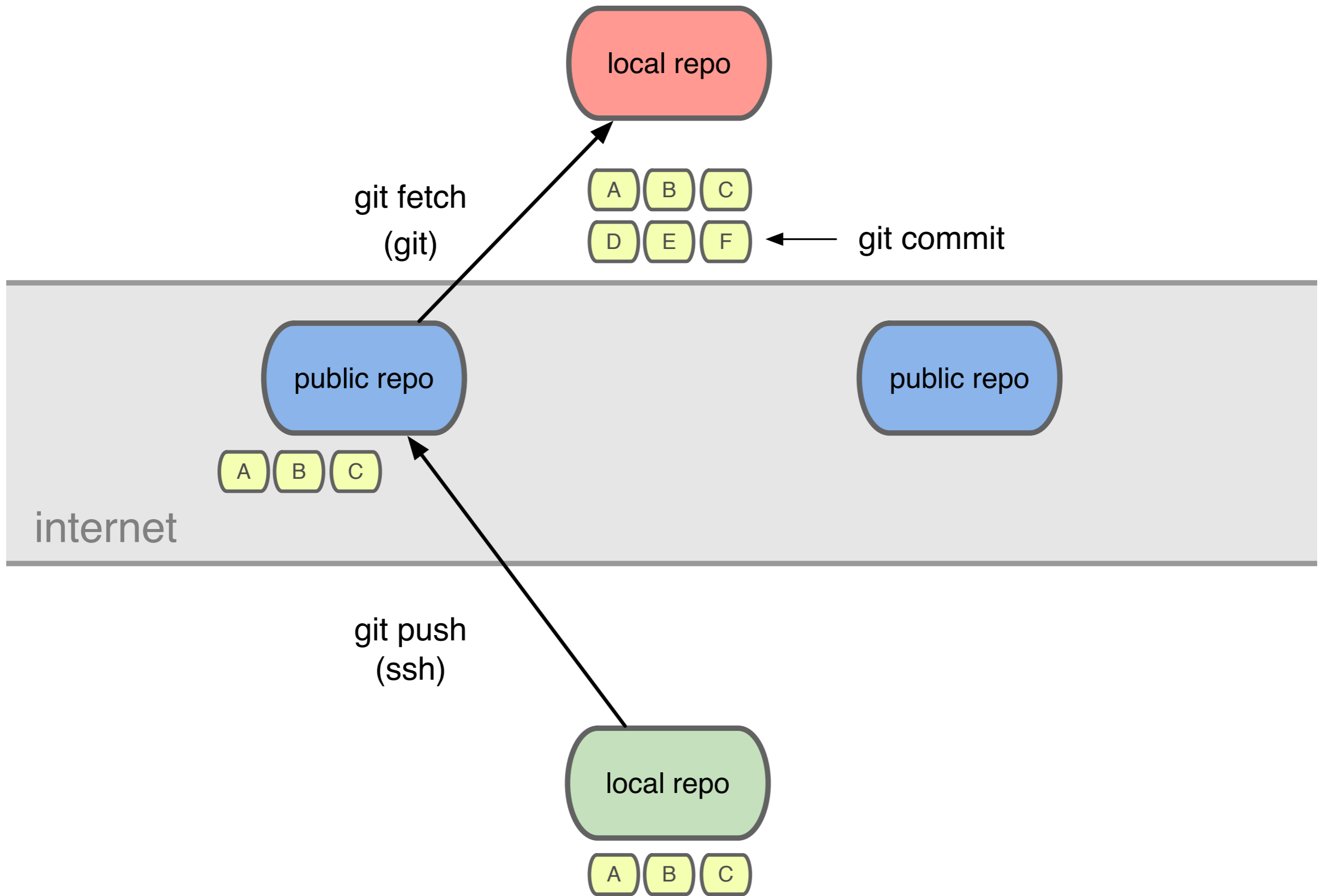
```
$>git remote
public
$>git remote show public
* remote public
  URL: git@github.com:schacon/simplegit.git
  Tracked remote branches
    master
$>git push public master
Counting objects: 7, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 365 bytes, done.
Total 4 (delta 2), reused 0 (delta 0)
refs/heads/master: f40831985e265eb8cca2d0e66c11ac48ccfe170d -> fc62
5543b195f18391886b9f663d5a7eca38e84
To git@github.com:schacon/simplegit.git
  f408319..fc62e55  master -> master
$>_
```

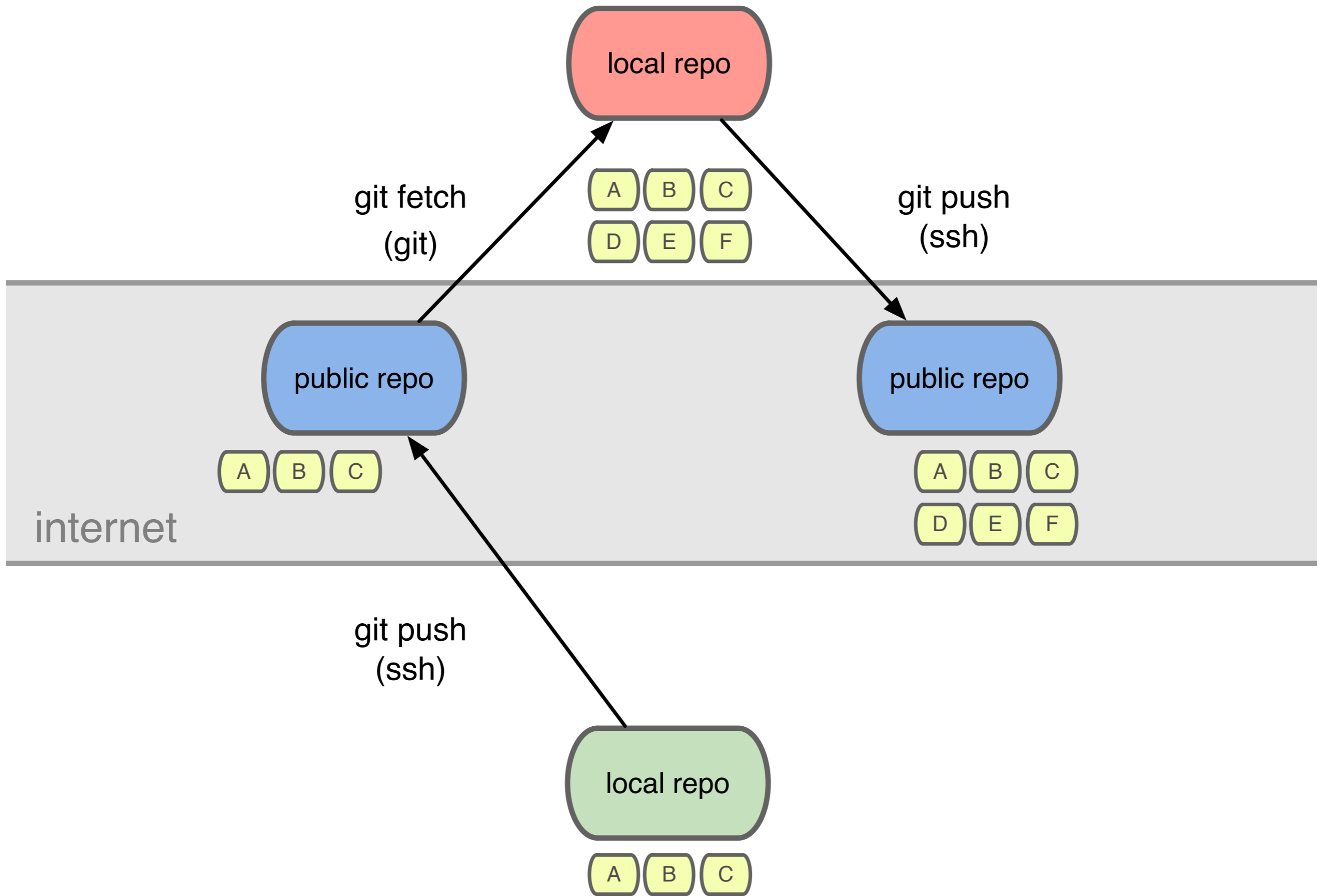
a who in the what now?

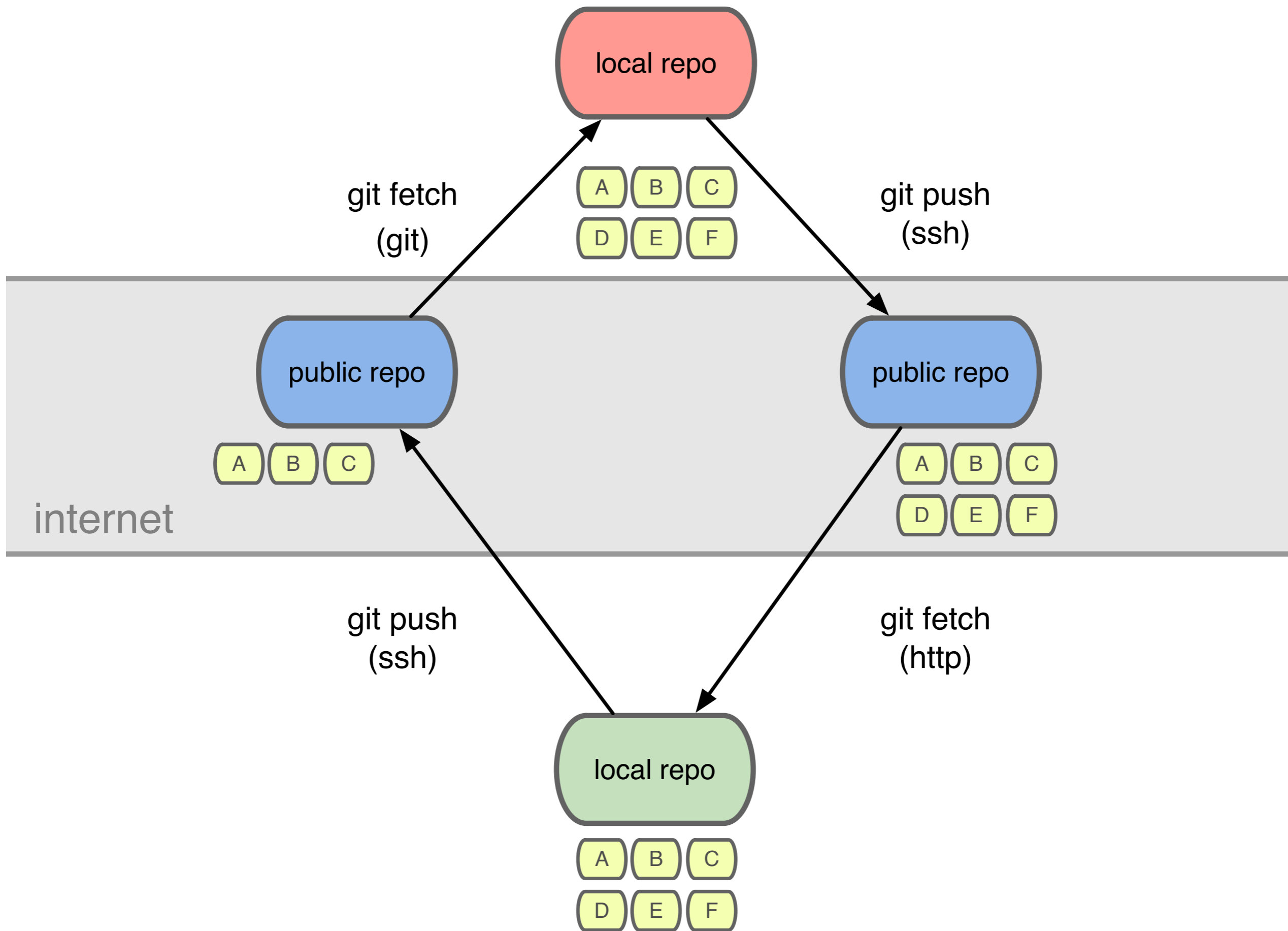


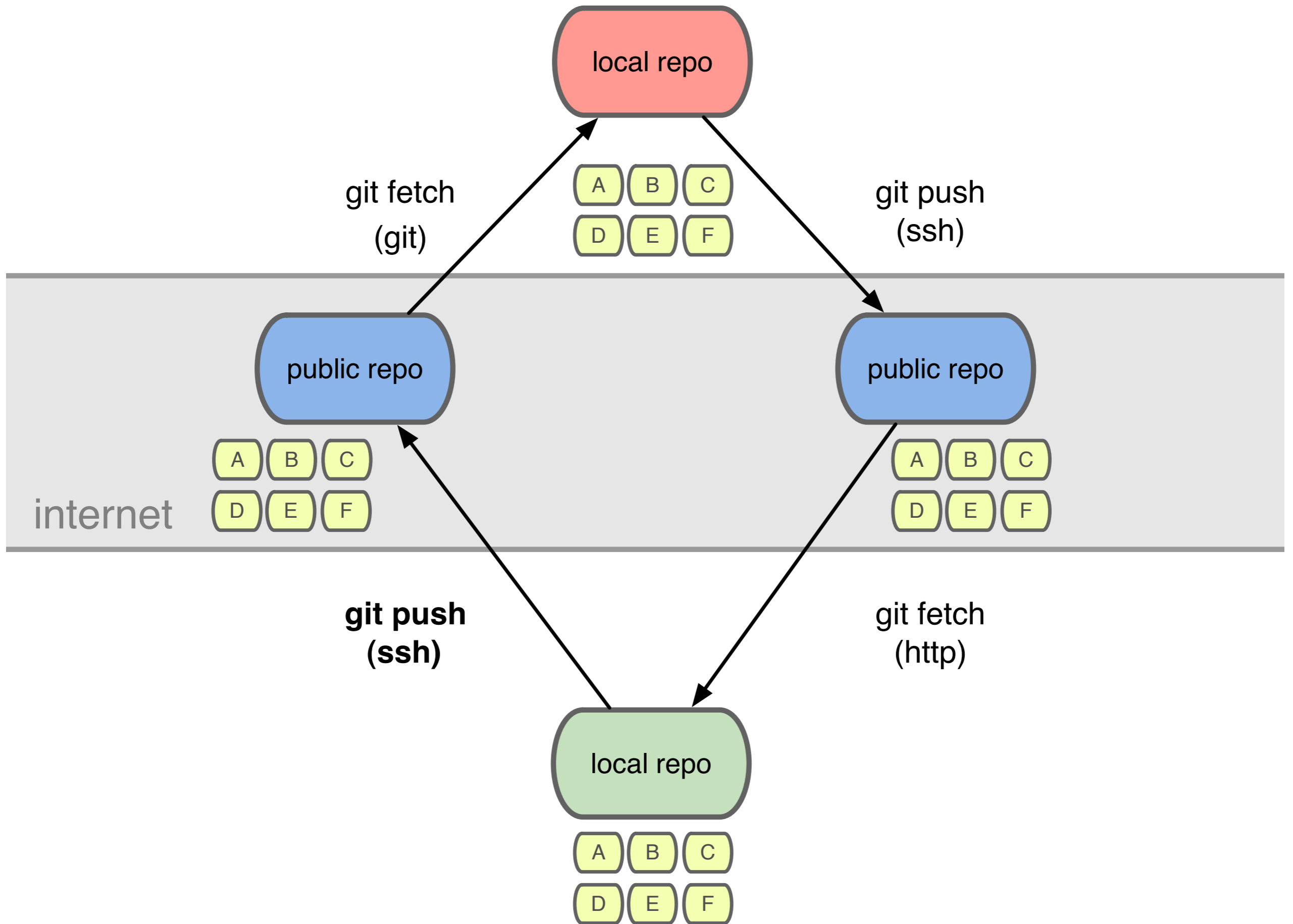








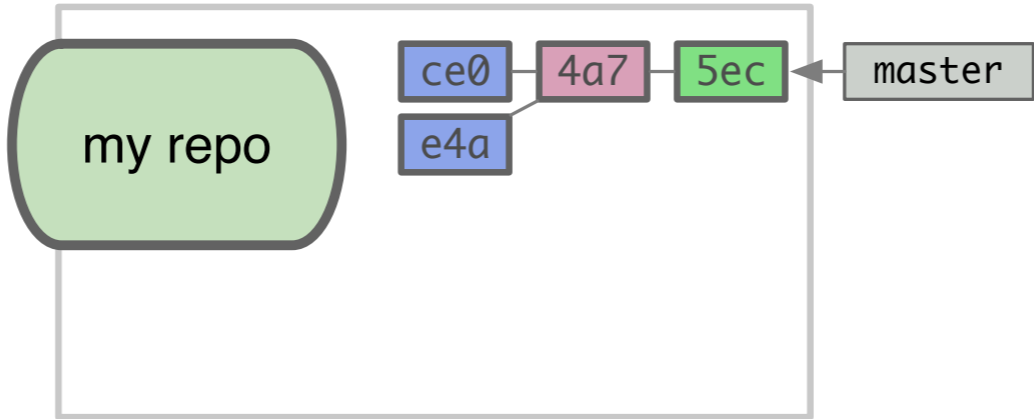




multiple remotes

developer
nick

developer
jessica

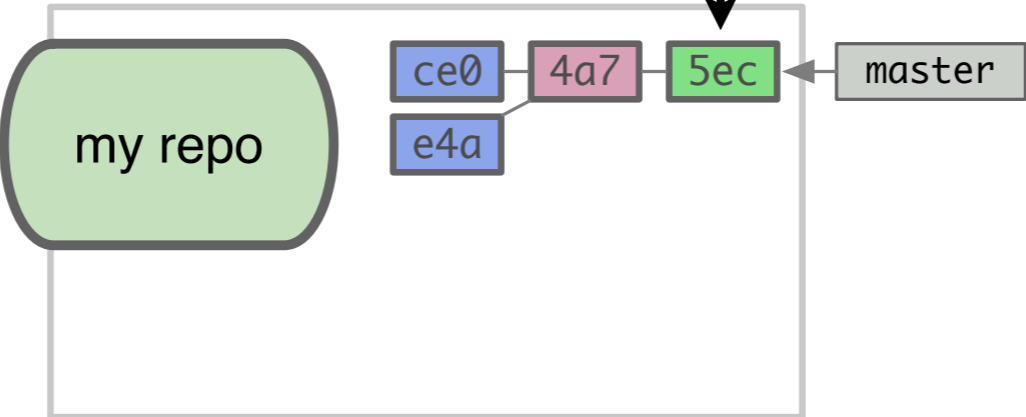


developer
nick

developer
jessica

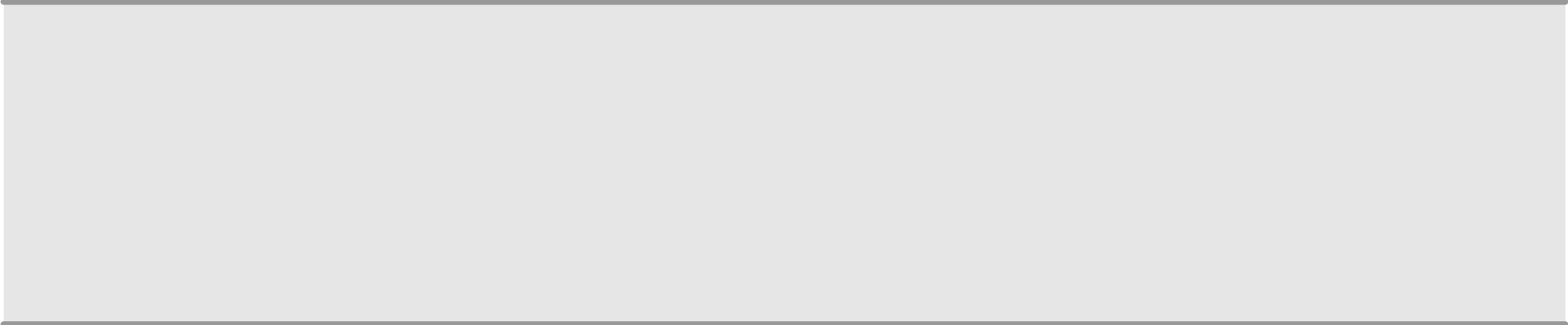


commit

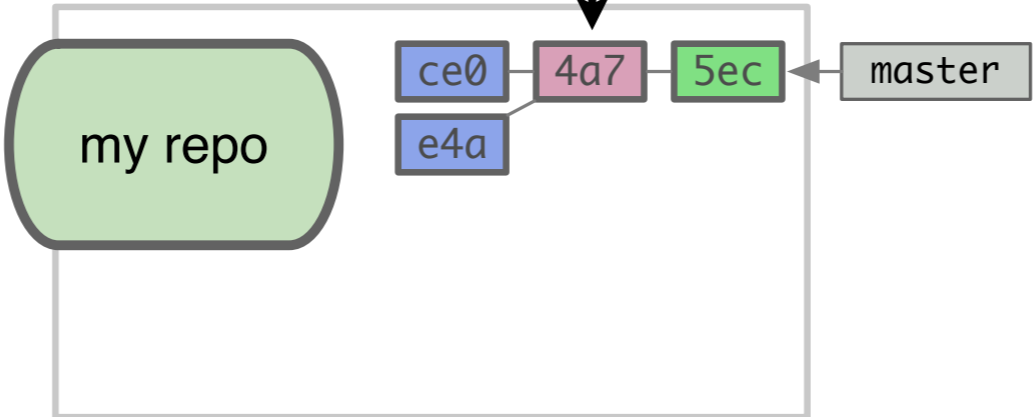


developer
nick

developer
jessica



tree

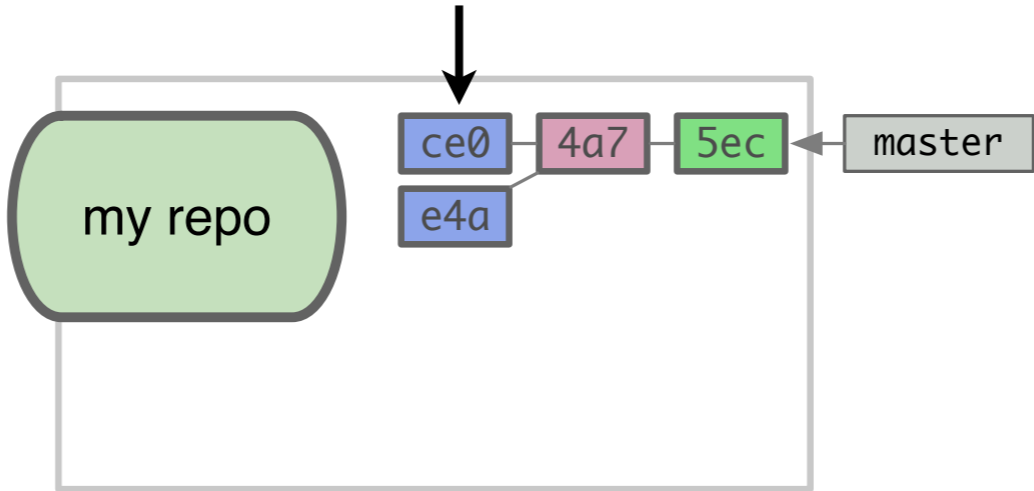


developer
nick

developer
jessica

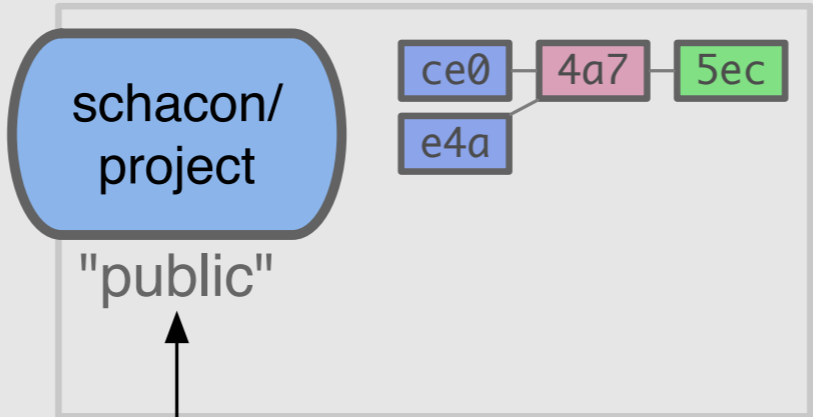


blobs

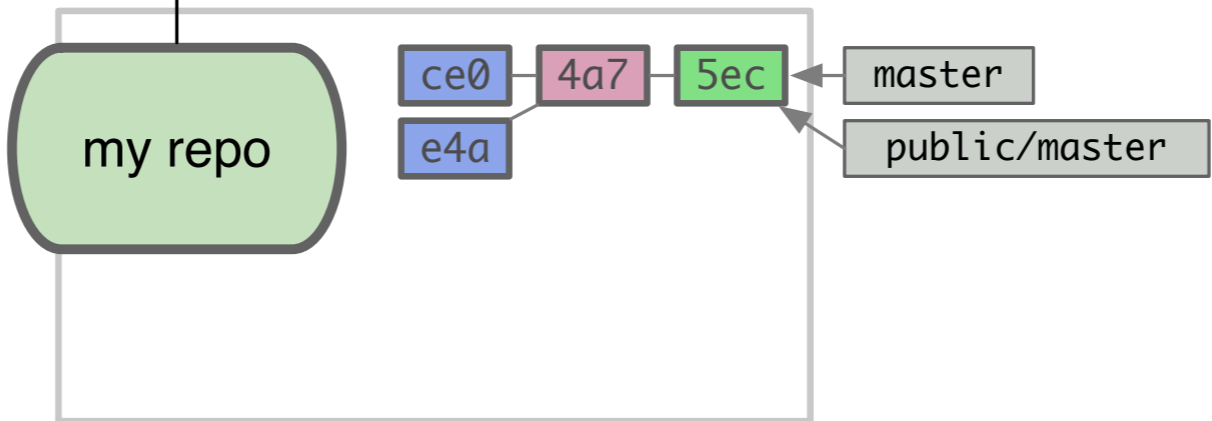


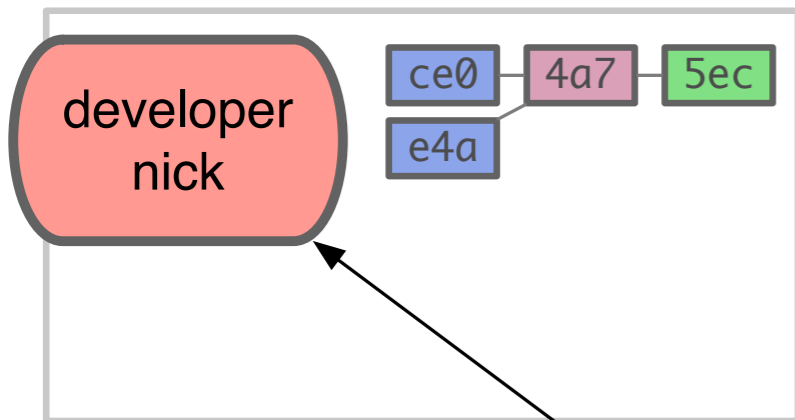
developer
nick

developer
jessica

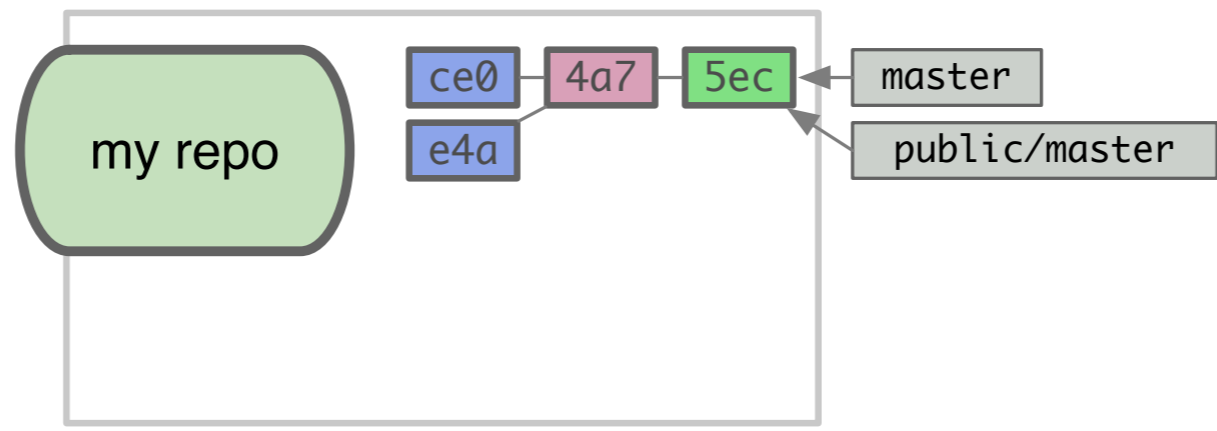
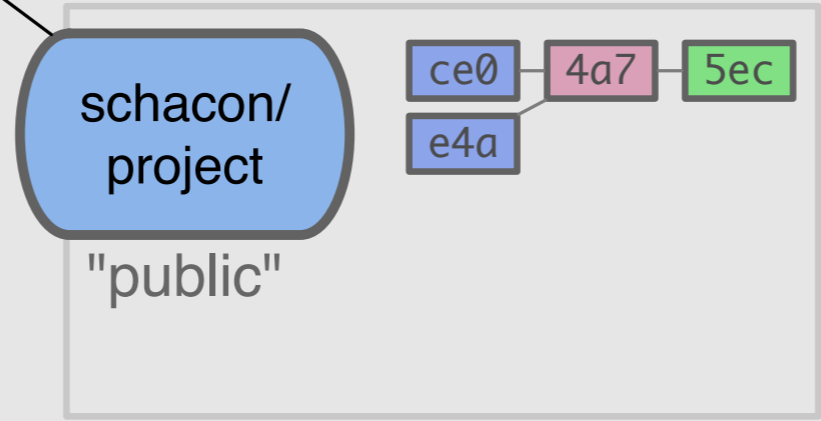


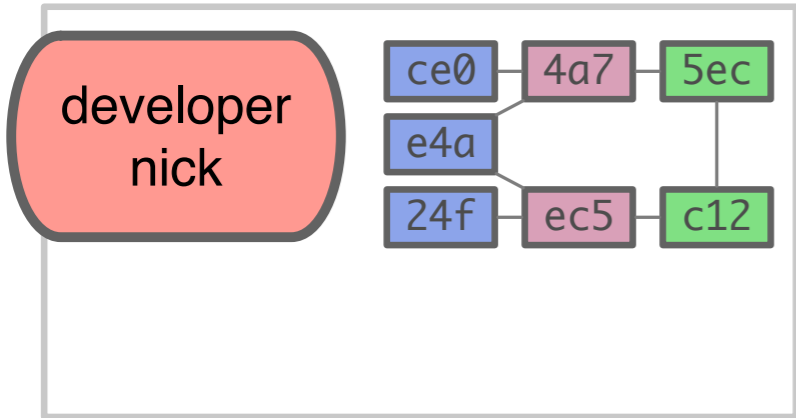
git push public



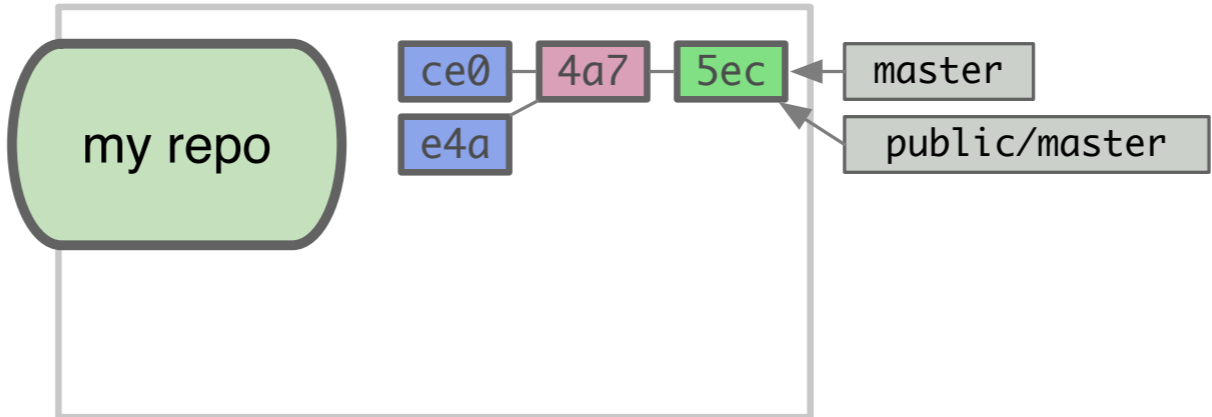
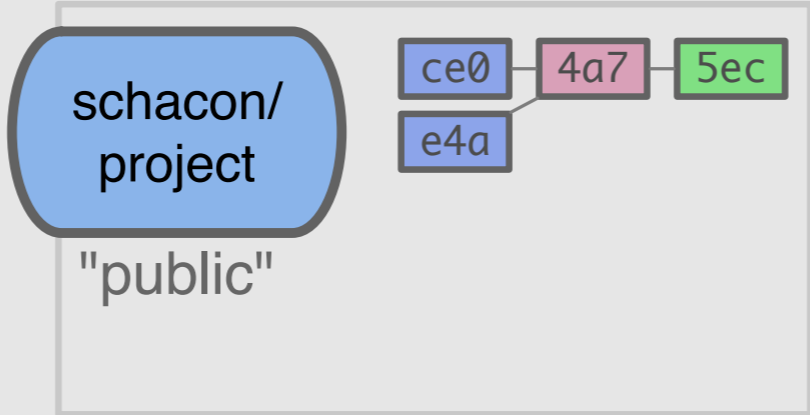


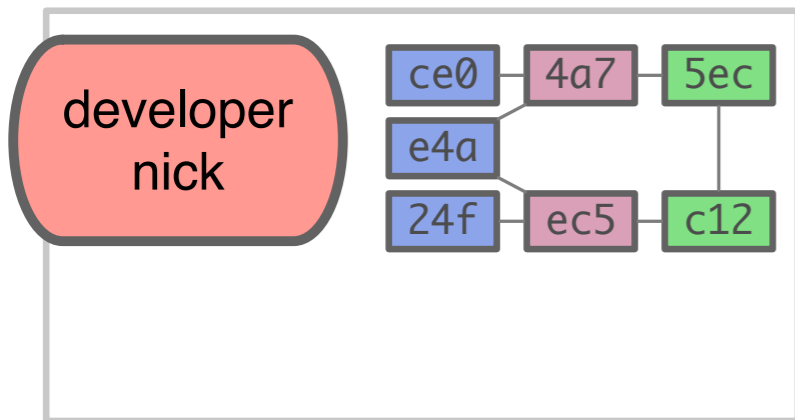
git clone (url)



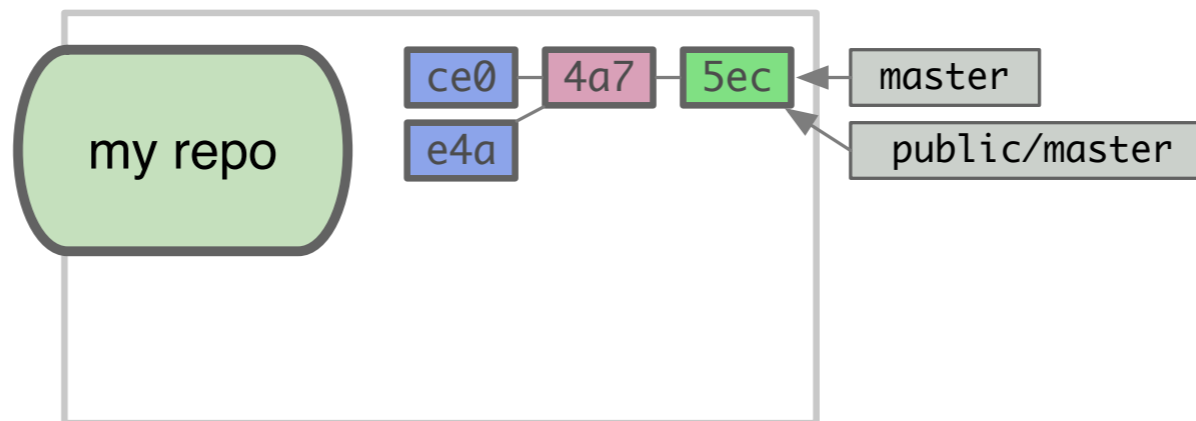
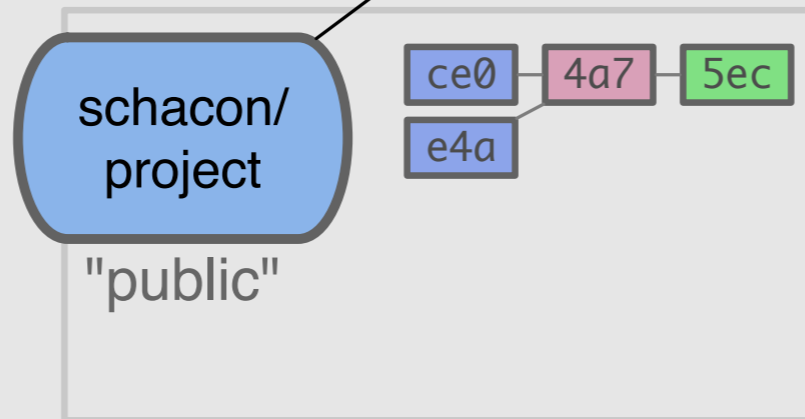
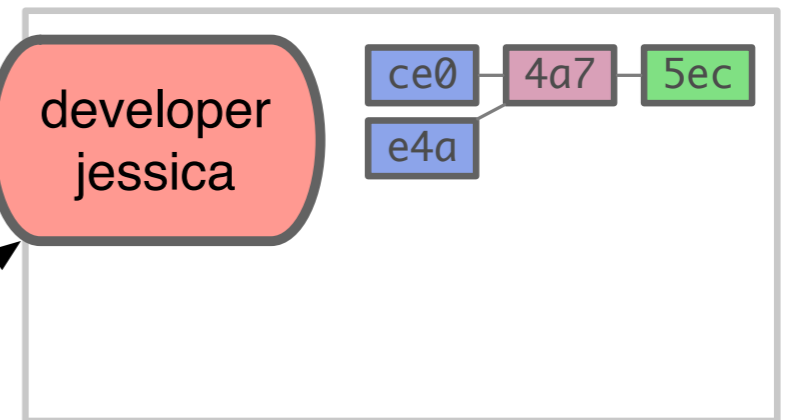


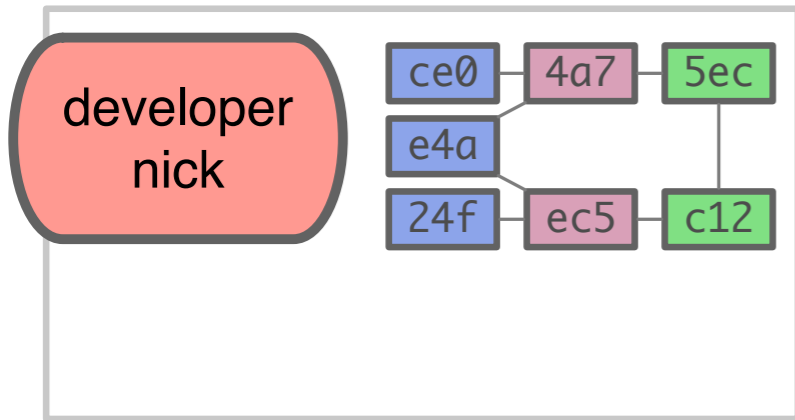
git commit



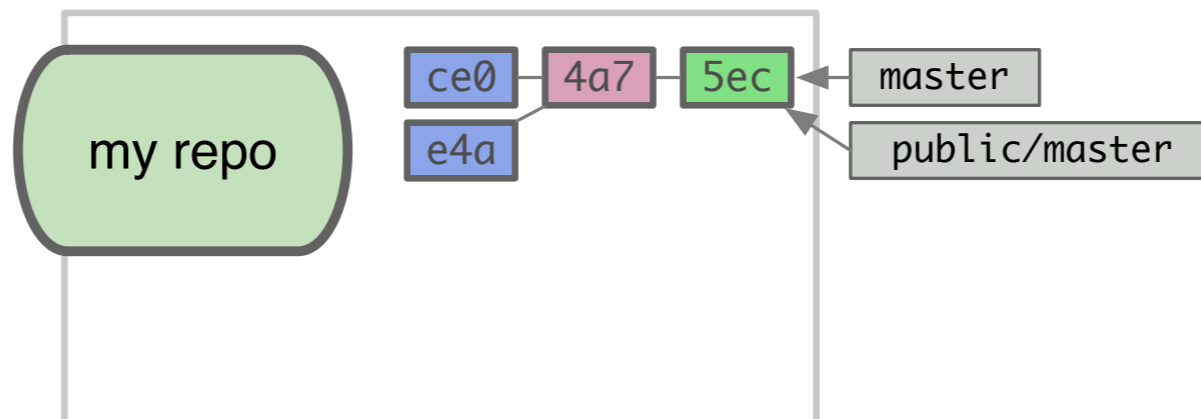
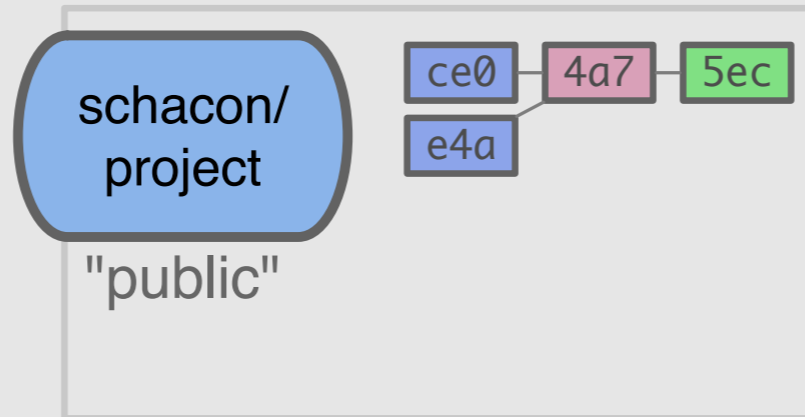
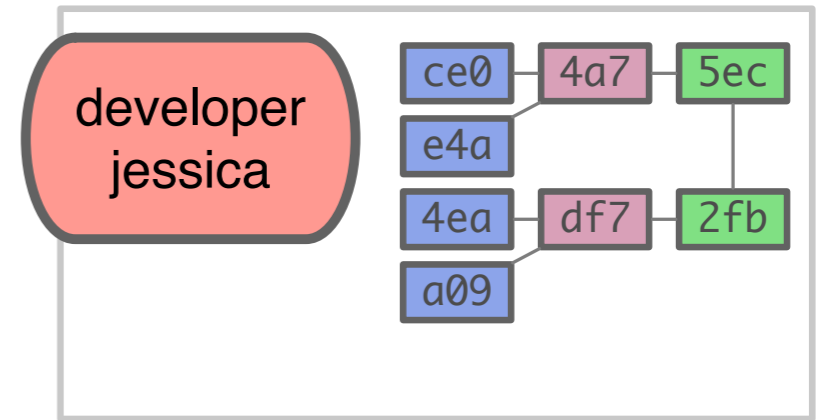


git clone (url)

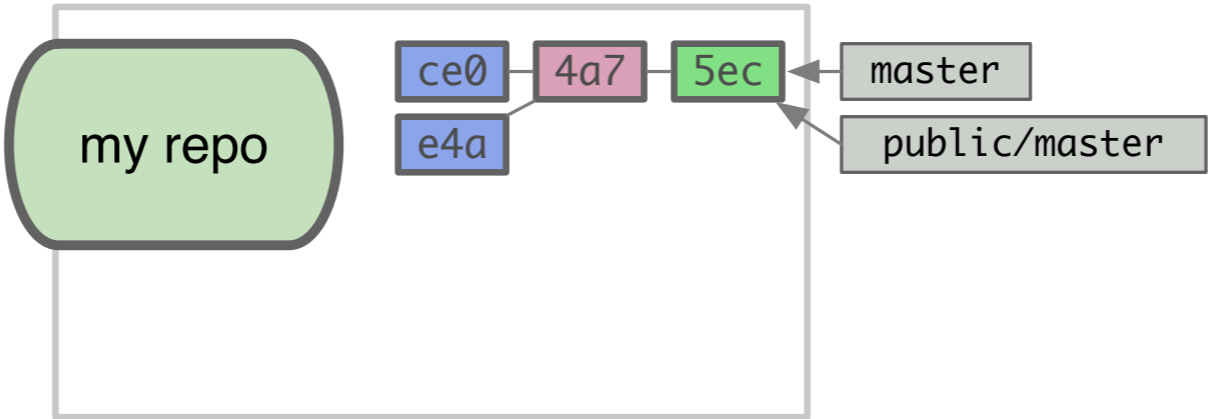
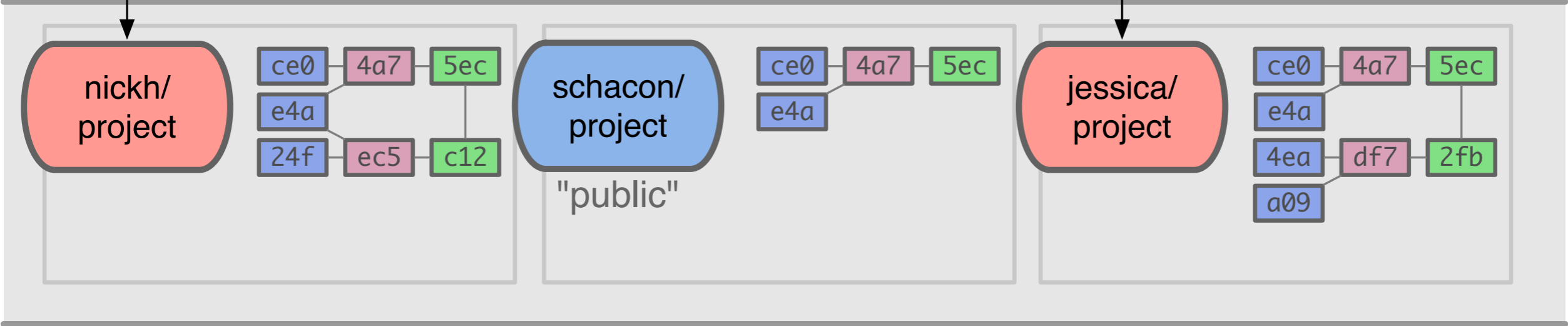
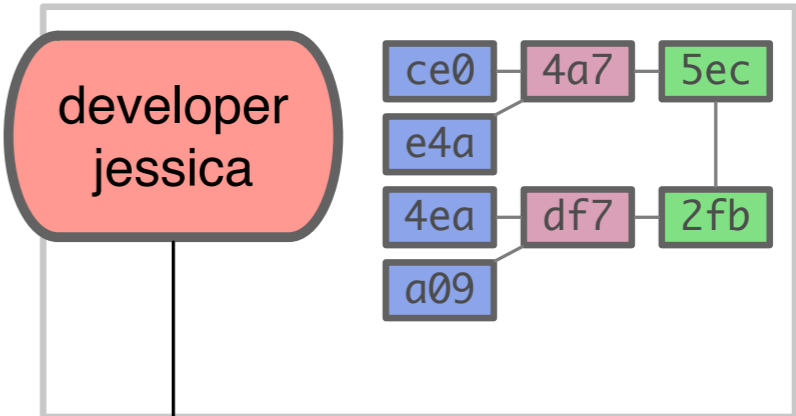
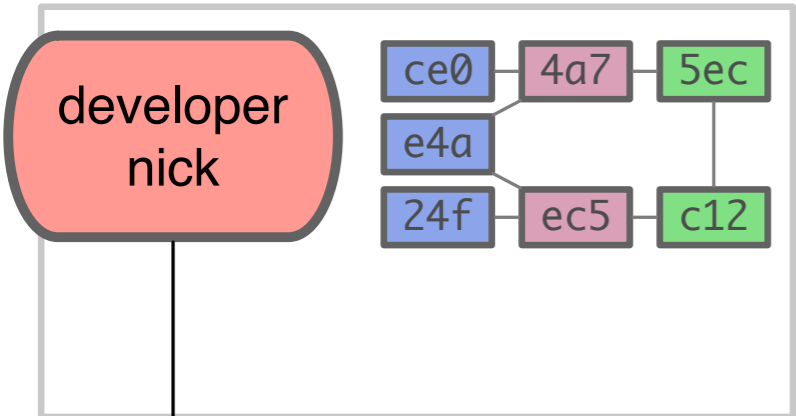


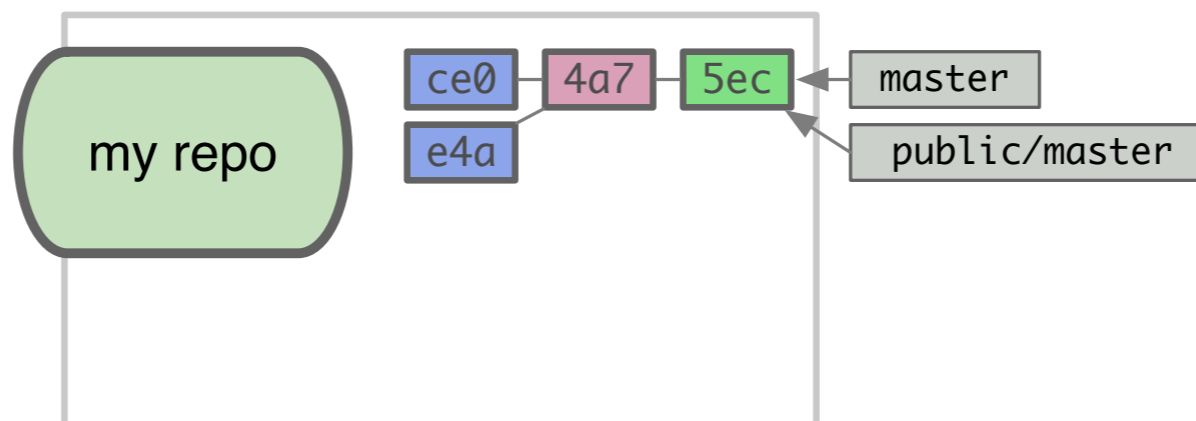
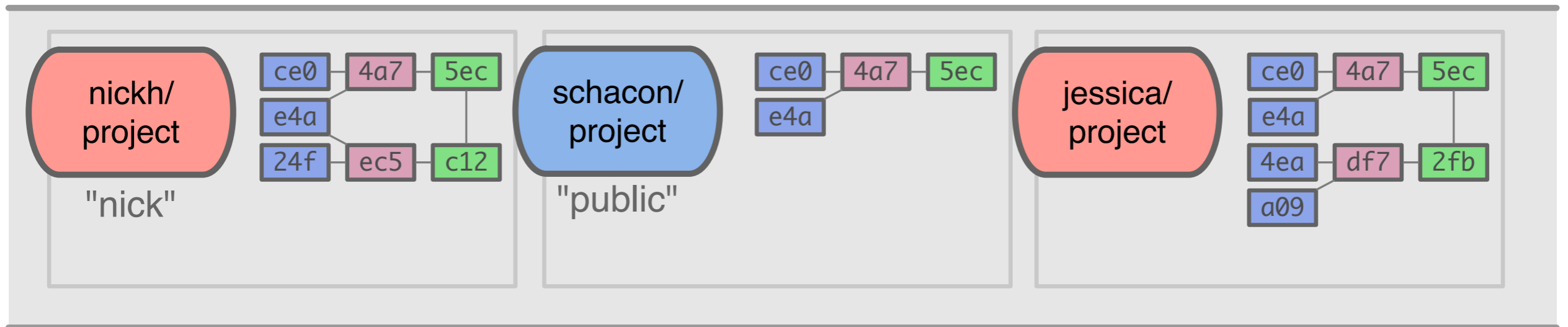
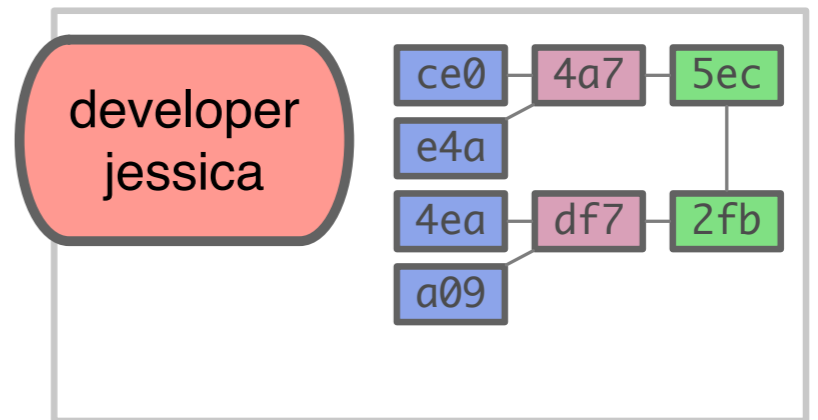
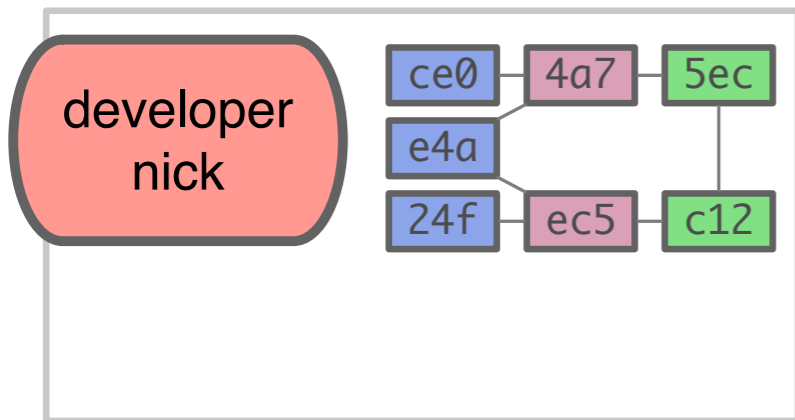


git commit

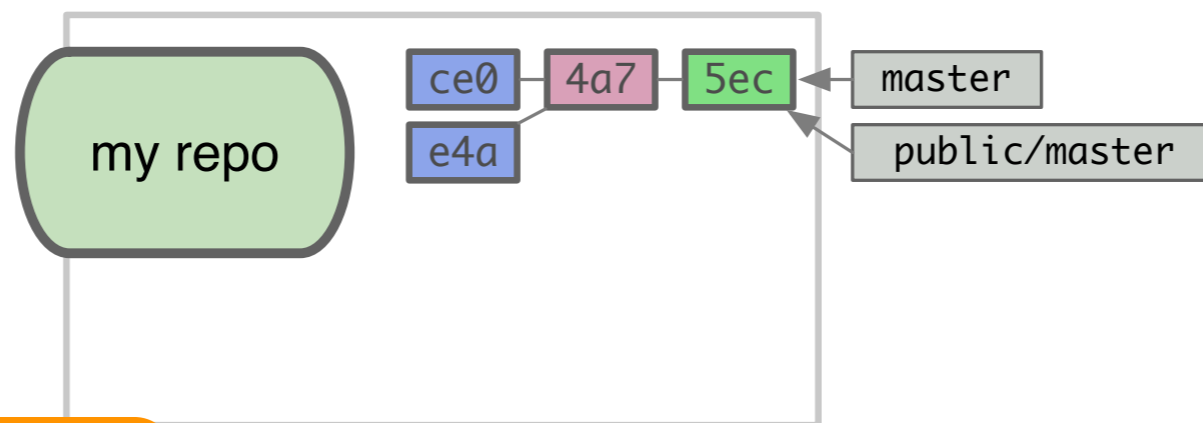
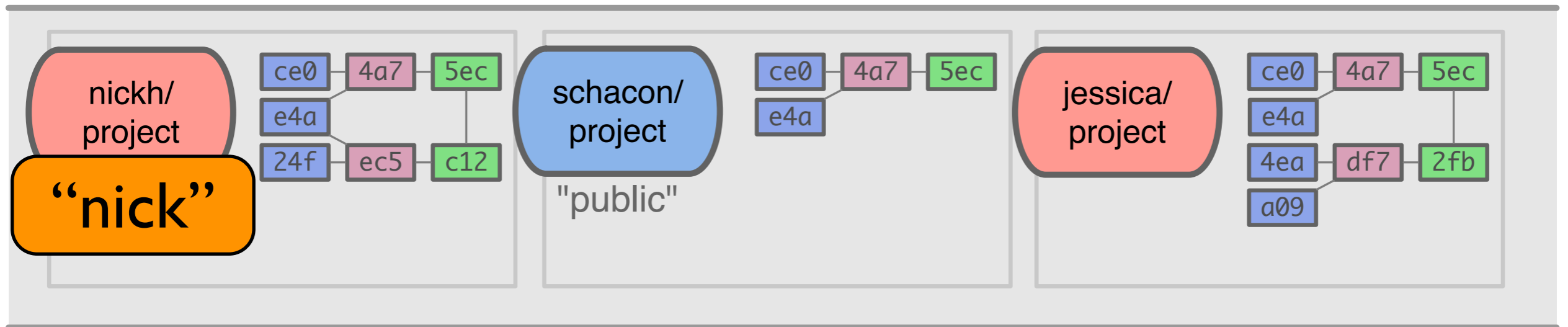
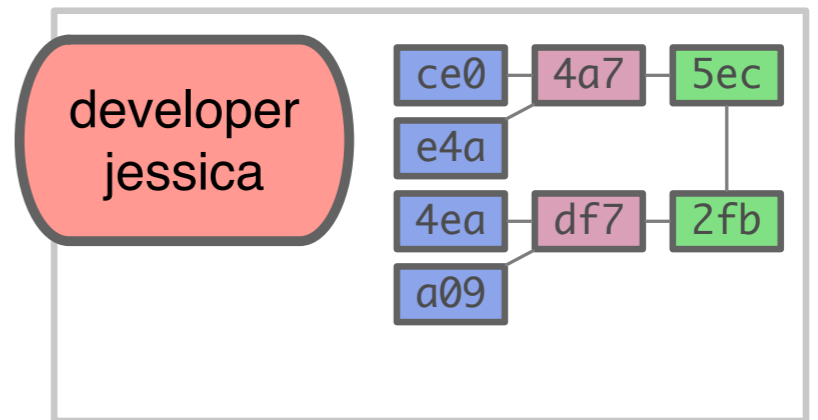
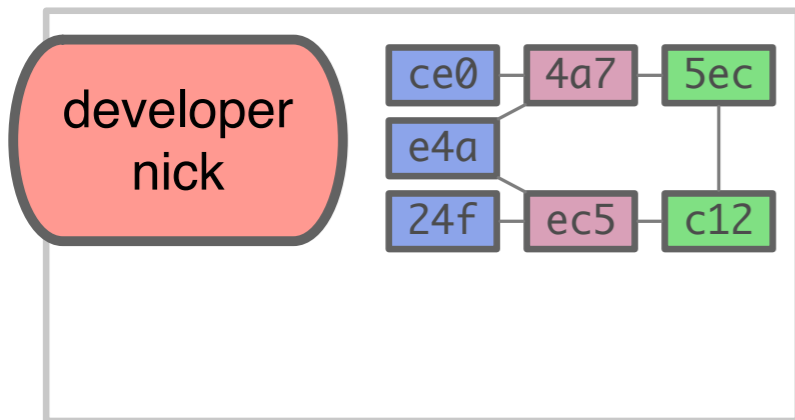


git push

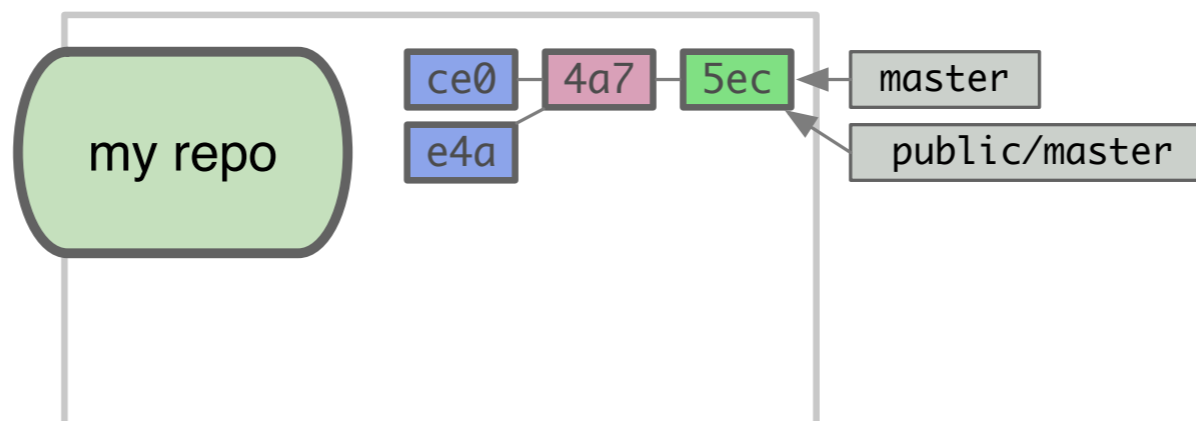
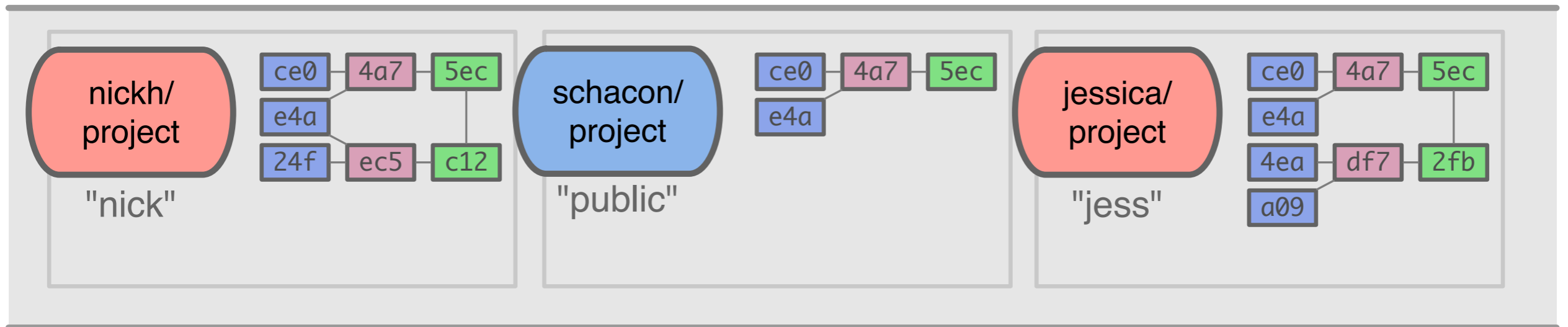
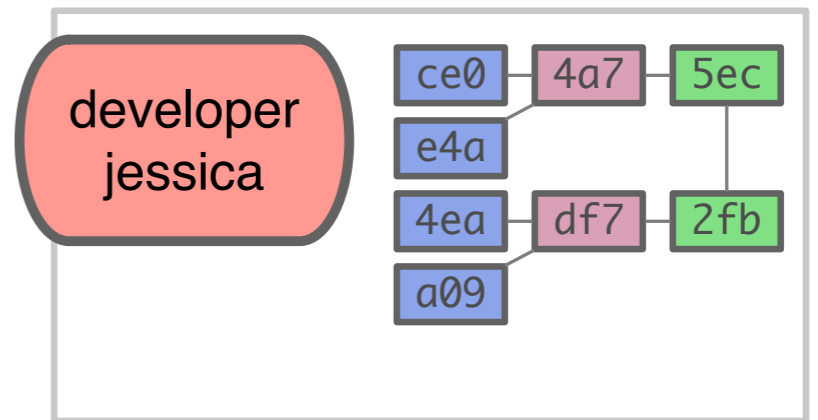
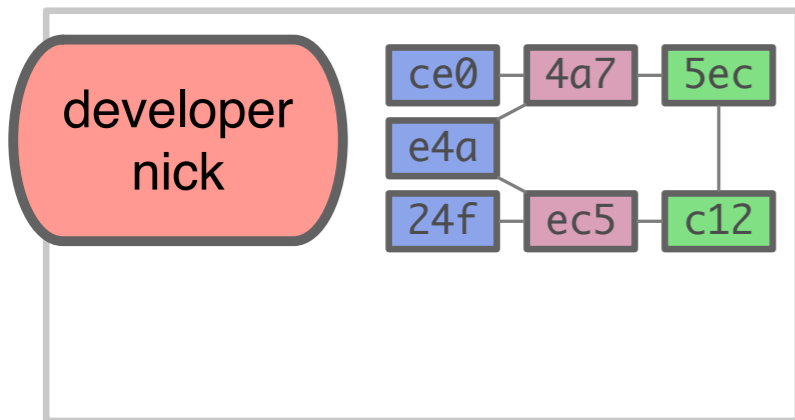




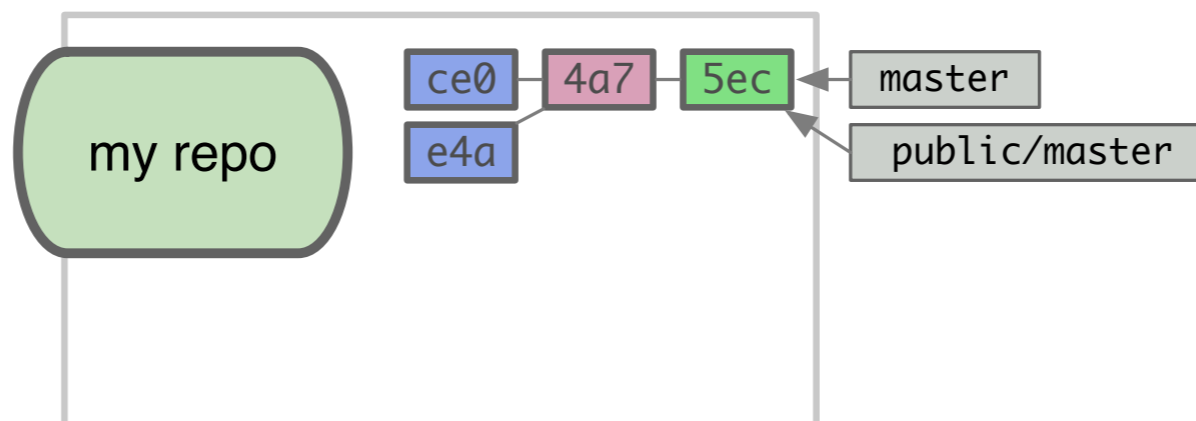
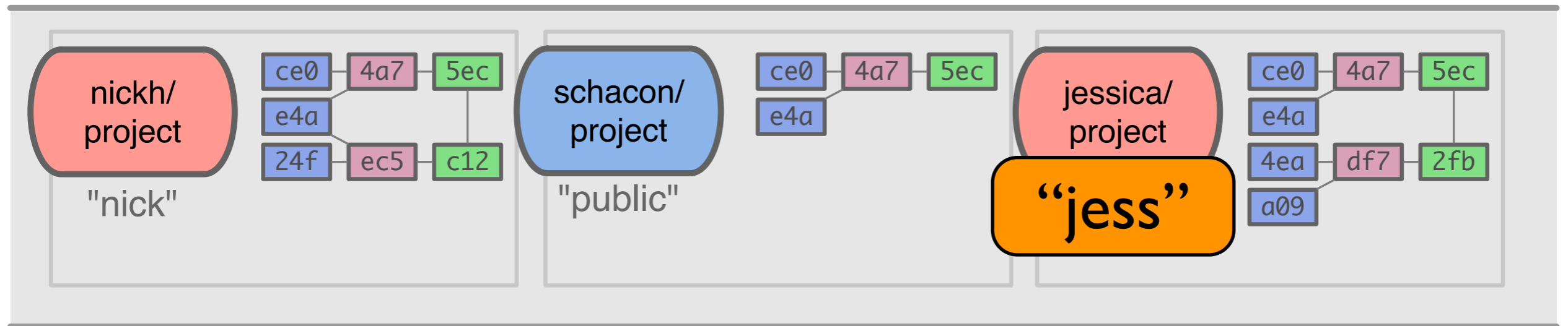
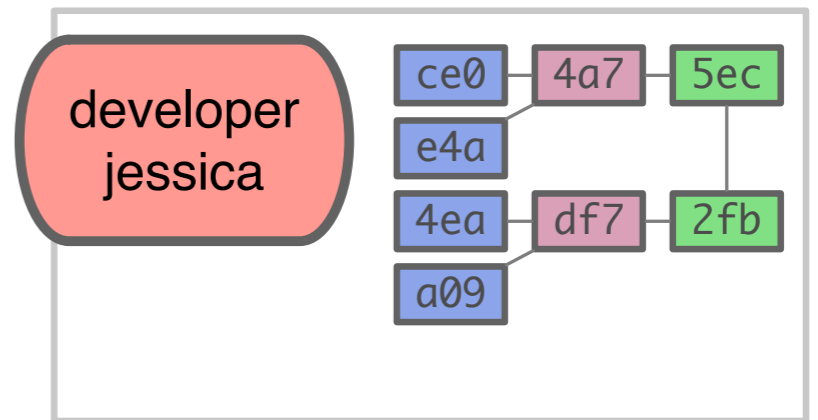
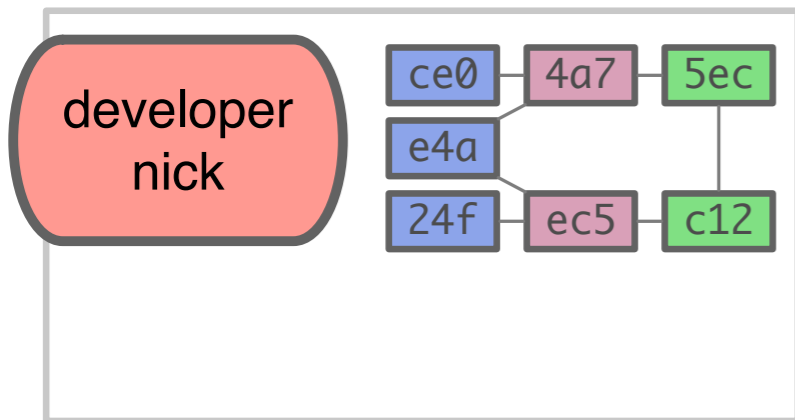
`git remote add nick git://github.com/nickh/project.git`



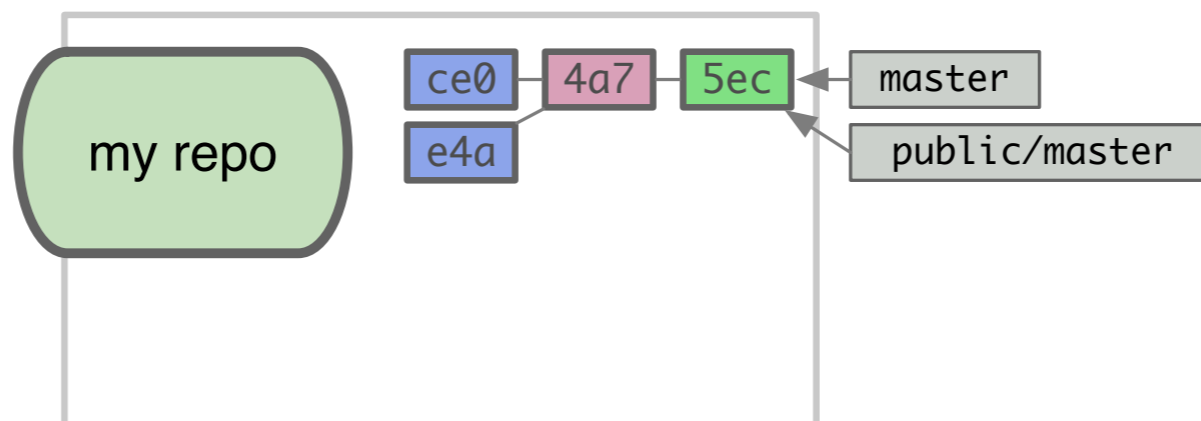
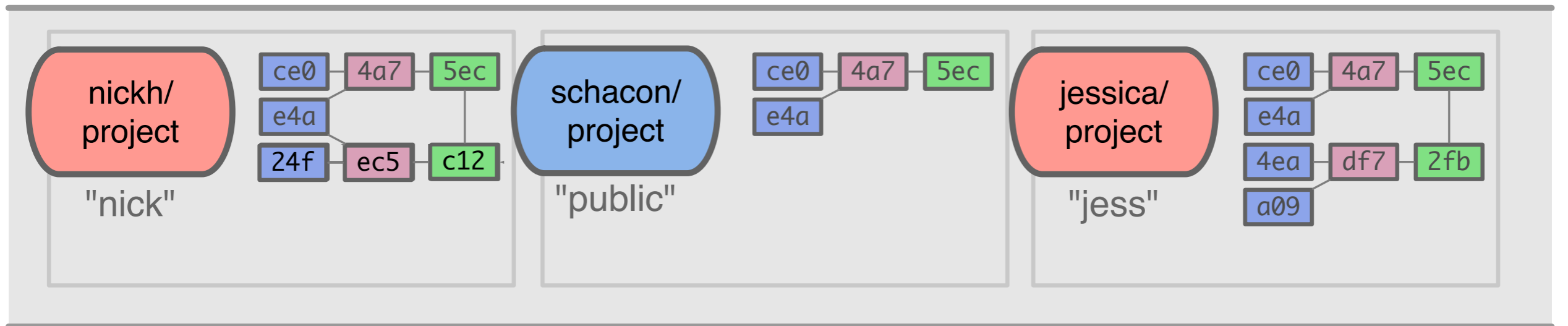
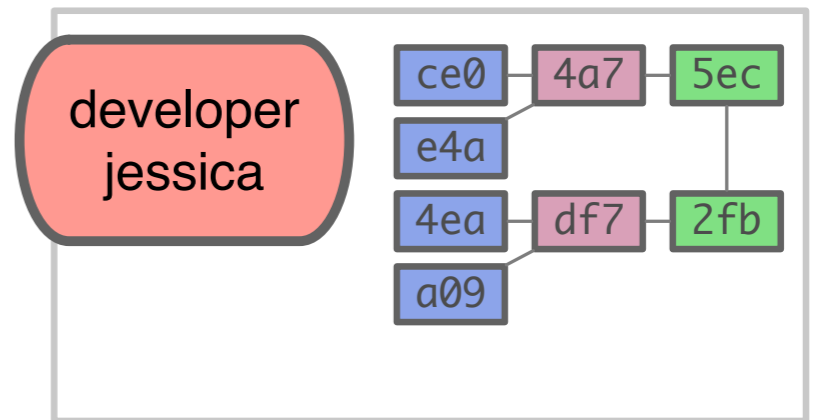
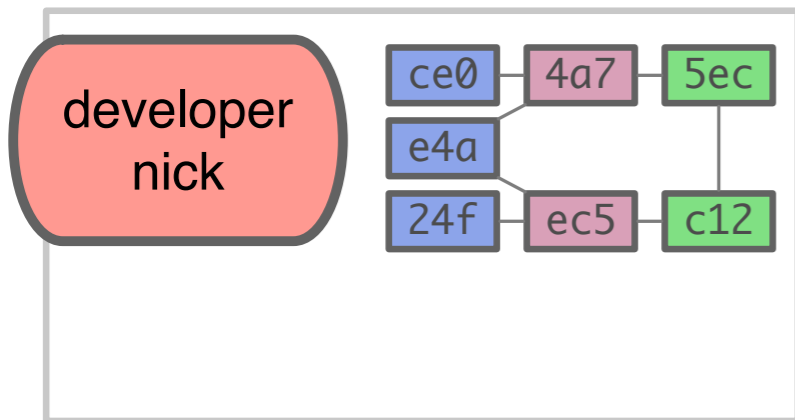
git remote add **nick** git://github.com/nickh/project.git



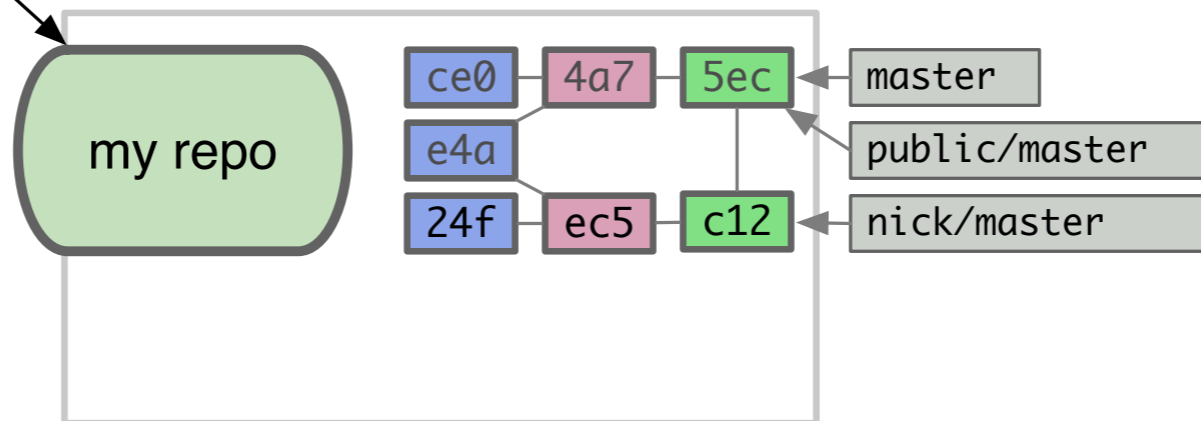
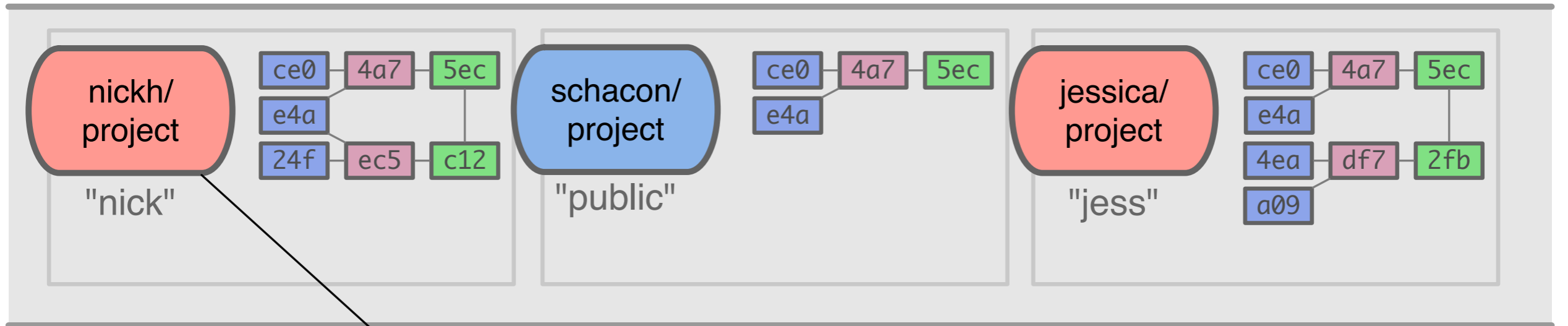
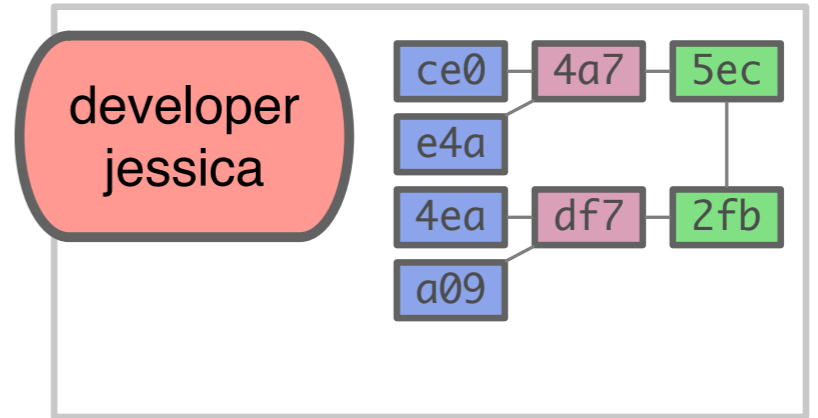
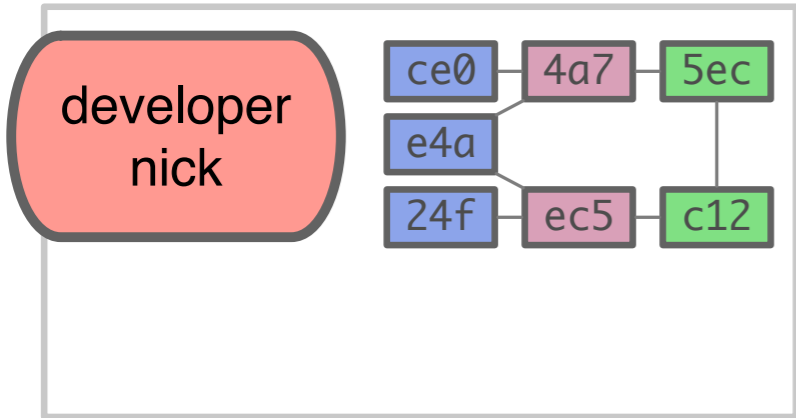
`git remote add jess git://github.com/jessica/project.git`

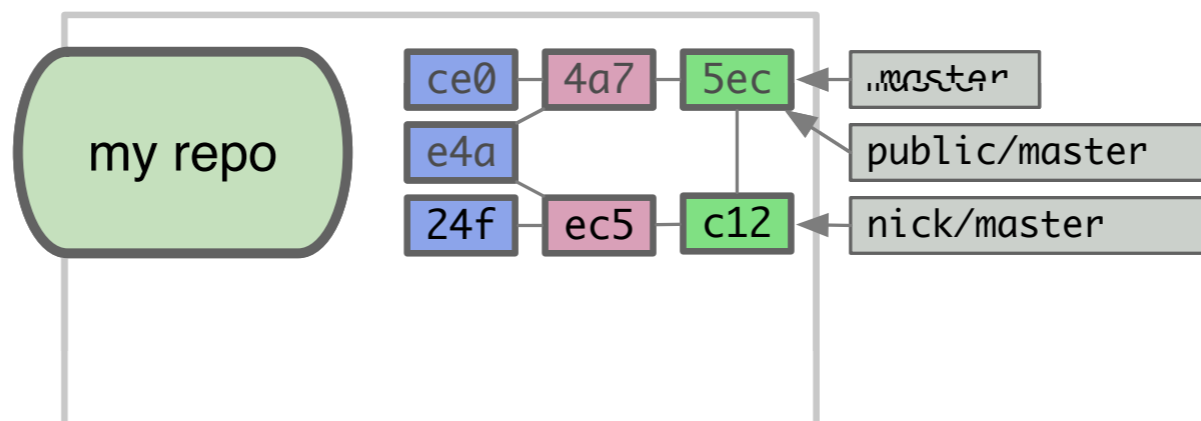
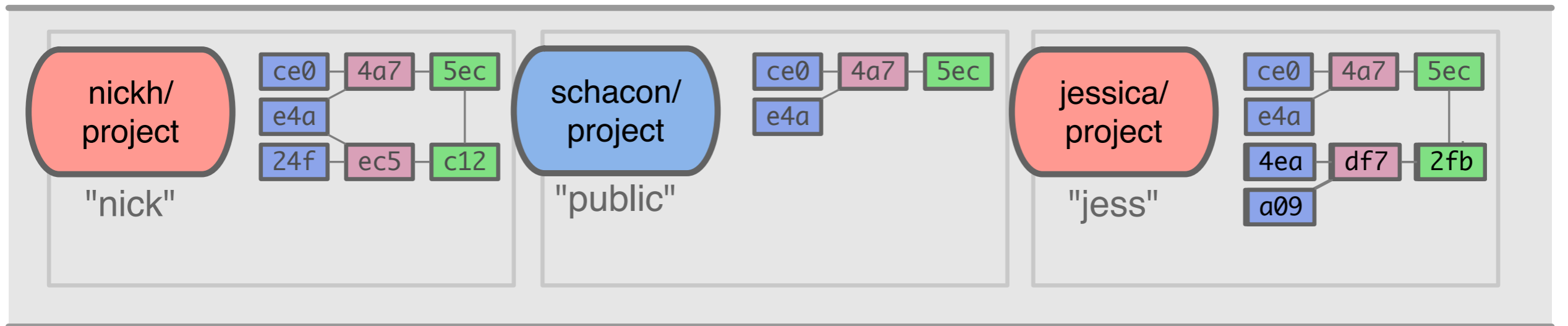
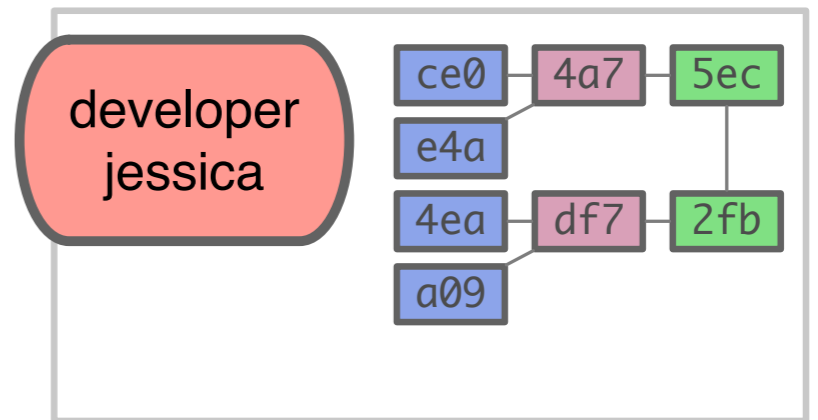
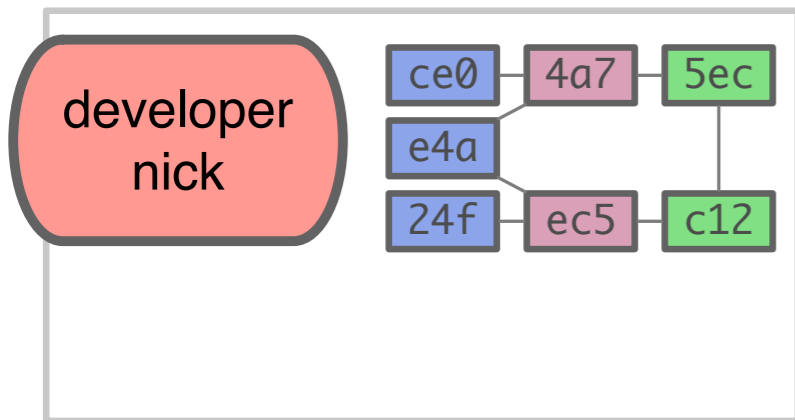


git remote add **jess** git://github.com/jessica/project.git

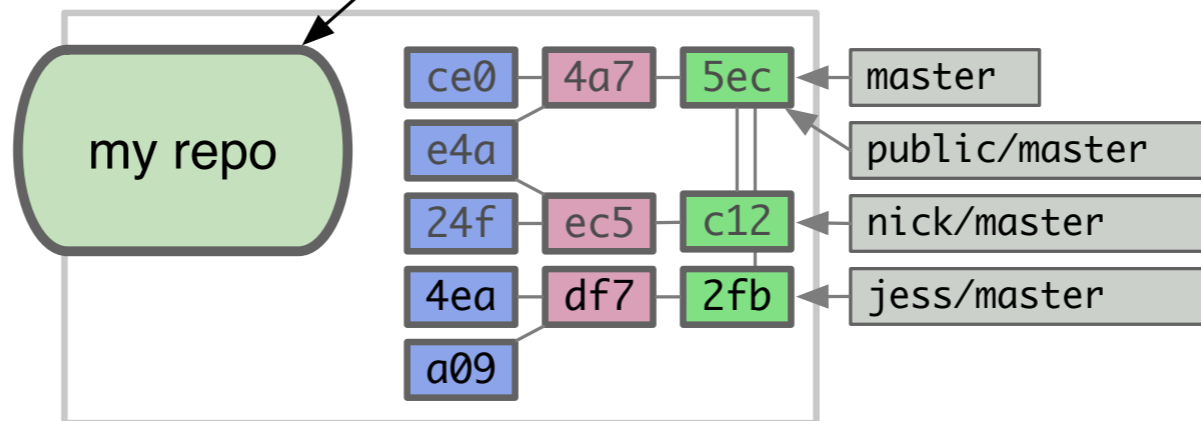
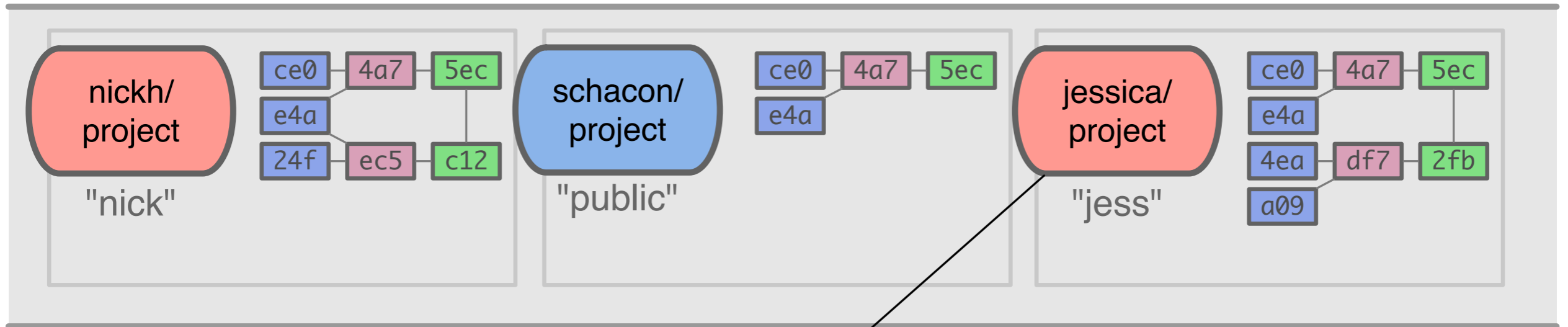
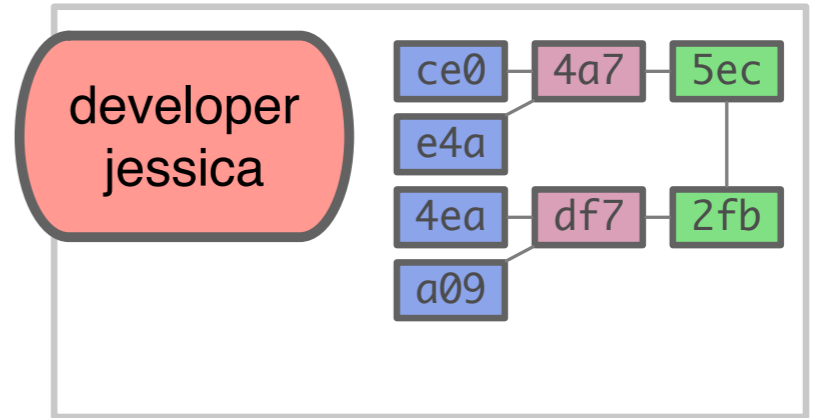
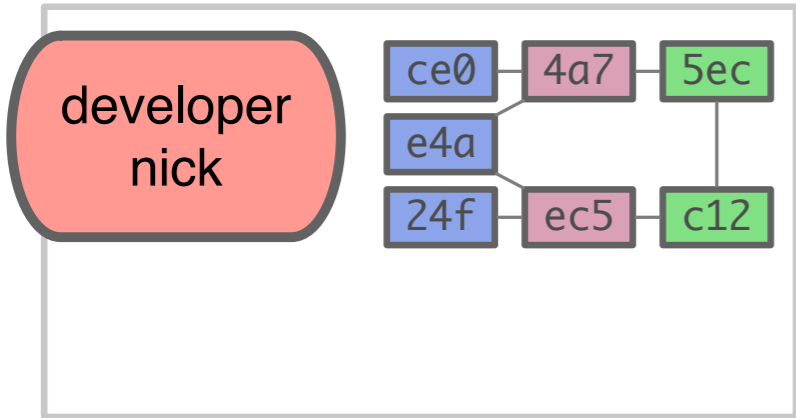


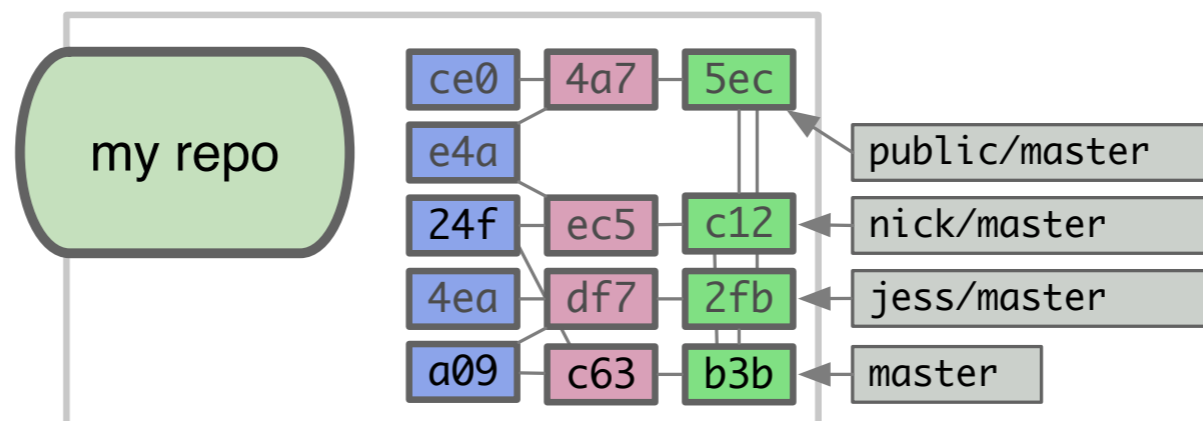
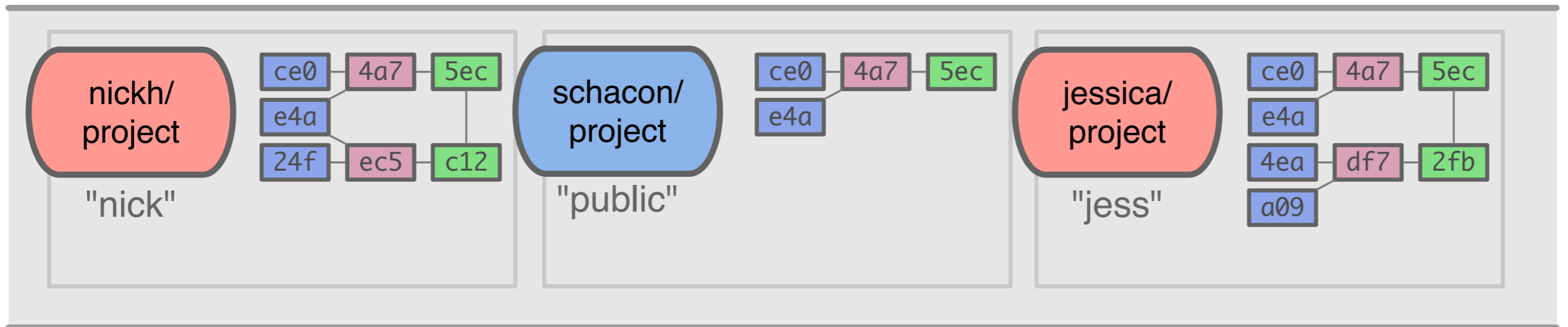
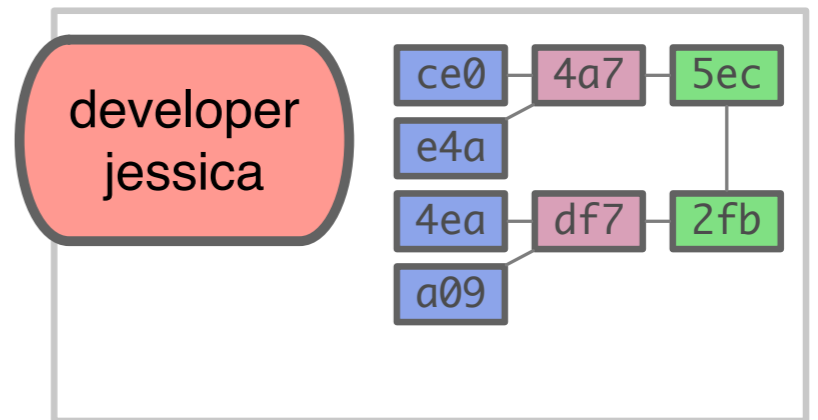
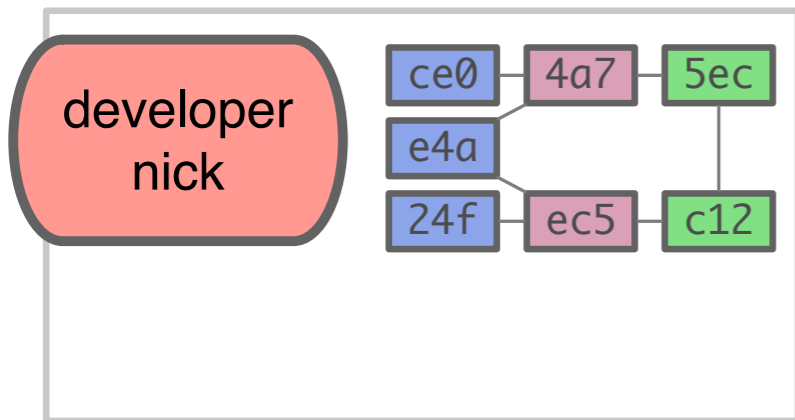
git fetch nick



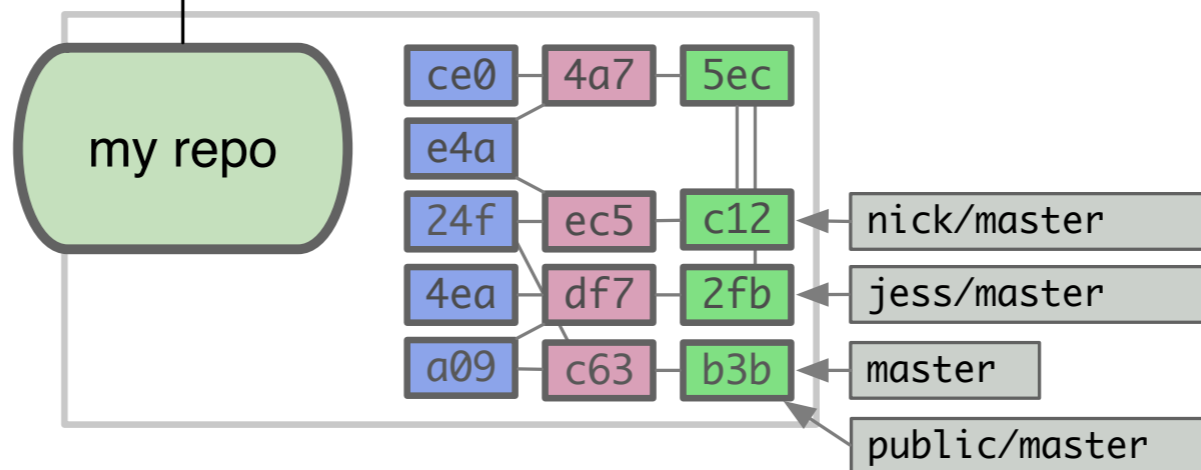
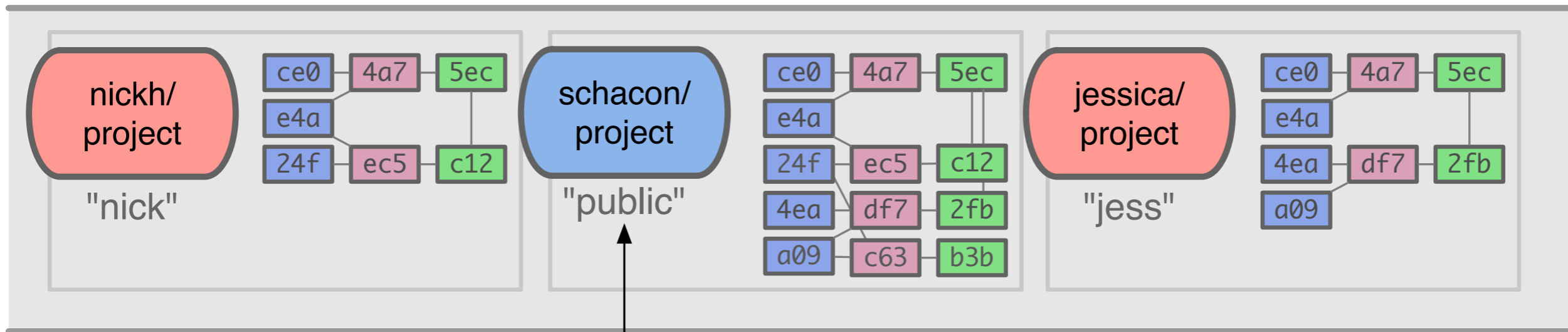
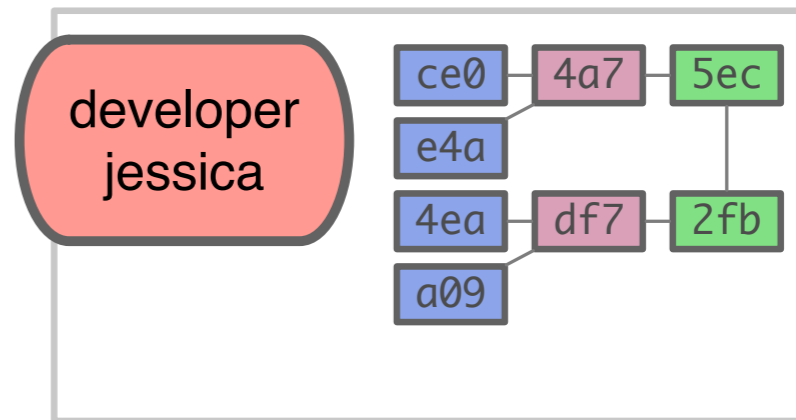
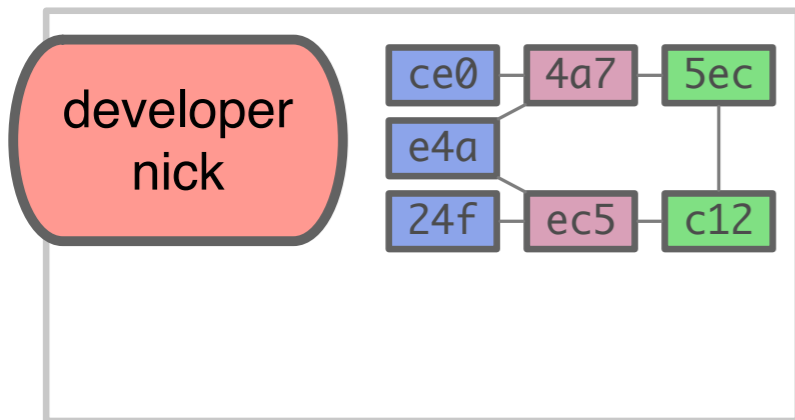


git fetch jess

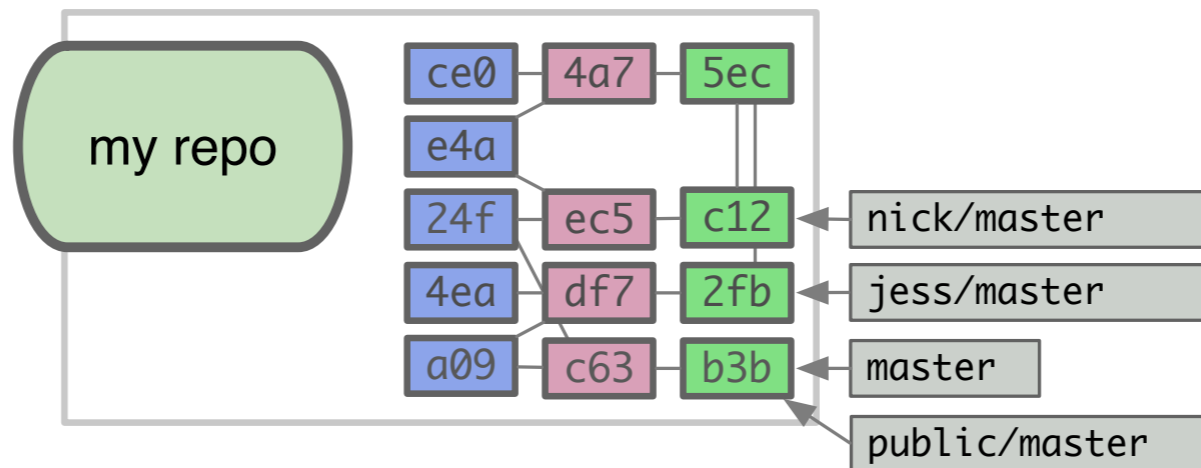
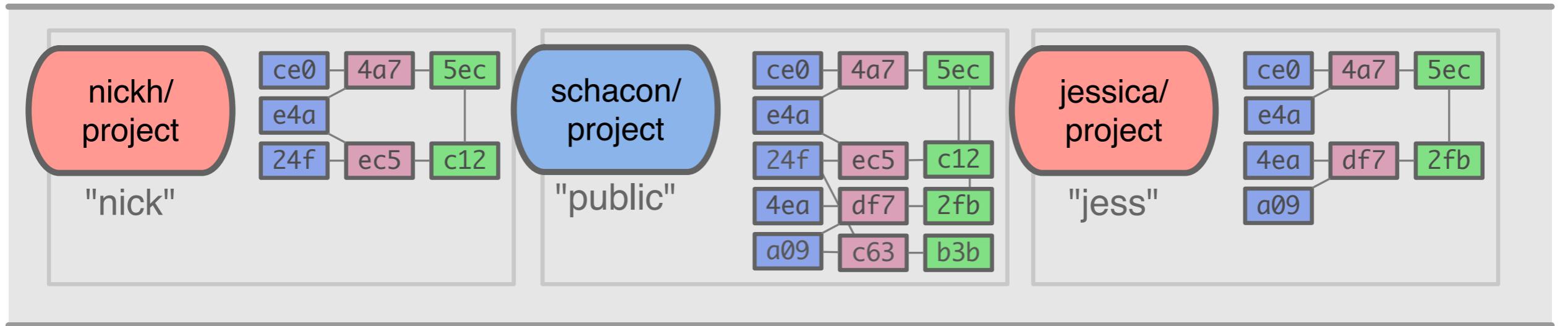
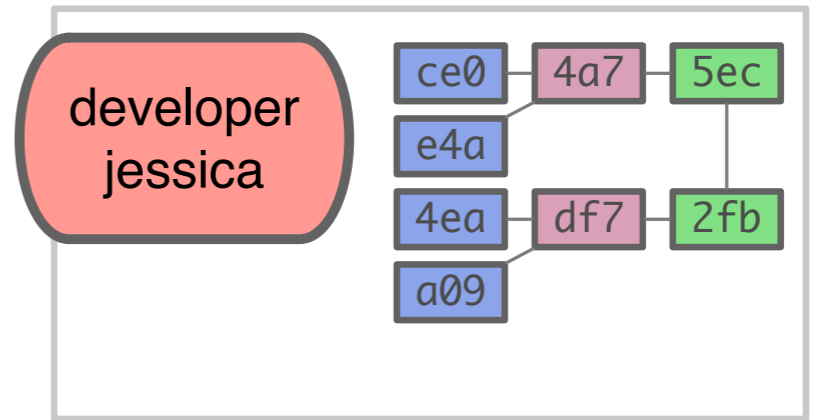
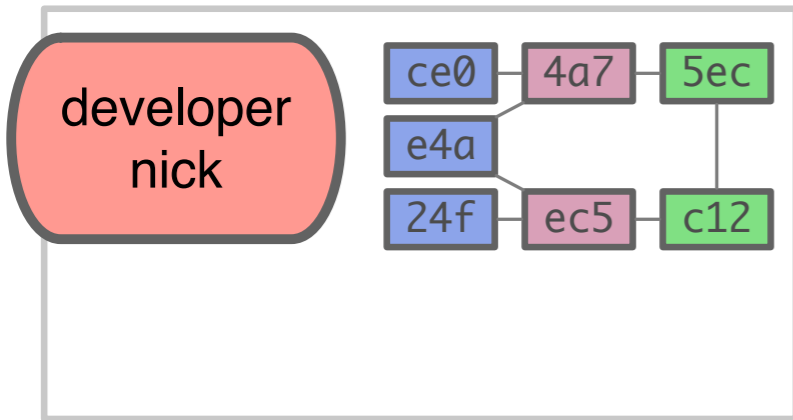


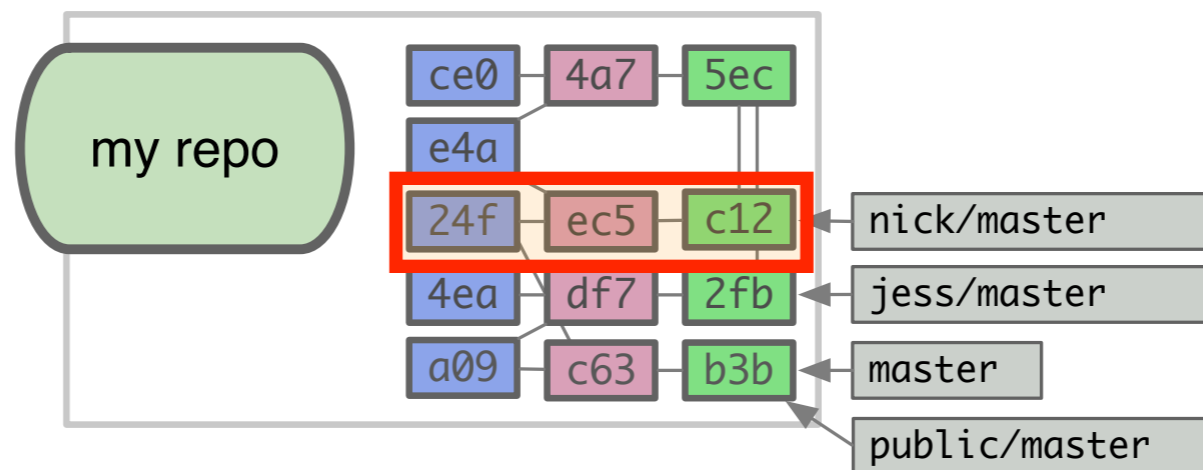
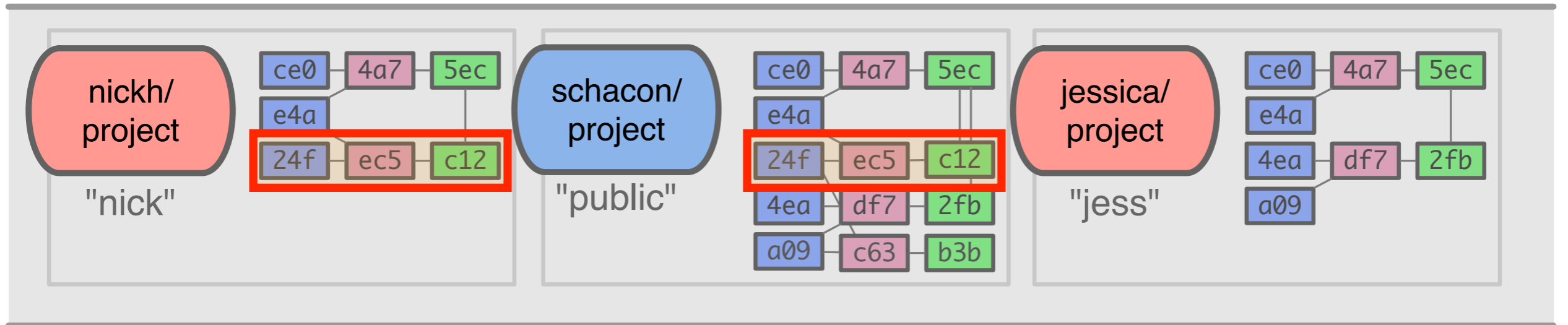
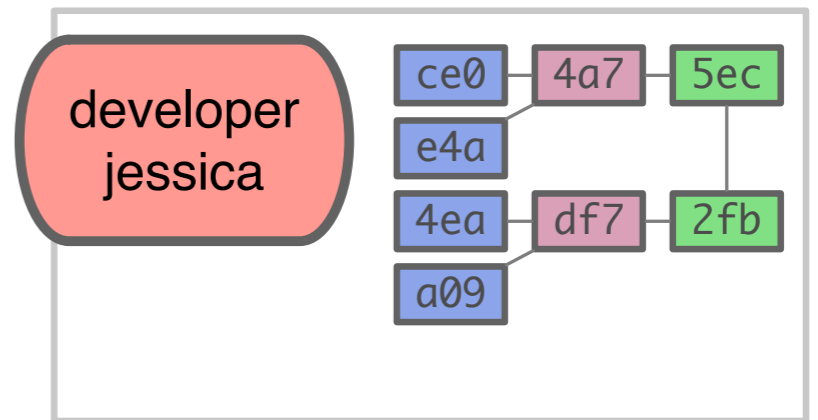
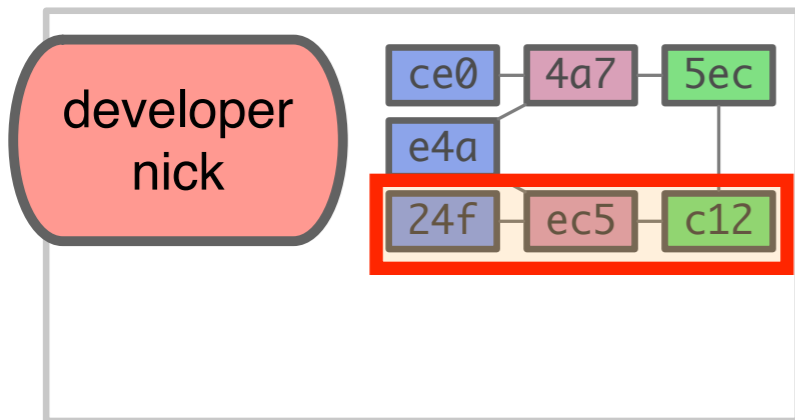


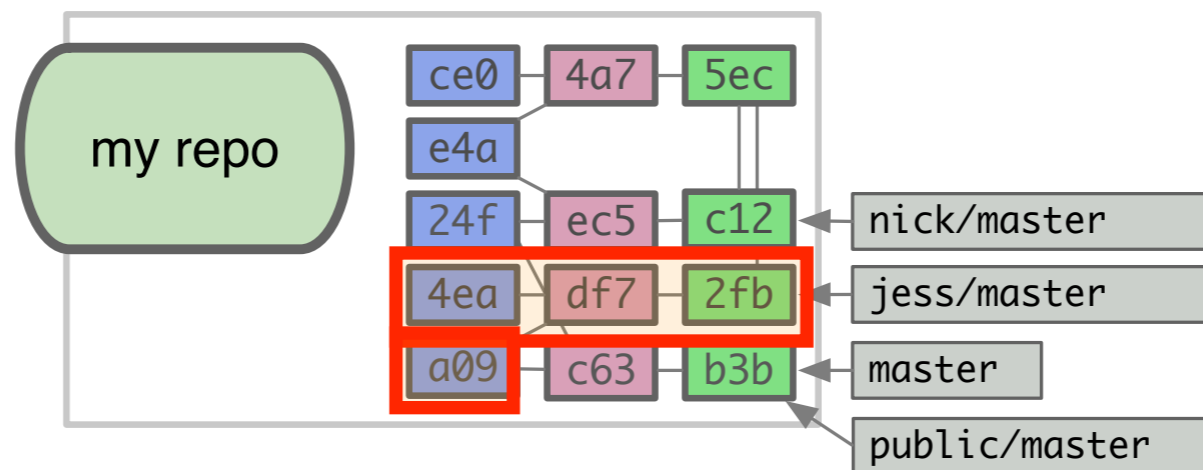
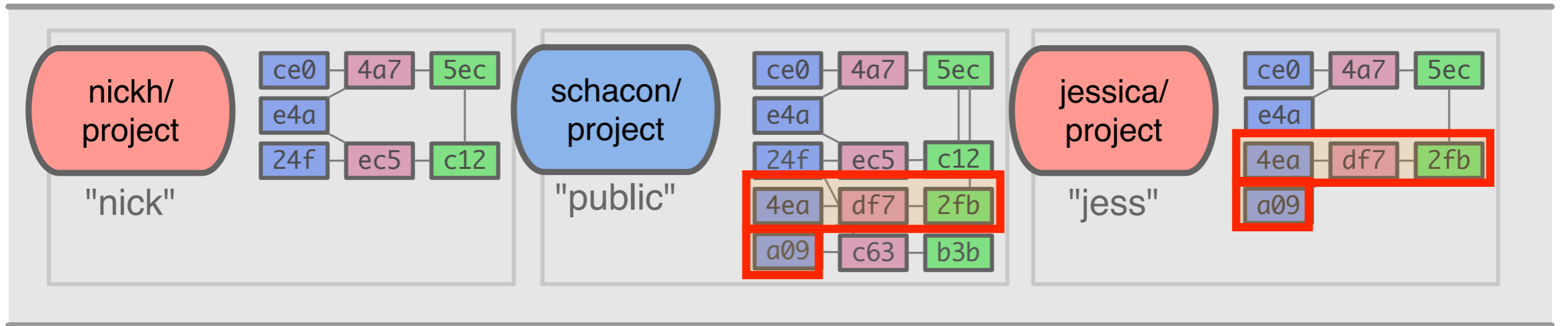
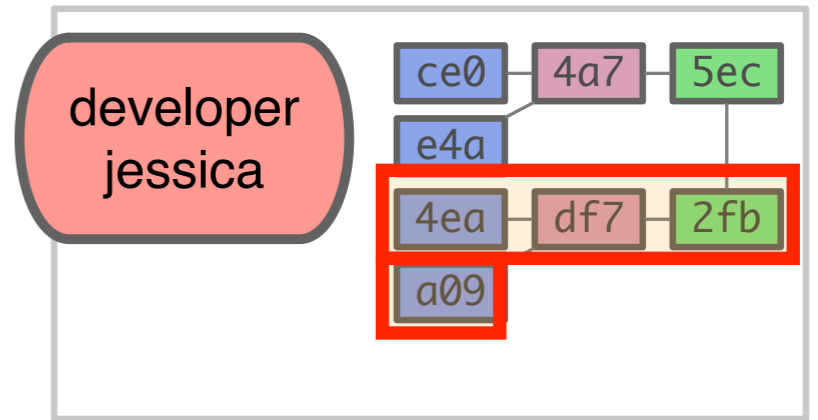
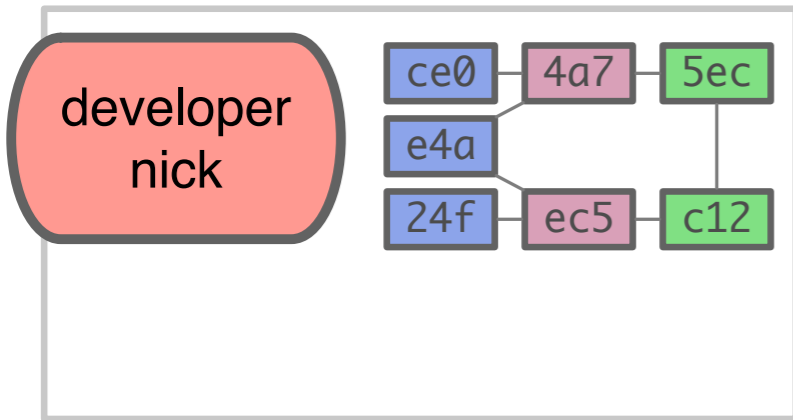
git merge nick jess

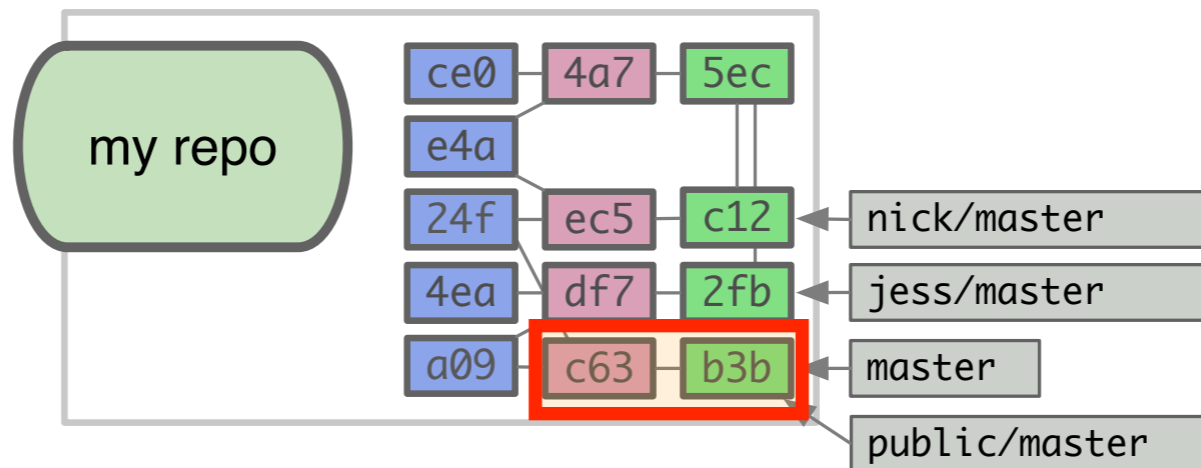
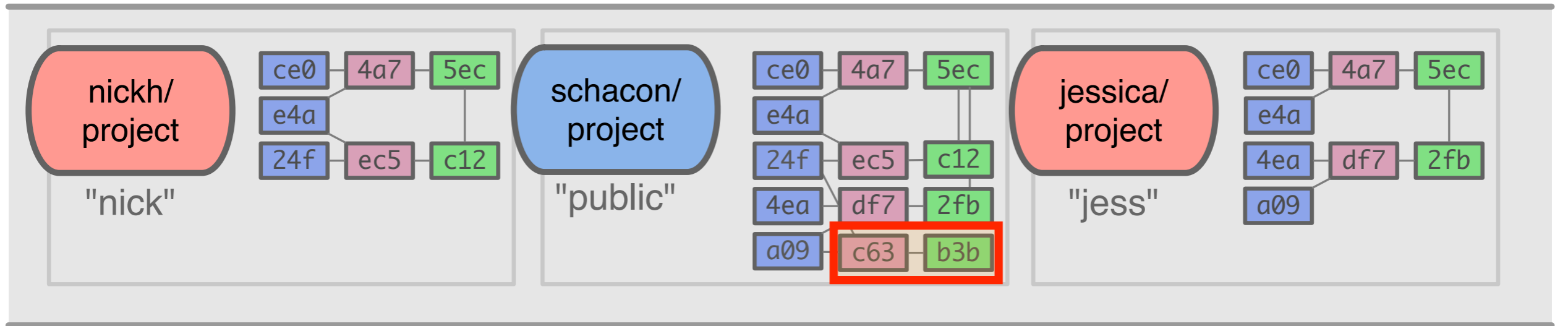
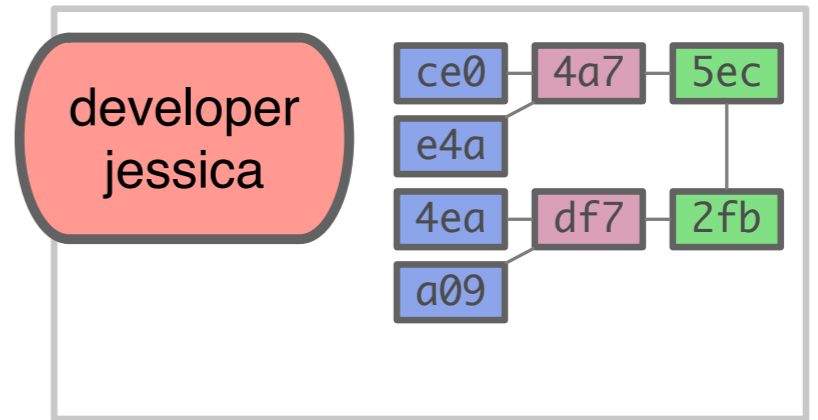
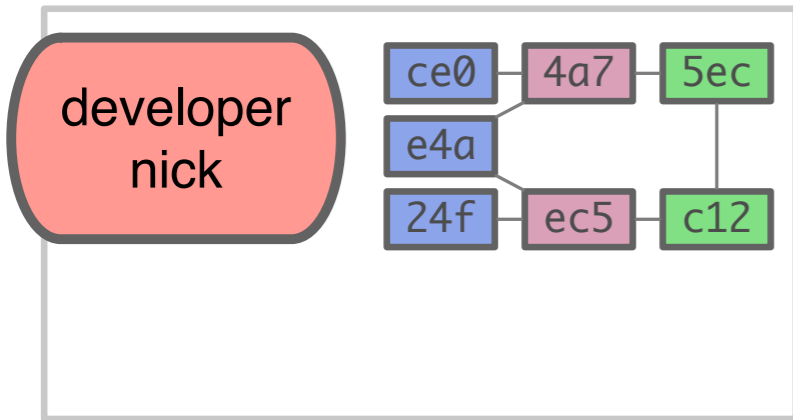


git push public



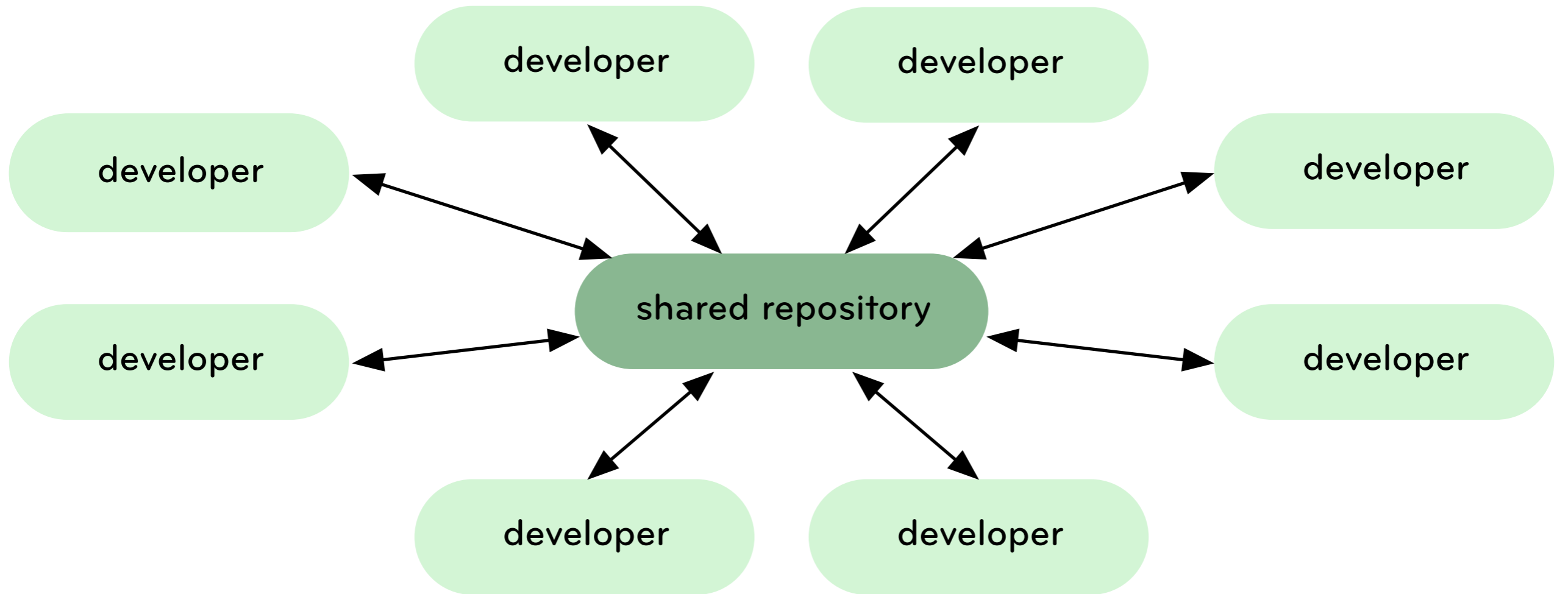


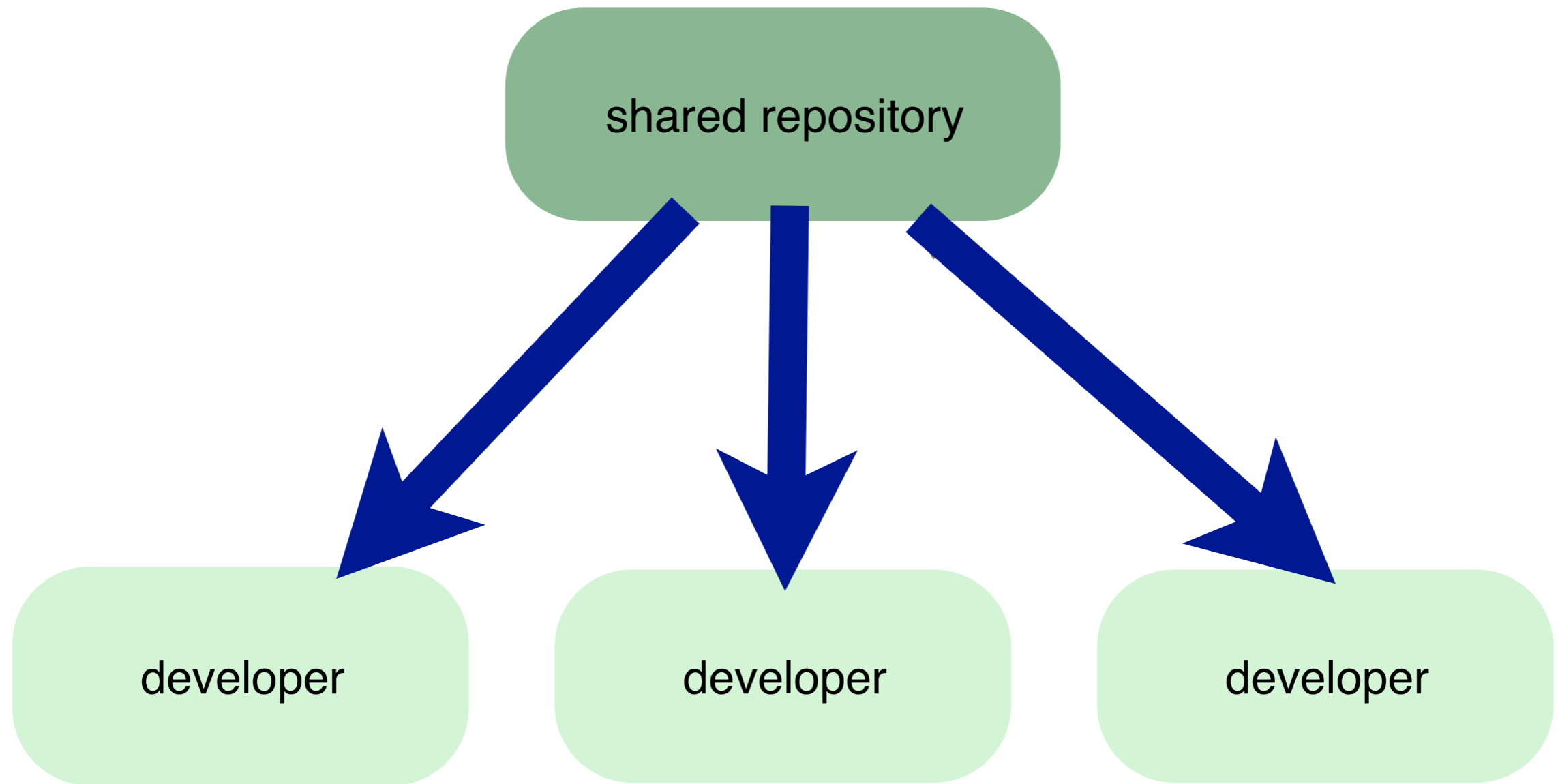




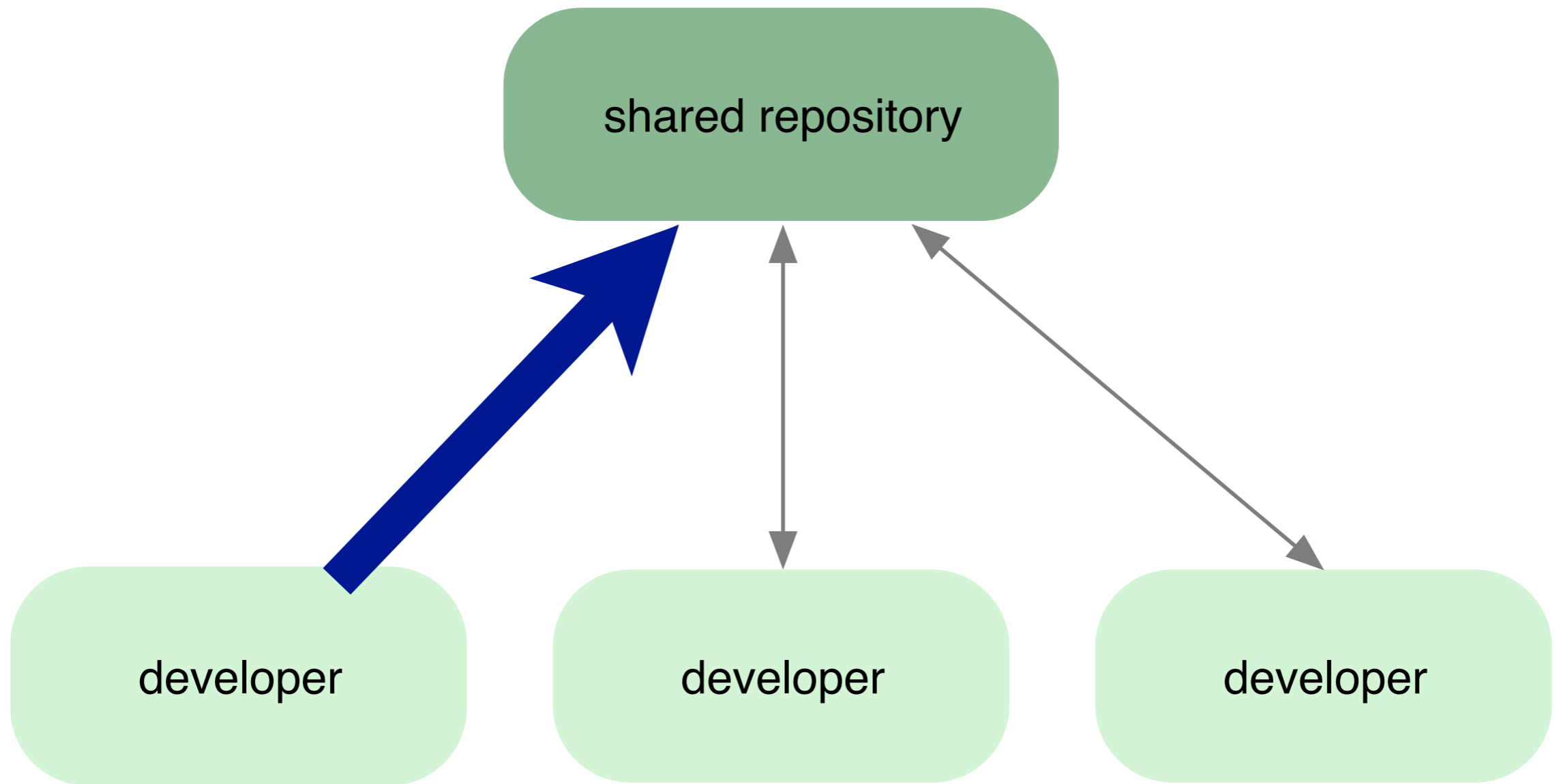
popular workflows

central repository model

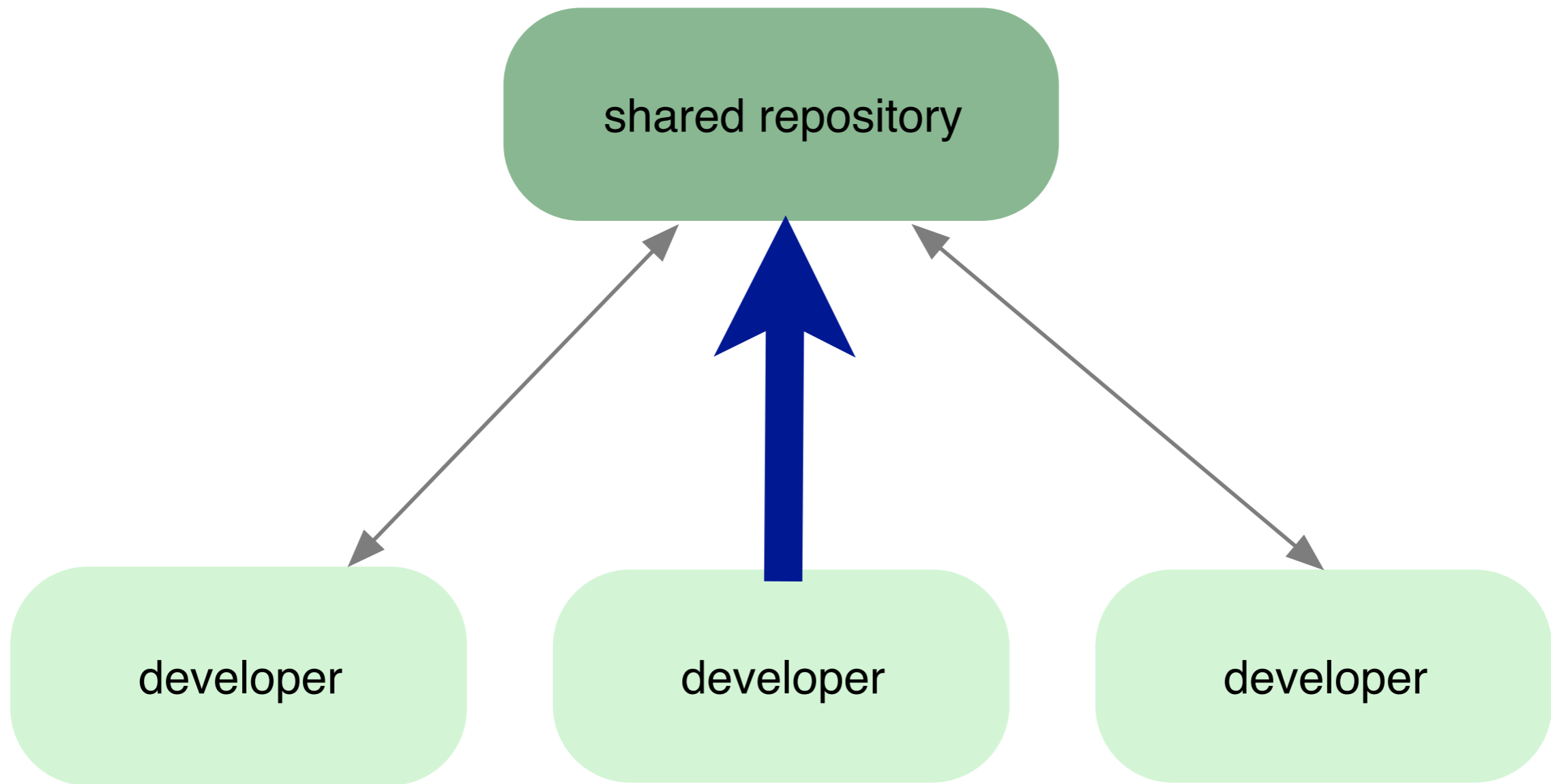




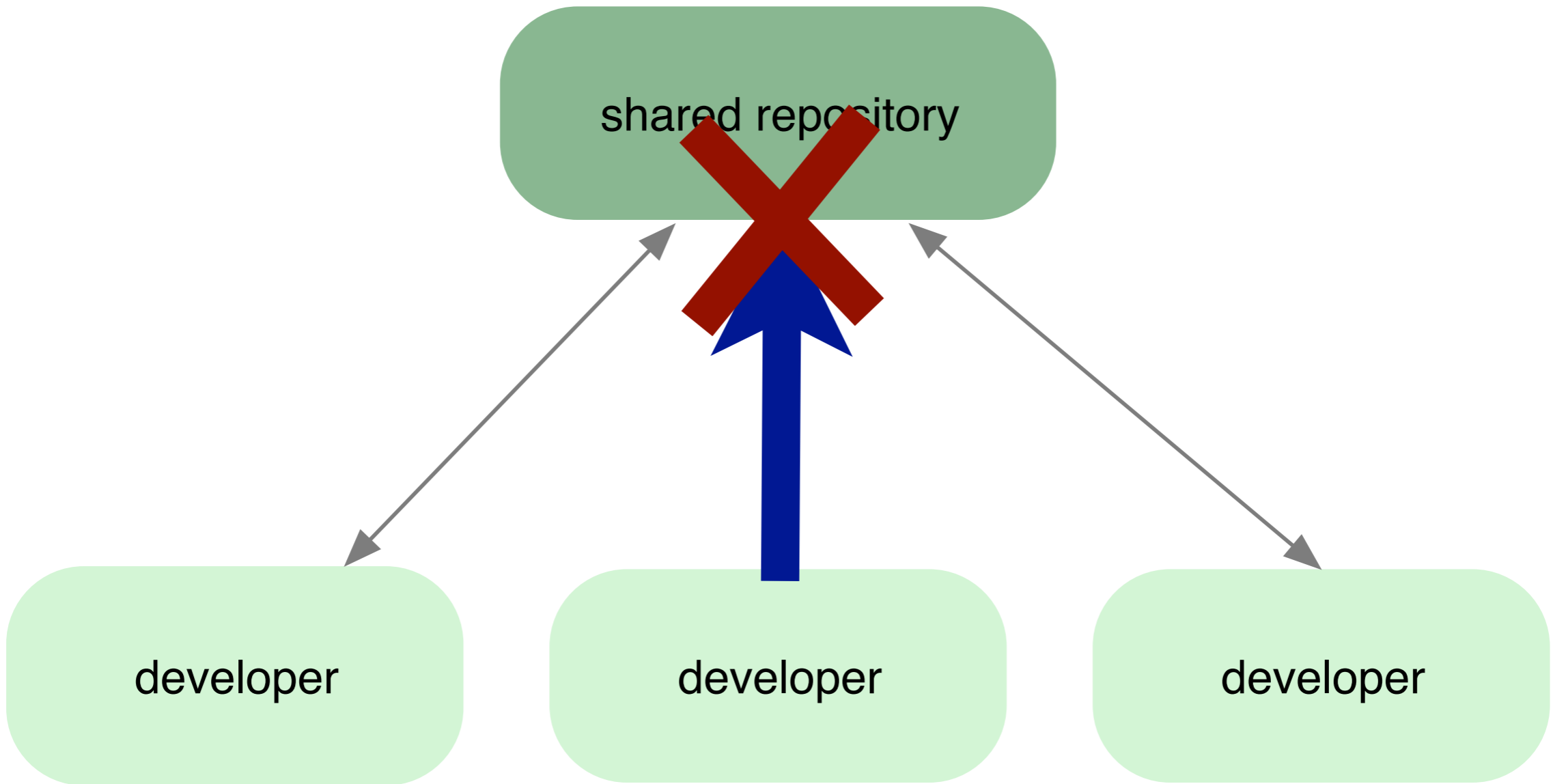
git clone

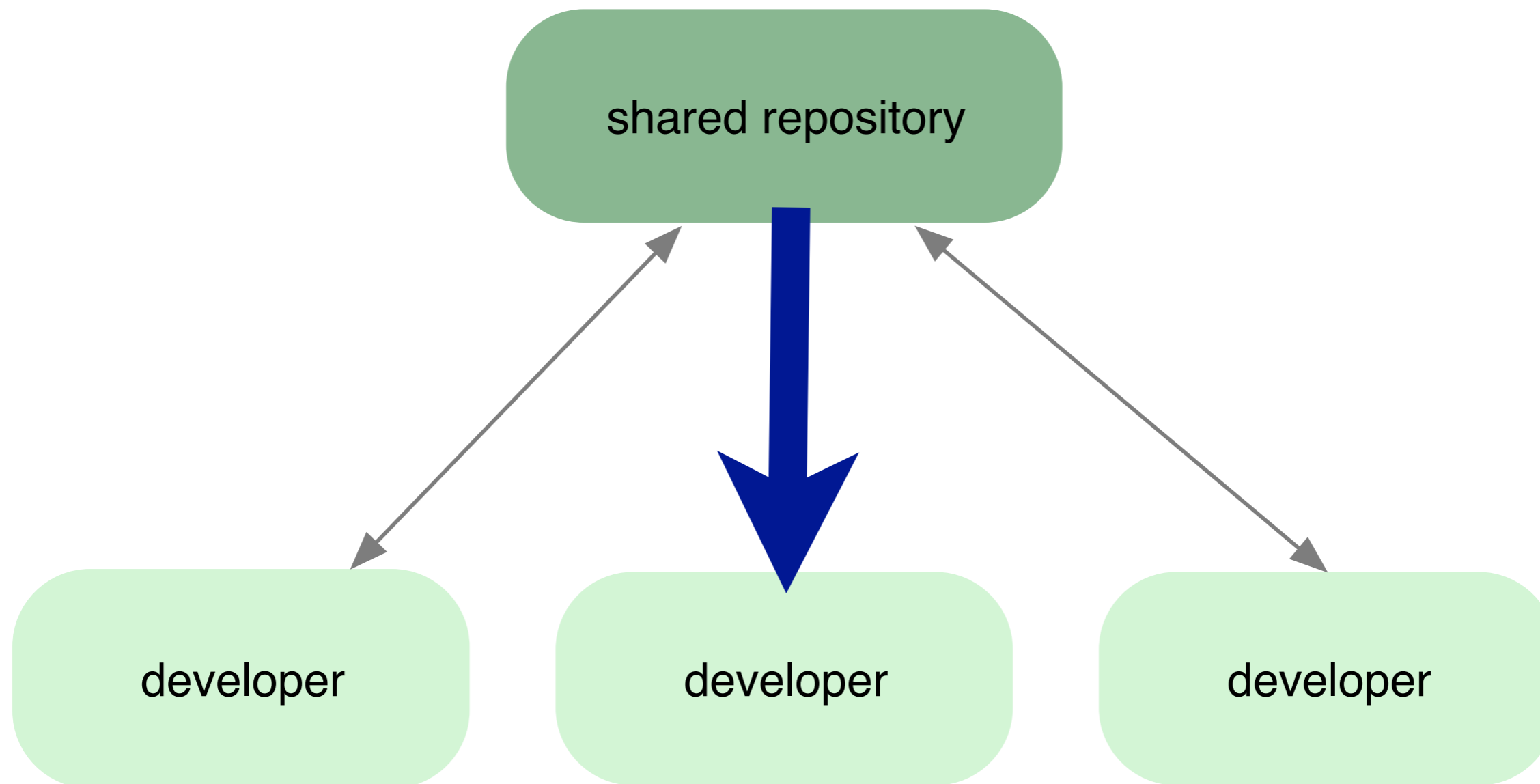


git push

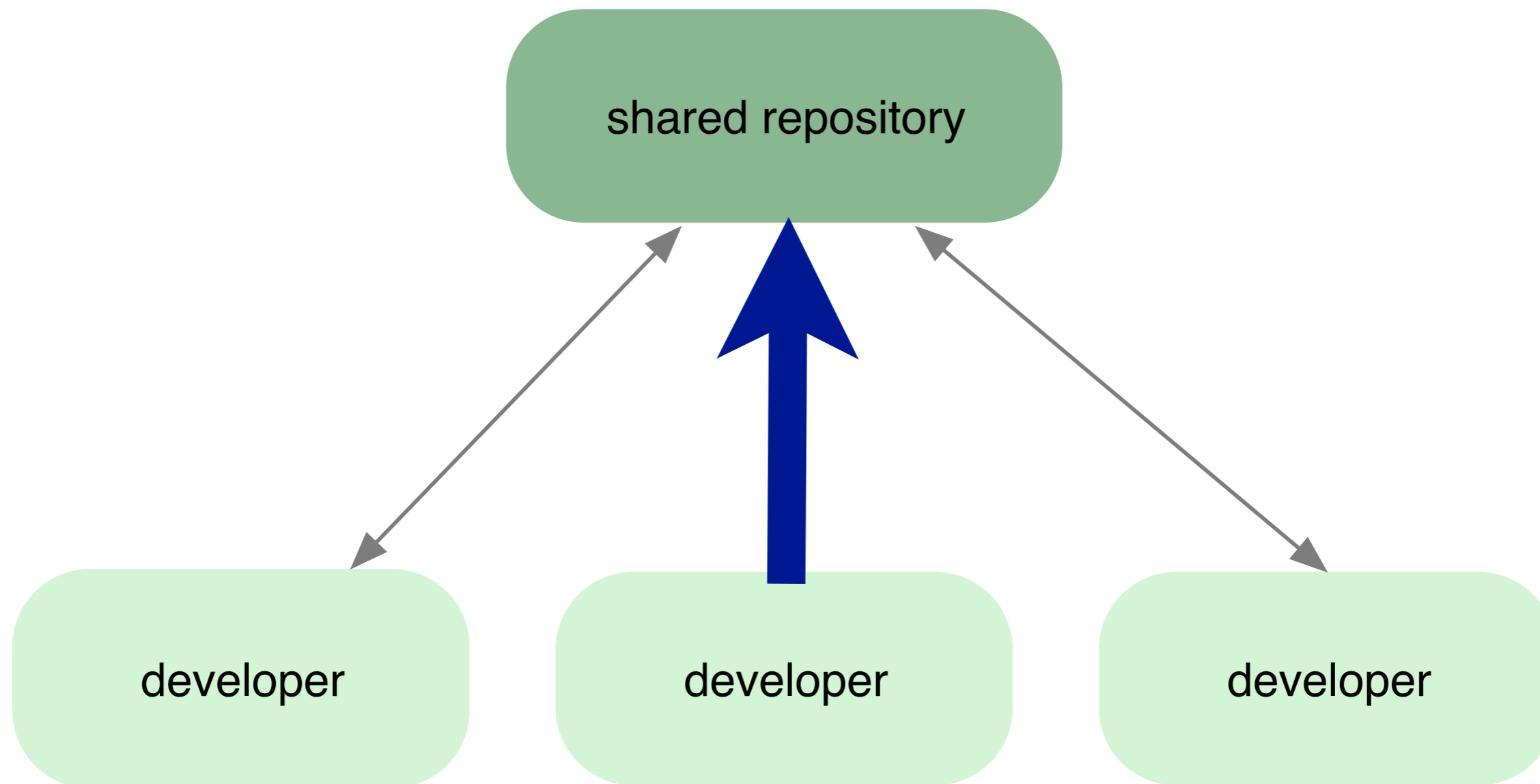


git push



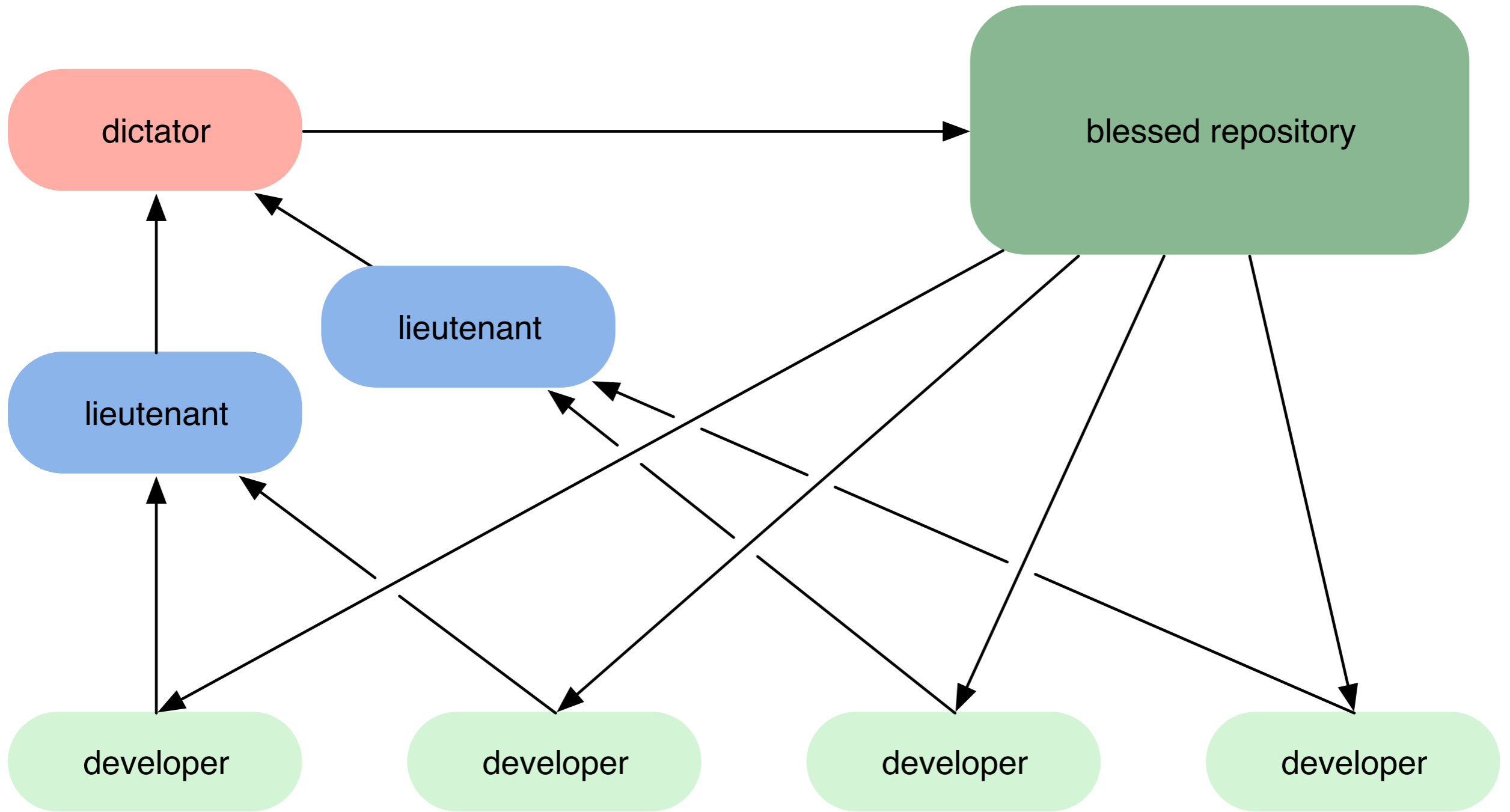


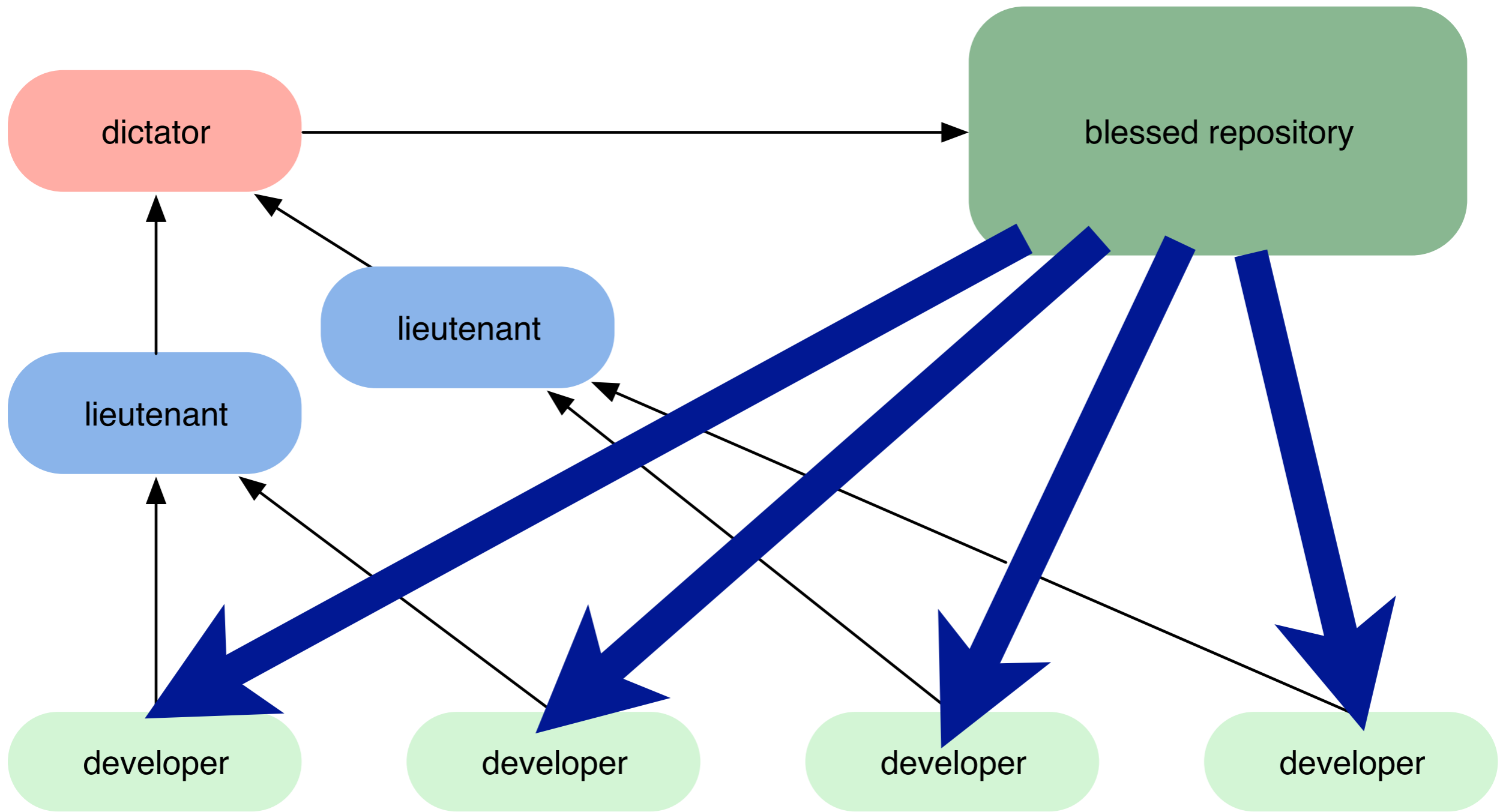
git fetch
git merge



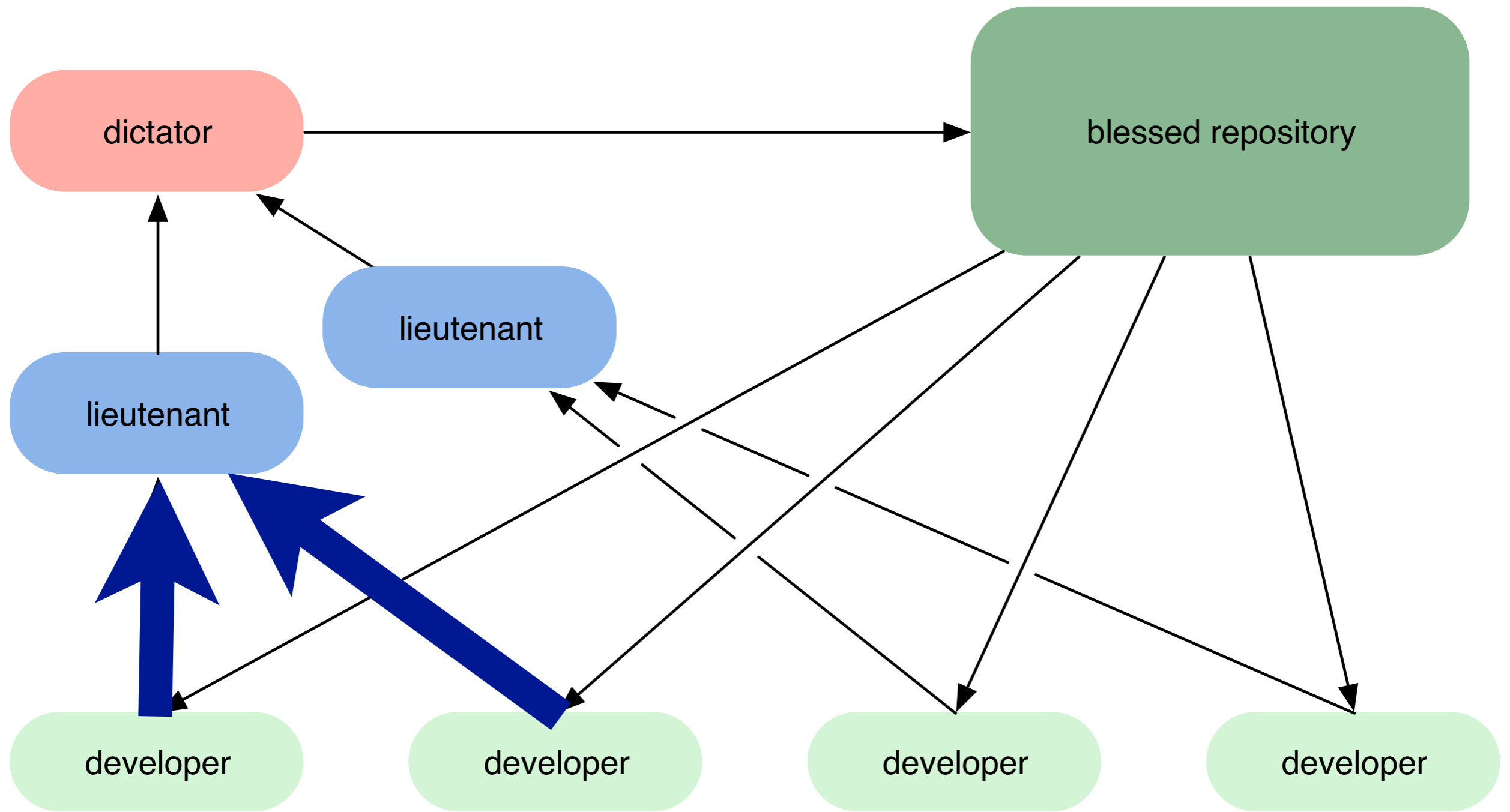
git push

dictator and lieutenants model

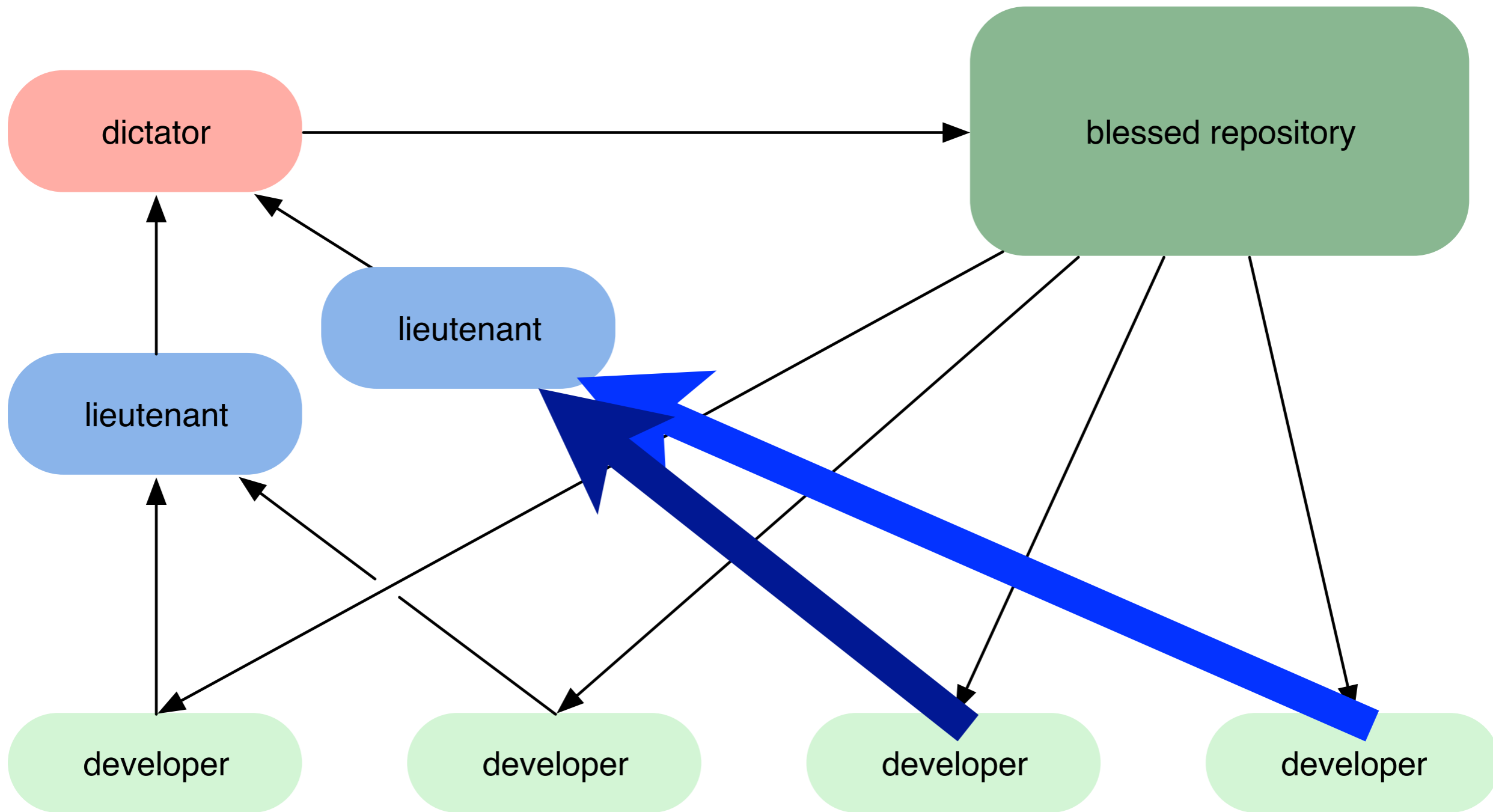




git clone

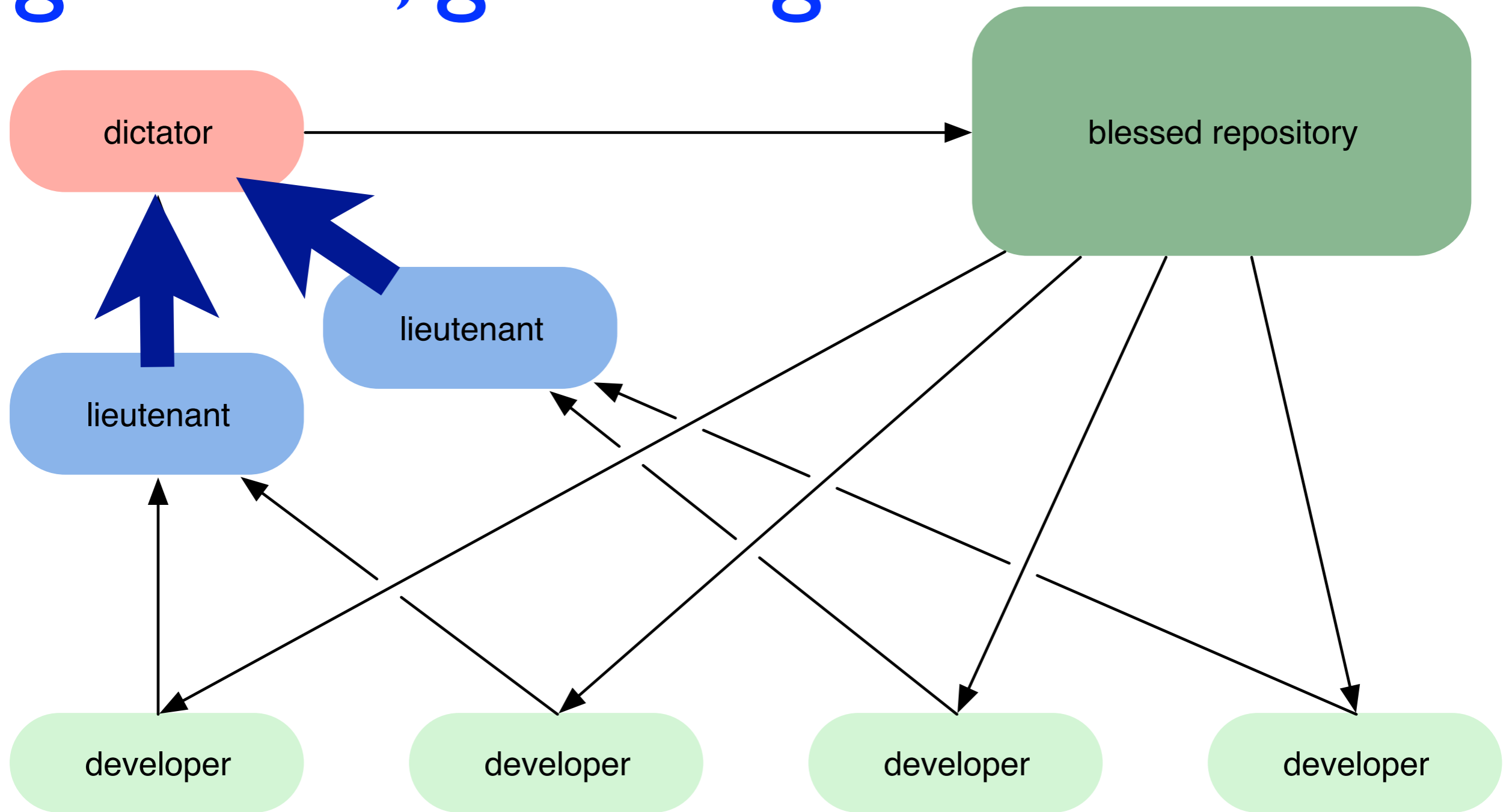


git fetch
git merge

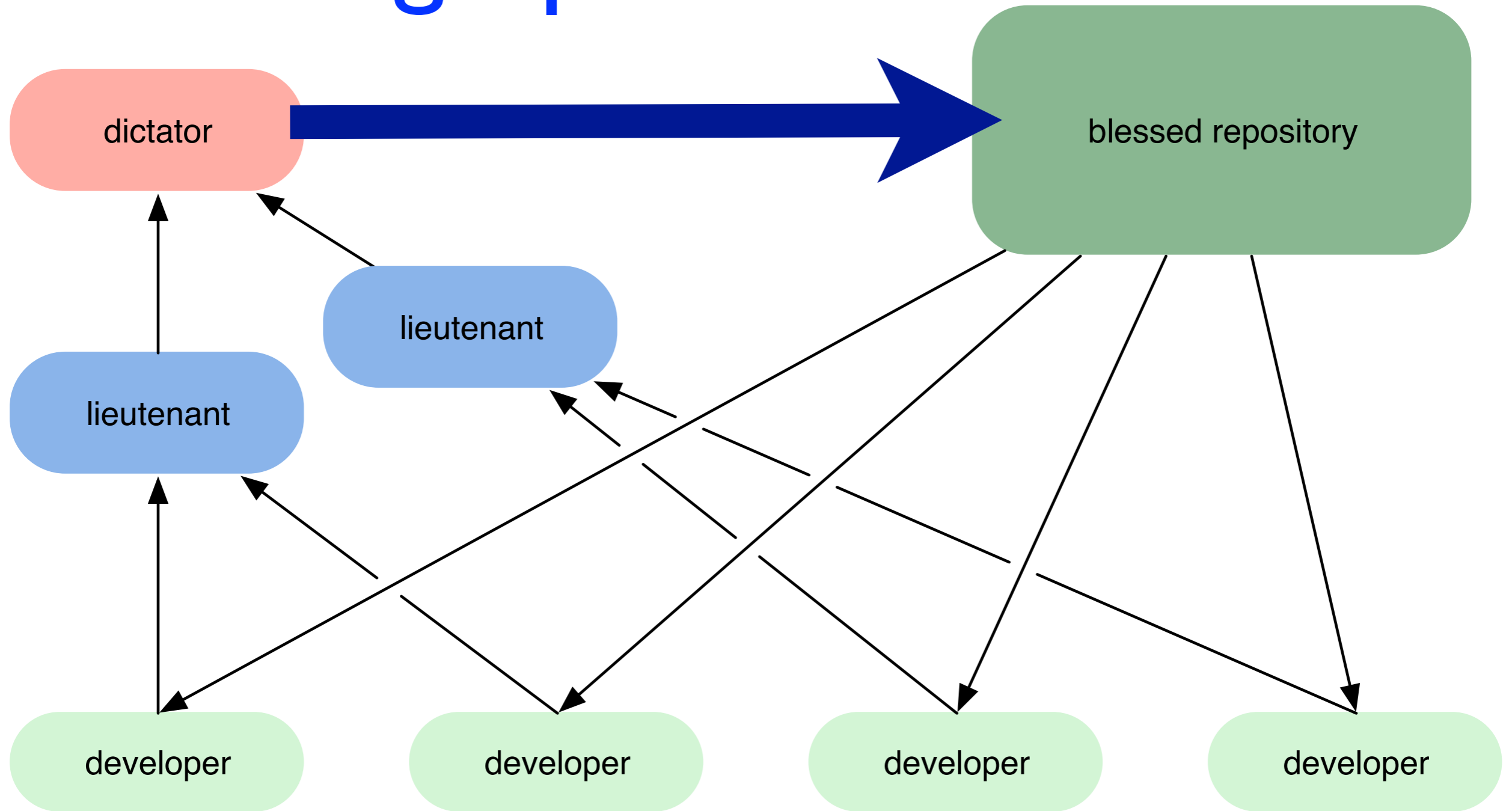


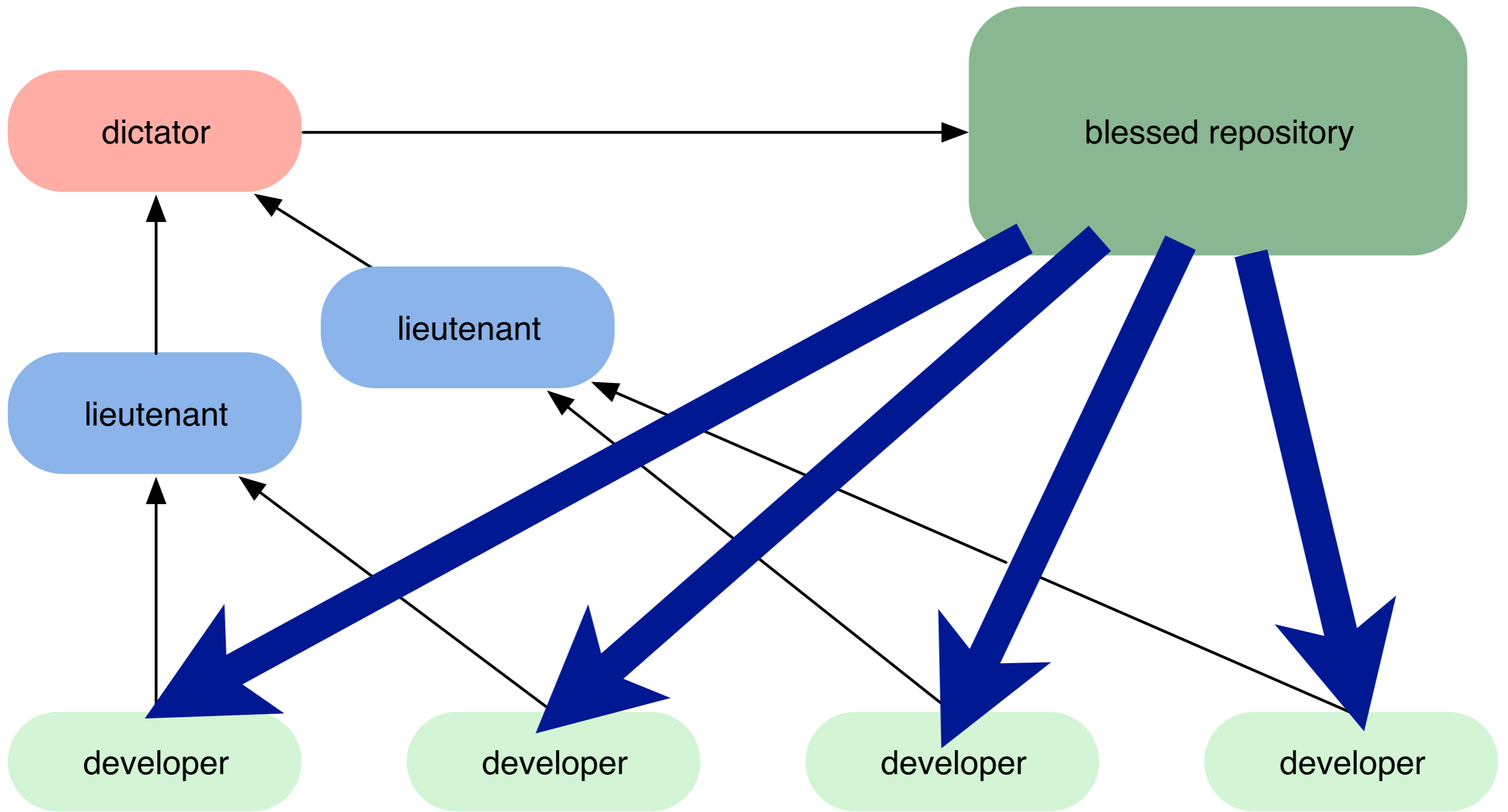
git fetch
git merge

git fetch; git merge



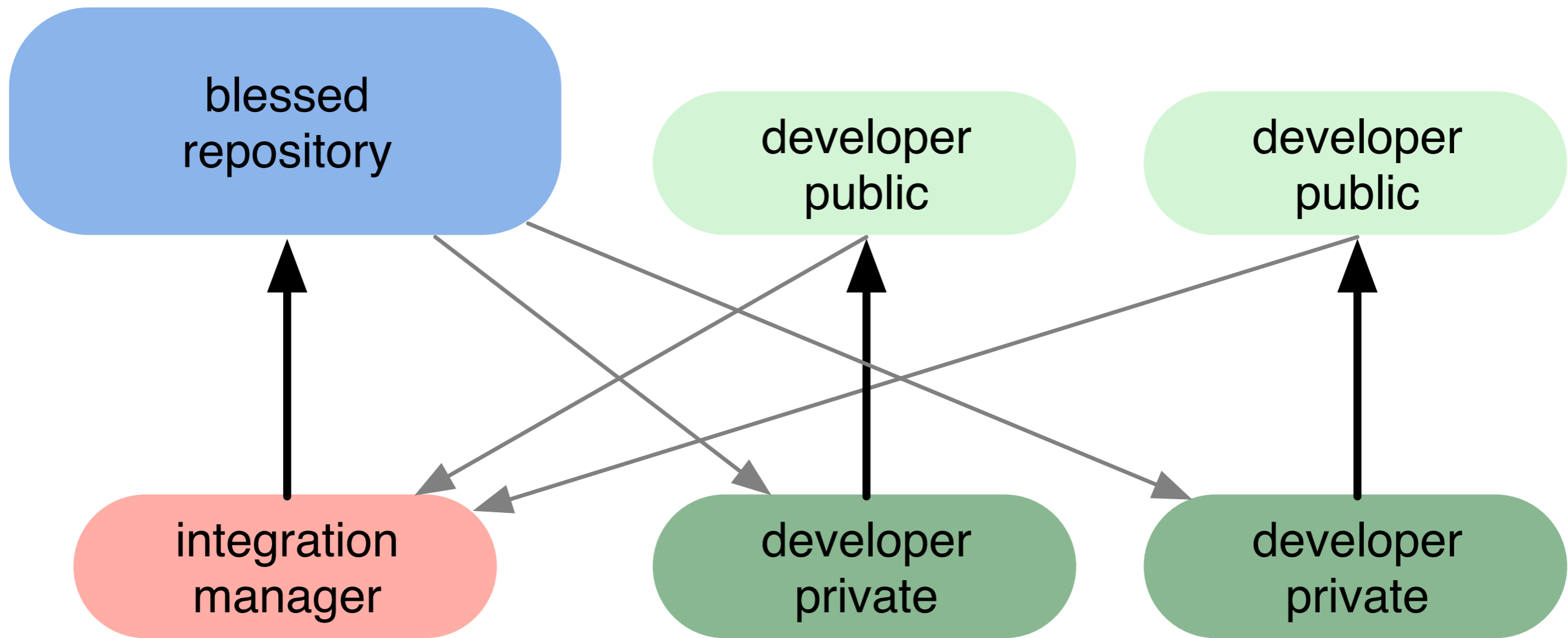
git push





git fetch

integration manager model



github.com

/schacon/ticgit

blessed
repository

/yob/ticgit

developer
public

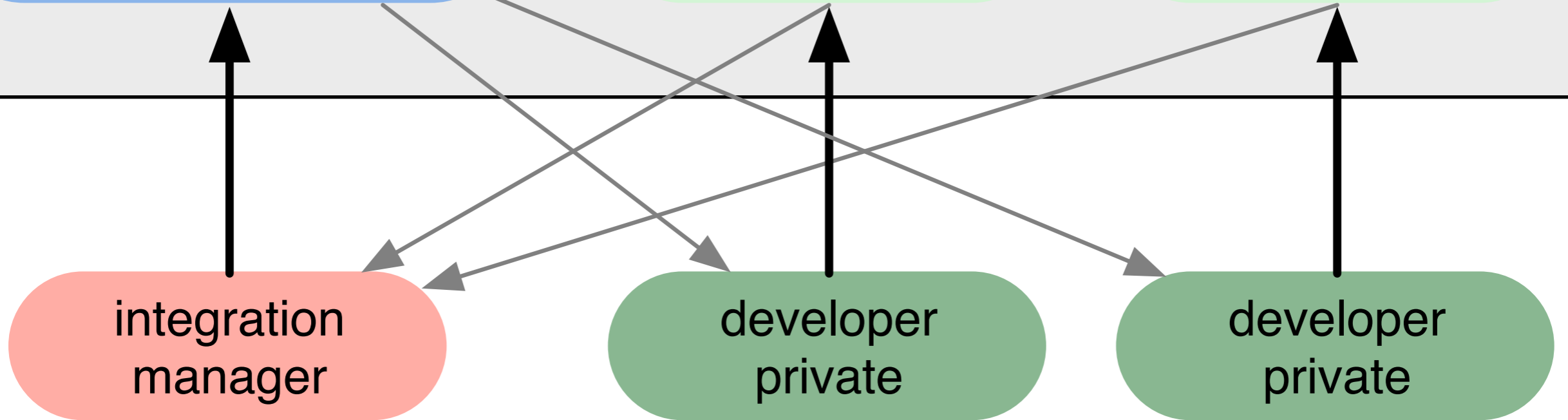
/pope/ticgit

developer
public

integration
manager

developer
private

developer
private



github.com

/schacon/ticgit

blessed
repository

“forks”

/yob/ticgit

developer
public

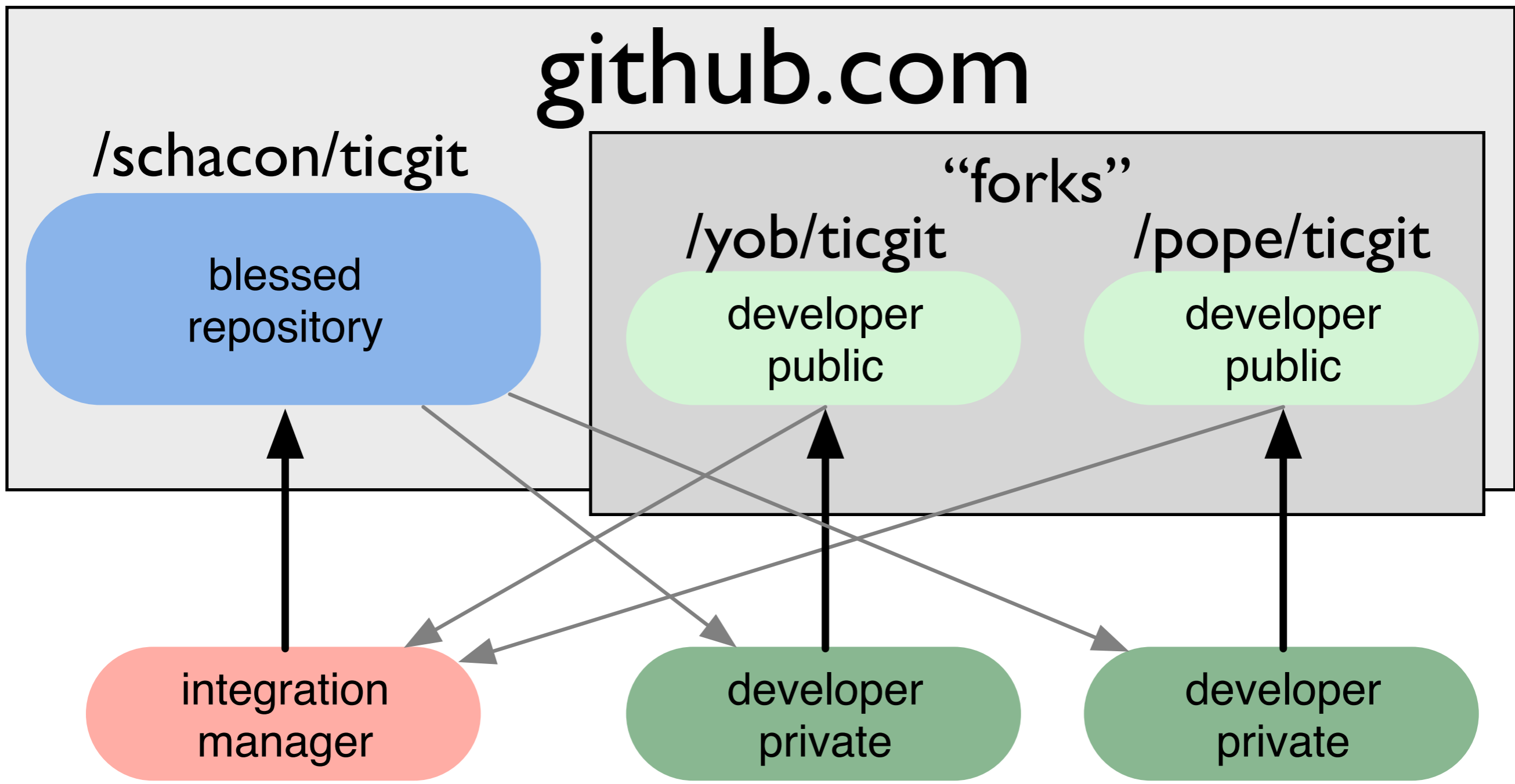
/pope/ticgit

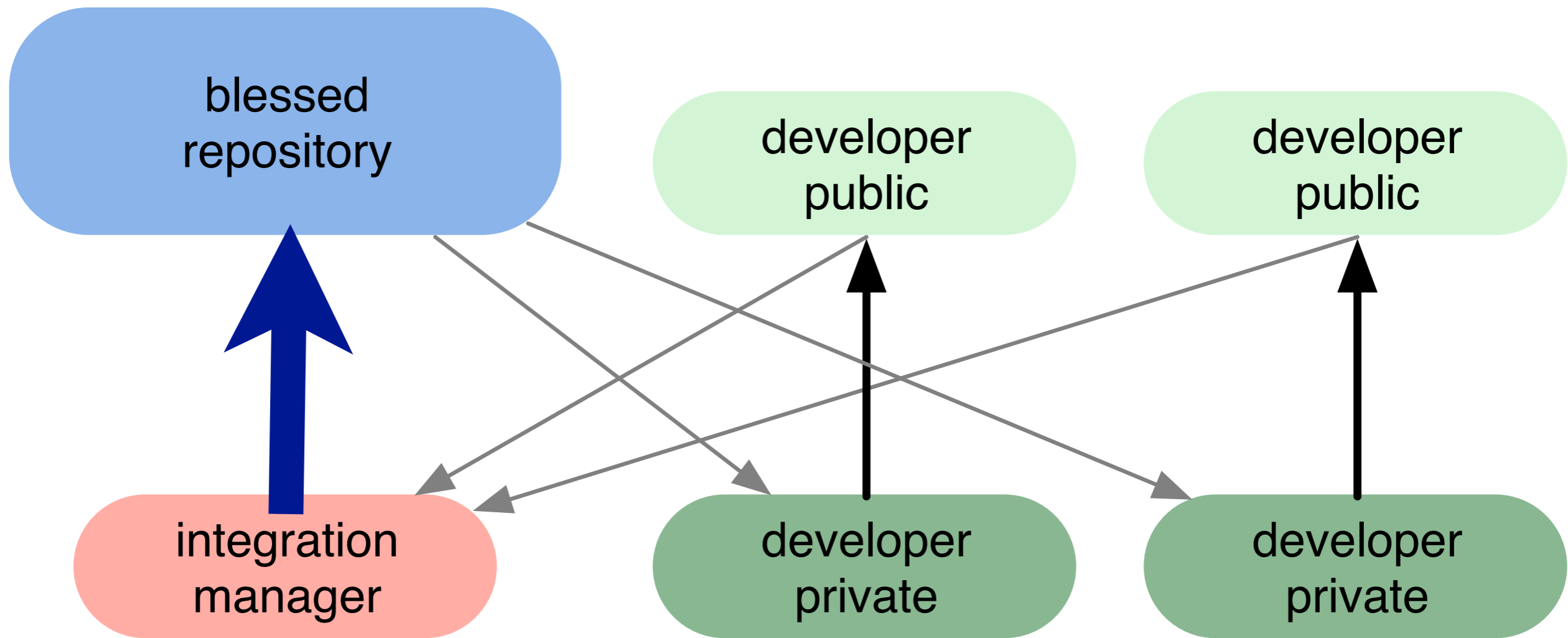
developer
public

integration
manager

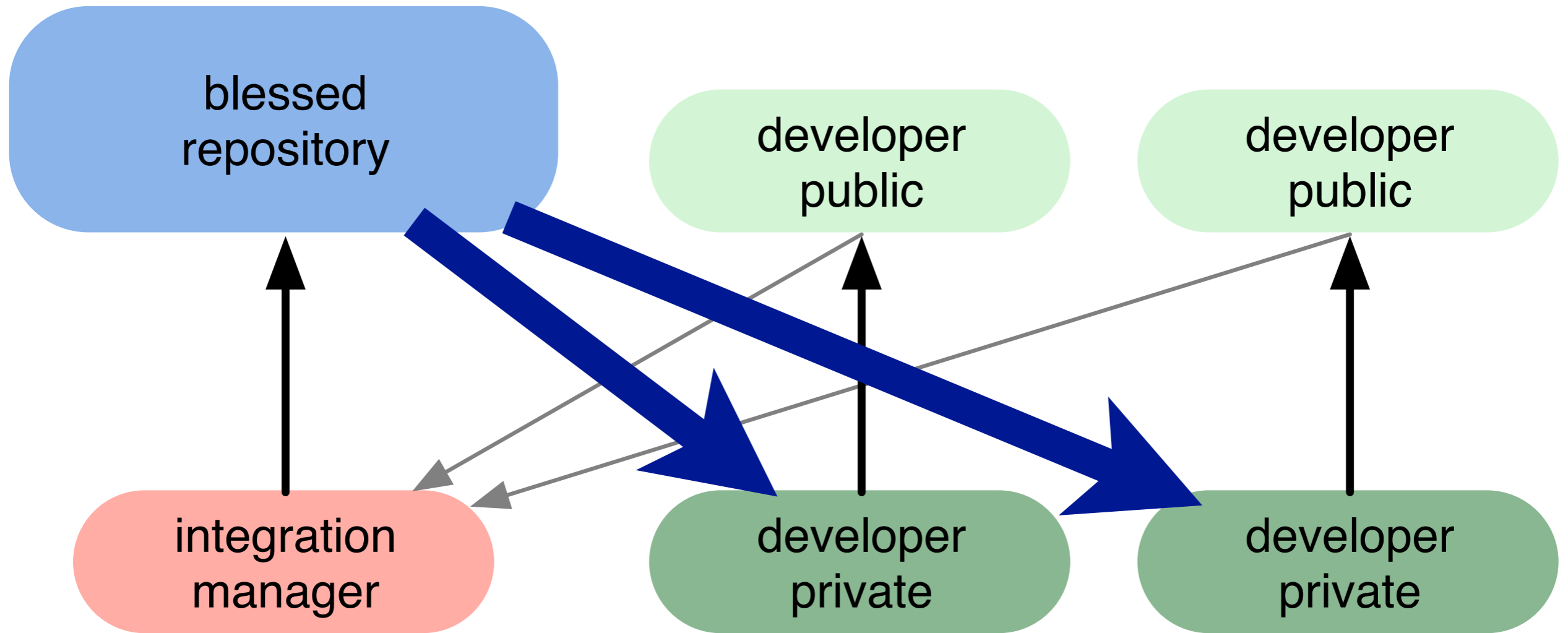
developer
private

developer
private

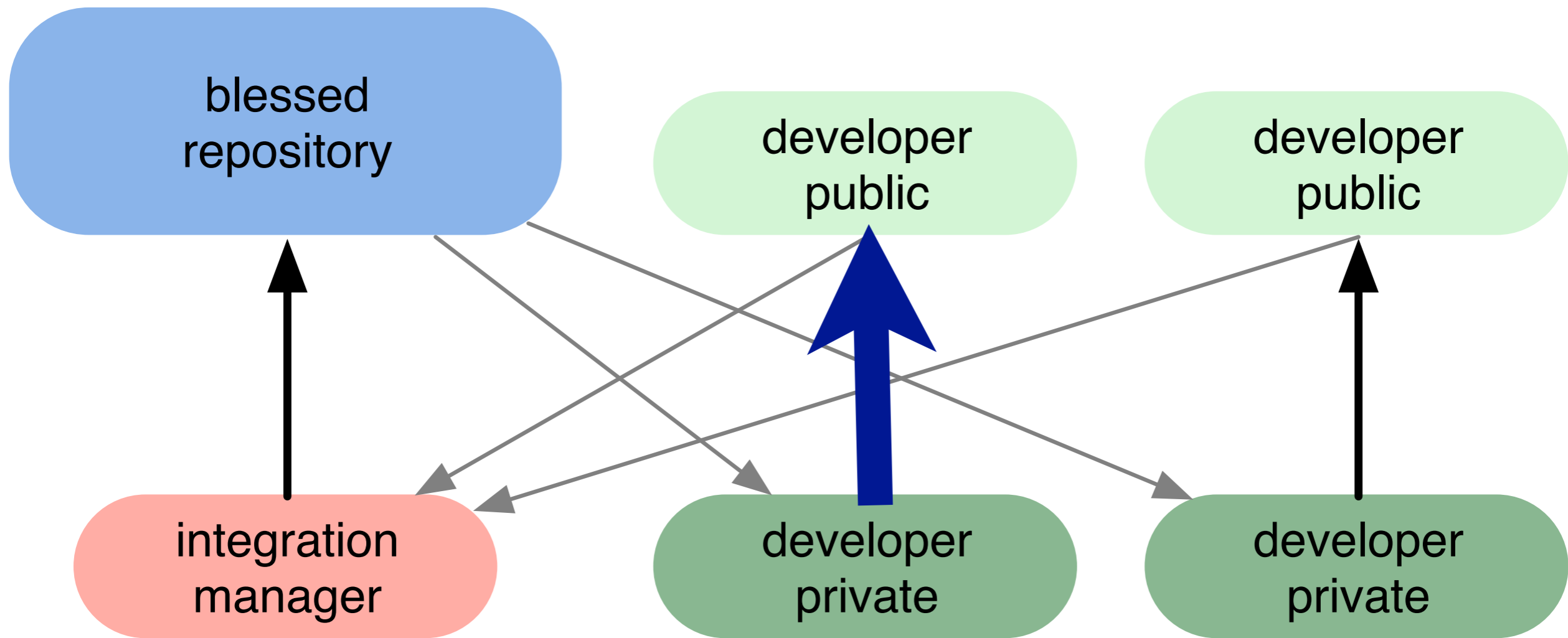




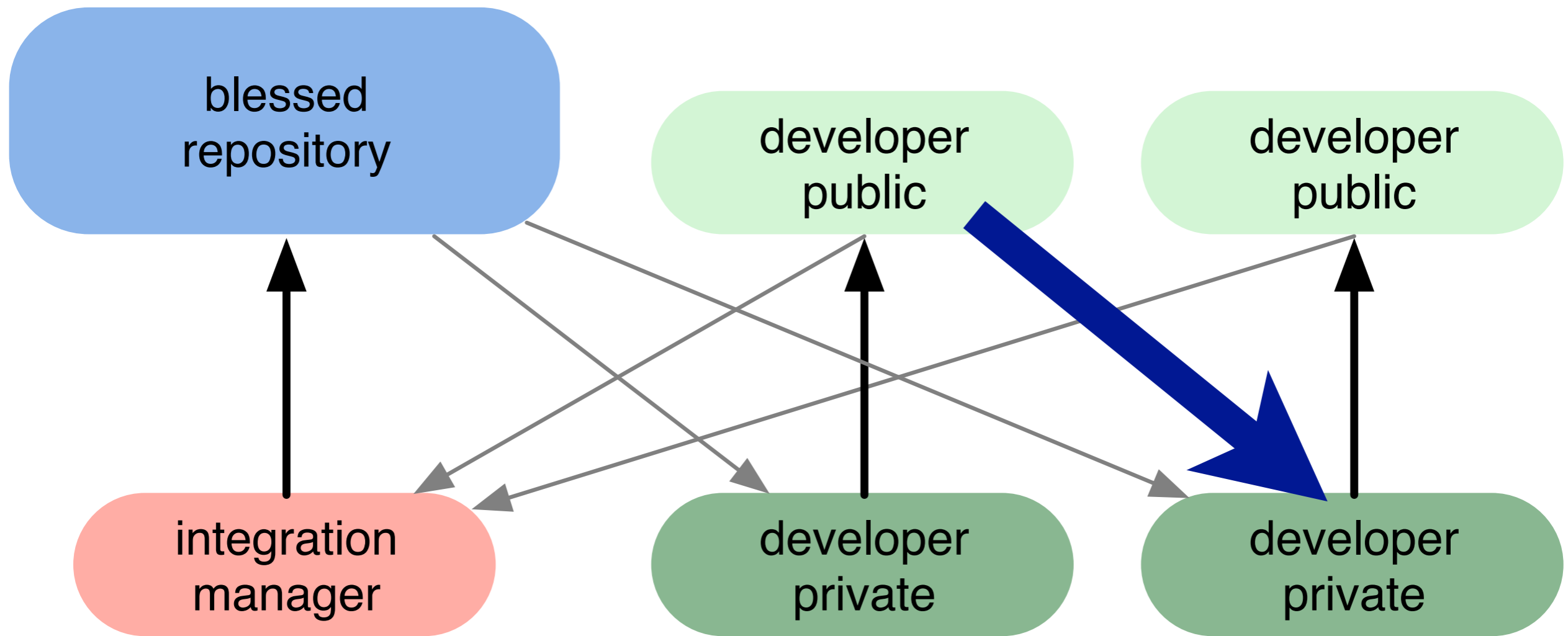
git push



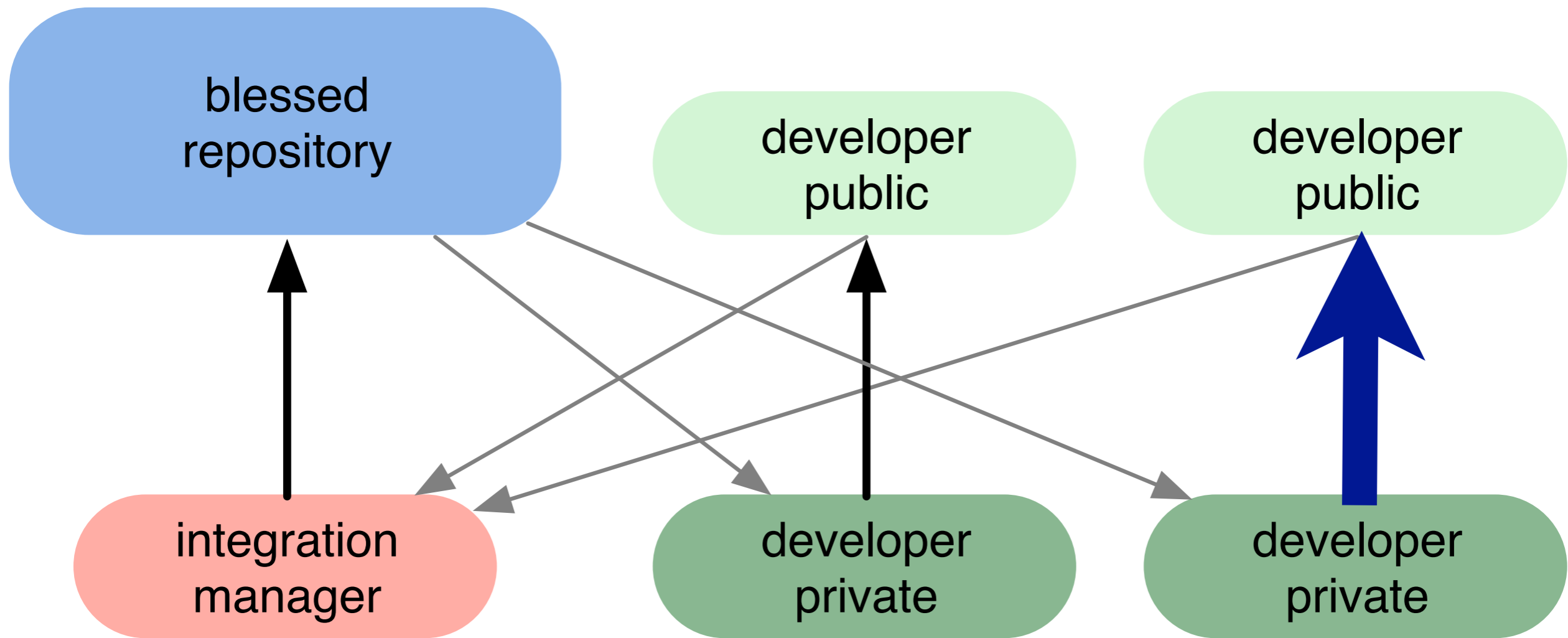
git clone



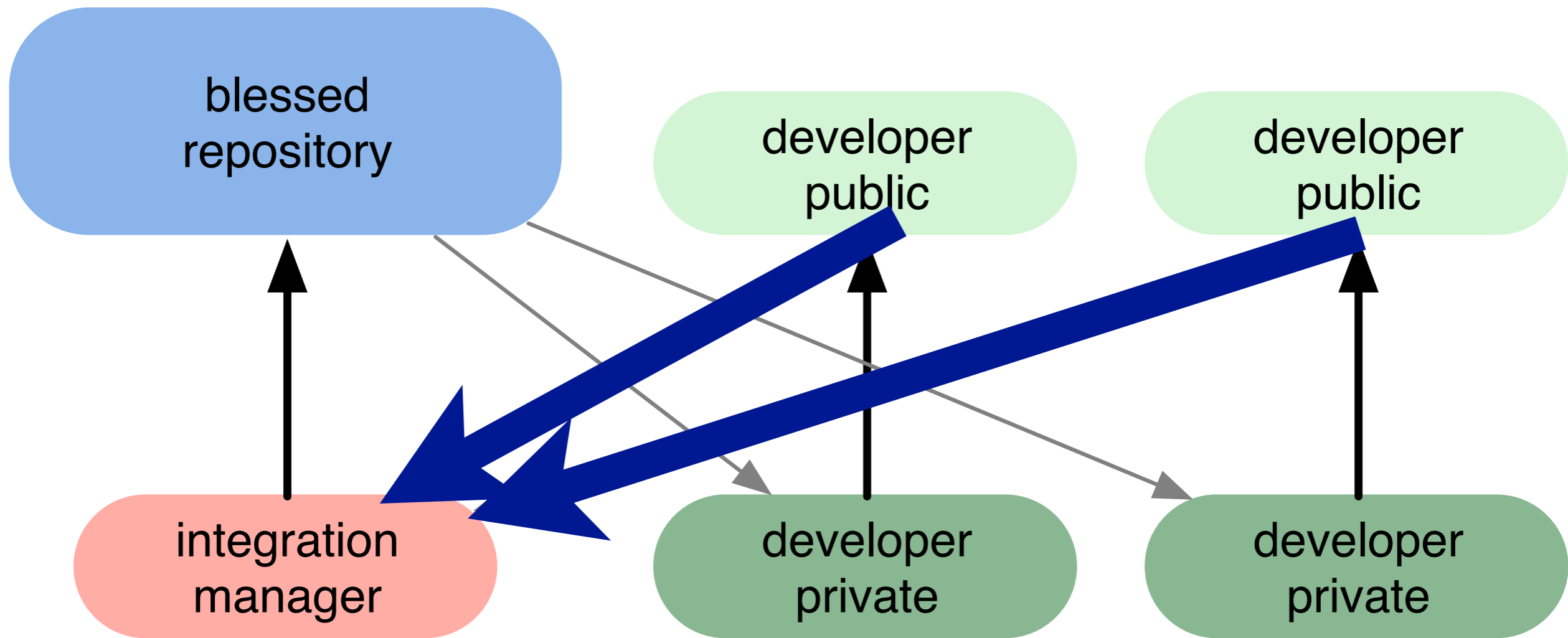
git push



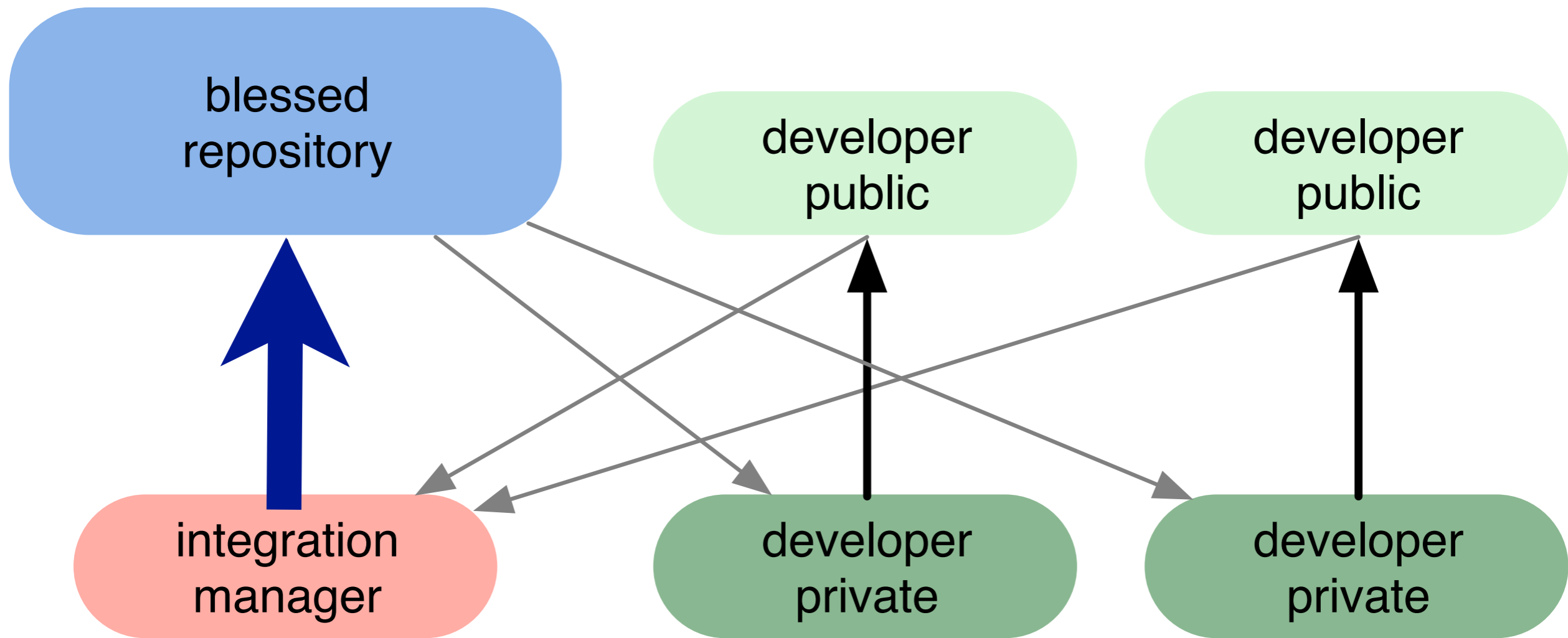
git fetch
git merge



git push



git fetch
git merge

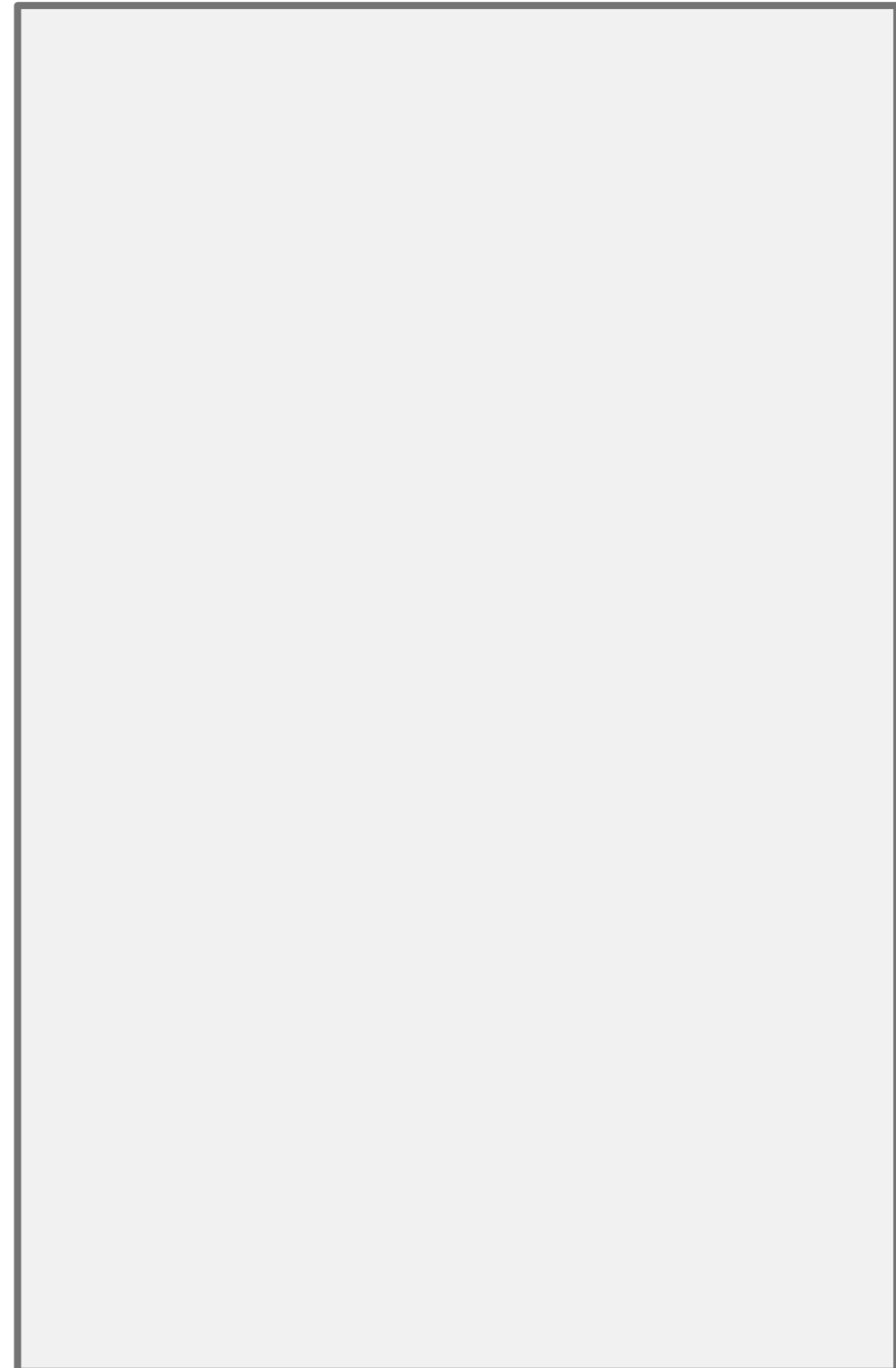
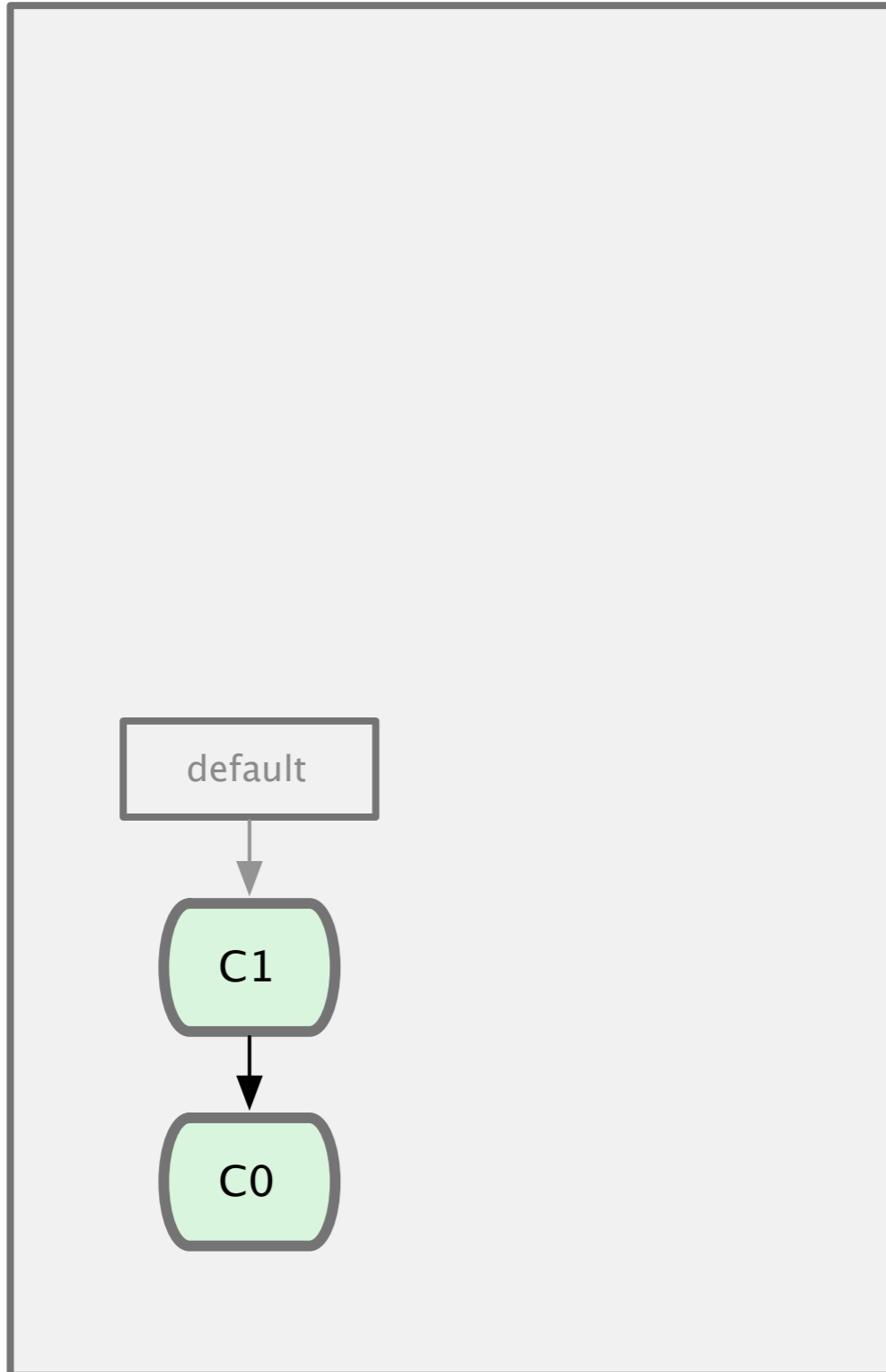


git push

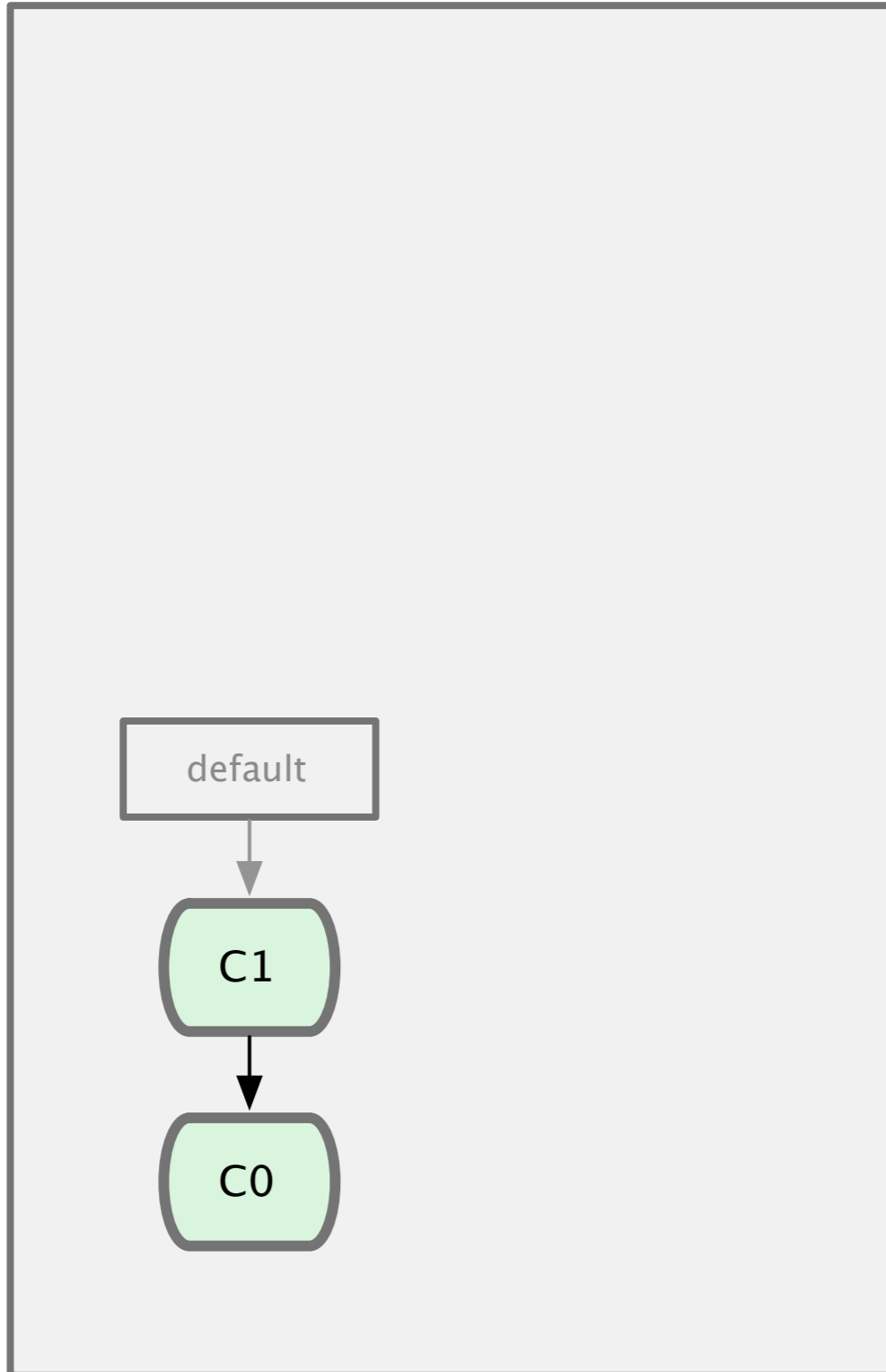
Remotes Are Branches

scott

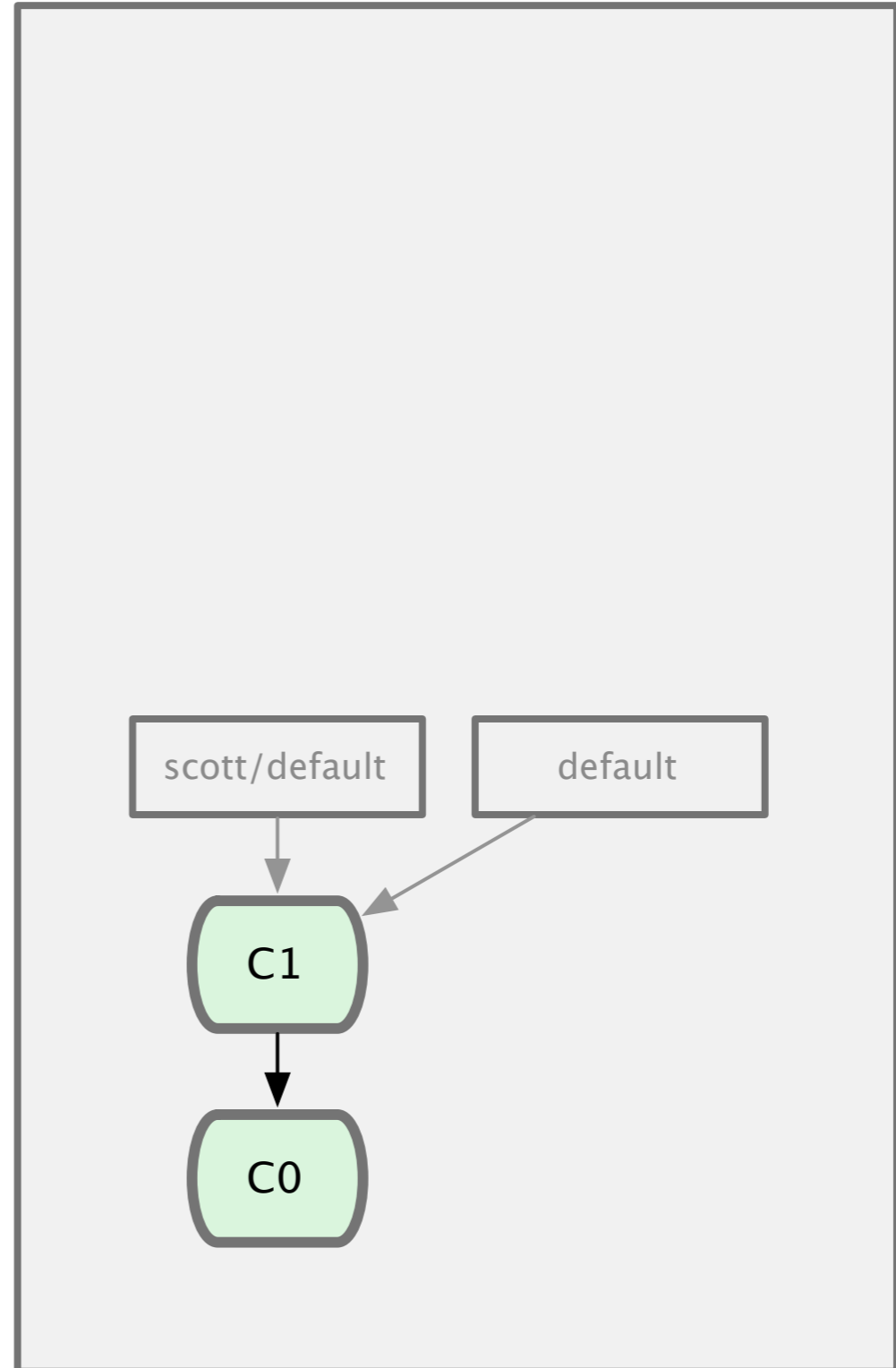
jessica



scott

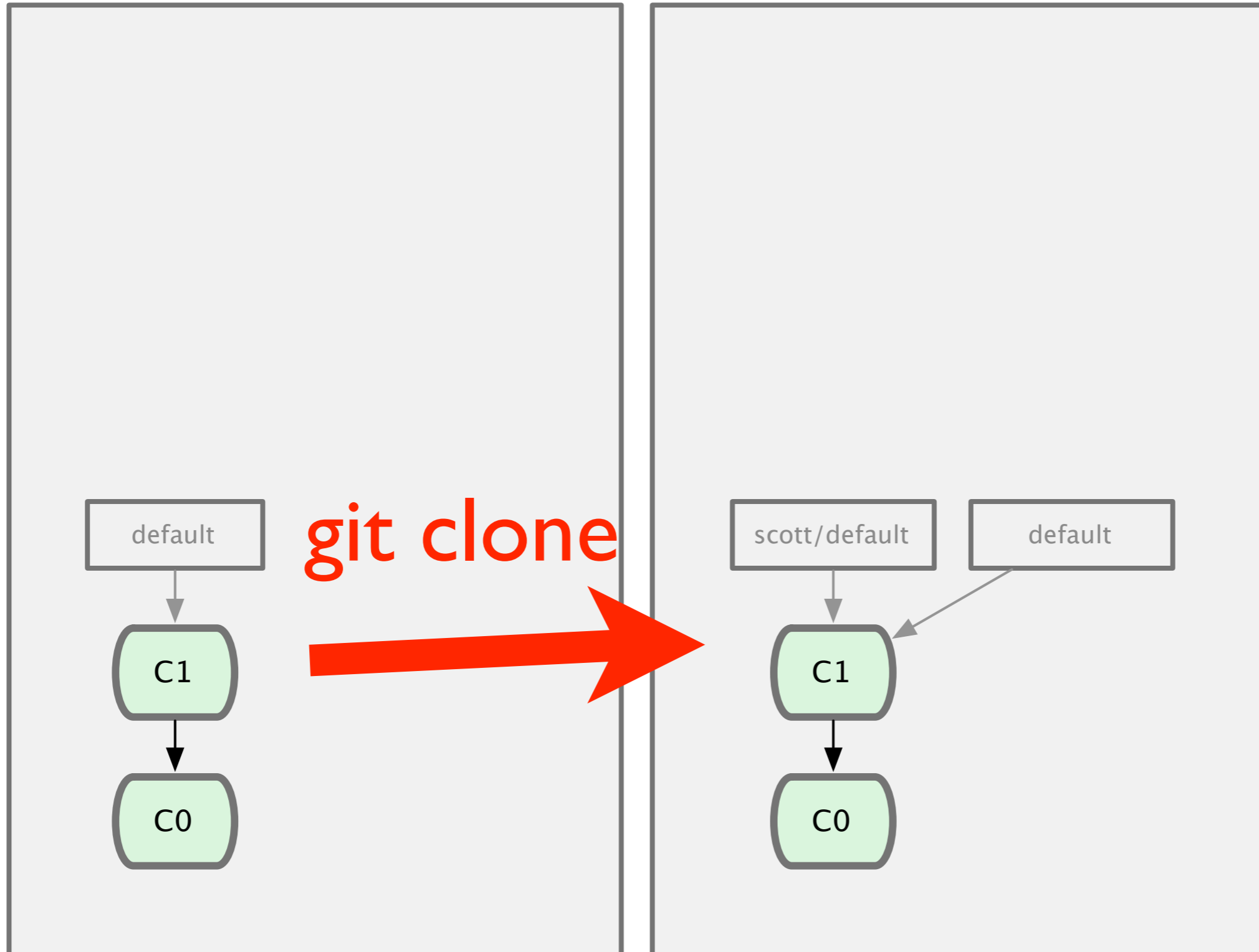


jessica

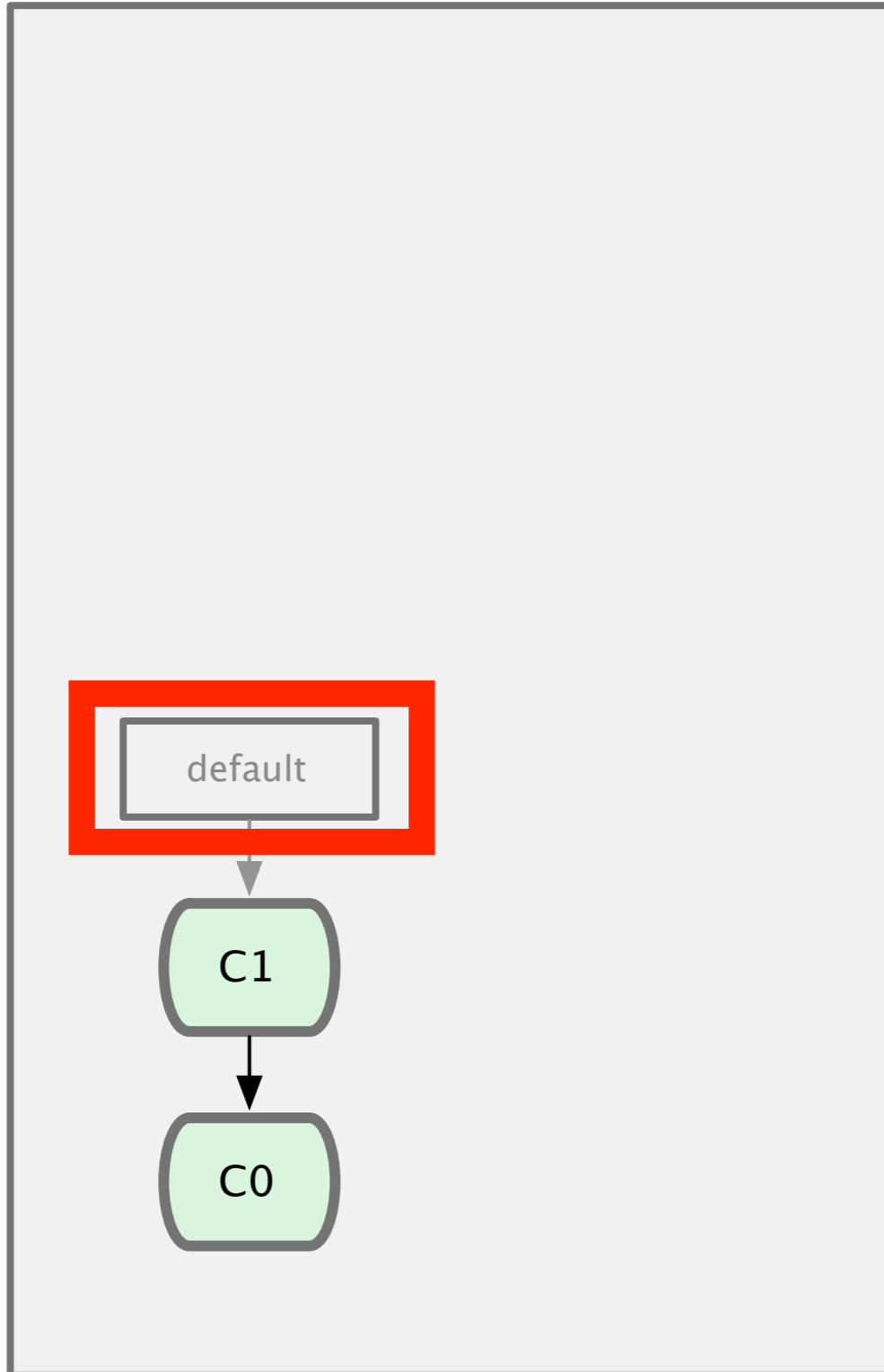


scott

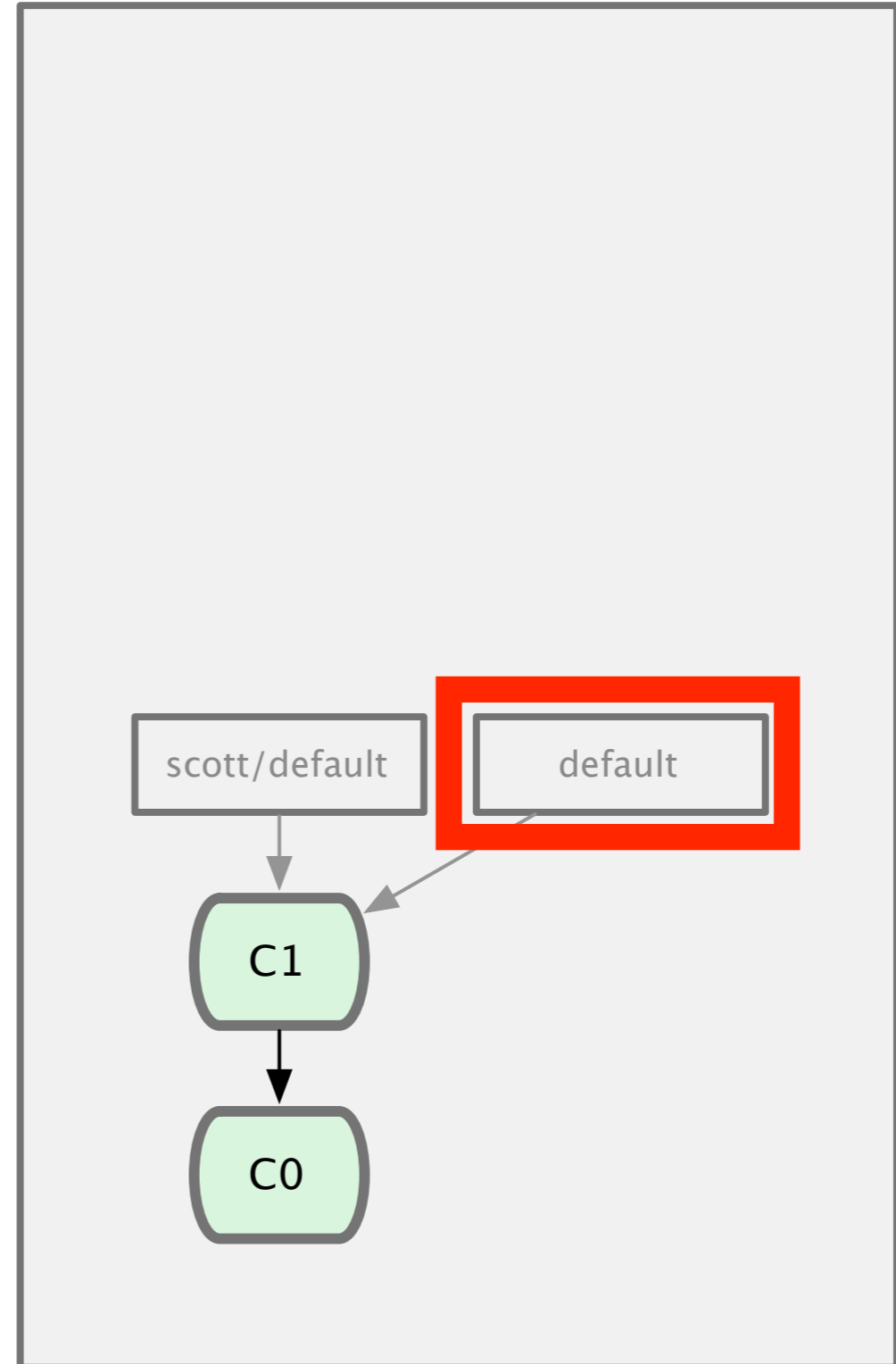
jessica



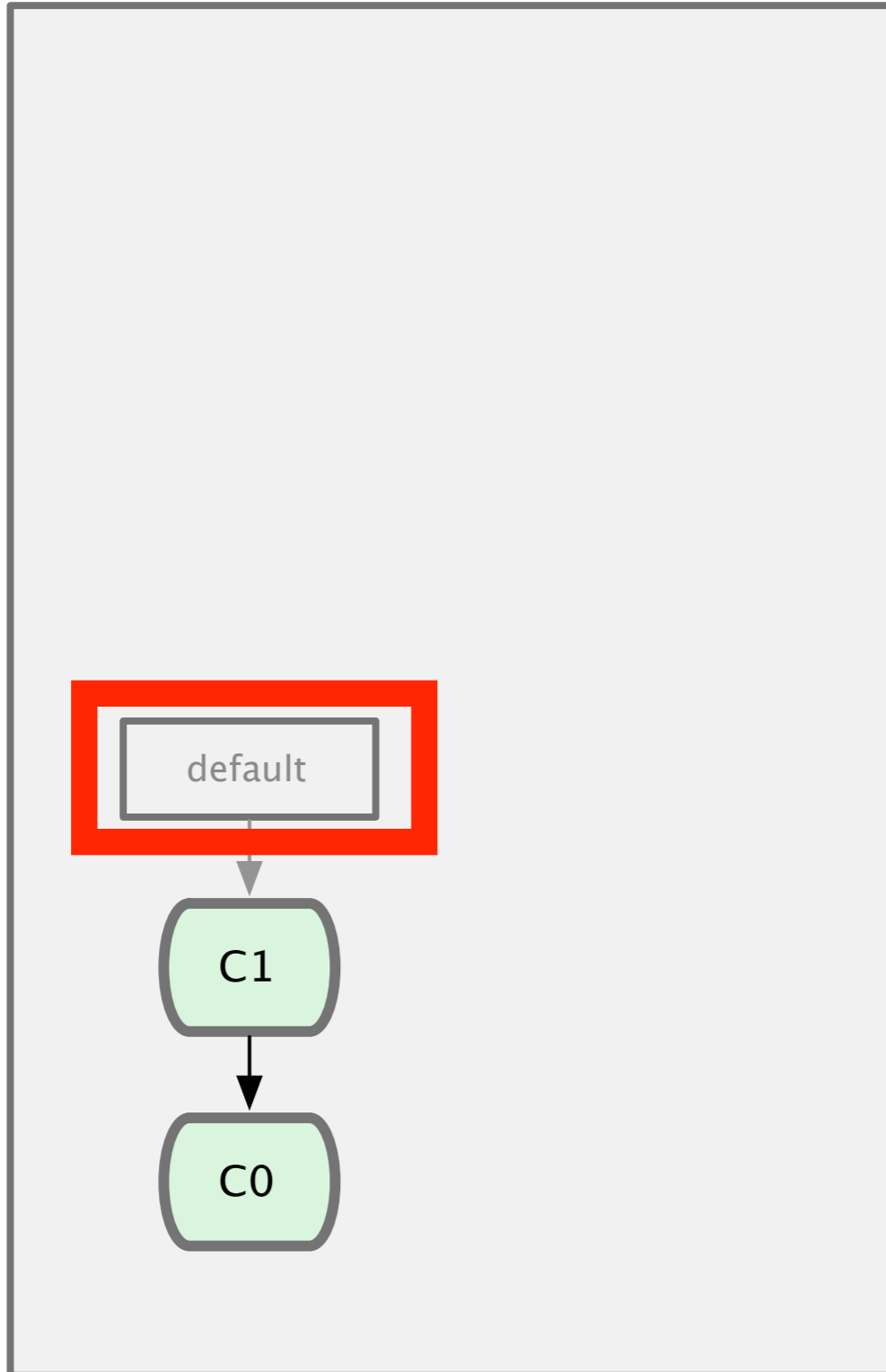
scott



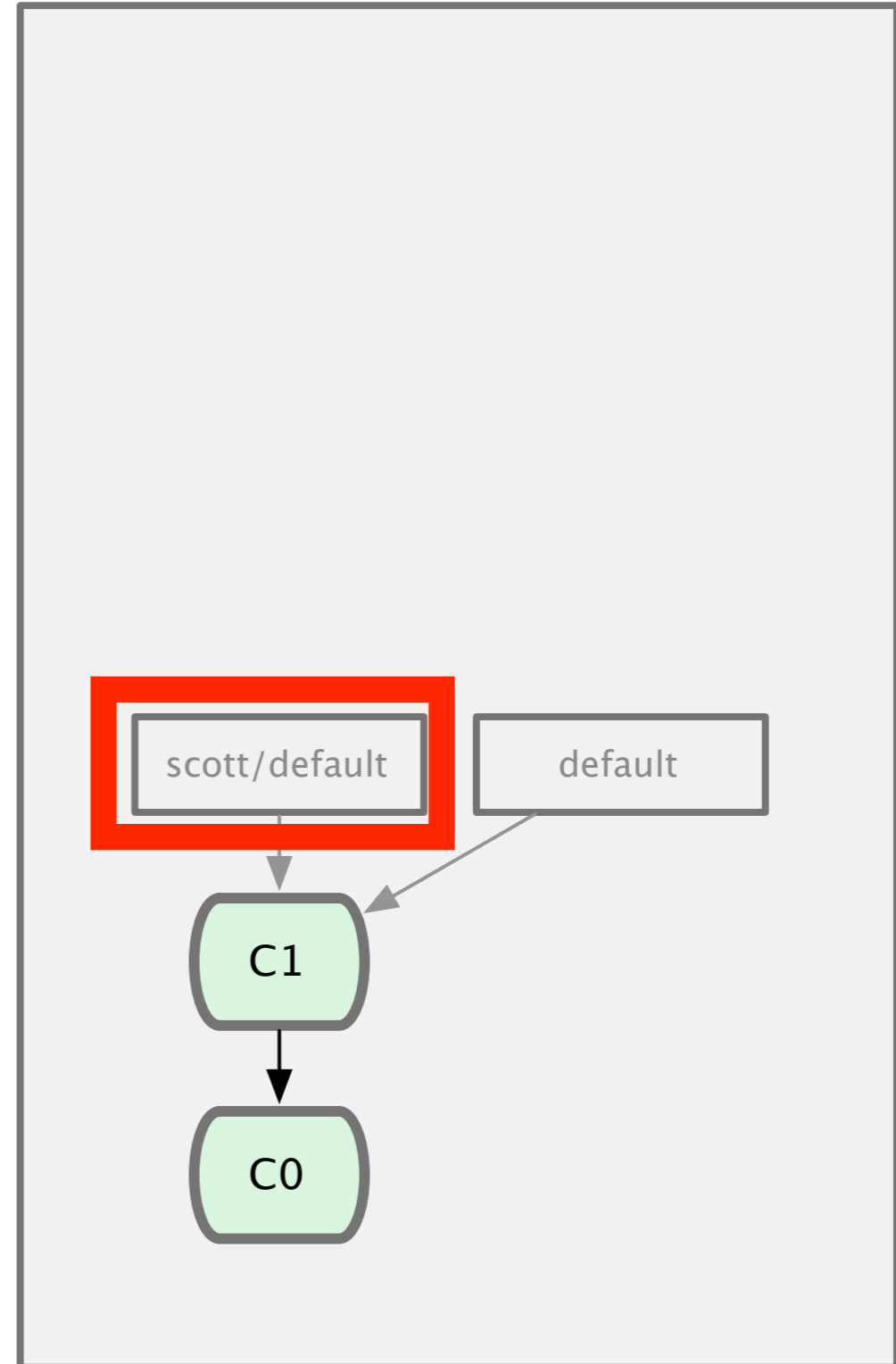
jessica



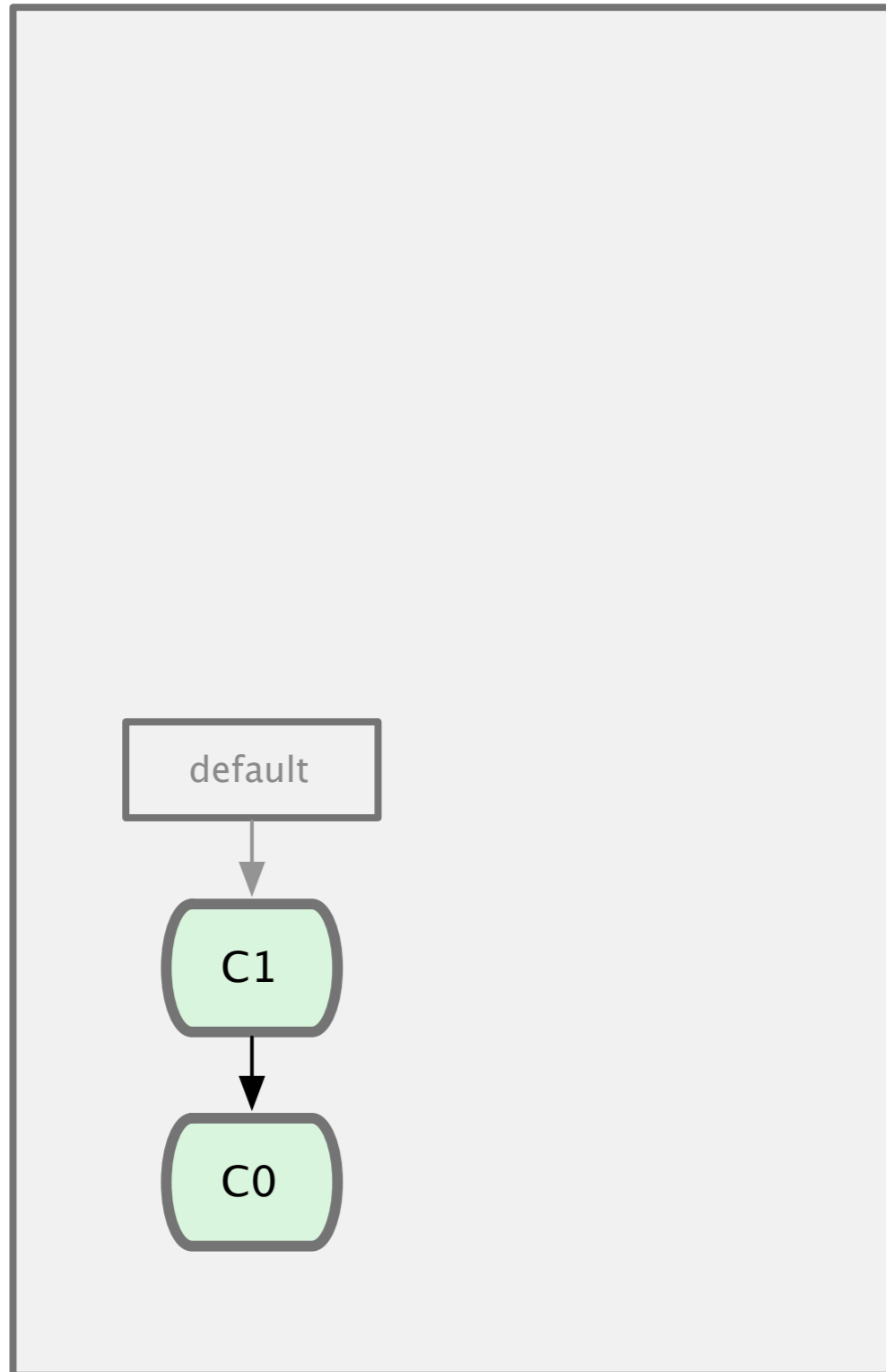
scott



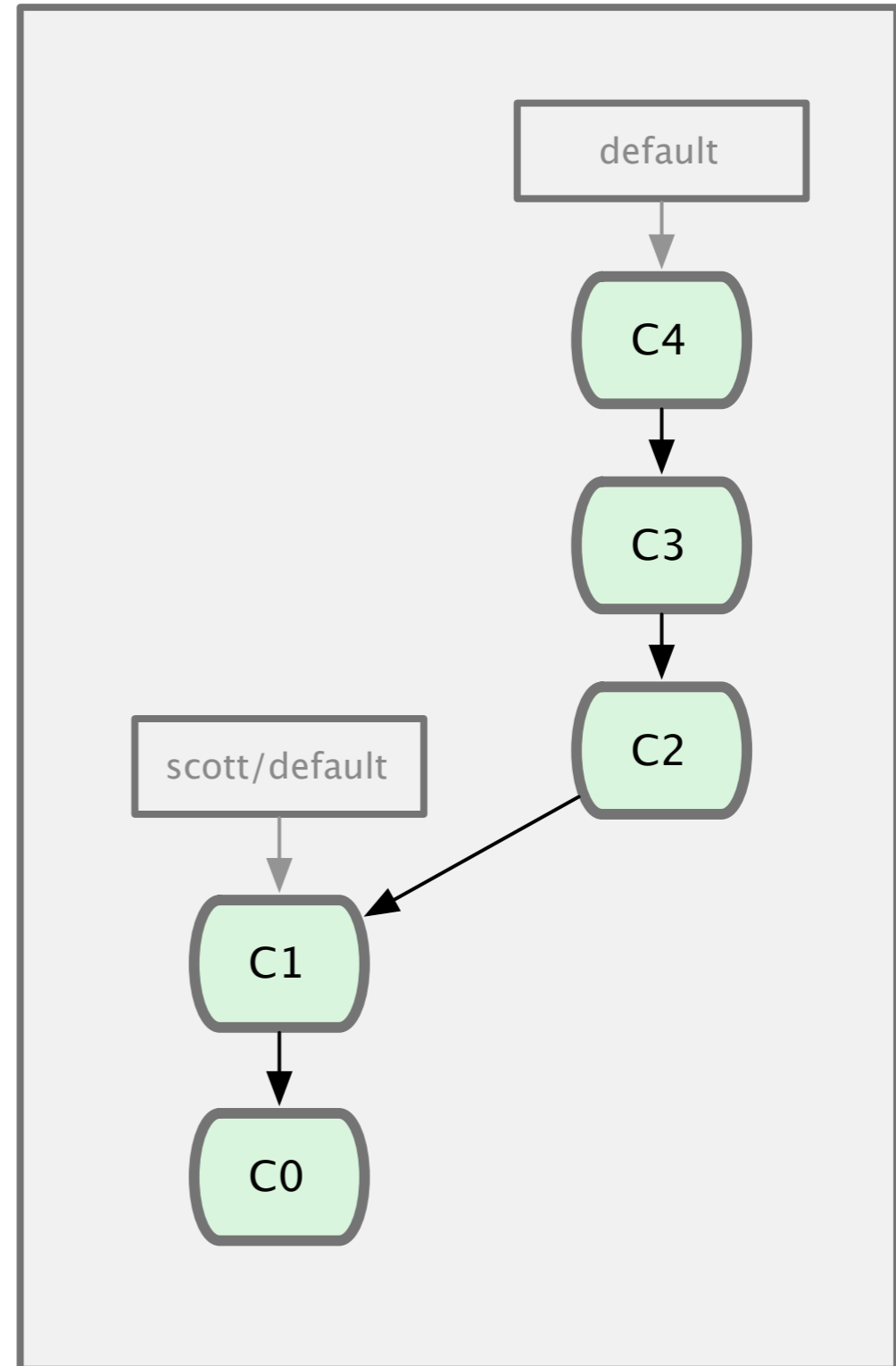
jessica



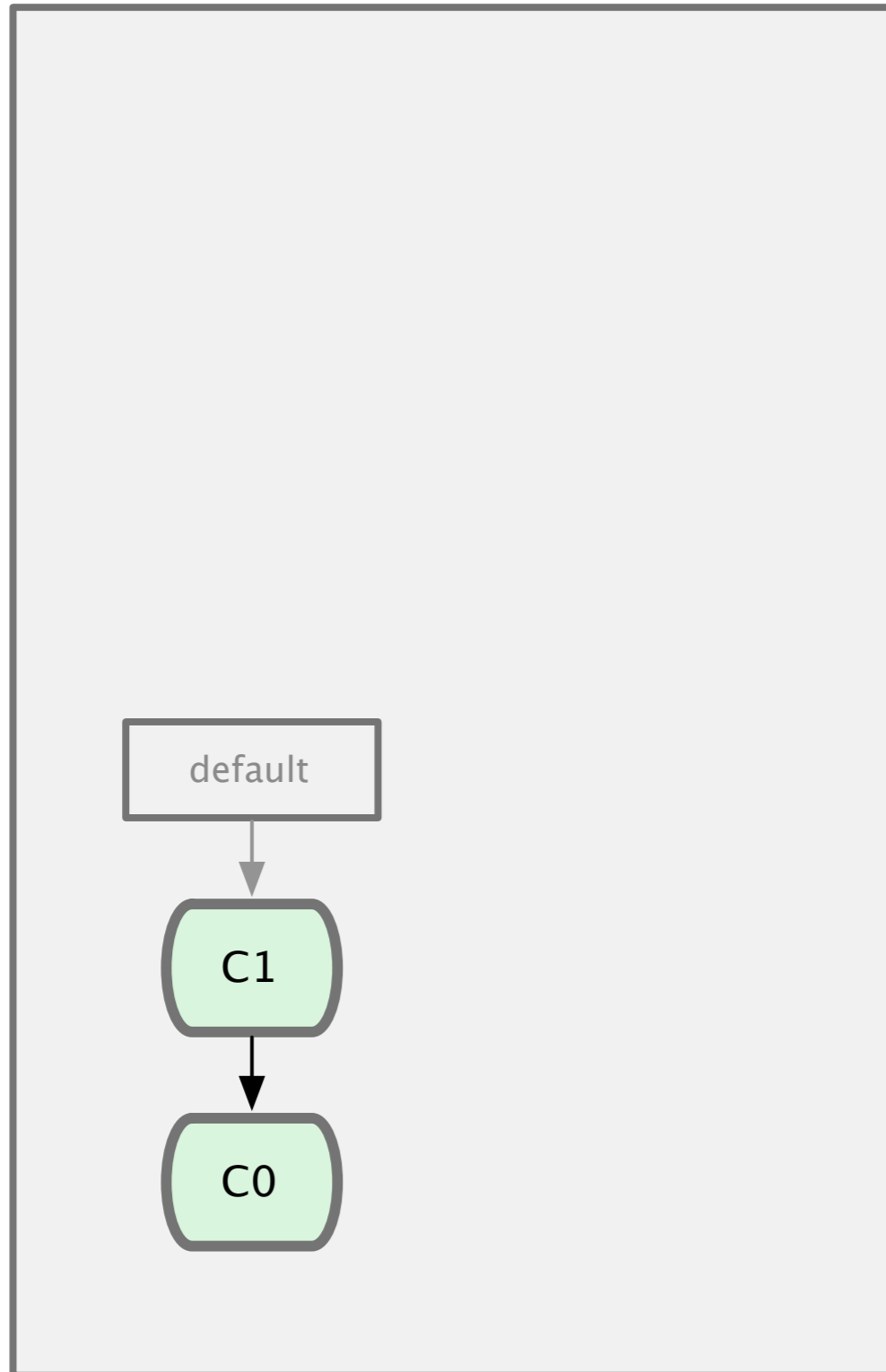
scott



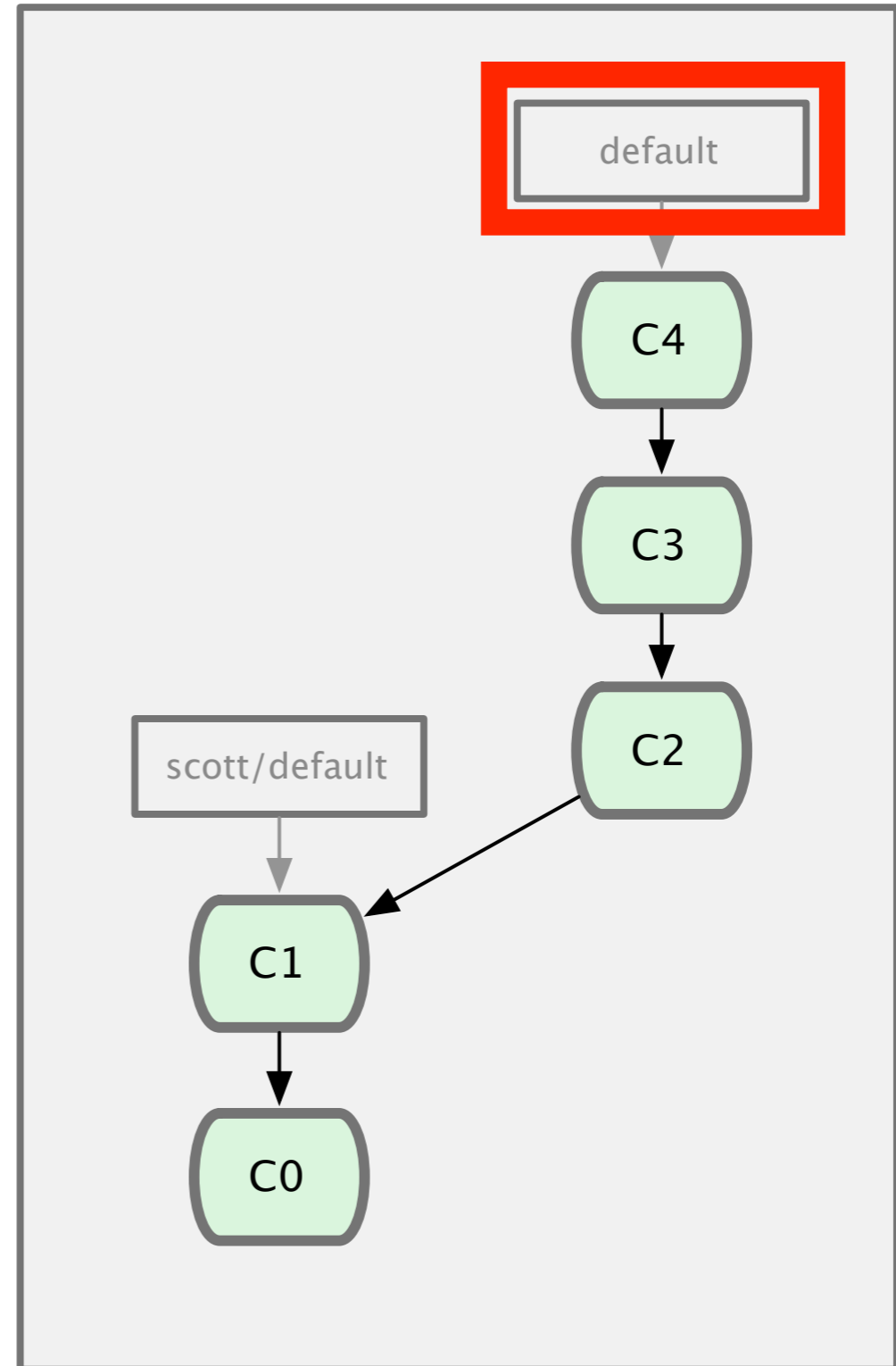
jessica



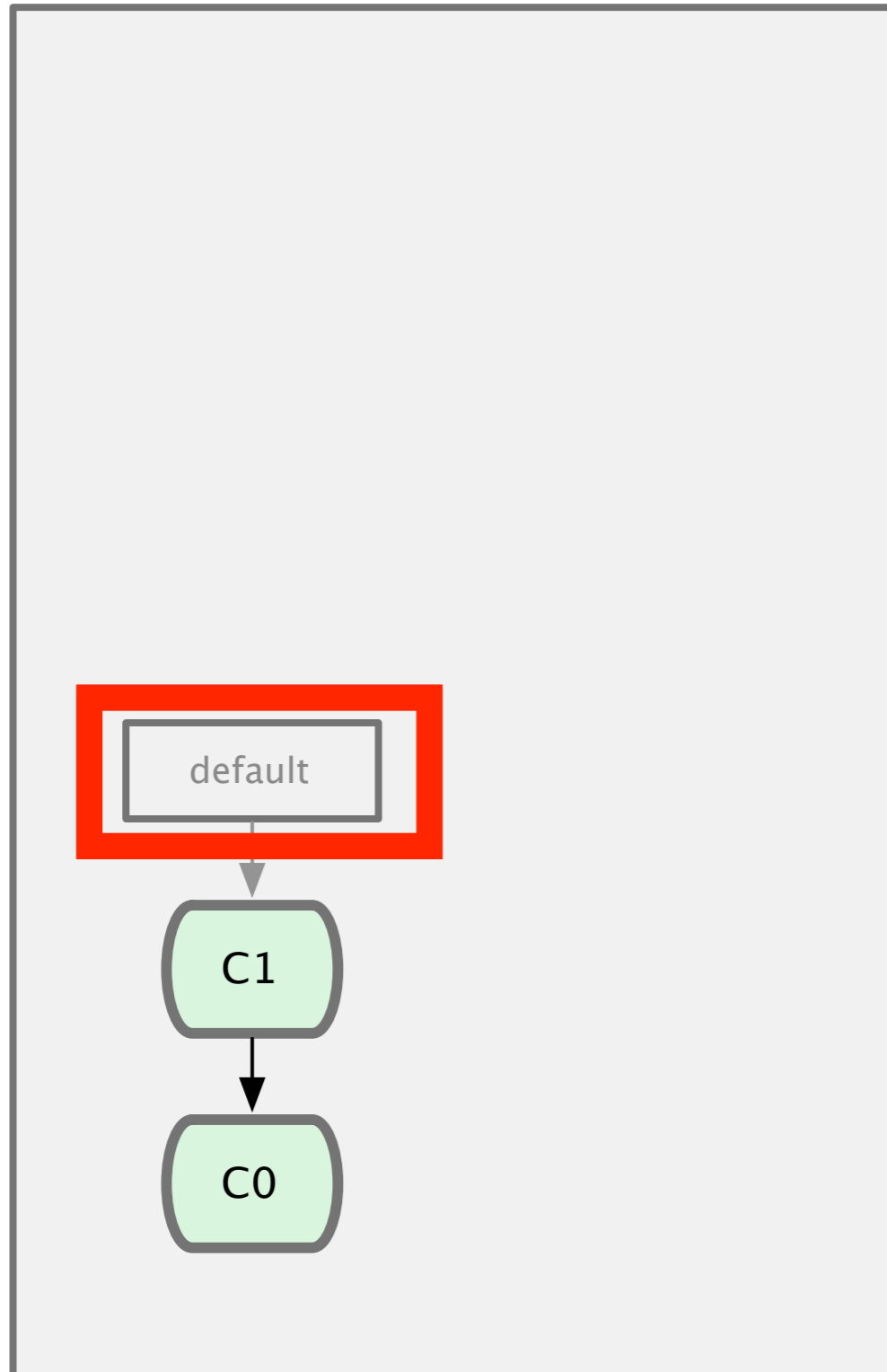
scott



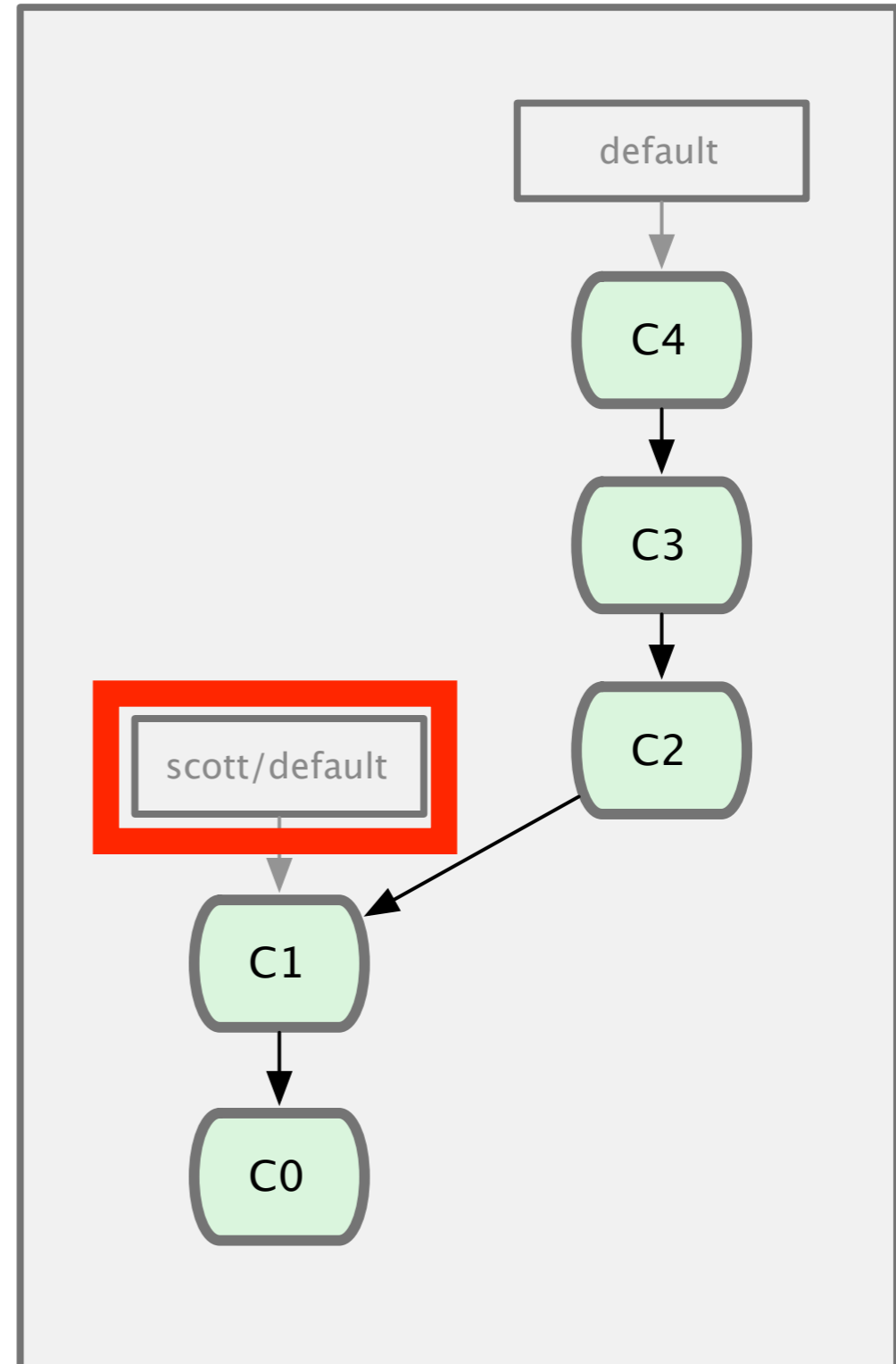
jessica



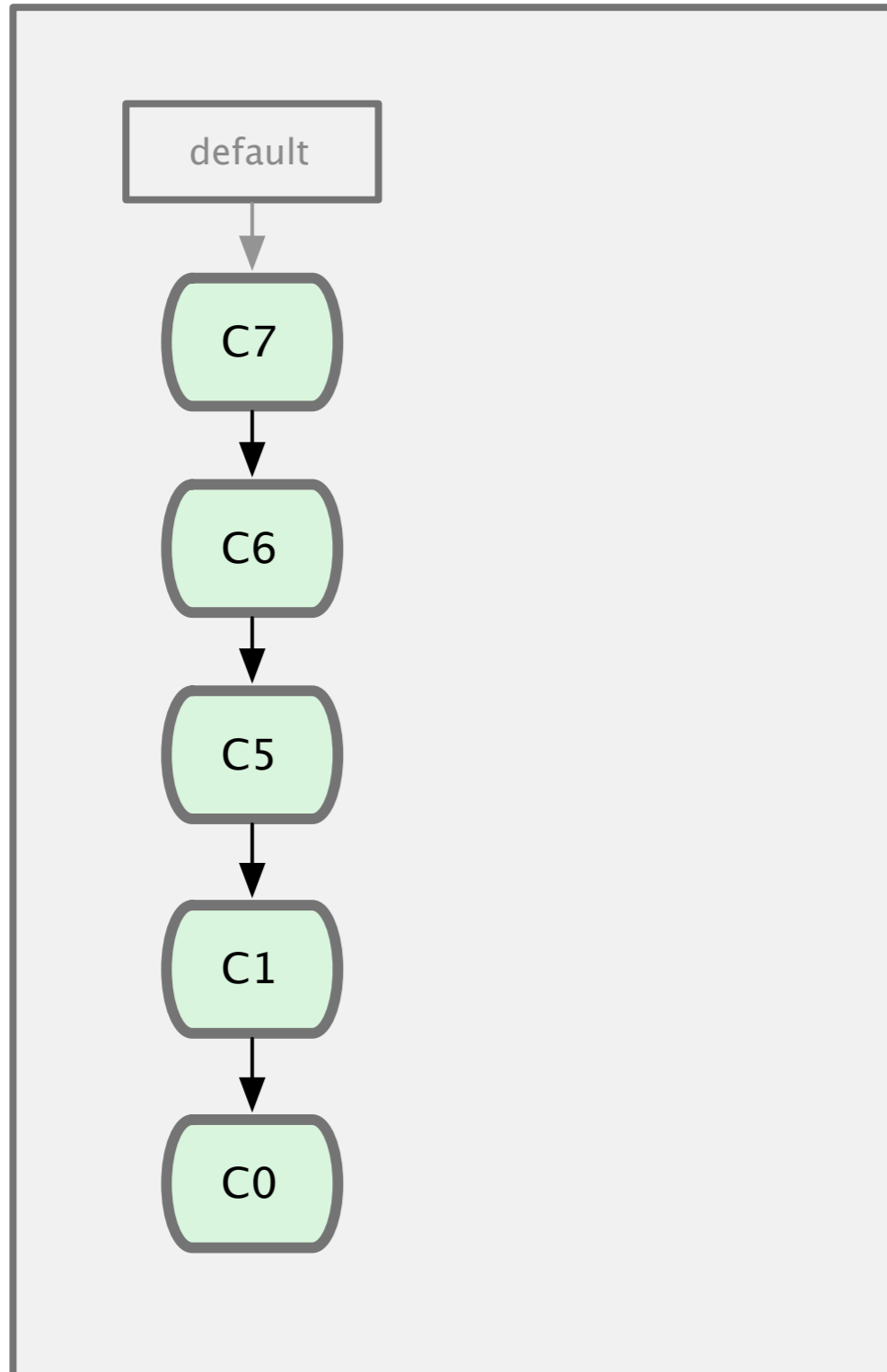
scott



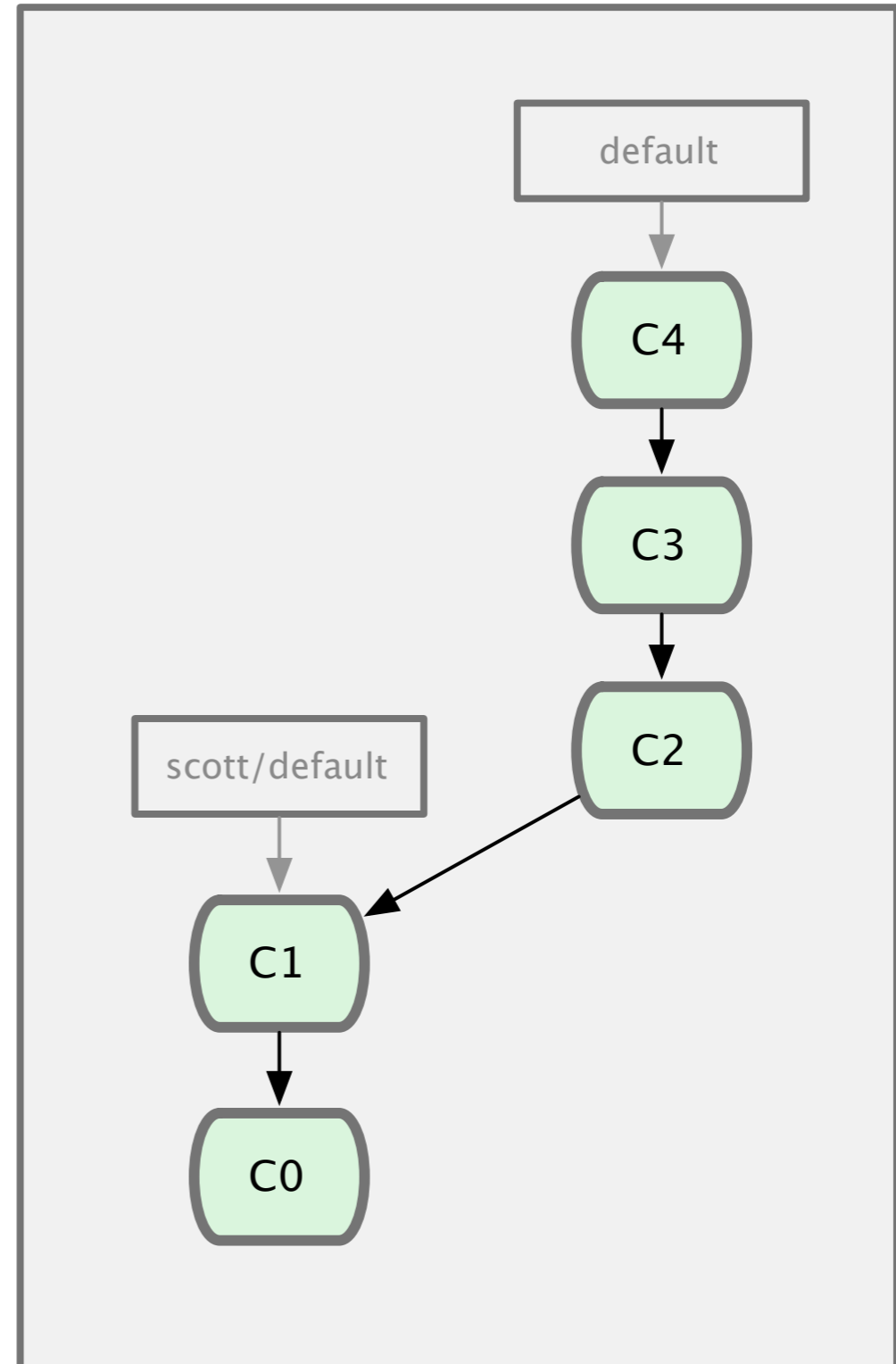
jessica



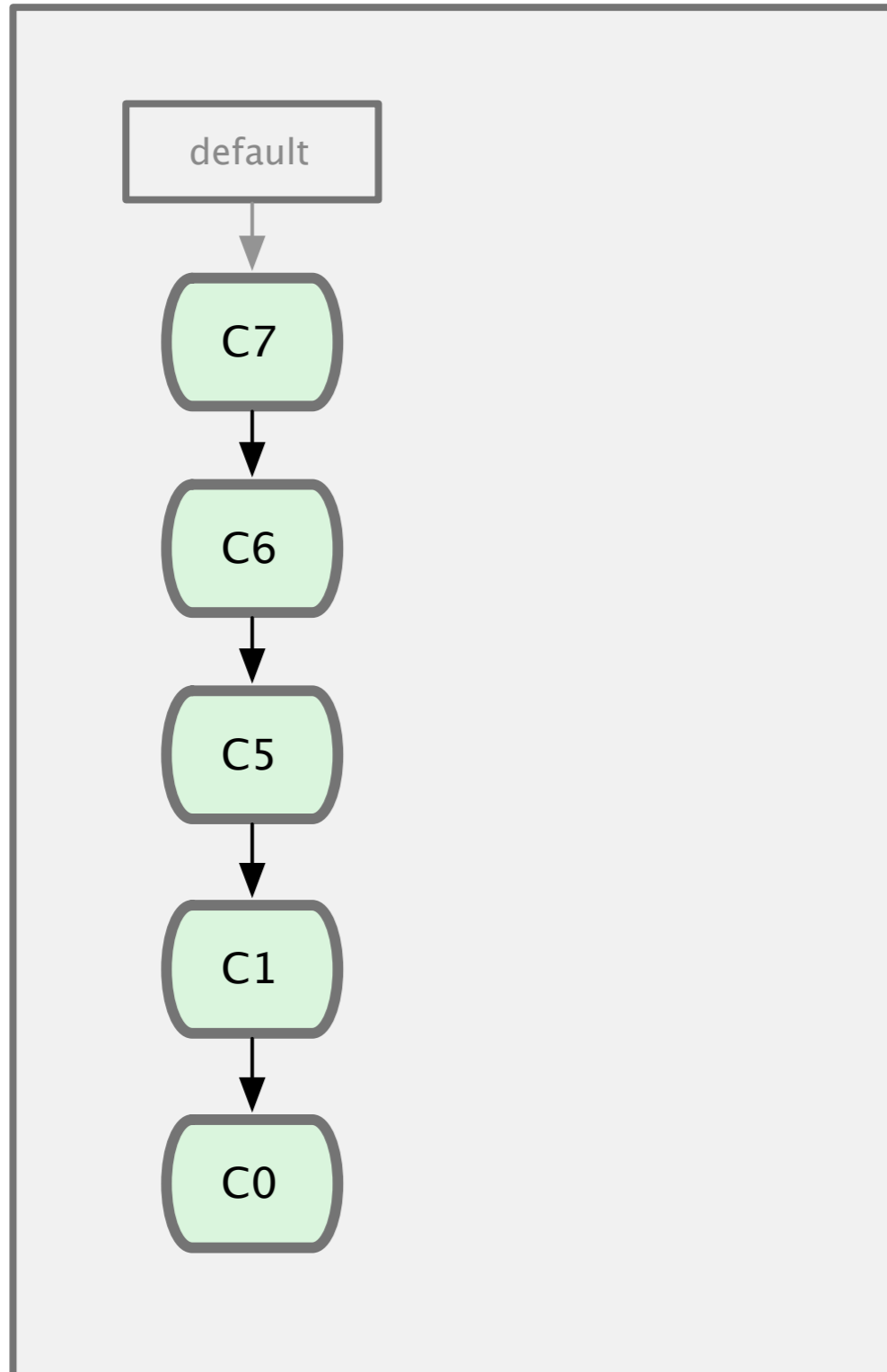
scott



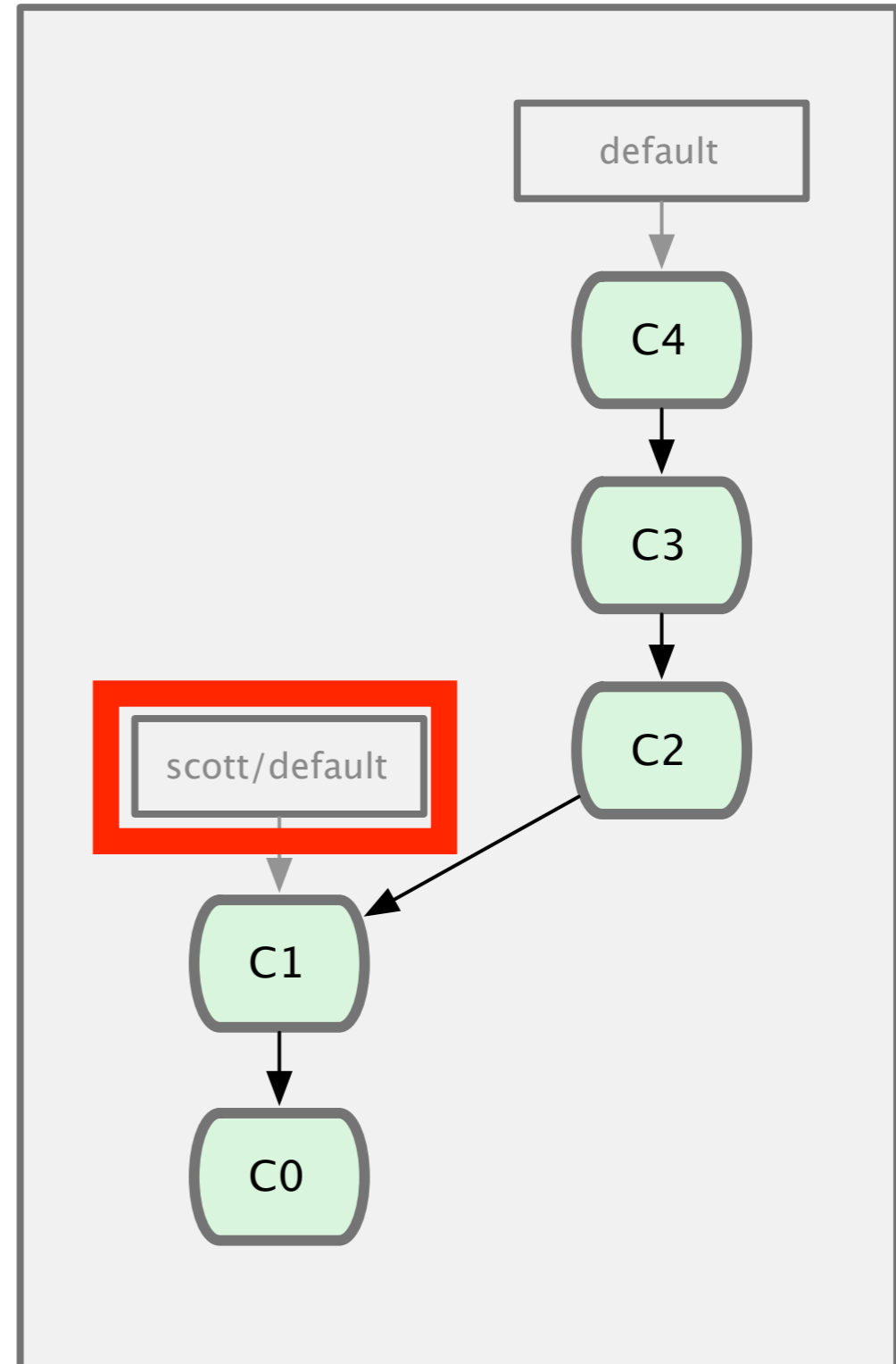
jessica



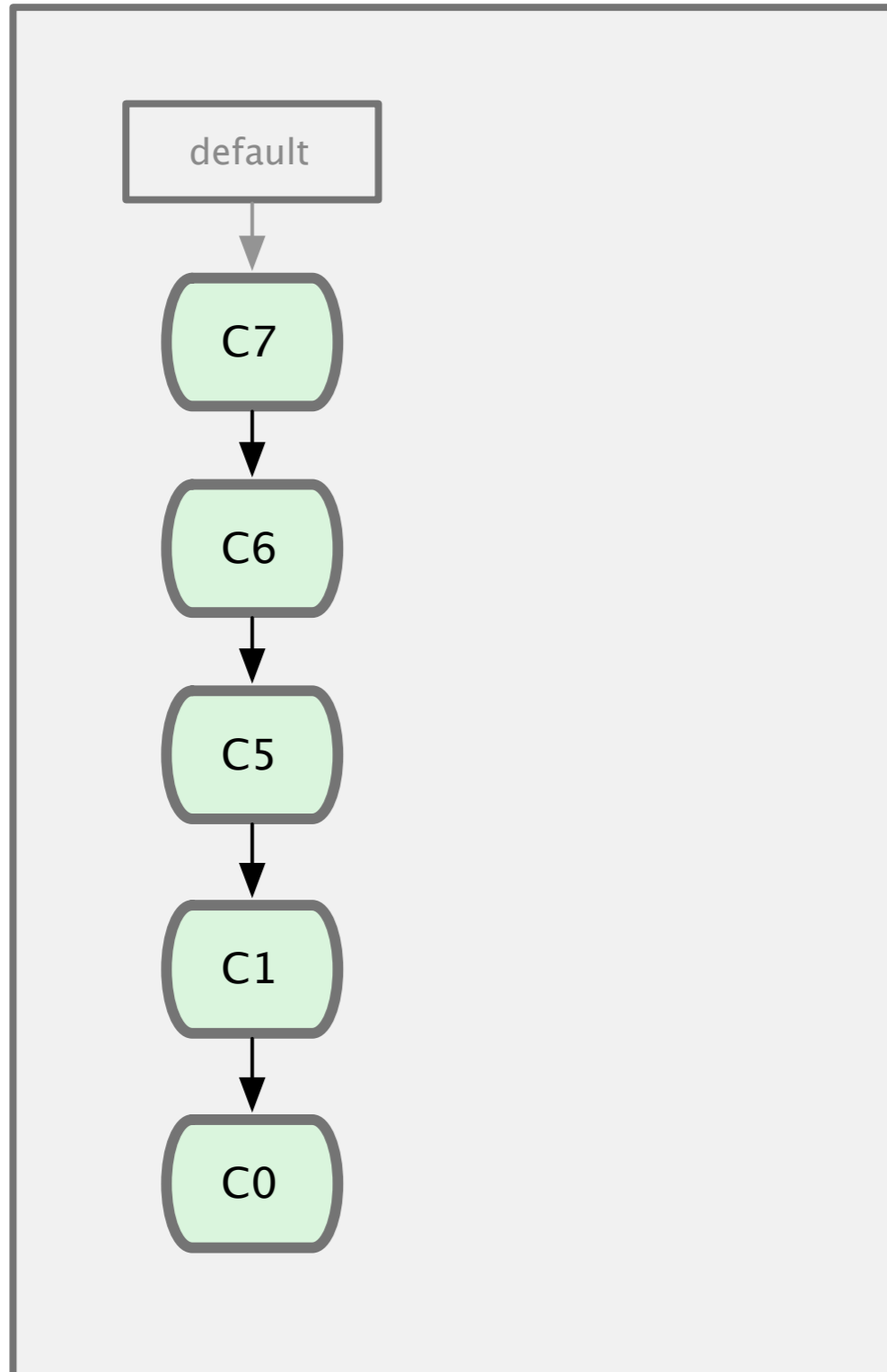
scott



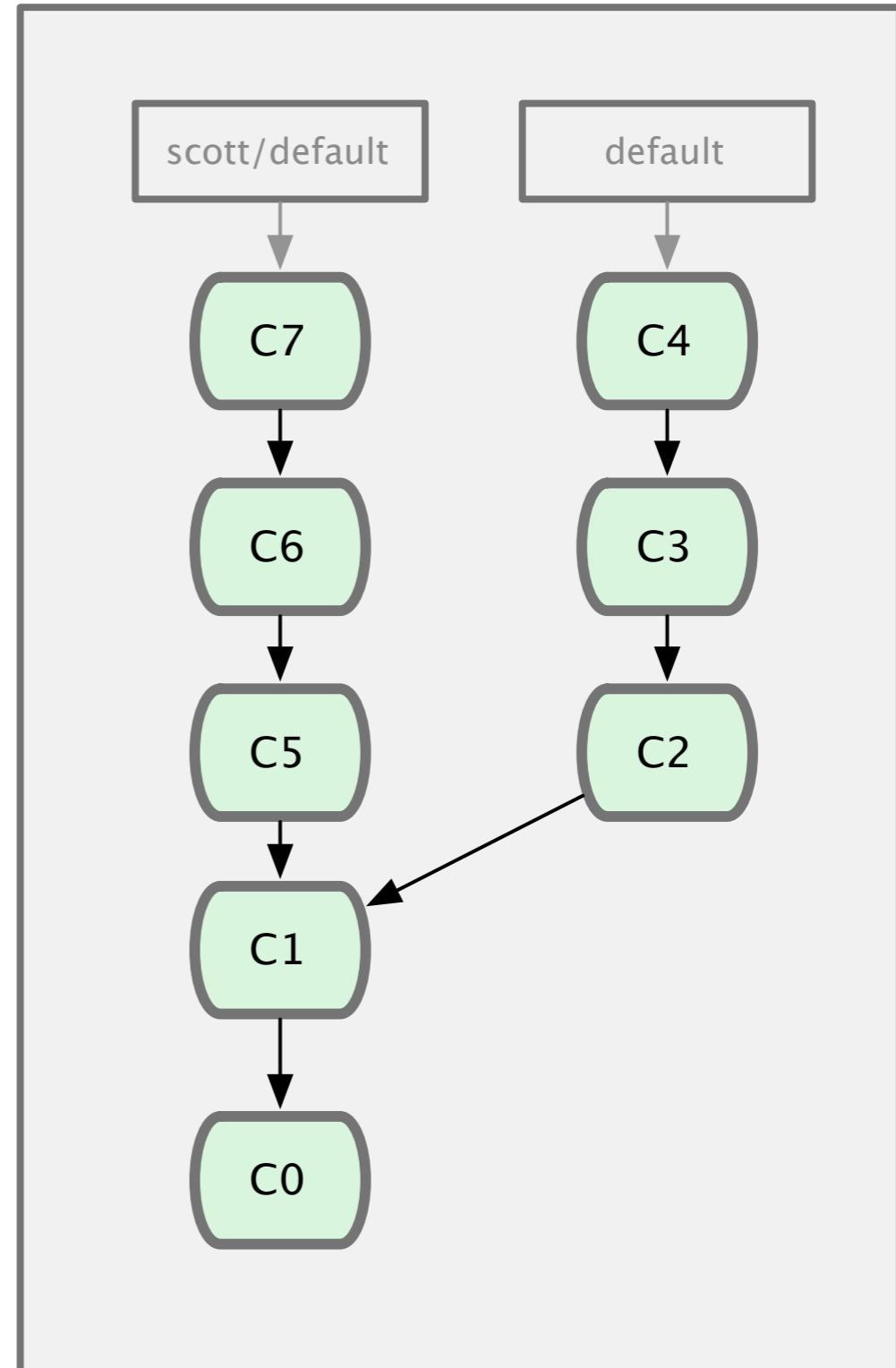
jessica



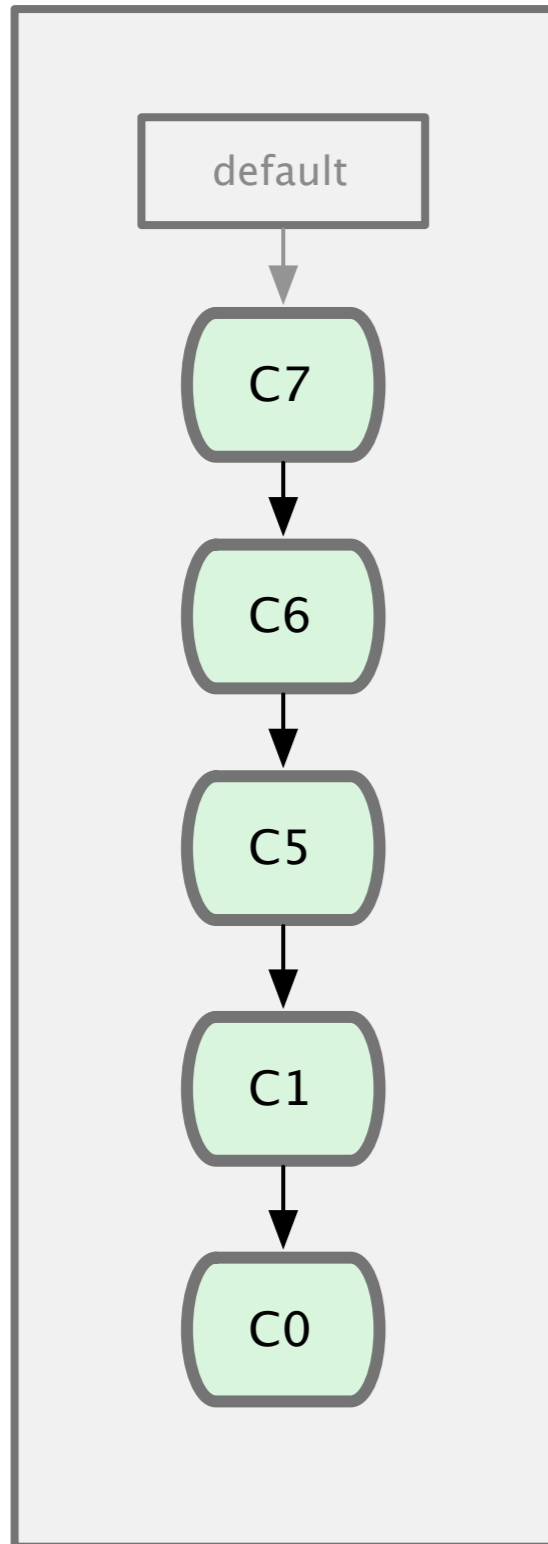
scott



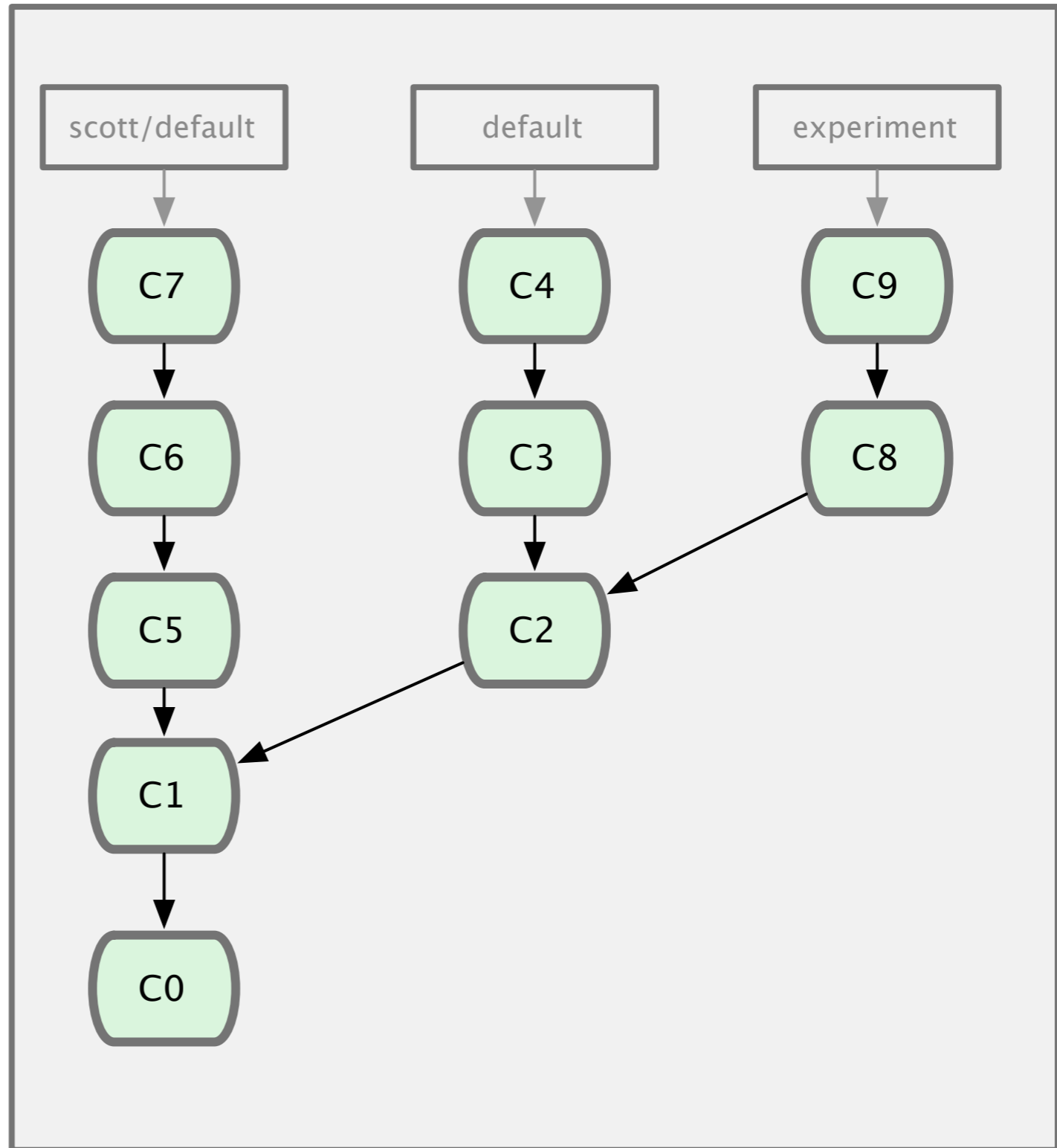
jessica



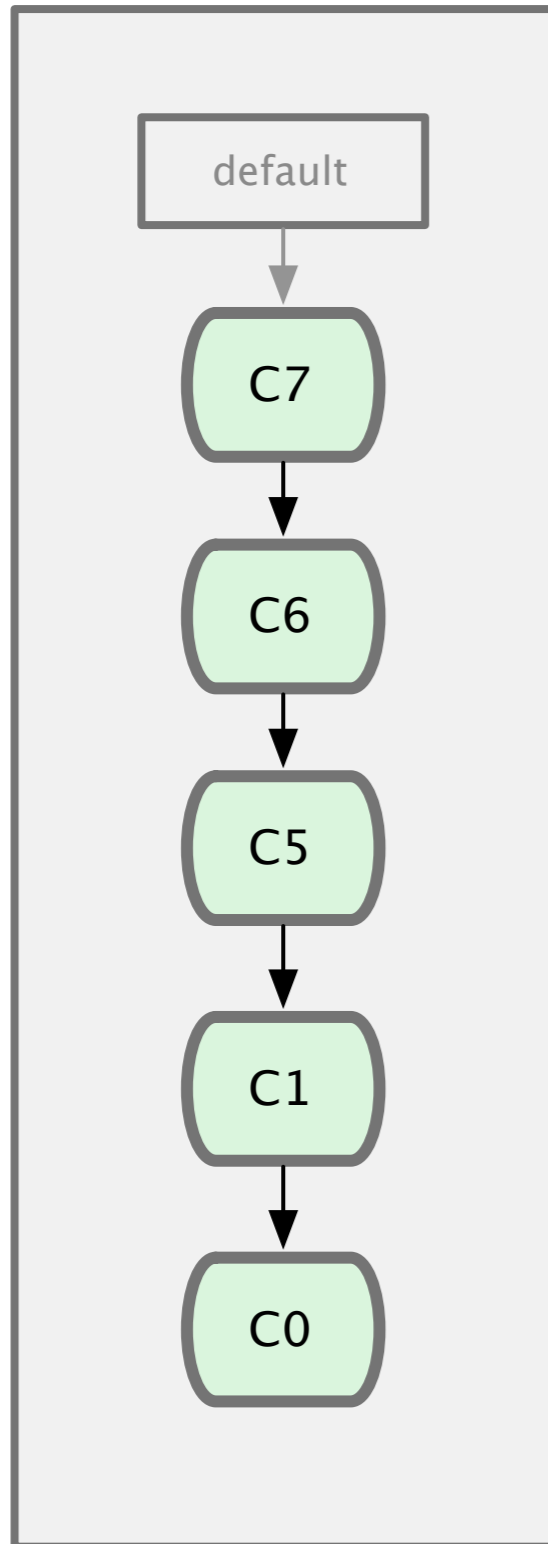
scott



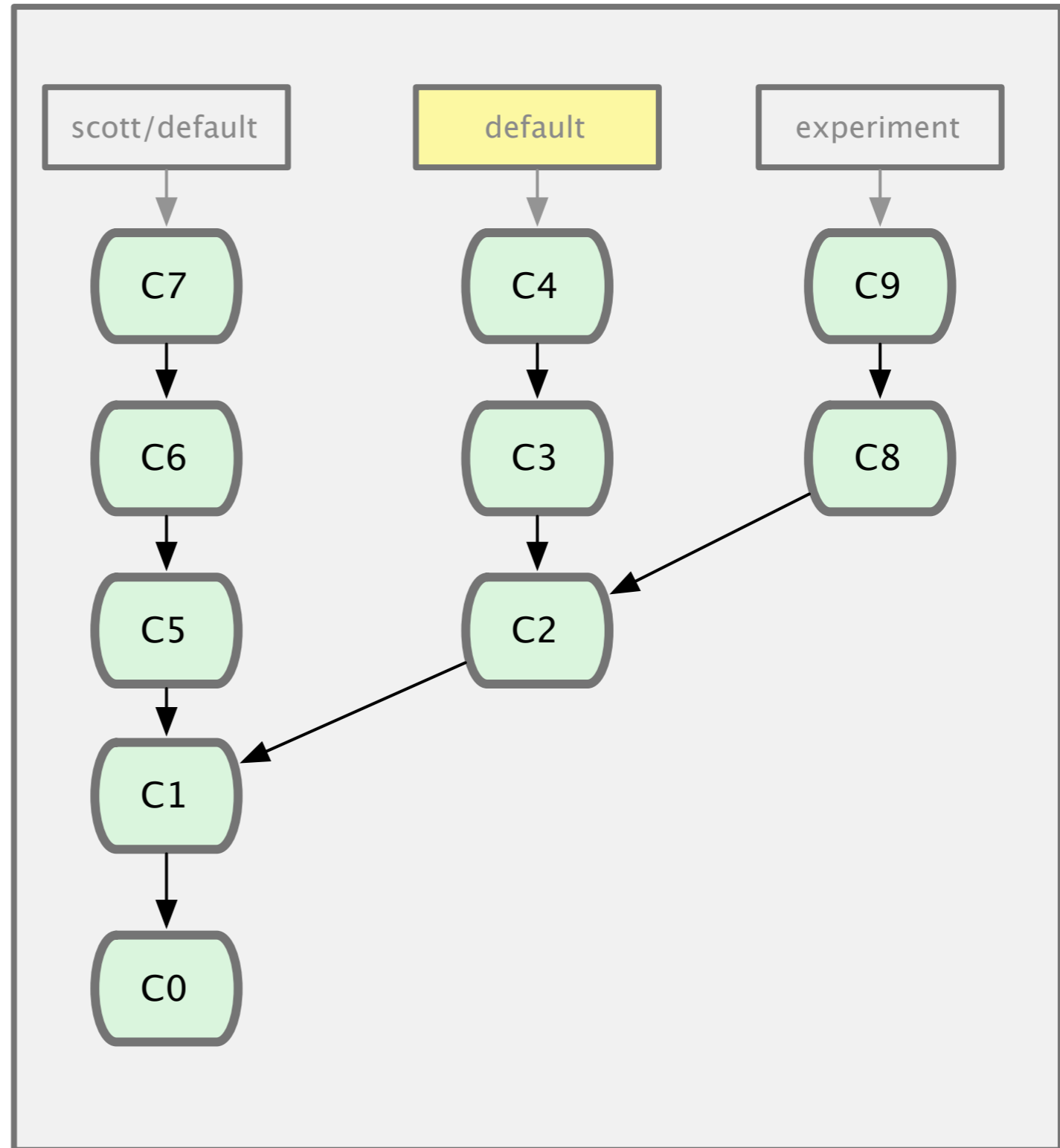
jessica



scott



jessica



scott

jessica

default

scott/default

default

experiment

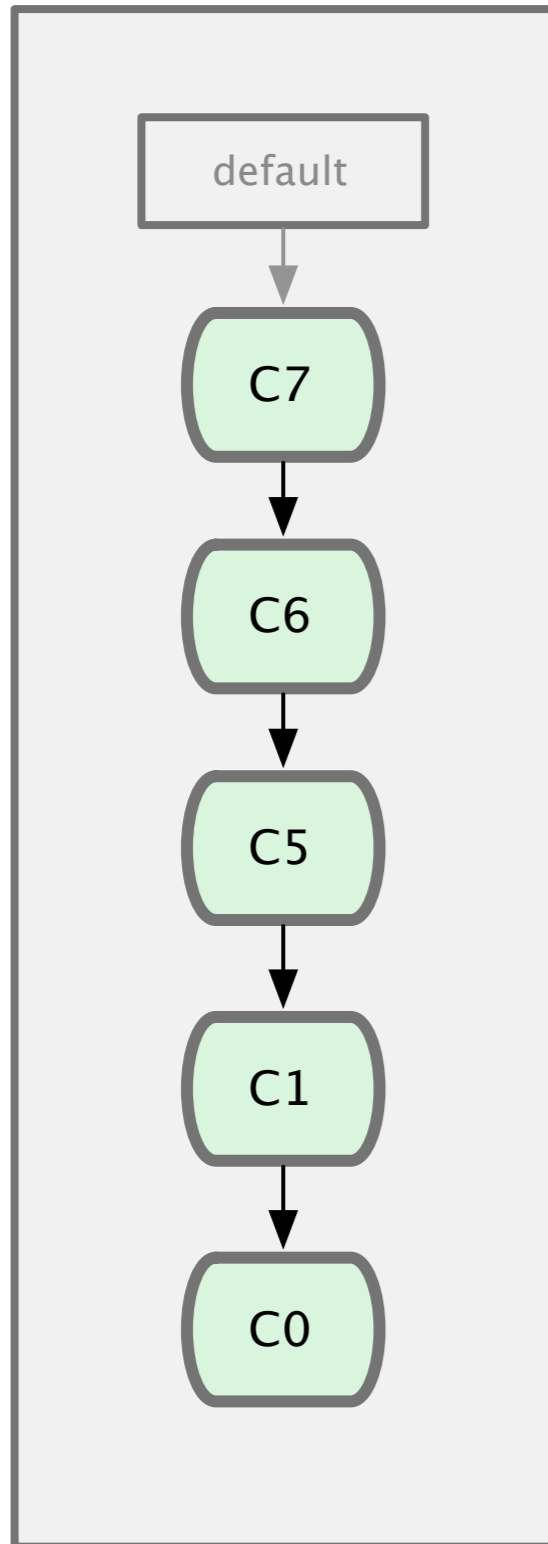
git merge experiment

git merge scott/default

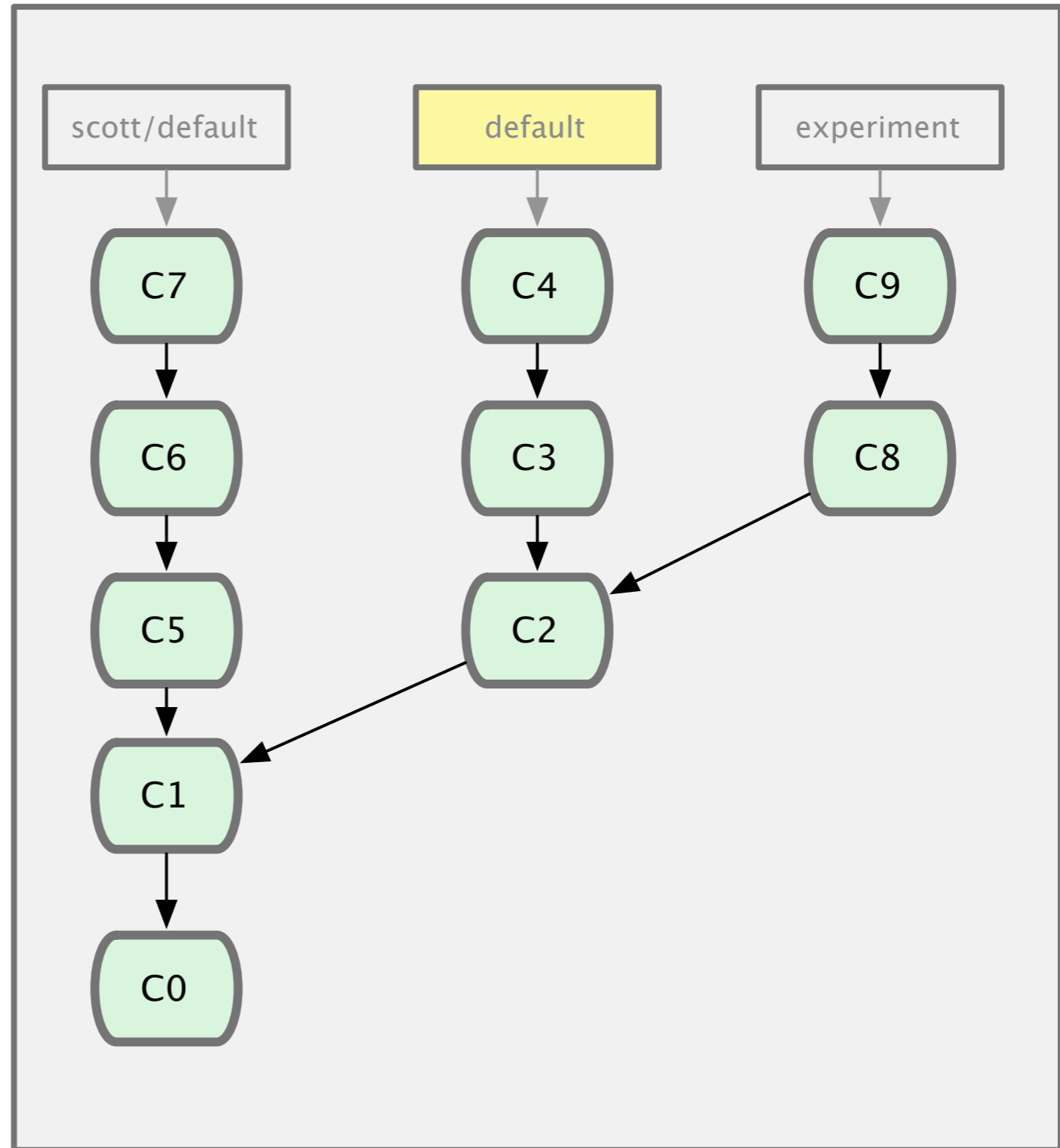
C0

C0

scott

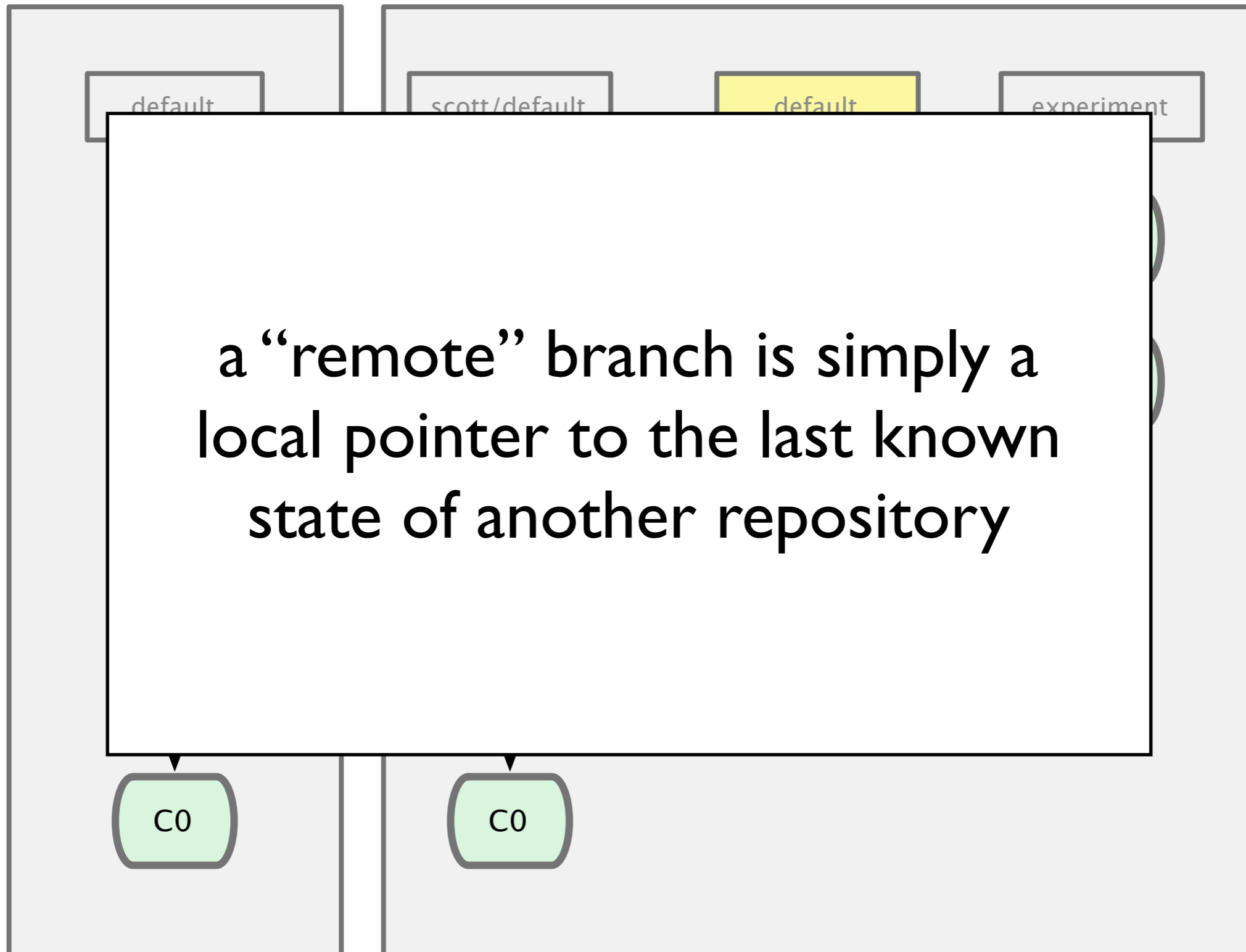


jessica



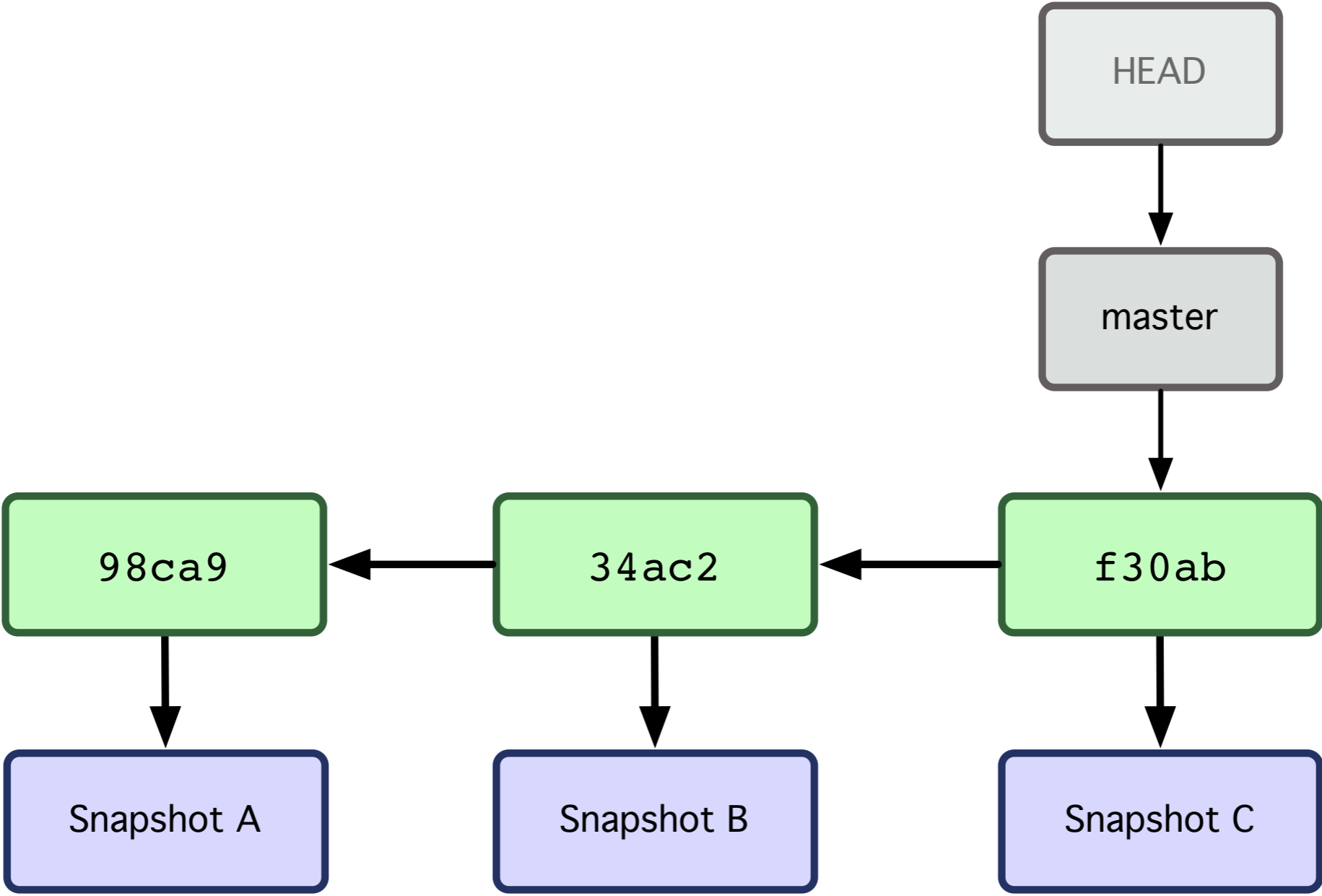
scott

jessica

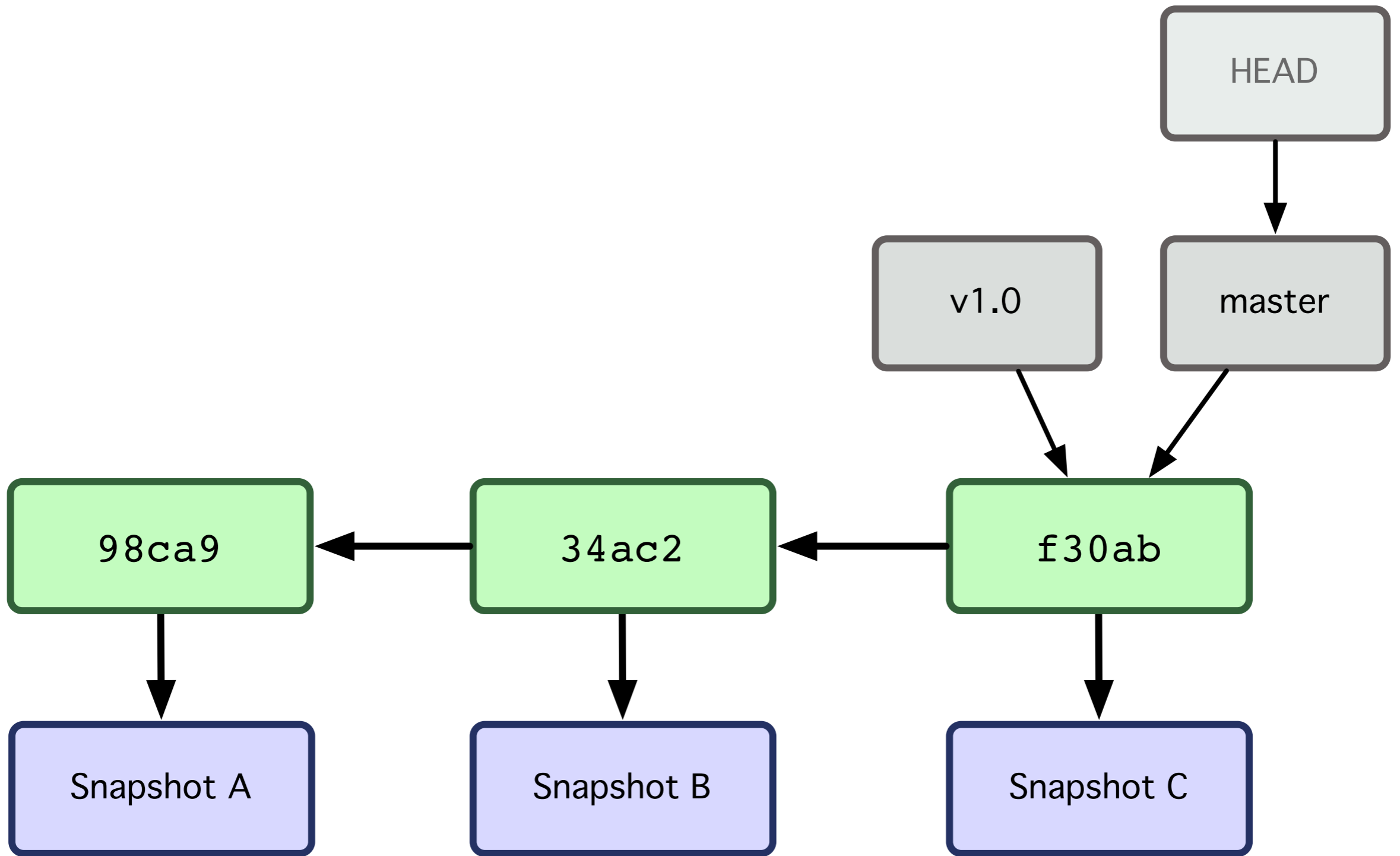


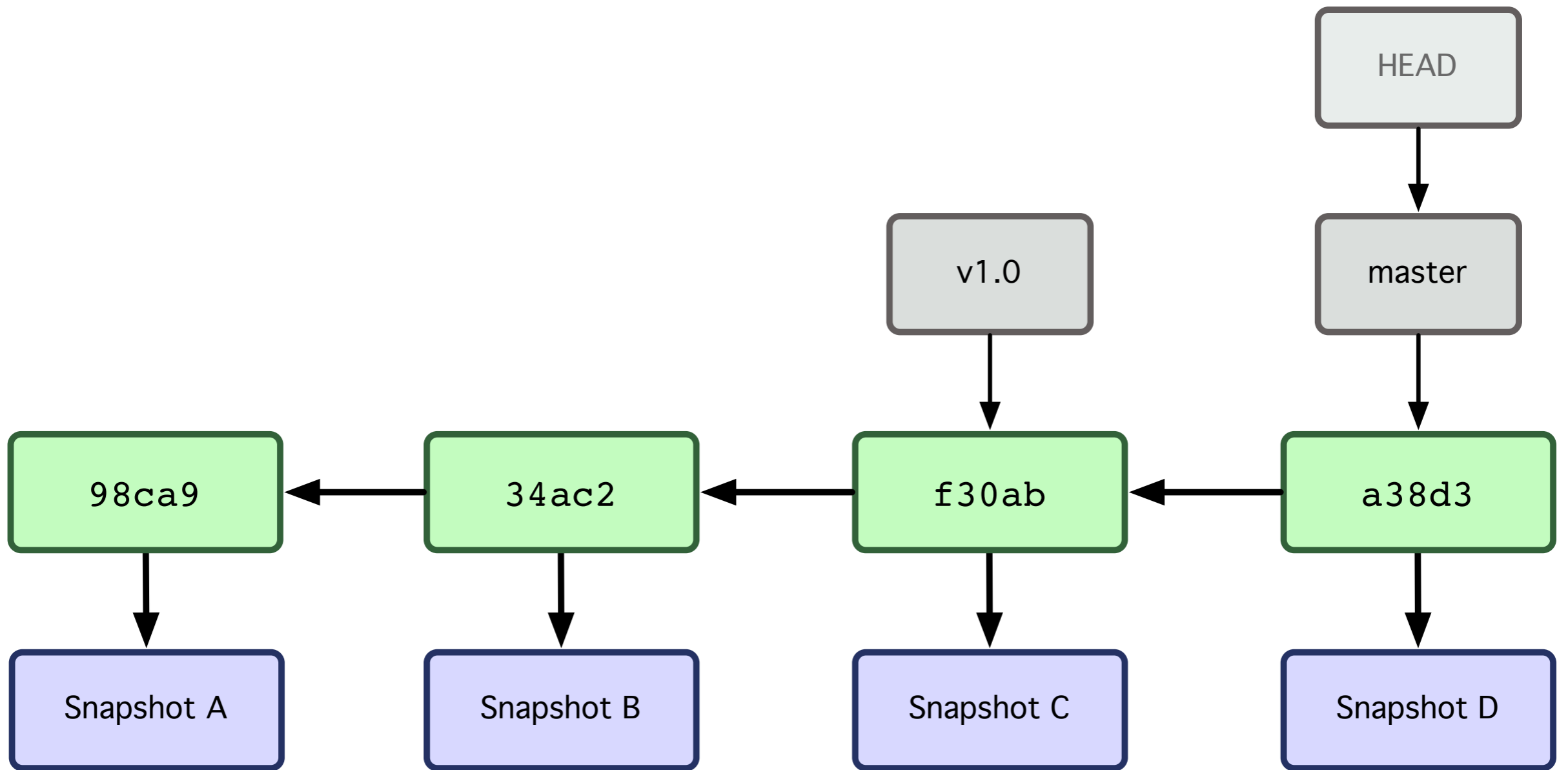
Tagging

lightweight tags



```
git tag v1.0
```

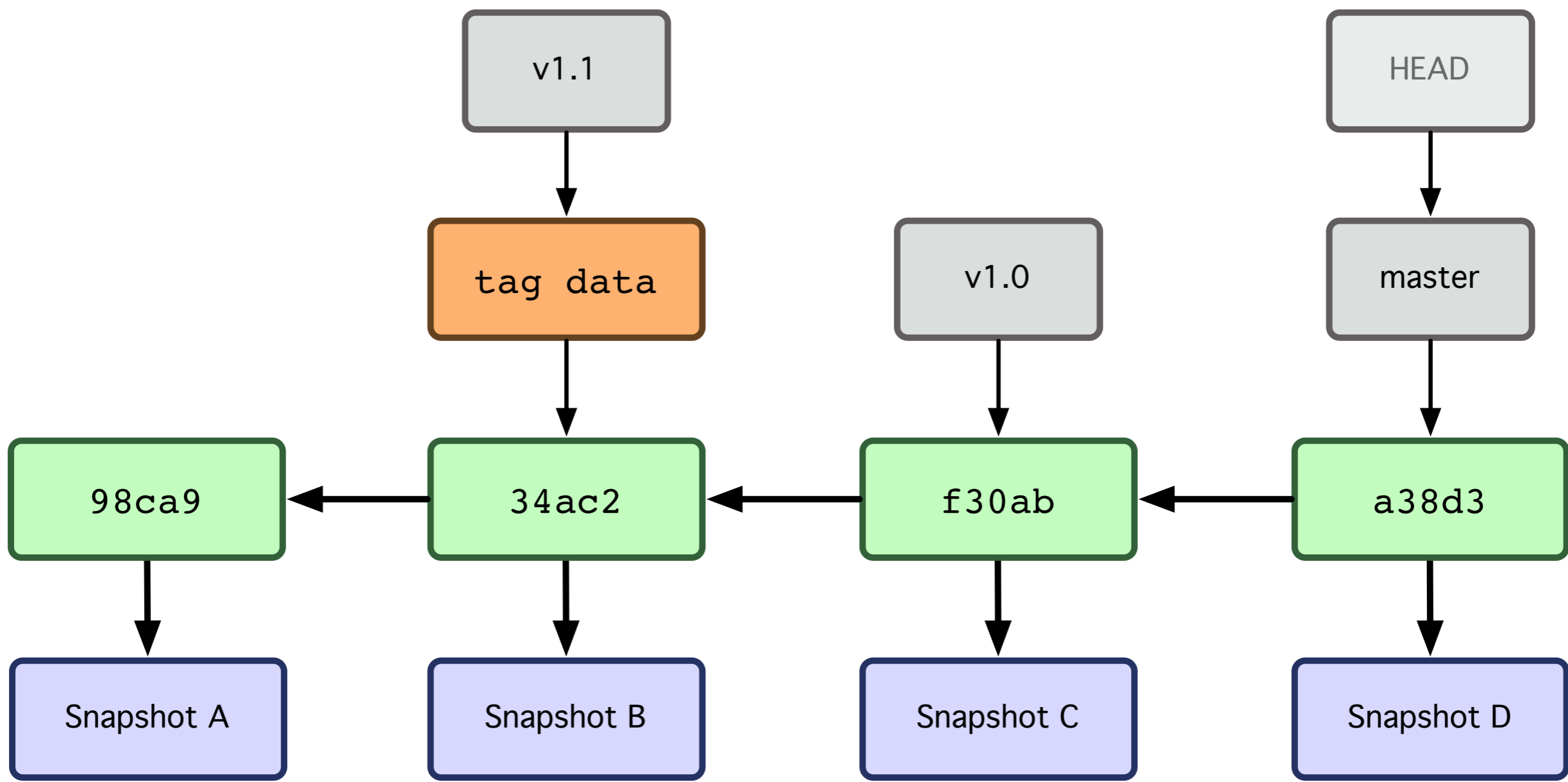


annotated tags

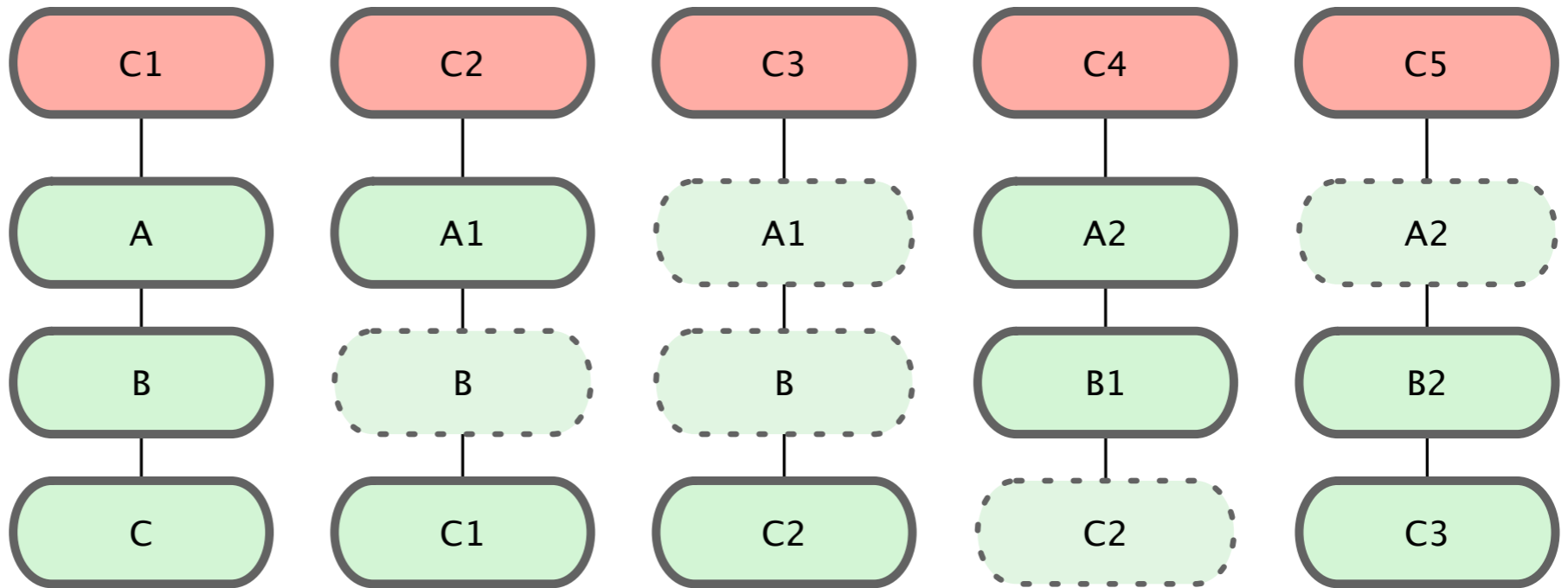
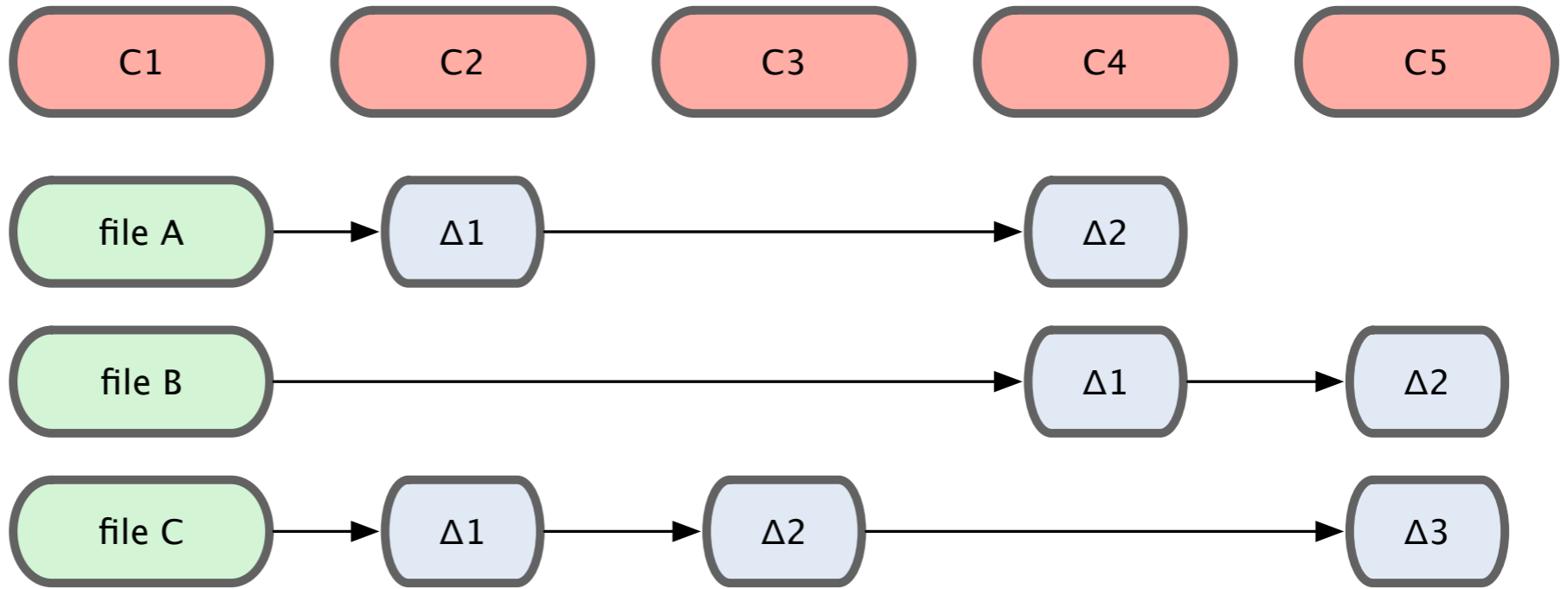
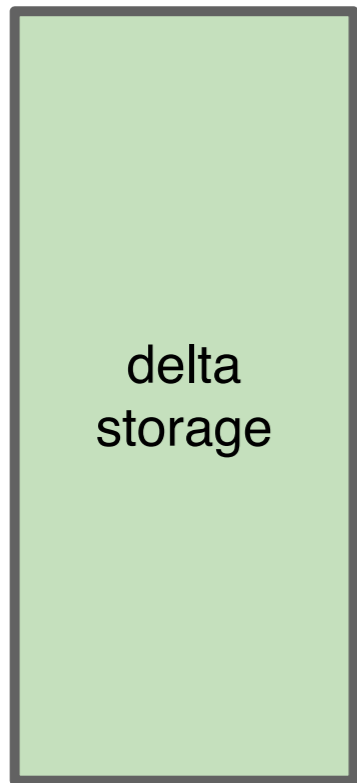
```
git tag -a v1.1 34ac2
```

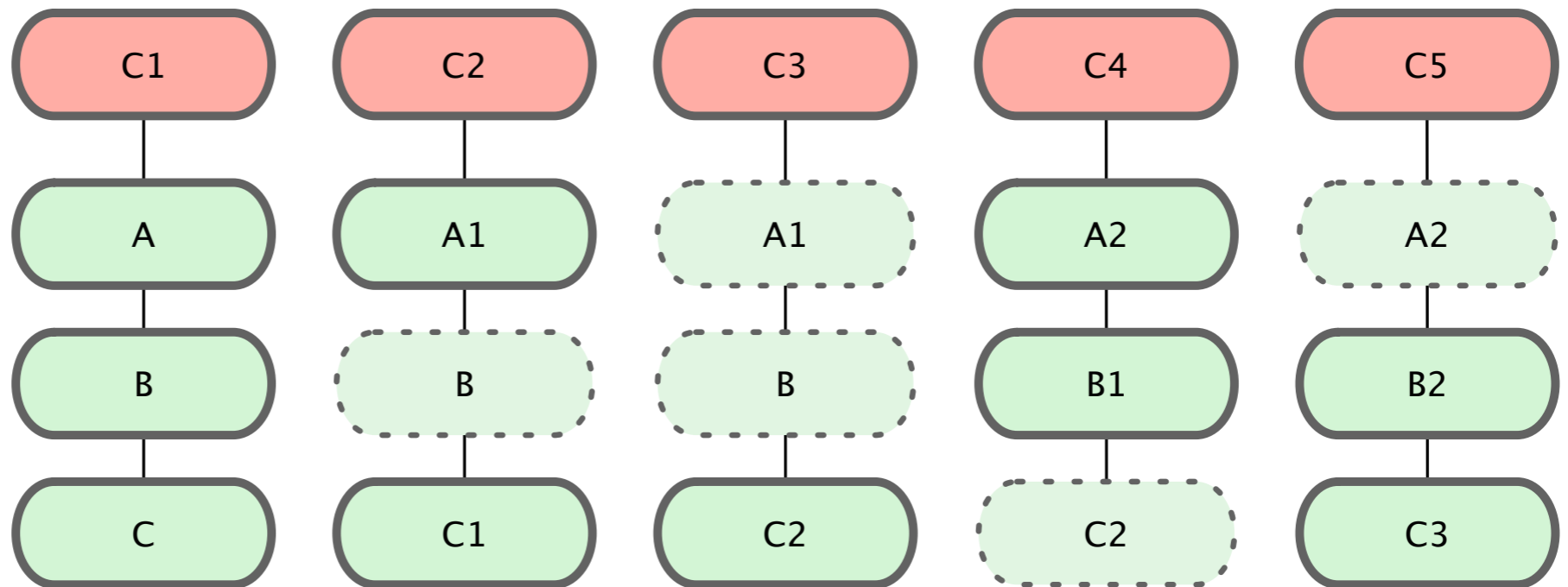
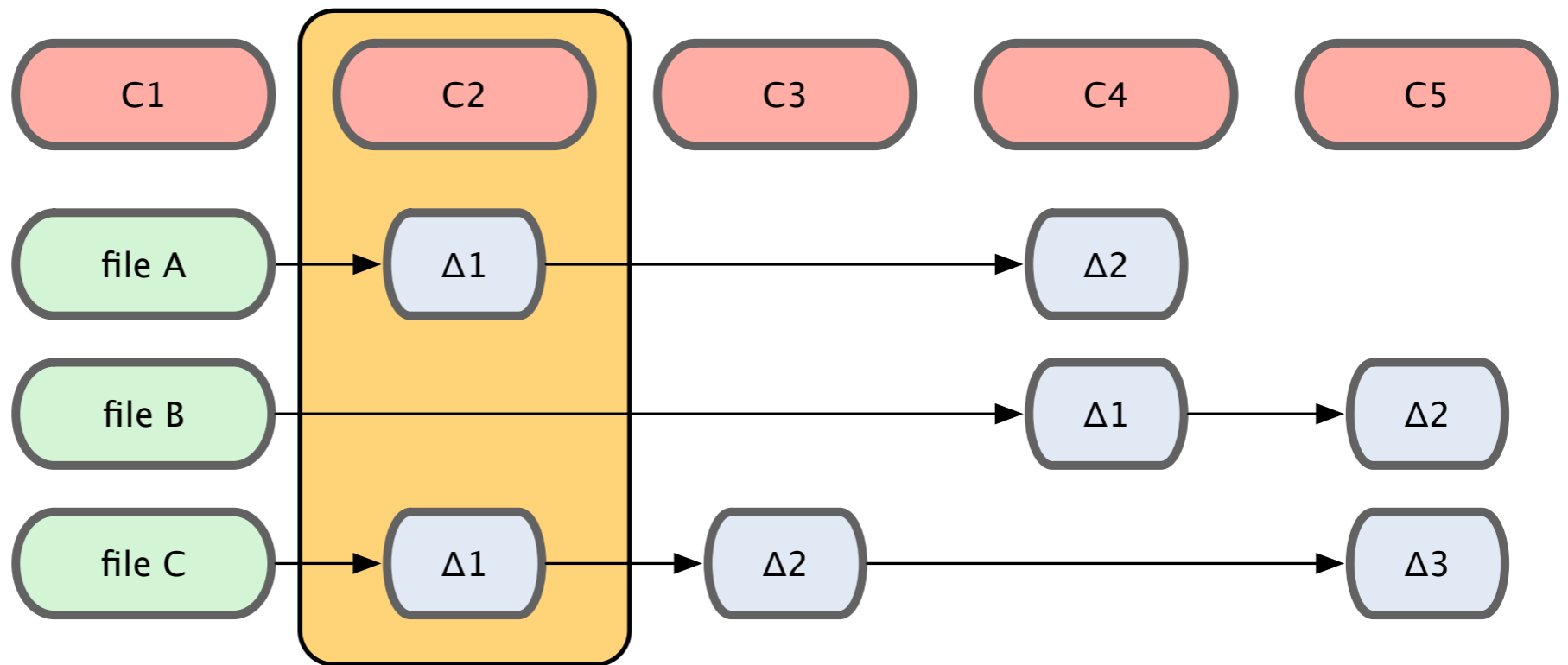
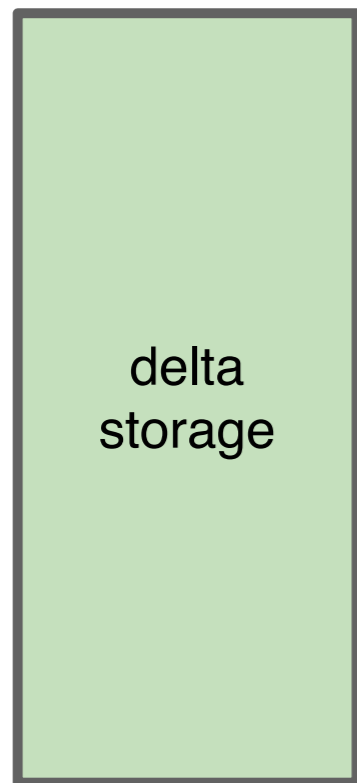
49e11..

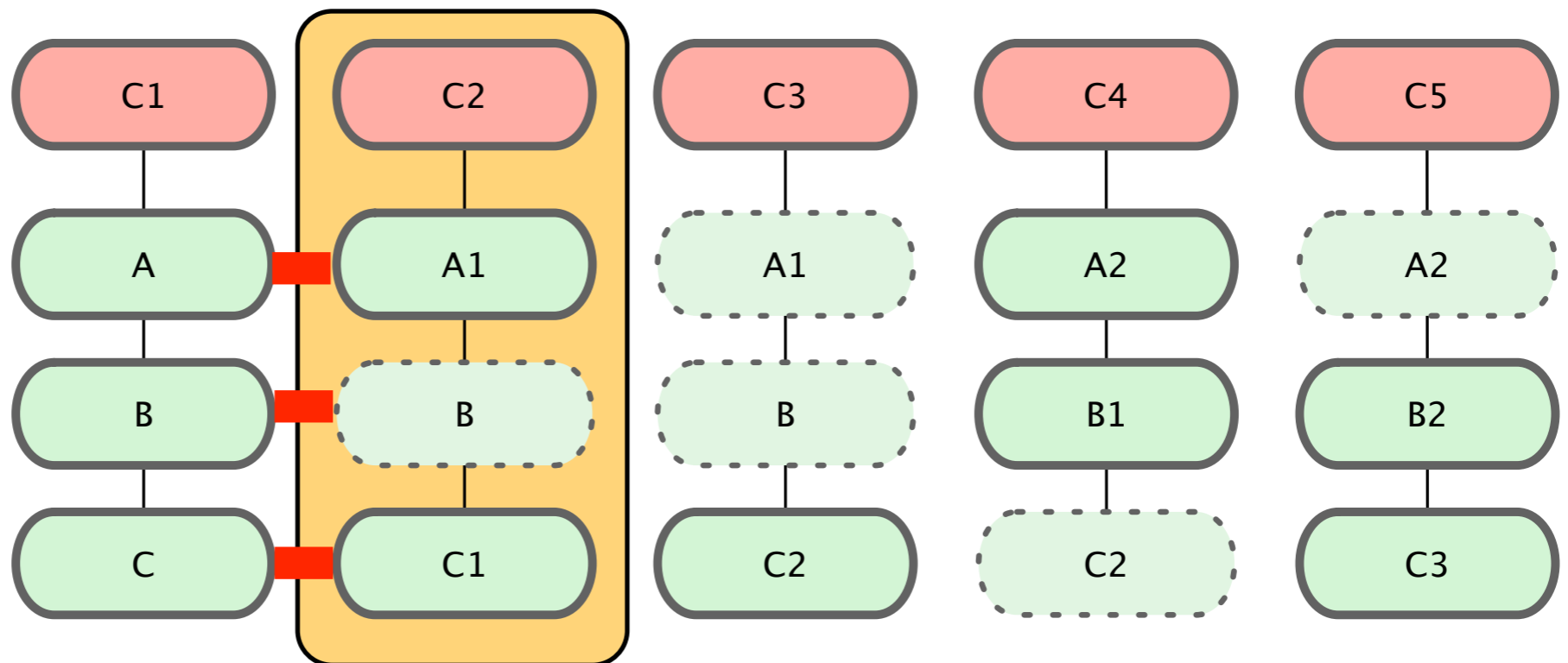
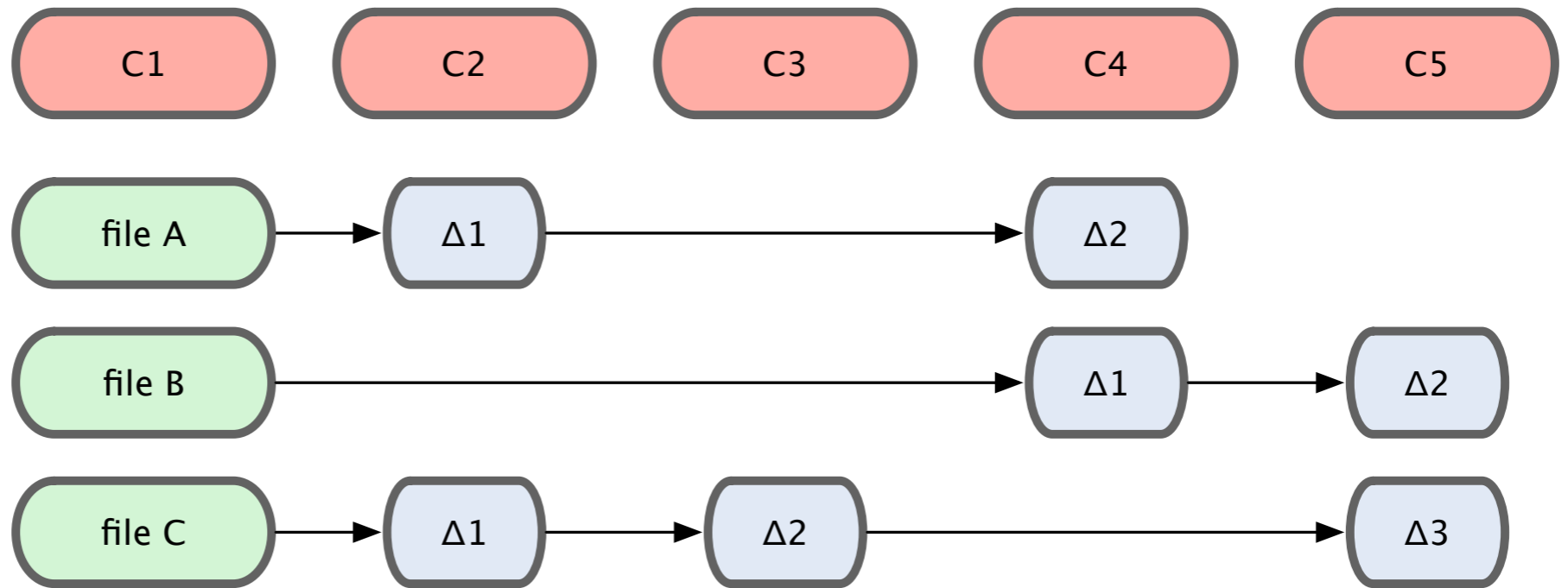
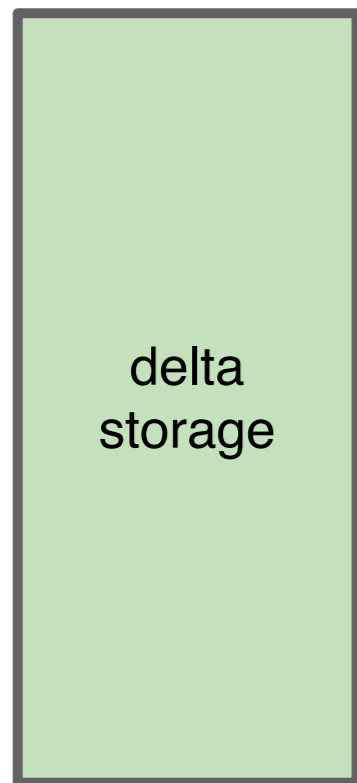
tag	size
object	ae668
type	commit
tagger	Scott
tag	v1.3
my tag message that explains this tag	



Changes







git diff

```
diff --git a/main.py b/main.py
index 6db8b97..b9bcc62 100755
--- a/main.py
+++ b/main.py
@@ -2,10 +2,12 @@
 import wsgiref.handlers
 from google.appengine.ext import webapp

+# this program prints out 'hello world'
+
 class MainHandler(webapp.RequestHandler):

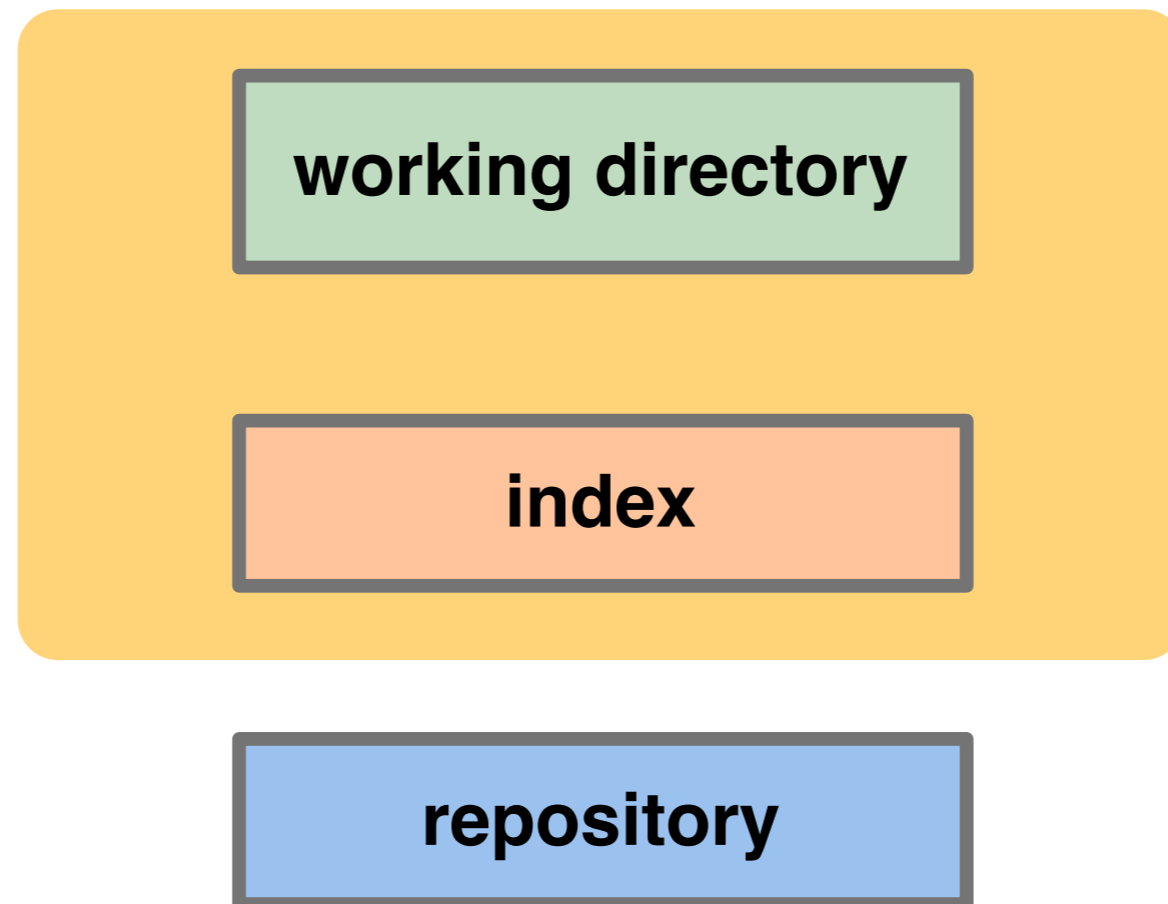
     def get(self):
-        self.response.out.write('Hello world!')
+        self.response.out.write('Hola mundo!')

 def main():
     application = webapp.WSGIApplication([('/', MainHandler)],
```

What is not yet staged?

git diff

git diff



What is staged?


```
git diff --cached
```

git diff --cached

working directory

index

repository

Unified Diff

```
git diff > change.patch
```

```
git diff > change.patch
```

```
patch -p1 < change.patch
```

```
git diff > change.patch
```

```
patch -p1 < change.patch
```

or

```
git apply change.patch
```

History

git log


```
$>git log_
```

```
$>git log
```

```
commit 310154e3c7db47d8bac935c2c43aee6afac11aae
```

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

```
Date: Sun Apr 13 10:49:15 2008 -0700
```

```
updated README formatting and added blame
```

```
commit f7f3f6dd8fd3fa40f052427c32785a0fa01aaa5f
```

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

```
Date: Sun Apr 13 10:45:01 2008 -0700
```

```
changed my name a bit
```

```
commit 710f0f8d2cdf5af87033b9ec08859a505f9a6af5
```

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

```
Date: Sun Apr 13 10:34:16 2008 -0700
```

```
added ls-files
```

```
commit c110d7ff8cfb86fd5cce9a8aee462678dbb4ef9b
```

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

```
Date: Sun Apr 6 12:13:36 2008 -0700
```

```
made the ls-tree function recursive and list trees
```

```
$>git log
```

```
commit 310154e3c7db47d8bac935c2c43aee6afac11aae
```

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

```
Date: Sun Apr 13 10:49:15 2008 -0700
```

```
updated README formatting and added blame
```

```
commit f7f3f6dd8fd3fa40f052427c32785a0fa01aaa5f
```

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

```
Date: Sun Apr 13 10:45:01 2008 -0700
```

```
changed my name a bit
```

```
commit 710f0f8d2cdf5af87033b9ec08859a505f9a6af5
```

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

```
Date: Sun Apr 13 10:34:16 2008 -0700
```

```
added ls-files
```

```
commit c110d7ff8cfb86fd5cce9a8aee462678dbb4ef9b
```

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

```
Date: Sun Apr 6 12:13:36 2008 -0700
```

```
made the ls-tree function recursive and list trees
```

```
$>git log
```

```
commit 310154e3c7db47d8bac935c2c43aee6afac11aae
```

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

```
Date: Sun Apr 13 10:49:15 2008 -0700
```

```
updated README formatting and added blame
```

```
commit f7f3f6dd8fd3fa40f052427c32785a0fa01aaa5f
```

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

```
Date: Sun Apr 13 10:45:01 2008 -0700
```

```
changed my name a bit
```

```
commit 710f0f8d2cdf5af87033b9ec08859a505f9a6af5
```

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

```
Date: Sun Apr 13 10:34:16 2008 -0700
```

```
added ls-files
```

```
commit c110d7ff8cfb86fd5c9a8aee462678dbb4ef9b
```

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

```
Date: Sun Apr 6 12:13:36 2008 -0700
```

```
made the ls-tree function recursive and list trees
```

```
$>git log
```

```
commit 310154e3c7db47d8bac935c2c43aee6afac11aae
```

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

```
Date: Sun Apr 13 10:49:15 2008 -0700
```

```
updated README formatting and added blame
```

```
commit f7f3f6dd8fd3fa40f052427c32785a0fa01aaa5f
```

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

```
Date: Sun Apr 13 10:45:01 2008 -0700
```

```
changed my name a bit
```

```
commit 710f0f8d2cdf5af87033b9ec08859a505f9a6af5
```

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

```
Date: Sun Apr 13 10:34:16 2008 -0700
```

```
added ls-files
```

```
commit c110d7ff8cfb86fd5cce9a8aee462678dbb4ef9b
```

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

```
Date: Sun Apr 6 12:13:36 2008 -0700
```

```
made the ls-tree function recursive and list trees
```

```
$>git log
```

```
commit 310154e3c7db47d8bac935c2c43aee6afac11aae
```

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

```
Date: Sun Apr 13 10:49:15 2008 -0700
```

```
updated README formatting and added blame
```

```
commit f7f3f6dd8fd3fa40f052427c32785a0fa01aaa5f
```

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

```
Date: Sun Apr 13 10:45:01 2008 -0700
```

```
changed my name a bit
```

```
commit 710f0f8d2cdf5af87033b9ec08859a505f9a6af5
```

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

```
Date: Sun Apr 13 10:34:16 2008 -0700
```

```
added ls-files
```

```
commit c110d7ff8cfb86fd5cce9a8aee462678dbb4ef9b
```

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

```
Date: Sun Apr 6 12:13:36 2008 -0700
```

```
made the ls-tree function recursive and list trees
```

```
$>git log
```

```
commit 310154e3c7db47d8bac935c2c43aee6afac11aae
```

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

```
Date: Sun Apr 13 10:49:15 2008 -0700
```

```
updated README formatting and added blame
```

```
commit f7f3f6dd8fd3fa40f052427c32785a0fa01aaa5f
```

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

```
Date: Sun Apr 13 10:45:01 2008 -0700
```

```
changed my name a bit
```

```
commit 710f0f8d2cdf5af87033b9ec08859a505f9a6af5
```

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

```
Date: Sun Apr 13 10:34:16 2008 -0700
```

```
added ls-files
```

```
commit c110d7ff8cfb86fd5cce9a8aee462678dbb4ef9b
```

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

```
Date: Sun Apr 6 12:13:36 2008 -0700
```

```
made the ls-tree function recursive and list trees
```

```
$>git log
```

```
commit 310154e3c7db47d8bac935c2c43aee6afac11aae
```

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

```
Date: Sun Apr 13 10:49:15 2008 -0700
```

```
updated README formatting and added blame
```

```
commit f7f3f6dd8fd3fa40f052427c32785a0fa01aaa5f
```

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

```
Date: Sun Apr 13 10:45:01 2008 -0700
```

```
changed my name a bit
```

```
commit 710f0f8d2cdf5af87033b9ec08859a505f9a6af5
```

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

```
Date: Sun Apr 13 10:34:16 2008 -0700
```

```
added ls-files
```

```
commit c110d7ff8cfb86fd5c9a8aee462678dbb4ef9b
```

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

```
Date: Sun Apr 6 12:13:36 2008 -0700
```

```
made the ls-tree function recursive and list trees
```


**What am I about to
submit?**

```
git log [branch] ..
```

```
git log origin/master..
```

```
git log origin/master..
```

what you've committed that isn't pushed yet

Stashing

**when a commit is
too much**

```
$ git status
```

```
$ git status
# On branch iss53
# 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/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```



```
$ git status
# On branch iss53
# 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/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")

$ git stash
```

```
$ git status
# On branch iss53
# 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/alarmlock/AlarmAlert.java
#   modified:   src/com/android/alarmlock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash
Saved working directory and index state "WIP on iss53: c8208f9...
Merge branch 'cupcake'"
HEAD is now at c8208f9 Merge branch 'cupcake'
(To restore them type "git stash apply")
```

```
$ git status
# On branch iss53
# 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/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash
Saved working directory and index state "WIP on iss53: c8208f9...
Merge branch 'cupcake'"
HEAD is now at c8208f9 Merge branch 'cupcake'
(To restore them type "git stash apply")
```

```
$ git status
# On branch iss53
nothing to commit (working directory clean)
```

```
$ git status
# On branch iss53
# 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/alarmlock/AlarmAlert.java
#   modified:   src/com/android/alarmlock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash
Saved working directory and index state "WIP on iss53: c8208f9...
Merge branch 'cupcake'"
HEAD is now at c8208f9 Merge branch 'cupcake'
(To restore them type "git stash apply")
```

```
$ git status
# On branch iss53
# 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/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash
Saved working directory and index state "WIP on iss53: c8208f9...
Merge branch 'cupcake'"
HEAD is now at c8208f9 Merge branch 'cupcake'
(To restore them type "git stash apply")
```

```
$ git status
```

```
$ git status
# On branch iss53
# 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/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash
Saved working directory and index state "WIP on iss53: c8208f9...
Merge branch 'cupcake'"
HEAD is now at c8208f9 Merge branch 'cupcake'
(To restore them type "git stash apply")
```

```
$ git status
# On branch iss53
nothing to commit (working directory clean)
```

```
$ git status
# On branch iss53
# 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/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash
Saved working directory and index state "WIP on iss53: c8208f9...
Merge branch 'cupcake'"
HEAD is now at c8208f9 Merge branch 'cupcake'
(To restore them type "git stash apply")
```

```
$ git status
# On branch iss53
nothing to commit (working directory clean)
```

```
$ git stash list
```



```
$ git stash list
```

```
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash list
```

```
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash apply
```

```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'

$ git stash apply
# On branch iss53
# 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/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash apply
# On branch iss53
# 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/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash list
```

```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash apply
# On branch iss53
# 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/alarmsclock/AlarmAlert.java
#   modified:   src/com/android/alarmsclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash apply
# On branch iss53
# 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/alarmsclock/AlarmAlert.java
#   modified:   src/com/android/alarmsclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash drop stash@{0}
```

```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash apply
# On branch iss53
# 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/alarmclock/AlarmAlert.java
#   modified:   src/com/android/alarmclock/AlarmAlertWakeLock.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git stash list
stash@{0}: WIP on iss53: c8208f9... Merge branch 'cupcake'
```

```
$ git stash drop stash@{0}
Dropped stash@{0} (e19b0e94032d0614f450238f91953fe5e7e454a8)
```

git stash

save

list

drop

pop

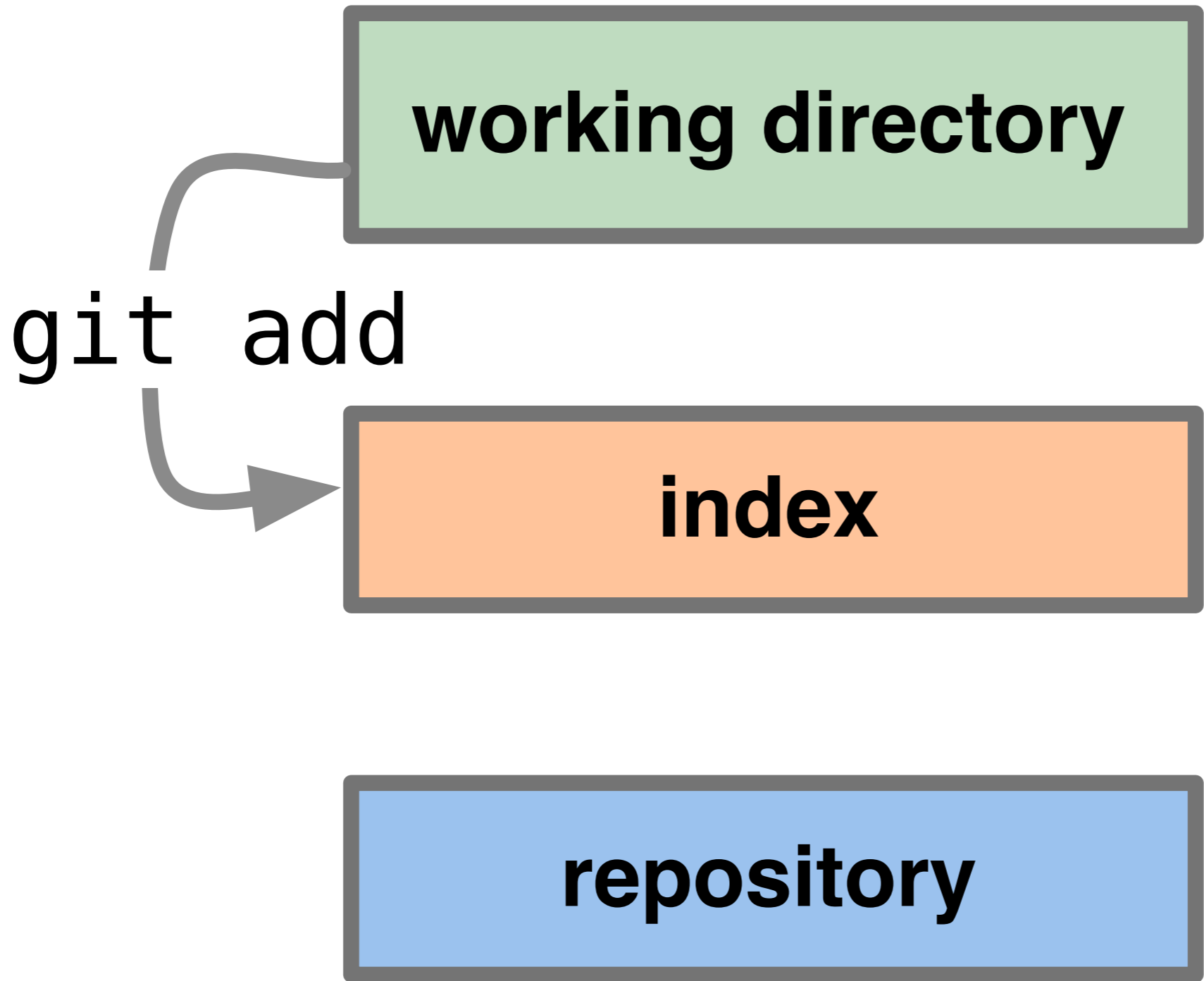
apply

Bulk Staging

working directory

index

repository

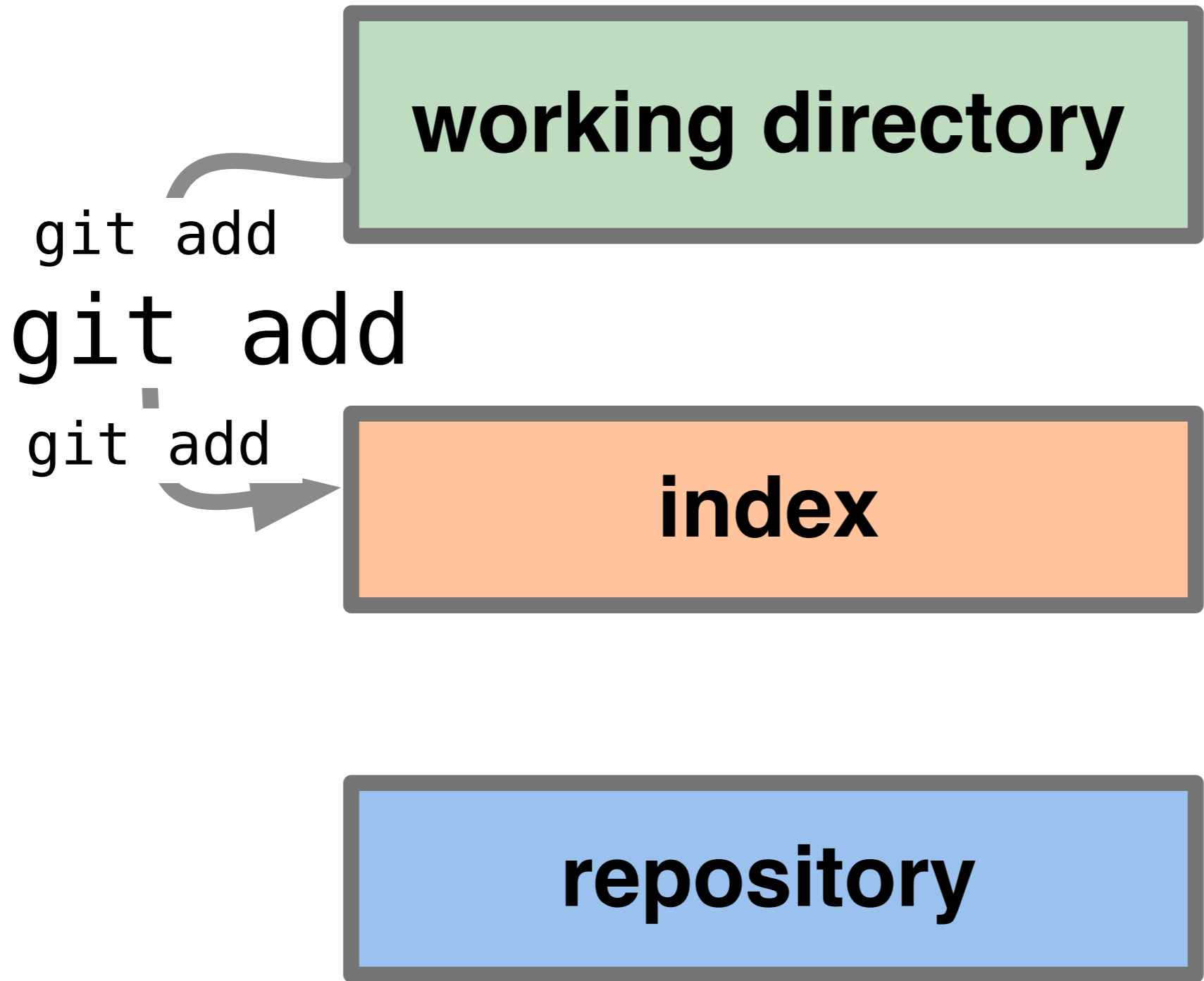


working directory

`git add`

index

repository



working directory

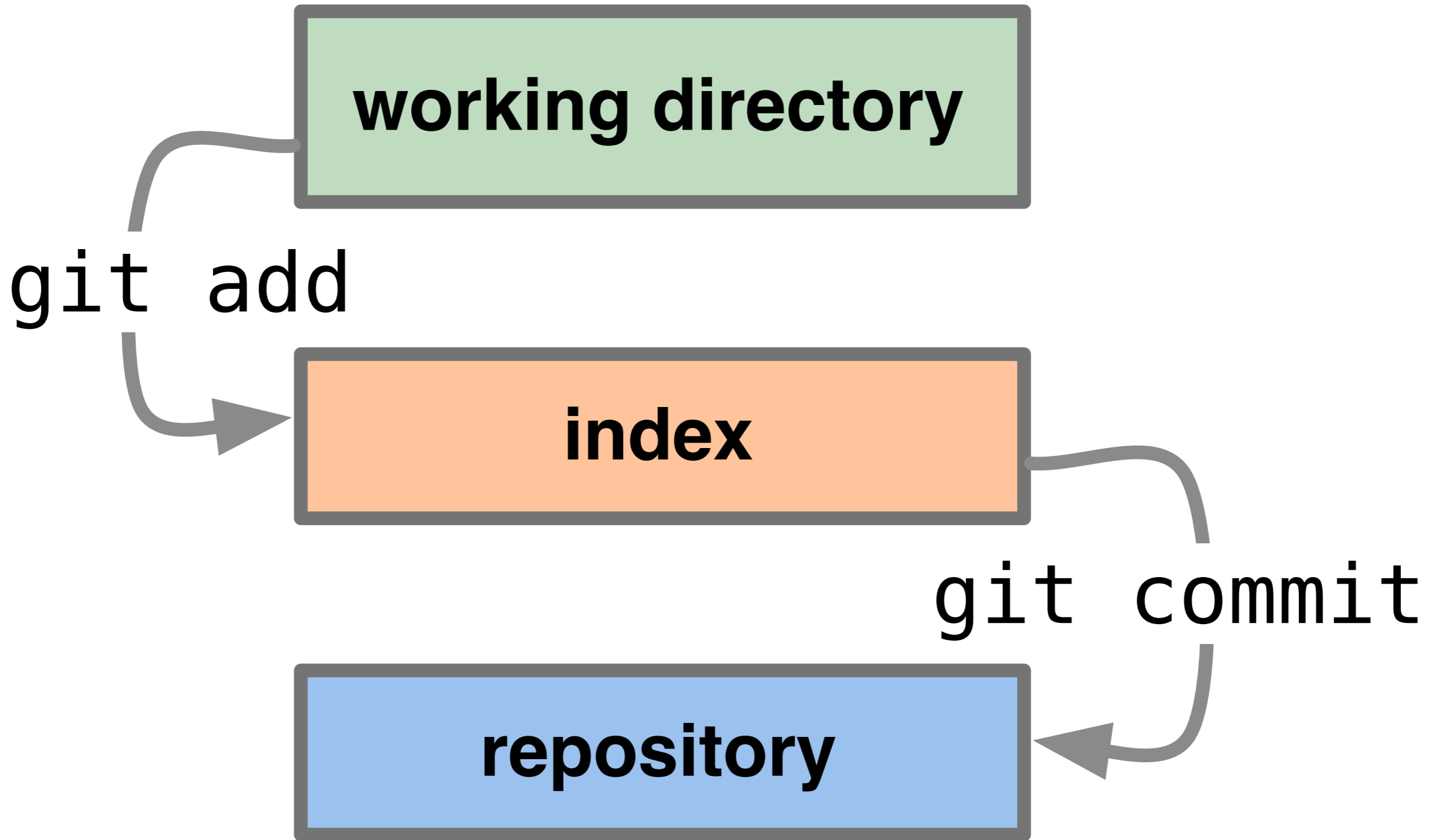
git add

git add

git add

index

repository



working directory

`git add`

index

`git commit`

repository

```
git add -u
```

```
git commit -a
```

working directory

index

repository

`git commit -a`

`git commit -a`

`git add -u`
`git commit`

Git Help

git [command] --help

git help [command]

git-scm.com

git-scm.com

The image shows the top portion of the git-scm.com website. At the top left is the Git logo, which consists of three red plus signs followed by the word "git" in a green, lowercase, sans-serif font. To the right of the logo, the word "git" is written in a large, bold, black, lowercase font, followed by the tagline "the fast version control system" in a smaller, grey, lowercase font. The background of this header section features a stylized illustration of a forest with green trees and a brown ground. On the right side of the header, there is a cartoon character that looks like a piece of bread or a bun, with a smiling face, wearing a green leafy hat, and holding a small green object. Below the header is a green navigation bar with several links: "Home", "About Git", "Documentation", "Download", "Tools & Hosting", and "Wiki". The "Documentation" link is circled in black. Below the navigation bar, there are three main content areas: "Git is..." on the left, "Projects using Git" in the middle, and "Download Git" on the right. The "Git is..." section contains text describing Git as an open source, distributed version control system. The "Projects using Git" section lists several projects: Git, Linux Kernel, and Perl. The "Download Git" section states that the latest stable release is v1.6.1.1 and provides a link to the release notes.

+++ git **git** the fast version control system

[Home](#) [About Git](#) [Documentation](#) [Download](#) [Tools & Hosting](#) [Wiki](#)

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.

Projects using Git


- [Git](#)
- [Linux Kernel](#)
- [Perl](#)

Download Git

The latest stable Git release is

v1.6.1.1

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

 [Git Community Book](#) : The git-scm.com community-built comprehensive online book

Tutorials

Short and Sweet

The [official Git tutorial](#) is a good place to get started.

[Everyday Git](#) in 20 commands is good for a useful minimum set of commands.

The [SVN Crash Course](#) might be helpful if you're coming from the SVN world.

[Git for the lazy](#) is a great guide for beginners.

Longer, More In Depth

[Git for Designers](#) – No knowledge of version control? No problem.

[Git for Computer Scientists](#) – A quick introduction to git internals for people who are not scared by words like Directed Acyclic Graph.

The [Git User's Manual](#) is a comprehensive resource, covering a lot of Git functionality.

[Git Magic](#) – An alternative online book with the source [online](#).

[The GitHub Guides](#) – Guides on a variety of Git and GitHub related topics

Reference

The official and comprehensive [reference manual](#) comes as part of the Git package itself

[Visual Git Cheat Sheet](#) – Everyone loves to cheat

[37signals' Git Resources](#) – Useful lightweight reference to keep handy

Books

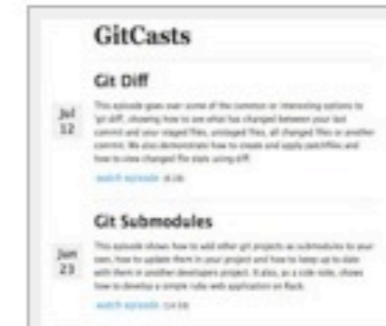
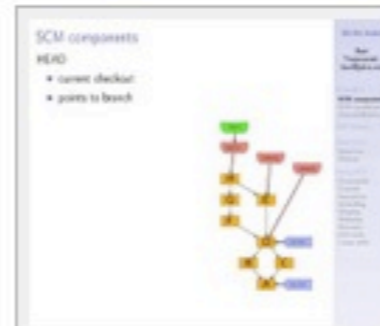



[Git Internals PDF](#)
by Scott Chacon



[Pragmatic Version Control Using Git](#)
by Travis Swicegood

Videos



 [Git Community Book](#) : The git-scm.com community-built comprehensive online book

Tutorials

Short and Sweet

The [official Git tutorial](#) is a good place to get started.

[Everyday Git](#) in 20 commands is good for a useful minimum set of commands.

The [SVN Crash Course](#) might be helpful if you're coming from the SVN world.

[Git for the lazy](#) is a great guide for beginners.

Longer, More In Depth

[Git for Designers](#) – No knowledge of version control? No problem.

[Git for Computer Scientists](#) – A quick introduction to git internals for people who are not scared by words like Directed Acyclic Graph.

The [Git User's Manual](#) is a comprehensive resource, covering a lot of Git functionality.

[Git Magic](#) – An alternative online book with the source [online](#).

The [GitHub Guides](#) – Guides on a variety of Git and GitHub related topics

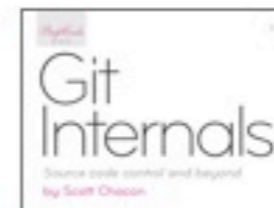
Reference

The official and comprehensive [reference manual](#) comes as part of the Git package itself

[Visual Git Cheat Sheet](#) – Everyone loves to cheat

[37signals' Git Resources](#) – Useful lightweight reference to keep handy

Books

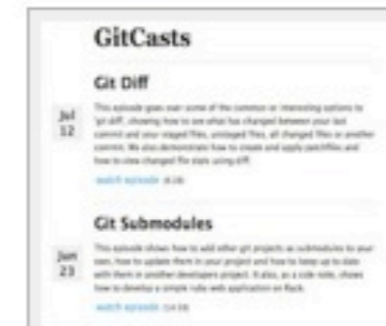
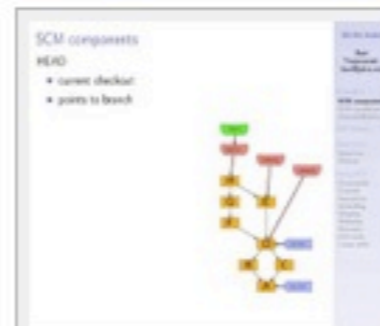



[Git Internals PDF](#)
by Scott Chacon



[Pragmatic Version Control Using Git](#)
by Travis Swicegood

Videos



 [Git Community Book](#) : The git-scm.com community-built comprehensive online book

Tutorials

Short and Sweet

The [official Git tutorial](#) is a good place to get started.

[Everyday Git](#) in 20 commands is good for a useful minimum set of commands.

The [SVN Crash Course](#) might be helpful if you're coming from the SVN world.

[Git for the lazy](#) is a great guide for beginners.

Longer, More in Depth

[Git for Designers](#) – No knowledge of version control? No problem.

[Git for Computer Scientists](#) – A quick introduction to git internals for people who are not scared by words like Directed Acyclic Graph.

The [Git User's Manual](#) is a comprehensive resource, covering a lot of Git functionality.

[Git Magic](#) – An alternative online book with the source [online](#).

[The GitHub Guides](#) – Guides on a variety of Git and GitHub related topics

Reference

The official and comprehensive [reference manual](#) comes as part of the Git package itself

[Visual Git Cheat Sheet](#) – Everyone loves to cheat

[37signals' Git Resources](#) – Useful lightweight reference to keep handy

Books

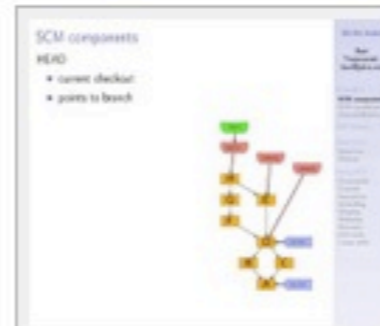



[Git Internals PDF](#)
by Scott Chacon



[Pragmatic Version Control Using Git](#)
by Travis Swicegood

Videos



 [Git Community Book](#) : The git-scm.com community-built comprehensive online book

Tutorials

Short and Sweet

The [official Git tutorial](#) is a good place to get started.

[Everyday Git](#) in 20 commands is good for a useful minimum set of commands.

The [SVN Crash Course](#) might be helpful if you're coming from the SVN world.

[Git for the lazy](#) is a great guide for beginners.

Longer, More In Depth

[Git for Designers](#) – No knowledge of version control? No problem.

[Git for Computer Scientists](#) – A quick introduction to git internals for people who are not scared by words like Directed Acyclic Graph.

The [Git User's Manual](#) is a comprehensive resource, covering a lot of Git functionality.

[Git Magic](#) – An alternative online book with the source [online](#).

[The GitHub Guides](#) – Guides on a variety of Git and GitHub related topics

Reference

The official and comprehensive [reference manual](#) comes as part of the Git package itself

[Visual Git Cheat Sheet](#) – Everyone loves to cheat

[37signals' Git Resources](#) – Useful lightweight reference to keep handy

Books

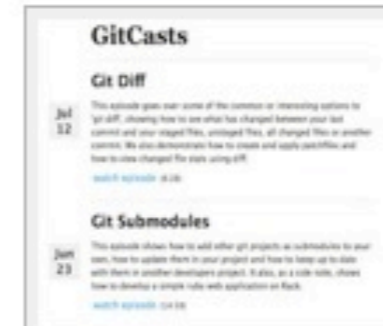
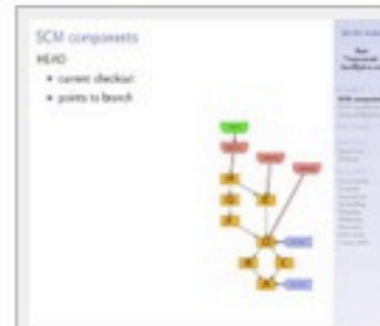



[Git Internals PDF](#)
by Scott Chacon



[Pragmatic Version Control Using Git](#)
by Travis Swicegood

videos



 [Git Community Book](#) : The git-scm.com community-built comprehensive online book

Tutorials

Short and Sweet

The [official Git tutorial](#) is a good place to get started.

[Everyday Git](#) in 20 commands is good for a useful minimum set of commands.

The [SVN Crash Course](#) might be helpful if you're coming from the SVN world.

[Git for the lazy](#) is a great guide for beginners.

Longer, More In Depth

[Git for Designers](#) – No knowledge of version control? No problem.

[Git for Computer Scientists](#) – A quick introduction to git internals for people who are not scared by words like Directed Acyclic Graph.

The [Git User's Manual](#) is a comprehensive resource, covering a lot of Git functionality.

[Git Magic](#) – An alternative online book with the source [online](#).

The [GitHub Guides](#) – Guides on a variety of Git and GitHub related topics

Reference

The official and comprehensive [reference manual](#) comes as part of the Git package itself

[Visual Git Cheat Sheet](#) – Everyone loves to cheat

[37signals' Git Resources](#) – Useful lightweight reference to keep handy

BOOKS

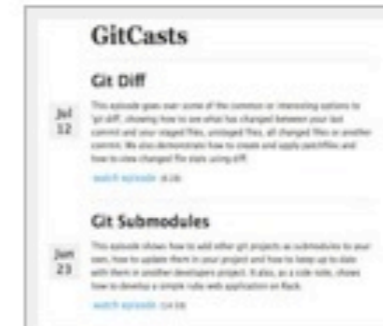
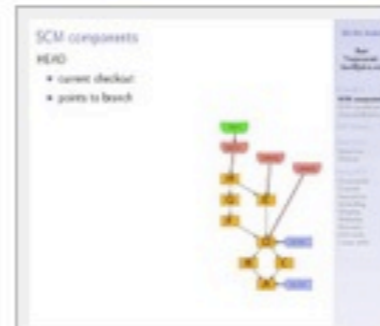



[Git Internals PDF](#)
by Scott Chacon



[Pragmatic Version Control Using Git](#)
by Travis Swicegood

Videos



 [Git Community Book](#) : The git-scm.com community-built comprehensive online book

Tutorials

Short and Sweet

The [official Git tutorial](#) is a good place to get started.

[Everyday Git](#) in 20 commands is good for a useful minimum set of commands.

The [SVN Crash Course](#) might be helpful if you're coming from the SVN world.

[Git for the lazy](#) is a great guide for beginners.

Longer, More In Depth

[Git for Designers](#) – No knowledge of version control? No problem.

[Git for Computer Scientists](#) – A quick introduction to git internals for people who are not scared by words like Directed Acyclic Graph.

The [Git User's Manual](#) is a comprehensive resource, covering a lot of Git functionality.

[Git Magic](#) – An alternative online book with the source [online](#).

The GitHub Guides – Guides on a variety of Git and GitHub related topics

Reference

The official and comprehensive [reference manual](#) comes as part of the Git package itself

[Visual Git Cheat Sheet](#) – Everyone loves to cheat

[37signals' Git Resources](#) – Useful lightweight reference to keep handy

Books

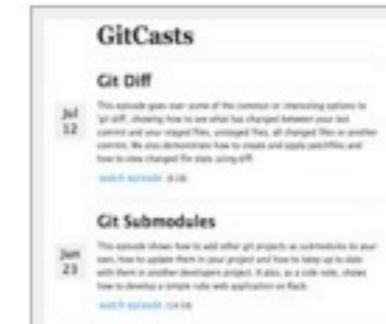
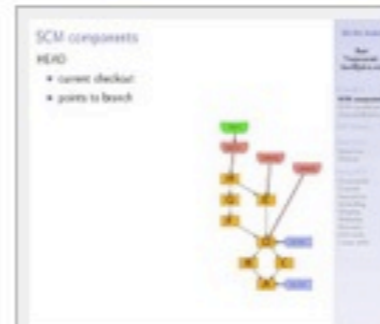



[Git Internals PDF](#)
by Scott Chacon



[Pragmatic Version Control Using Git](#)
by Travis Swicegood

Videos



 [Git Community Book](#) : The git-scm.com community-built comprehensive online book

Tutorials

Short and Sweet

The [official Git tutorial](#) is a good place to get started.

[Everyday Git](#) in 20 commands is good for a useful minimum set of commands.

The [SVN Crash Course](#) might be helpful if you're coming from the SVN world.

[Git for the lazy](#) is a great guide for beginners.

Longer, More In Depth

[Git for Designers](#) – No knowledge of version control? No problem.

[Git for Computer Scientists](#) – A quick introduction to git internals for people who are not scared by words like Directed Acyclic Graph.

The [Git User's Manual](#) is a comprehensive resource, covering a lot of Git functionality.

[Git Magic](#) – An alternative online book with the source [online](#).

[The GitHub Guides](#) – Guides on a variety of Git and GitHub related topics

Reference

The official and comprehensive [reference manual](#) comes as part of the Git package itself

[Visual Git Cheat Sheet](#) – Everyone loves to cheat

[37signals' Git Resources](#) – Useful lightweight reference to keep handy

Books

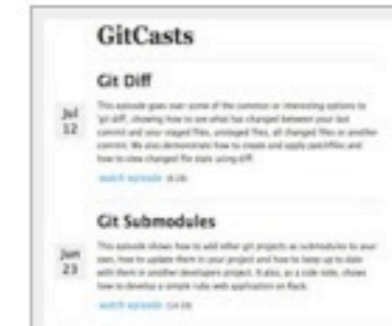
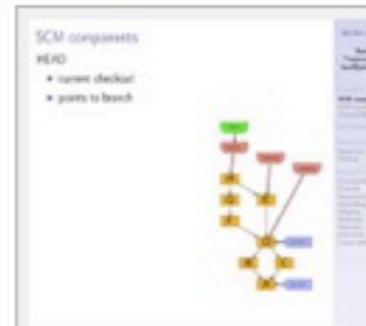


[Git Internals PDF](#)
by Scott Chacon



[Pragmatic Version Control Using Git](#)
by Travis Swicegood

Videos



that's it!

merci

Resources

- git-scm.com
- progit.org
- github.com
- #git / #github on IRC
- gitcasts.com
- schacon@gmail.com

Patricks Irish Pub
33, rue de Montreuil
8pm, tonight

git ninja training

Patch Staging

```
git add -p
```

```
$ git status
```

```
$ git status
# On branch master
# 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:   ticgit.gemspec
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git add -p
```

```
$ git add -p
diff --git a/ticgit.gemspec b/ticgit.gemspec
index 9c32bd4..a44667b 100644
--- a/ticgit.gemspec
+++ b/ticgit.gemspec
@@ -1,7 +1,7 @@
  Gem::Specification.new do |s|
    s.platform = Gem::Platform::RUBY
    s.name = "ticgit"
-   s.version = "0.3.5"
+   s.version = "0.3.6"
    s.date = "2008-05-10"
    s.author = "Scott Chacon"
    s.email = "schacon@gmail.com"
Stage this hunk [y,n,a,d,/ ,j,J,g,e,?]? y
```

```
$ git add -p
diff --git a/ticgit.gemspec b/ticgit.gemspec
index 9c32bd4..a44667b 100644
--- a/ticgit.gemspec
+++ b/ticgit.gemspec
@@ -1,7 +1,7 @@
Gem::Specification.new do |s|
  s.platform = Gem::Platform::RUBY
  s.name = "ticgit"
-  s.version = "0.3.5"
+  s.version = "0.3.6"
  s.date = "2008-05-10"
  s.author = "Scott Chacon"
  s.email = "schacon@gmail.com"
```

```
Stage this hunk [y,n,a,d,/ ,j,J,g,e,?]?
```



```
$ git add -p
diff --git a/ticgit.gemspec b/ticgit.gemspec
index 9c32bd4..a44667b 100644
```

```
--- a/ticgit.gemspec
+++ b/ticgit.gemspec
```

```
@@ -1,7 +1,7 @@
```

```
  Gem::Specification.new do |s|
```

```
    s.platform = Gem::Platform::RUBY
```

```
    s.name = "ticgit"
```

```
-    s.version = "0.3.5"
```

```
+    s.version = "0.3.6"
```

```
    s.date = "2008-05-10"
```

```
    s.author = "Scott Chacon"
```

```
    s.email = "schacon@gmail.com"
```

```
Stage this hunk [y,n,a,d,/ ,j,J,g,e,?]? 
```

```
@@ -9,9 +9,10 @@ Gem::Specification.new do |s|
  s.files = ["lib/ticgit/base.rb", "lib/ticgit",
"lib/ticgit/comment.rb", "lib/ticgit/ticket.rb", "lib/
ticgit.rb", "bin/ti", "bin/ticgitweb"]
```

```
  s.bindir = 'bin'
- s.executables << "ti"
- s.executables << "ticgitweb"
+ s.executables = ["ti", "ticgitweb"]
+ s.default_executable = %q{ti}
  s.homepage = "http://github/schacon/ticgit"
```

```
  s.require_paths = ["lib", "bin"]
+ s.specification_version = 2 if
s.respond_to? :specification_version=
end
```

```
Stage this hunk [y,n,a,d,/ ,K,g,s,e,?]?
```

```
@@ -9,9 +9,10 @@ Gem::Specification.new do |s|
  s.files = ["lib/ticgit/base.rb", "lib/ticgit",
"lib/ticgit/comment.rb", "lib/ticgit/ticket.rb", "lib/
ticgit.rb", "bin/ti", "bin/ticgitweb"]

  s.bindir = 'bin'
- s.executables << "ti"
- s.executables << "ticgitweb"
+ s.executables = ["ti", "ticgitweb"]
+ s.default_executable = %q{ti}
  s.homepage = "http://github/schacon/ticgit"

  s.require_paths = ["lib", "bin"]
+ s.specification_version = 2 if
s.respond_to? :specification_version=
end
```

Stage this hunk [y,n,a,d,/ ,K,g,s,e,?]?

```
@@ -9,9 +9,10 @@ Gem::Specification.new do |s|
  s.files = ["lib/ticgit/base.rb", "lib/ticgit",
"lib/ticgit/comment.rb", "lib/ticgit/ticket.rb", "lib/
ticgit.rb", "bin/ti", "bin/ticgitweb"]

  s.bindir = 'bin'
- s.executables << "ti"
- s.executables << "ticgitweb"
+ s.executables = ["ti", "ticgitweb"]
+ s.default_executable = %q{ti}
  s.homepage = "http://github/schacon/ticgit"

  s.require_paths = ["lib", "bin"]
+ s.specification_version = 2 if
s.respond_to? :specification_version=
end
```

Stage this hunk [y,n,a,d,/ ,K,g,s,e,?]?

```
$ git status
```

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#   modified:   ticgit.gemspec
#
# 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:   ticgit.gemspec
#
```

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:   ticgit.gemspec ←
#
# 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:   ticgit.gemspec
#
```

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#   modified:   ticgit.gemspec
#
# 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:   ticgit.gemspec ←
```


partially stage files

```
git add -p
```

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)
960decdb (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) "
3bd62c21 (Stephen R. van den Berg       2008-08-14 20:02:20 +0200   23) "
73a7a656 (Jens Axboe                2007-07-27 14:00:29 -0700   24) "
49ba83fb (Jon Loeliger             2006-09-19 20:31:51 -0500   25) "
678dac6b (Tilman Sauerbeck         2006-08-22 19:37:41 +0200   26) "
d9edcbd6 (Junio C Hamano            2006-09-07 01:40:04 -0700   27) "
dd467629 (Jon Loeliger             2006-09-26 09:47:43 -0500   28) "
dd467629 (Jon Loeliger             2006-09-26 09:47:43 -0500   29) "
dd467629 (Jon Loeliger             2006-09-26 09:47:43 -0500   30) "
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 pref
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)
960decdb (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) "
3bd62c21 (Stephen R. van den Berg 2008-08-14 20:02:20 +0200  23) "
73a7a656 (Jens Axboe             2007-07-27 14:00:29 -0700  24) "
49ba83fb (Jon Loeliger            2006-09-19 20:31:51 -0500  25) "
678dac6b (Tilman Sauerbeck       2006-08-22 19:37:41 +0200  26) "
d9edcbd6 (Junio C Hamano          2006-09-07 01:40:04 -0700  27) "
dd467629 (Jon Loeliger            2006-09-26 09:47:43 -0500  28) "
dd467629 (Jon Loeliger            2006-09-26 09:47:43 -0500  29) "
dd467629 (Jon Loeliger            2006-09-26 09:47:43 -0500  30) "
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 pref
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)
960decbb (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) "
3bd62c21 (Stephen R. van den Berg 2008-08-14 20:02:20 +0200 23) "
73a7a656 (Jens Axboe 2007-07-27 14:00:29 -0700 24) "
49ba83fb (Jon Loeliger 2006-09-19 20:31:51 -0500 25) "
678dac6b (Tilman Sauerbeck 2006-08-22 19:37:41 +0200 26) "
d9edcbd6 (Junio C Hamano 2006-09-07 01:40:04 -0700 27) "
dd467629 (Jon Loeliger 2006-09-26 09:47:43 -0500 28) "
dd467629 (Jon Loeliger 2006-09-26 09:47:43 -0500 29) "
dd467629 (Jon Loeliger 2006-09-26 09:47:43 -0500 30) "
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 pref
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)
960decdb (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) "
3bd62c21 (Stephen R. van den Berg 2008-08-14 20:02:20 +0200 23) "
73a7a656 (Jens Axboe 2007-07-27 14:00:29 -0700 24) "
49ba83fb (Jon Loeliger 2006-09-19 20:31:51 -0500 25) "
678dac6b (Tilman Sauerbeck 2006-08-22 19:37:41 +0200 26) "
d9edcbd6 (Junio C Hamano 2006-09-07 01:40:04 -0700 27) "
dd467629 (Jon Loeliger 2006-09-26 09:47:43 -0500 28) "
dd467629 (Jon Loeliger 2006-09-26 09:47:43 -0500 29) "
dd467629 (Jon Loeliger 2006-09-26 09:47:43 -0500 30) "
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 pref
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
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
695dffe2 (Johannes Schindelin       2006-09-28 12:00:35 +0200
695dffe2 (Johannes Schindelin       2006-09-28 12:00:35 +0200
695dffe2 (Johannes Schindelin       2006-09-28 12:00:35 +0200
415e7b87 (Patrick Welche              2007-10-18 18:17:39 +0100
415e7b87 (Patrick Welche              2007-10-18 18:17:39 +0100
415e7b87 (Patrick Welche              2007-10-18 18:17:39 +0100
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
960decdb (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 Rienties            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)
960decdb (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) "
3bd62c21 (Stephen R. van den Berg     2008-08-14 20:02:20 +0200   23) "
73a7a656 (Jens Axboe              2007-07-27 14:00:29 -0700   24) "
49ba83fb (Jon Loeliger             2006-09-19 20:31:51 -0500   25) "
678dac6b (Tilman Sauerbeck        2006-08-22 19:37:41 +0200   26) "
d9edcbd6 (Junio C Hamano            2006-09-07 01:40:04 -0700   27) "
dd467629 (Jon Loeliger             2006-09-26 09:47:43 -0500   28) "
dd467629 (Jon Loeliger             2006-09-26 09:47:43 -0500   29) "
dd467629 (Jon Loeliger             2006-09-26 09:47:43 -0500   30) "
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 pref
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)
```

Bisecting

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 3acb4c2c666ed6cb91cb0b983efe9d50cf8cfbe
```

```
$ 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
```

```
[ecb6e1bc347ccec5f9350d878ce677feb13d3b2] 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
[ecb6e1bc347ccec5f9350d878ce677feb13d3b2] 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
[ecb6e1bc347ccec5f9350d878ce677feb13d3b2] 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
[ecb6e1bc347ccec5f9350d878ce677feb13d3b2] 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
[ecb6e1bc347ccec5f9350d878ce677feb13d3b2] 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
f24d3c6ebcfc639b1a3814550e62d60b8e68a8e4 M config
```

```
$ git bisect start
$ git bisect bad
$ git bisect good 3acb4c2c6666ed6cb91cb0b983efe9d50cf8cfbe
Bisecting: 6 revisions left to test after this
[ecb6e1bc347ccec5f9350d878ce677feb13d3b2] 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
f24d3c6ebcfc639b1a3814550e62d60b8e68a8e4 M config
```

```
$ git bisect reset
```

no mas