

# Git Crash Course

Scott Chacon

# Introduction

# Introduction

Who is this guy?

What will we cover?

How will the day go?

Me

Scott Chacon



**github**  
SOCIAL CODE HOSTING

# github.com/schacon



**schacon**



0

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

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

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

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

😊 **asgit**

😊 **fireeaglet**

😊 **git-lighthouse**

😊 **git-ruby**

😊 **git-source**

😊 **munger**

😊 **ruby-git**

😊 **simplegit**

😊 **ticgit**

# github.com/schacon



**schacon**



0

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

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

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

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

 [asgit](#)


 [fireeaglet](#)

 [git-lighthouse](#)

 [git-ruby](#)

 [git-source](#)

 [munger](#)

 [ruby-git](#)

 [simplegit](#)

 [ticgit](#)



PeepCode  
press

\$9

# Git Internals

*Source code control and beyond*

by Scott Chacon

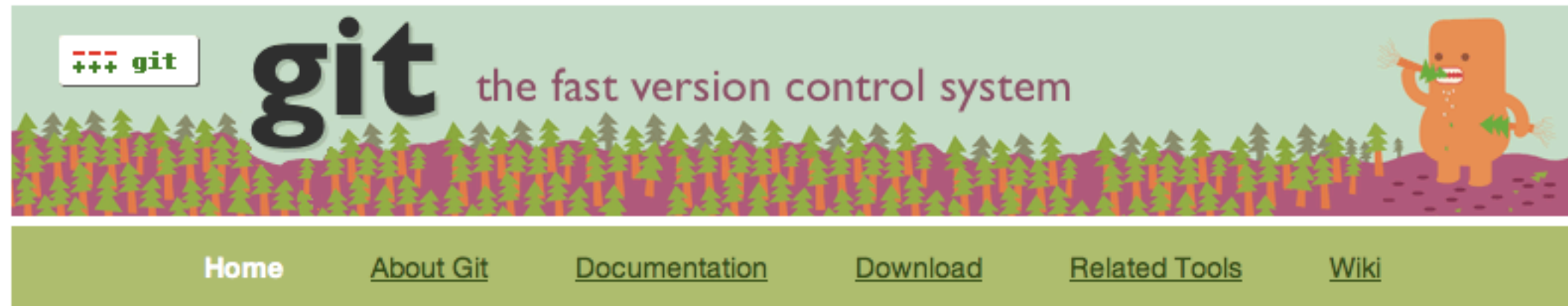
THE EXPERT'S VOICE®

# Pro Git

Patrick Aljord and Scott Chacon

apress®

# git-scm.com



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



# book.git-scm.com



## The Git Community Book

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

If you see anything out of date, have a suggestion on how to improve it, or would like to help add to the book, please see the [How to Contribute](#) page, or just send [our maintainer](#) a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

## Download

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



## New to Git?

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

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

## 1. Introduction

[Welcome to Git](#)

[The Git Object Model](#)

[Git Directory and Working Directory](#)

[The Git Index](#)

## 2. First Time

[Installing Git](#)

## 5. Advanced Git

[Creating New Empty Branches](#)

[Modifying your History](#)

[Advanced Branching And Merging](#)

[Finding Issues - Git Bisect](#)

[Finding Issues - Git Blame](#)

[Git and Email](#)

[Customizing Git](#)

# gitcasts.com

[Blog](#) [About](#) [Episode Index](#)

[Comments RSS](#) [High-Res RSS](#) [iPhone RSS](#)

## GitCasts

### Browsing Git Objects

May  
23

In this episode, I show how to browse and inspect raw Git objects. The major tools covered are the `git cat-file` and `git ls-tree` commands to inspect the object contents, and then I cover some of the included graphical browsers, `gitk` and `gitweb`.

[watch episode](#) (7:26)

### Git Log

May  
19

This episode is on `git-log`, which demonstrates most of the major features and options to the `git-log` command. It includes showing the `stat`, `short-stat` and `name-stat` options, the `--pretty` options, the `since` and `until` limiters, the `path` limiter and author field searching.

[watch episode](#) (6:20)

### Interactive Adding

May  
19

This episode demonstrates how to use the `git interactive add` command. It covers all of the major features of interactive adding, including `status`, `update`, `revert`, `add untracked`, `patch` and `diff`.



To support GitCasts, and to have a nice searchable PDF of a lot of this stuff, please consider buying my [Git Internals book](#) published by Peepcode Press.



[GitHub](#): Social Git hosting, a home for your codes.

[schacon@gmail.com](mailto:schacon@gmail.com)

**What will we cover?**

# What will we cover?

What is Git?

Basic Git Usage

Intermediate Git Usage

Git Branching

Working with Repo & Gerrit

Advanced Git Usage

Git Internals

Git Maintenance

Running a Git Server



How will the day go?

# How will the day go?

What is Git?

Basic Git Usage

Intermediate Git Usage

Git Branching

Working with Repo & Gerrit

Advanced Git Usage

Git Internals

Git Maintenance

Running a Git Server

# How will the day go?

9:00-10:30

What is Git?

Basic Git Usage

Intermediate Git Usage

1:30

Git Branching

Working with Repo & Gerrit

Advanced Git Usage

Git Internals

Git Maintenance

Running a Git Server

# How will the day go?

9:00-10:30

What is Git?

Basic Git Usage

Intermediate Git Usage

Git Branching

Working with Repo & Gerrit

Advanced Git Usage

Git Internals

Git Maintenance

Running a Git Server

1:30

Break (:15)

# How will the day go?

9:00-10:30

What is Git?

Basic Git Usage

Intermediate Git Usage

1:30

Break (:15)

10:45-12:30

Git Branching

Working with Repo & Gerrit

1:30

Advanced Git Usage

Git Internals

Git Maintenance

Running a Git Server

# How will the day go?

9:00-10:30

What is Git?

Basic Git Usage

Intermediate Git Usage

1:30

Break (:15)

10:45-12:30

Git Branching

Working with Repo & Gerrit

1:30

Lunch (1:00)

Advanced Git Usage

Git Internals

Git Maintenance

Running a Git Server

# How will the day go?

9:00-10:30

What is Git?

Basic Git Usage

Intermediate Git Usage

1:30

Break (:15)

10:45-12:30

Git Branching

Working with Repo & Gerrit

1:30

Lunch (1:00)

1:15-3:00

Advanced Git Usage

Git Internals

1:45

Git Maintenance

Running a Git Server

# How will the day go?

9:00-10:30

What is Git?

Basic Git Usage

Intermediate Git Usage

1:30

Break (:15)

10:45-12:30

Git Branching

Working with Repo & Gerrit

1:30

Lunch (1:00)

1:15-3:00

Advanced Git Usage

Git Internals

1:45

Break (:15)

Git Maintenance

Running a Git Server



# How will the day go?

9:00-10:30

What is Git?

Basic Git Usage

Intermediate Git Usage

1:30

Break (:15)

10:45-12:30

Git Branching

Working with Repo & Gerrit

1:30

Lunch (1:00)

1:15-3:00

Advanced Git Usage

Git Internals

1:45

Break (:15)

3:15-5:00

Git Maintenance

Running a Git Server

1:45

# How will the day go?

9:00-10:30

What is Git?

Basic Git Usage

Intermediate Git Usage

1:30

Break (:15)

10:45-12:30

Git Branching

Working with Repo & Gerrit

1:30

Lunch (1:00)

1:15-3:00

Advanced Git Usage

Git Internals

1:45

Break (:15)

3:15-5:00

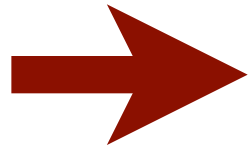
Git Maintenance

Running a Git Server

1:45



# Where are we?



## **What is Git?**

Basic Git Usage

Intermediate Git Usage

Git Branching

Working with Repo & Gerrit

Advanced Git Usage

Git Internals

Git Maintenance


Running a Git Server


# What is Git?

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

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

# git-scm.com

 **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**

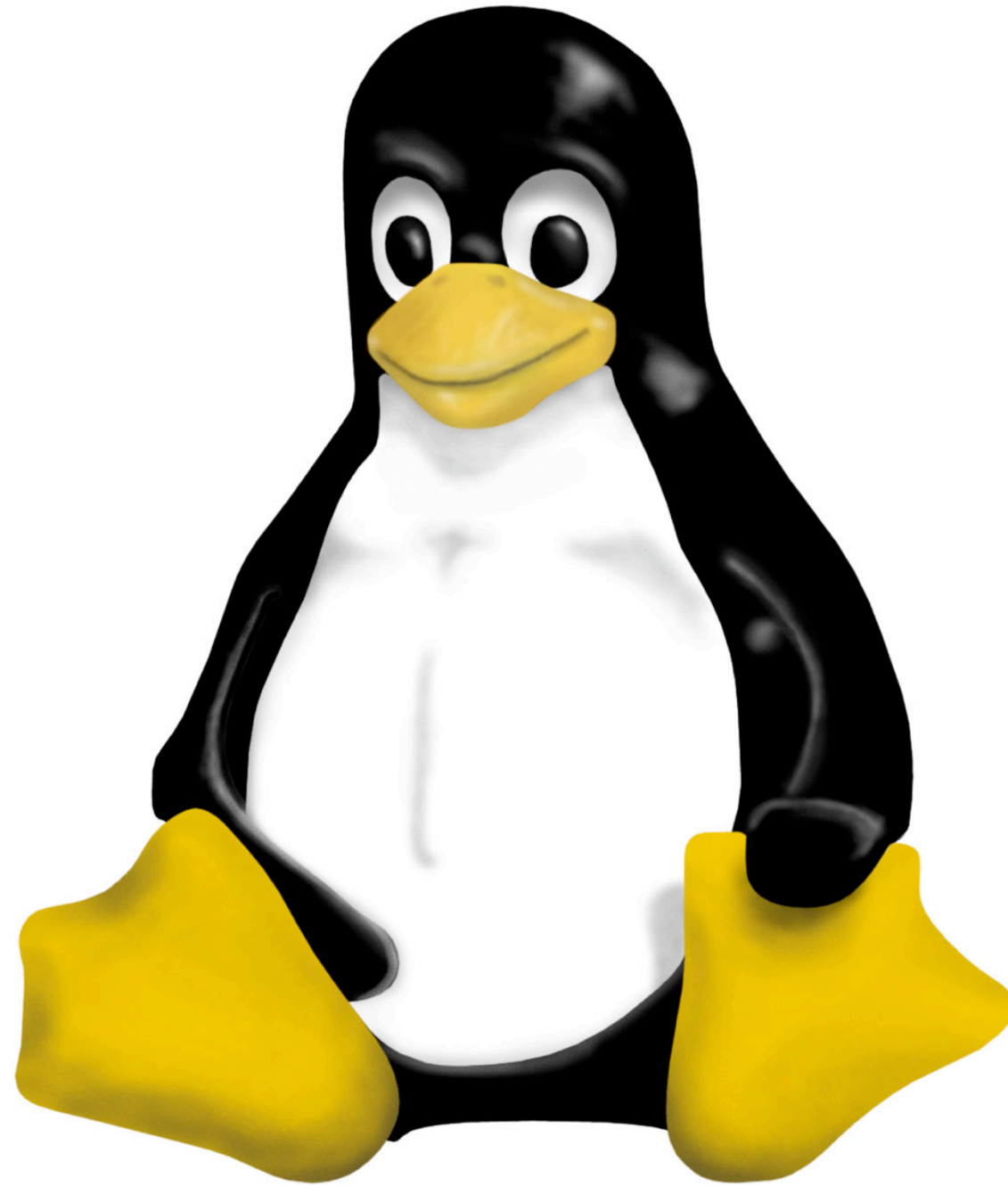
**which means**

everything is fast

every clone is a backup

work offline

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



**Git is Different**

Simple





git object format

“plumbing” commands

git object format

“porcelain” commands

“plumbing” commands

git object format

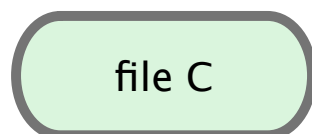
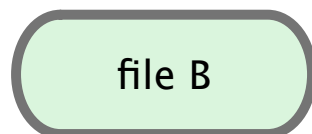
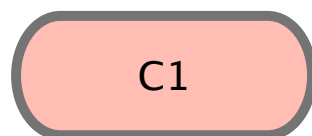
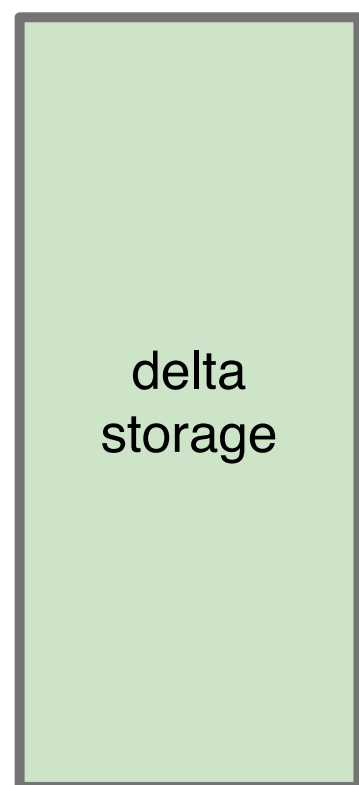
Immutable

**(almost) never removes data**

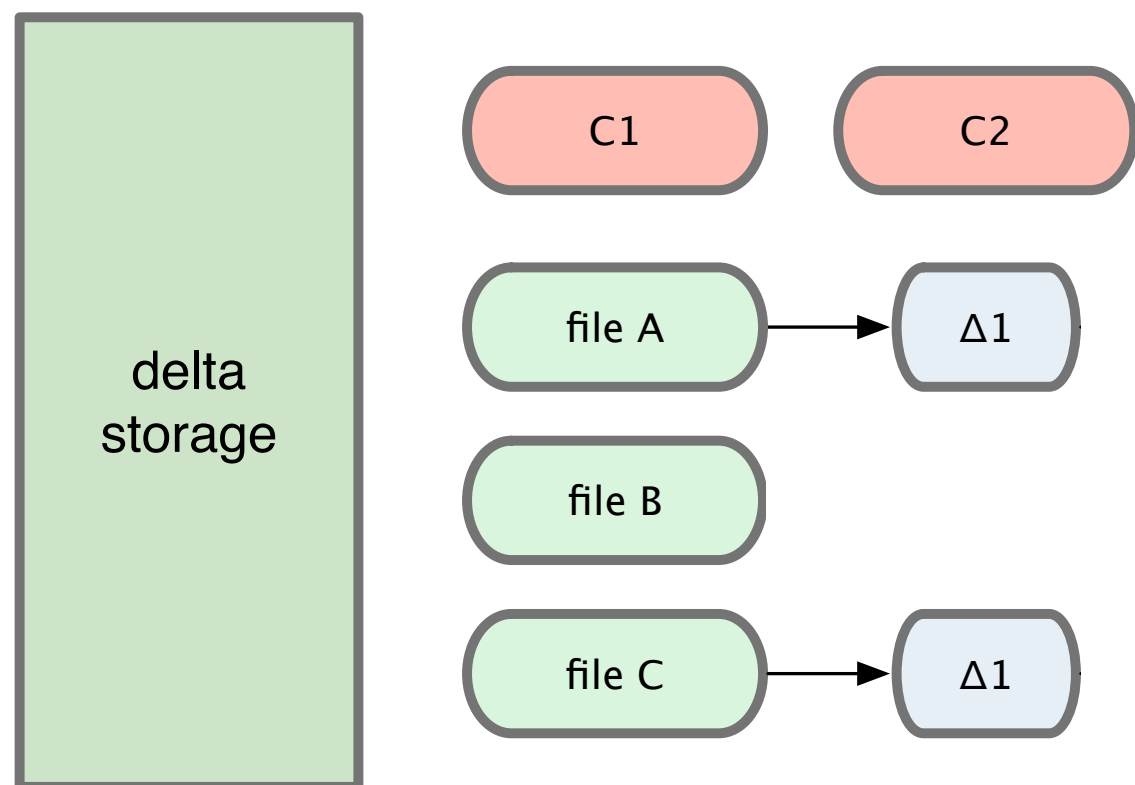
Snapshots, not Patches

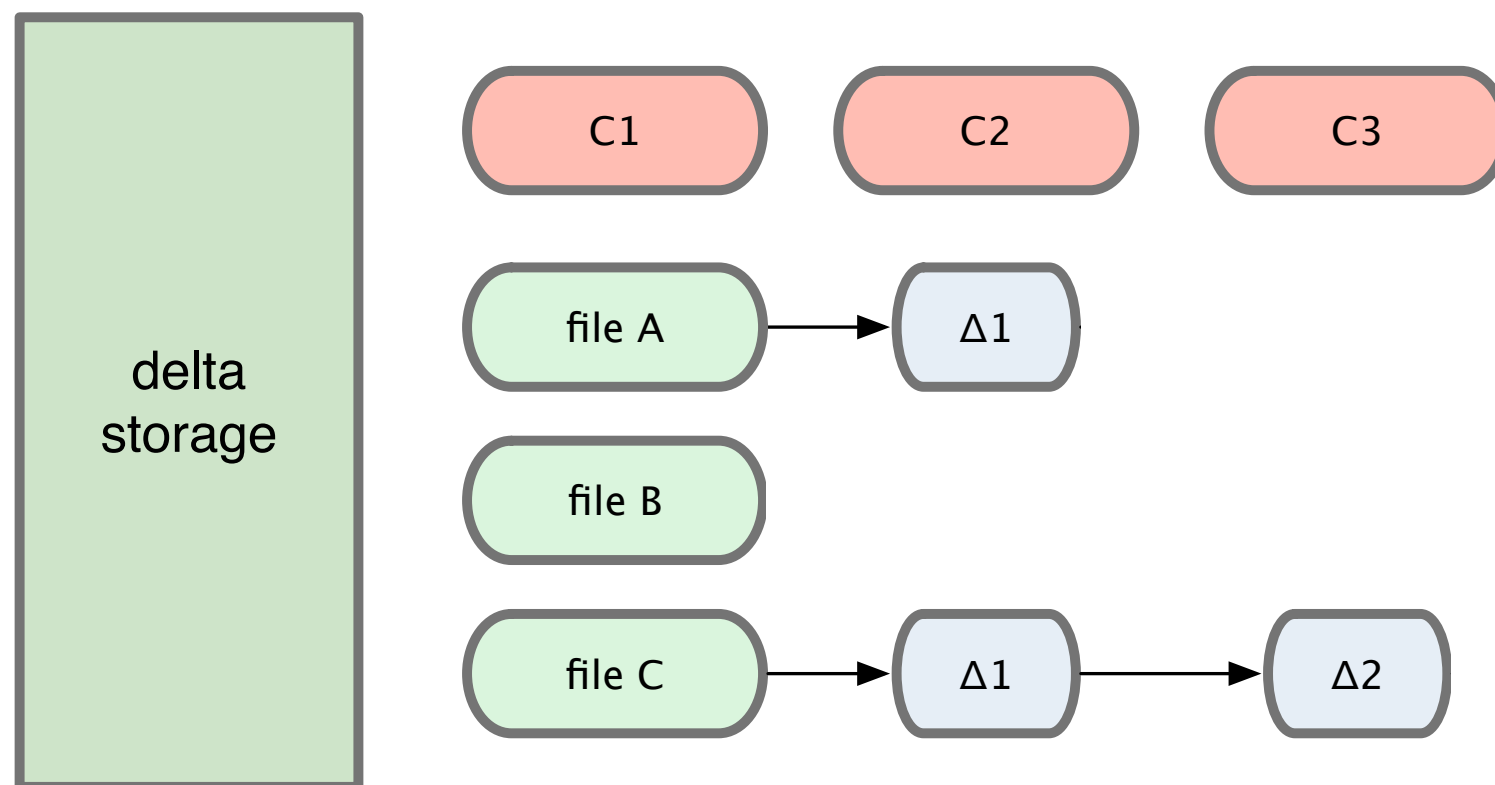


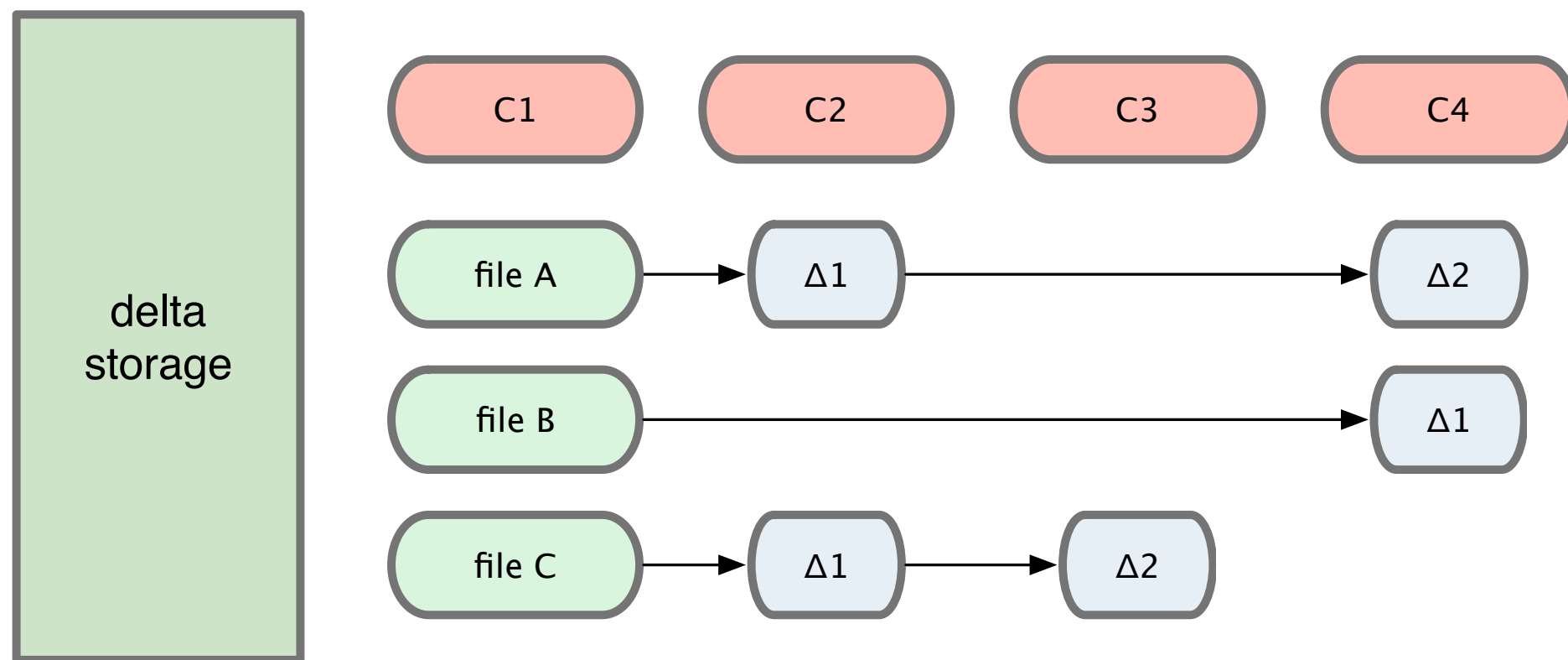
delta  
storage

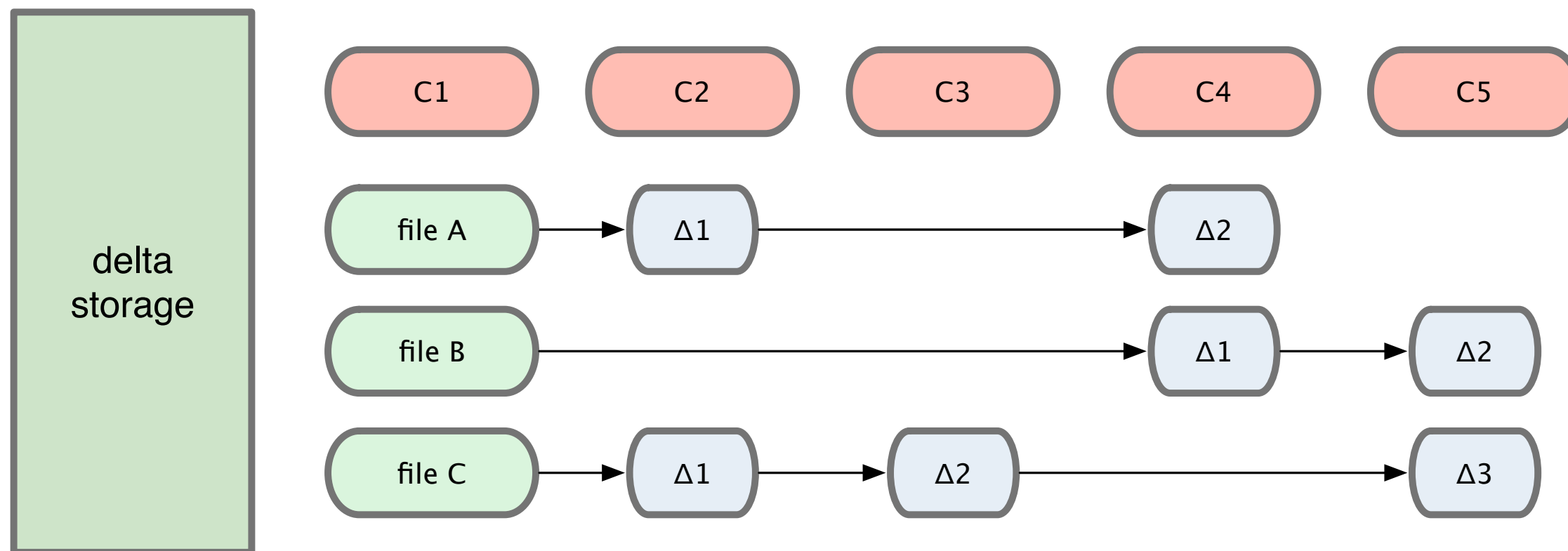


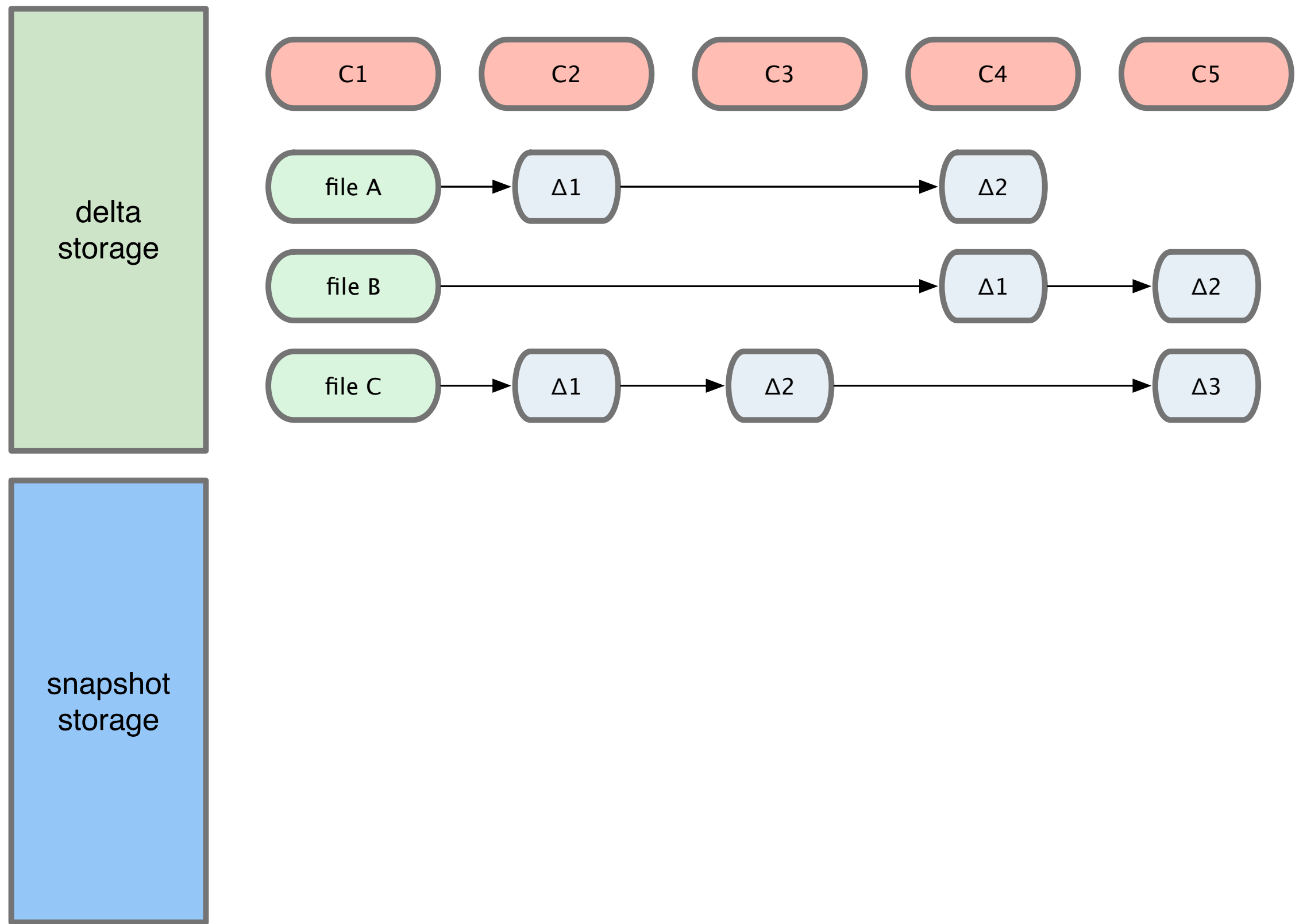


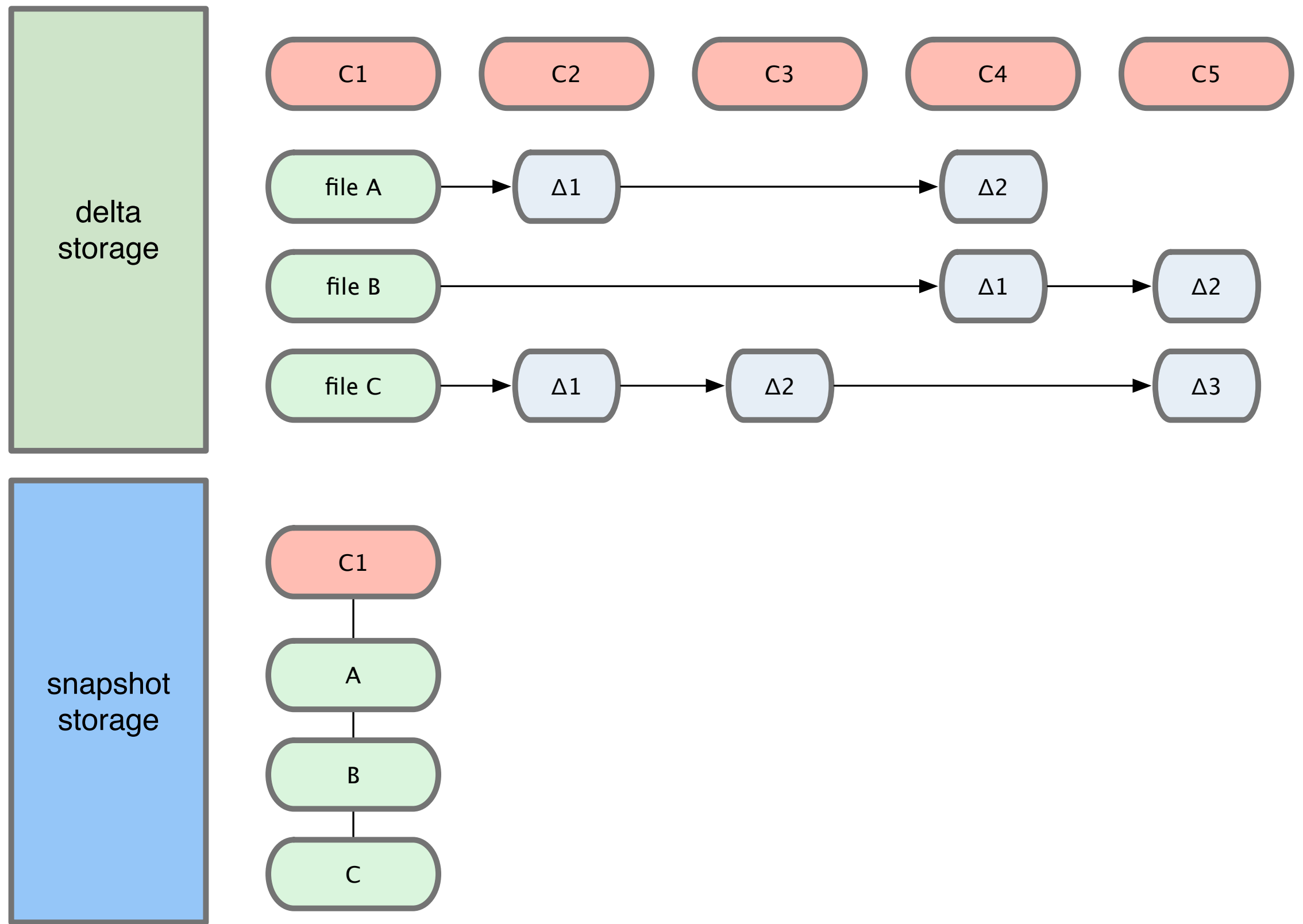


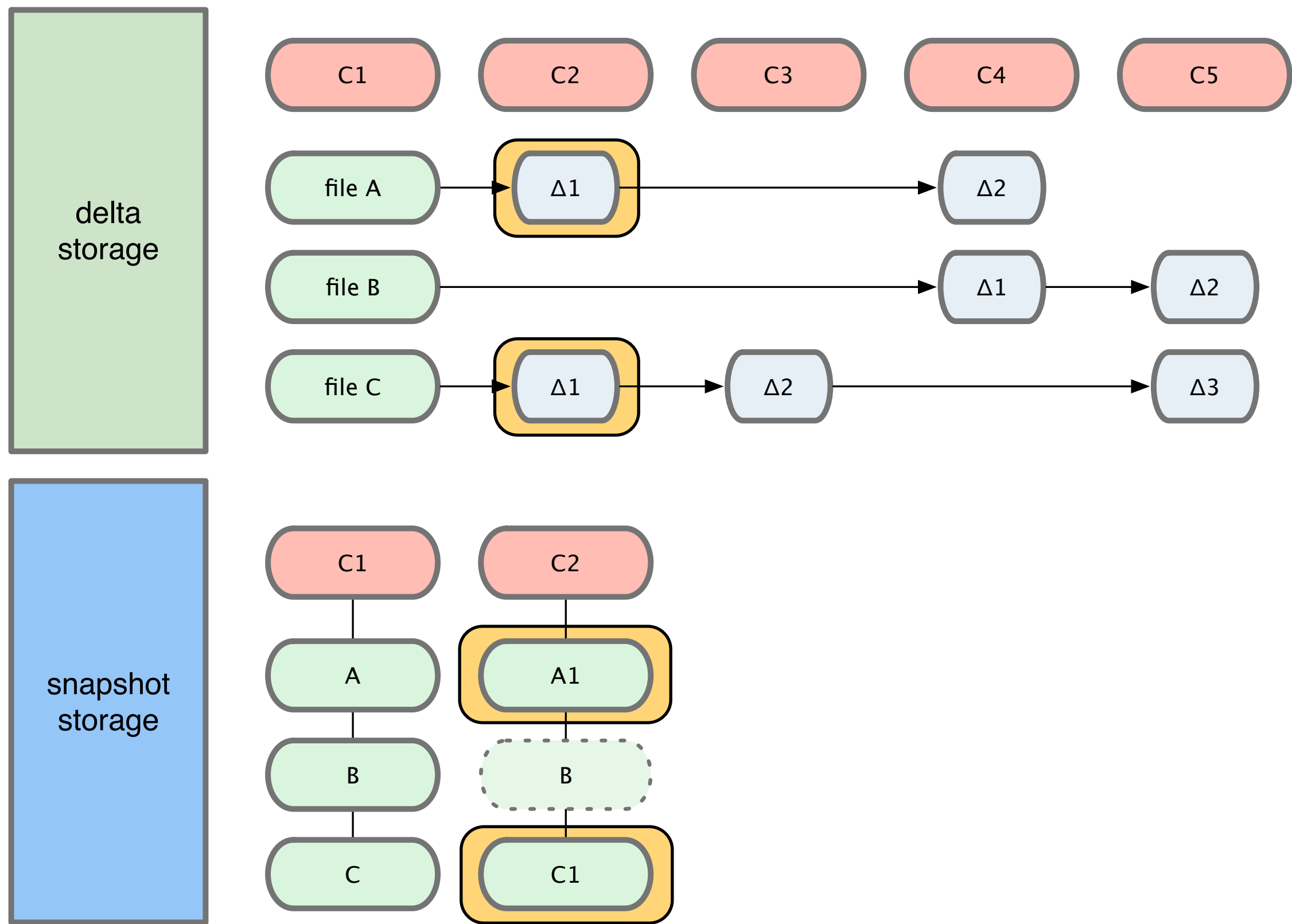


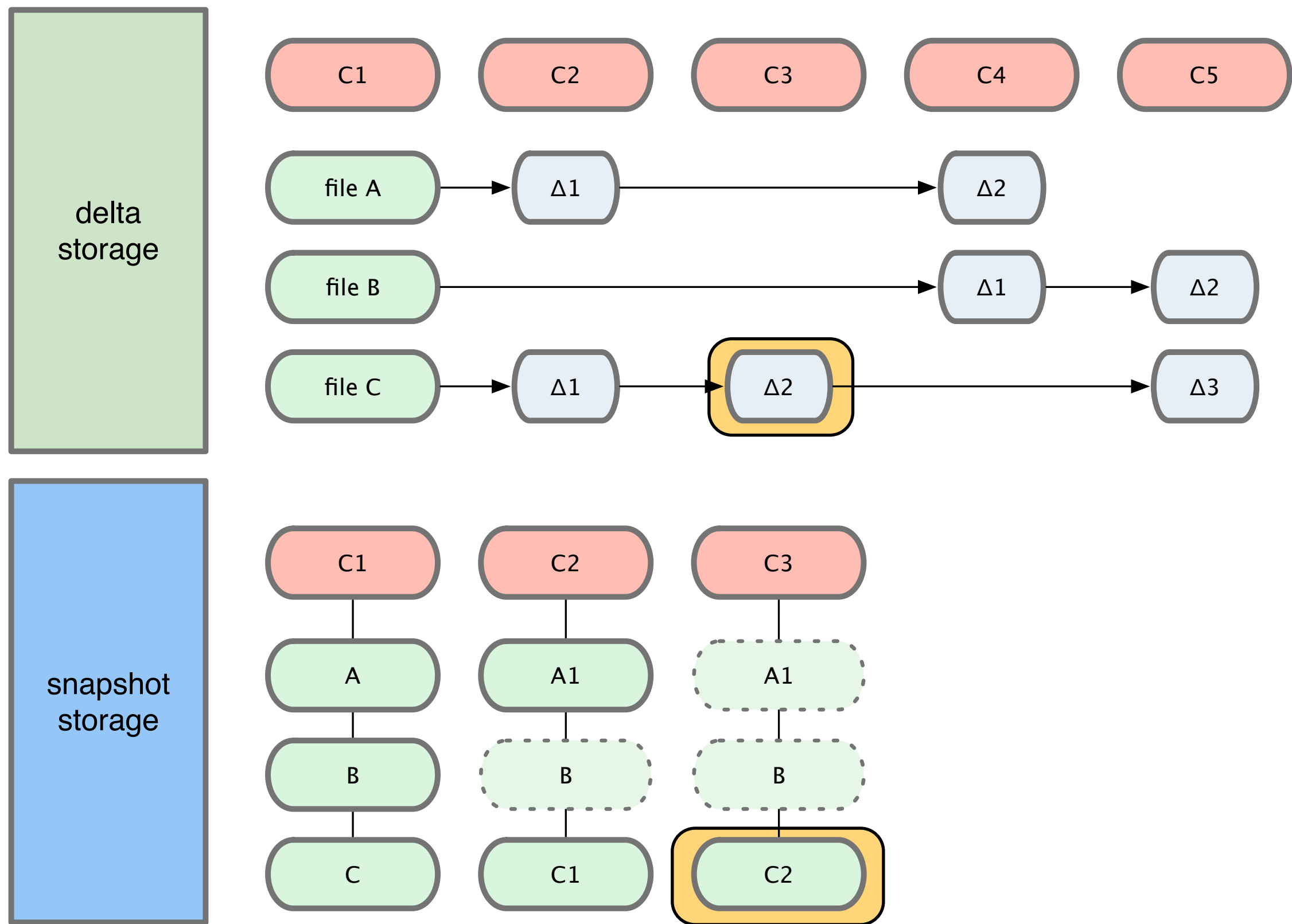




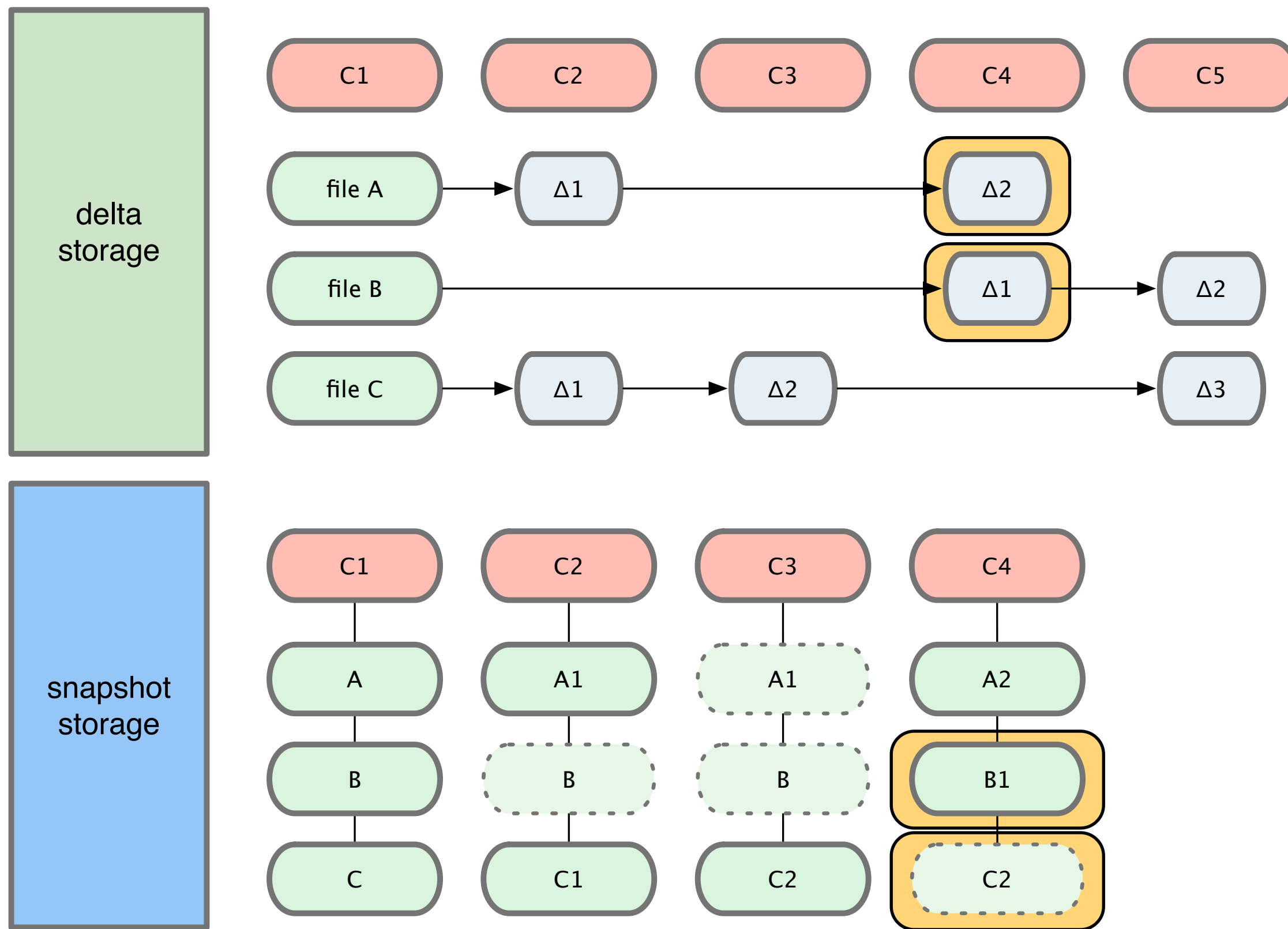


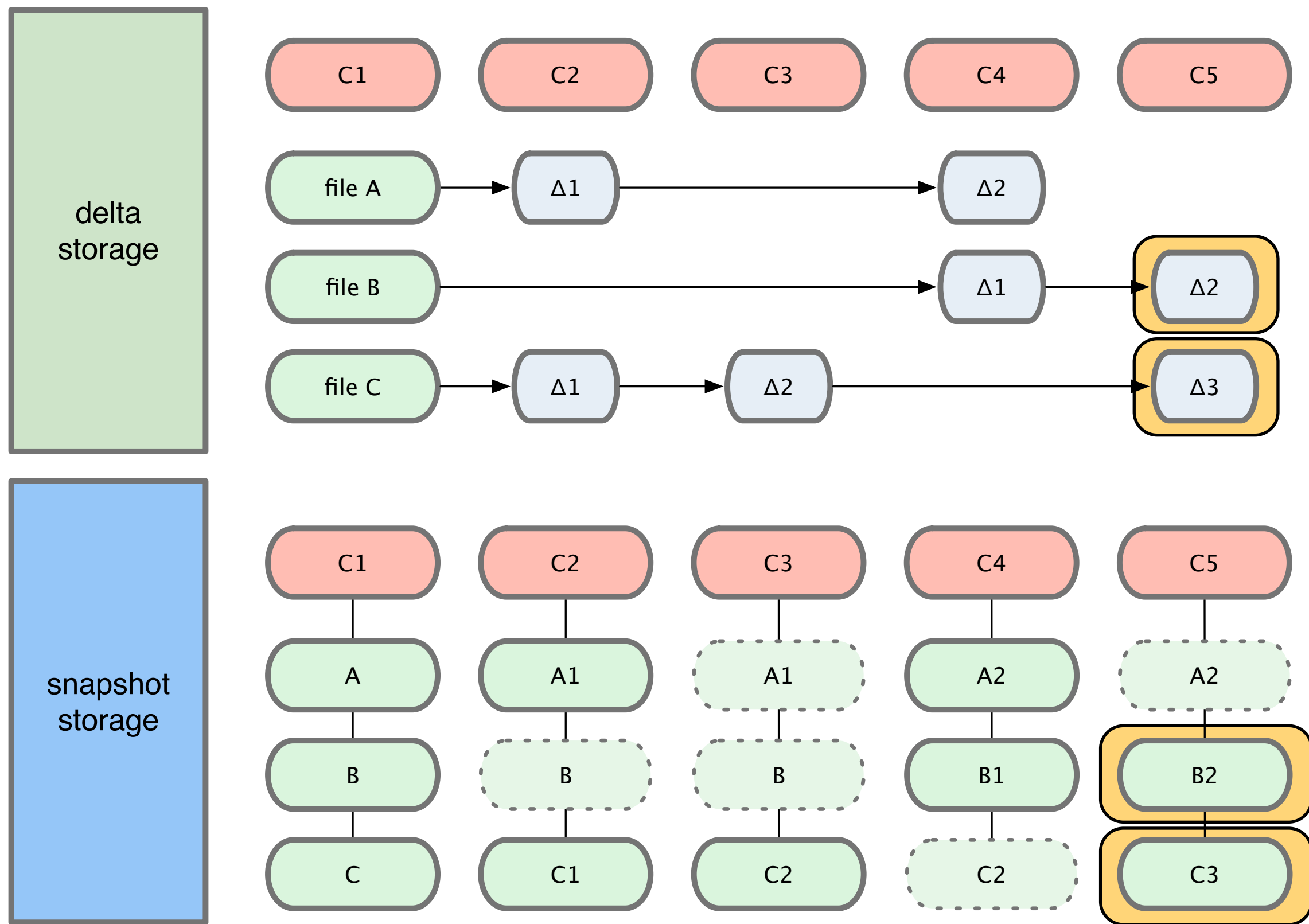


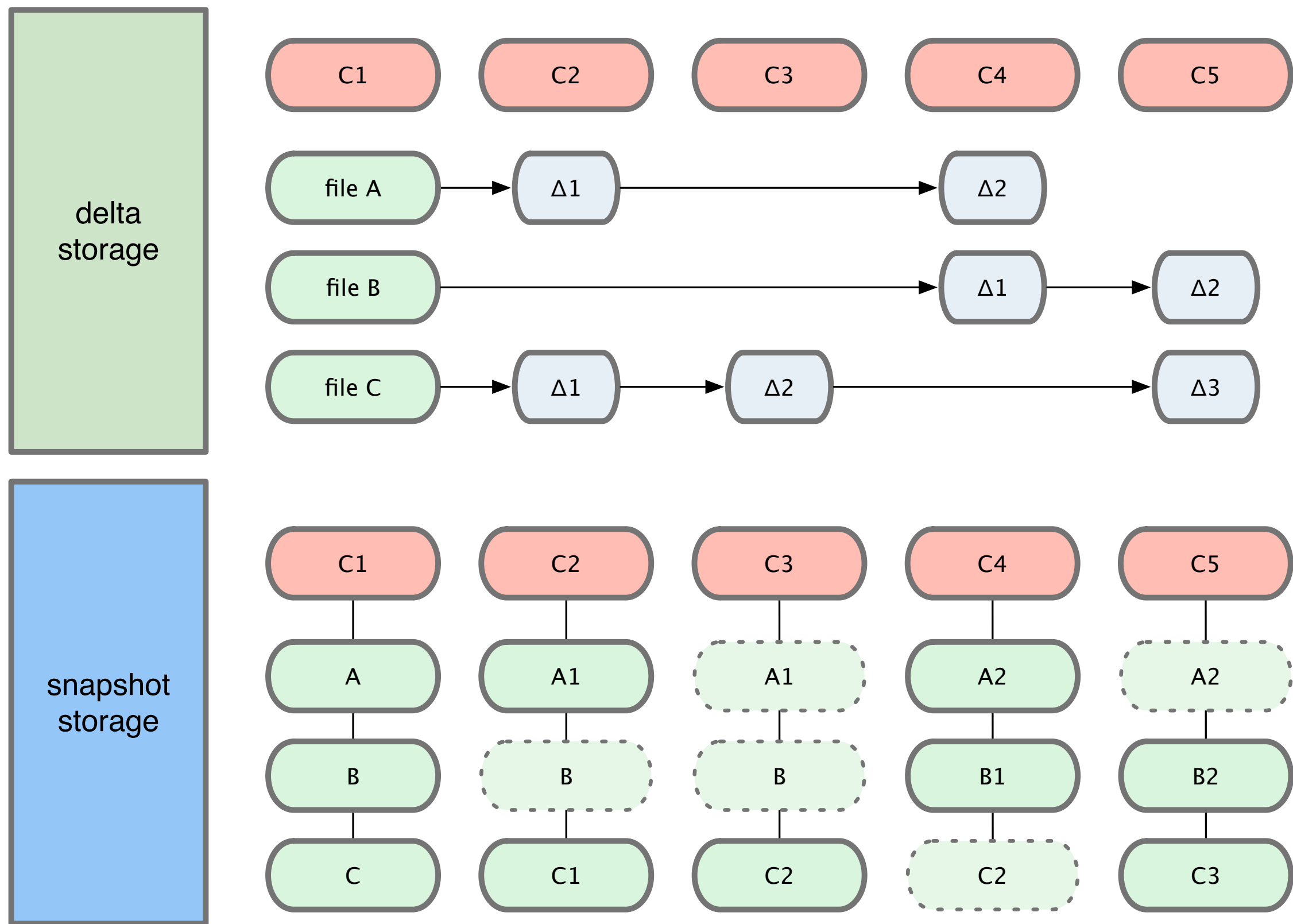


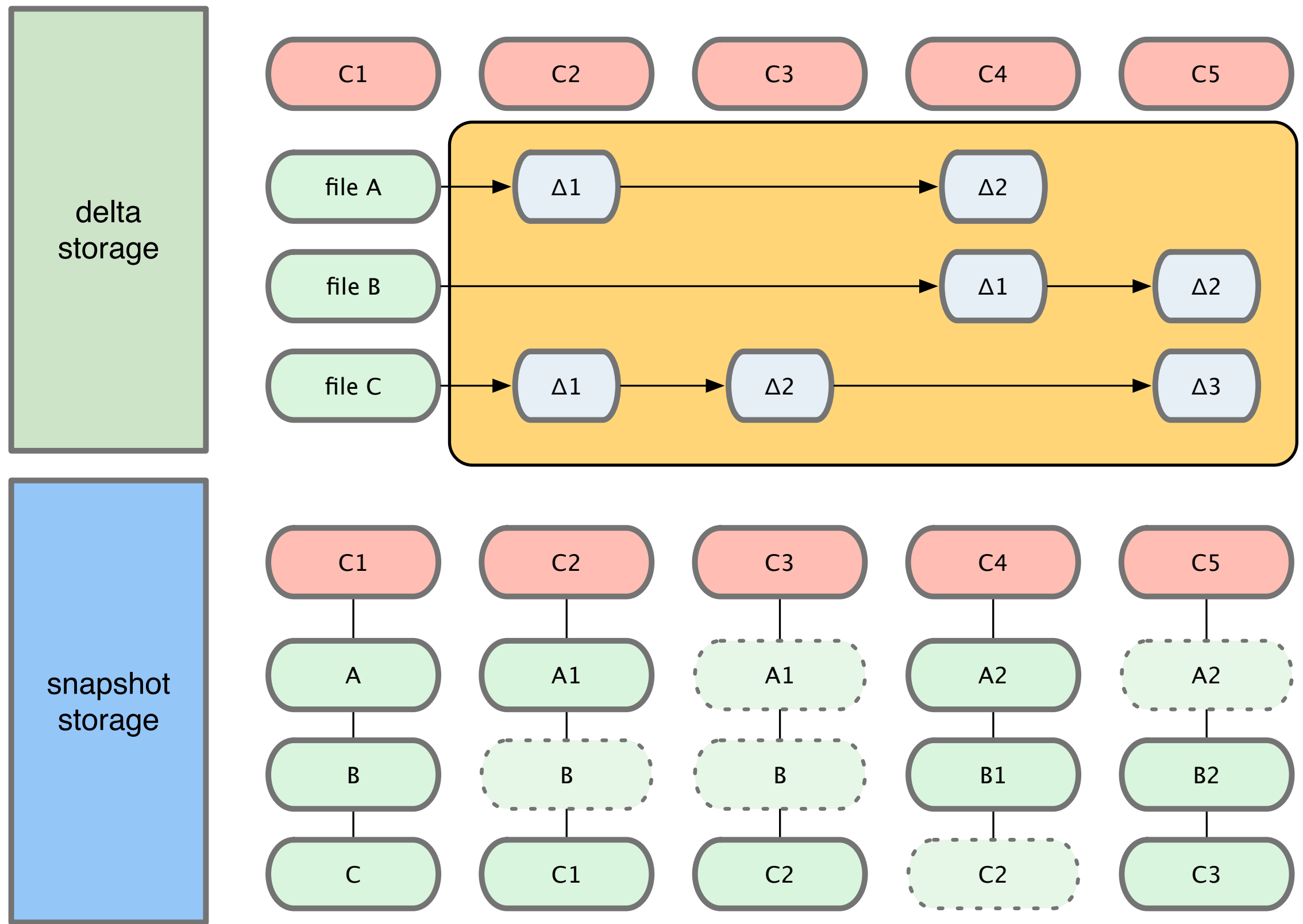


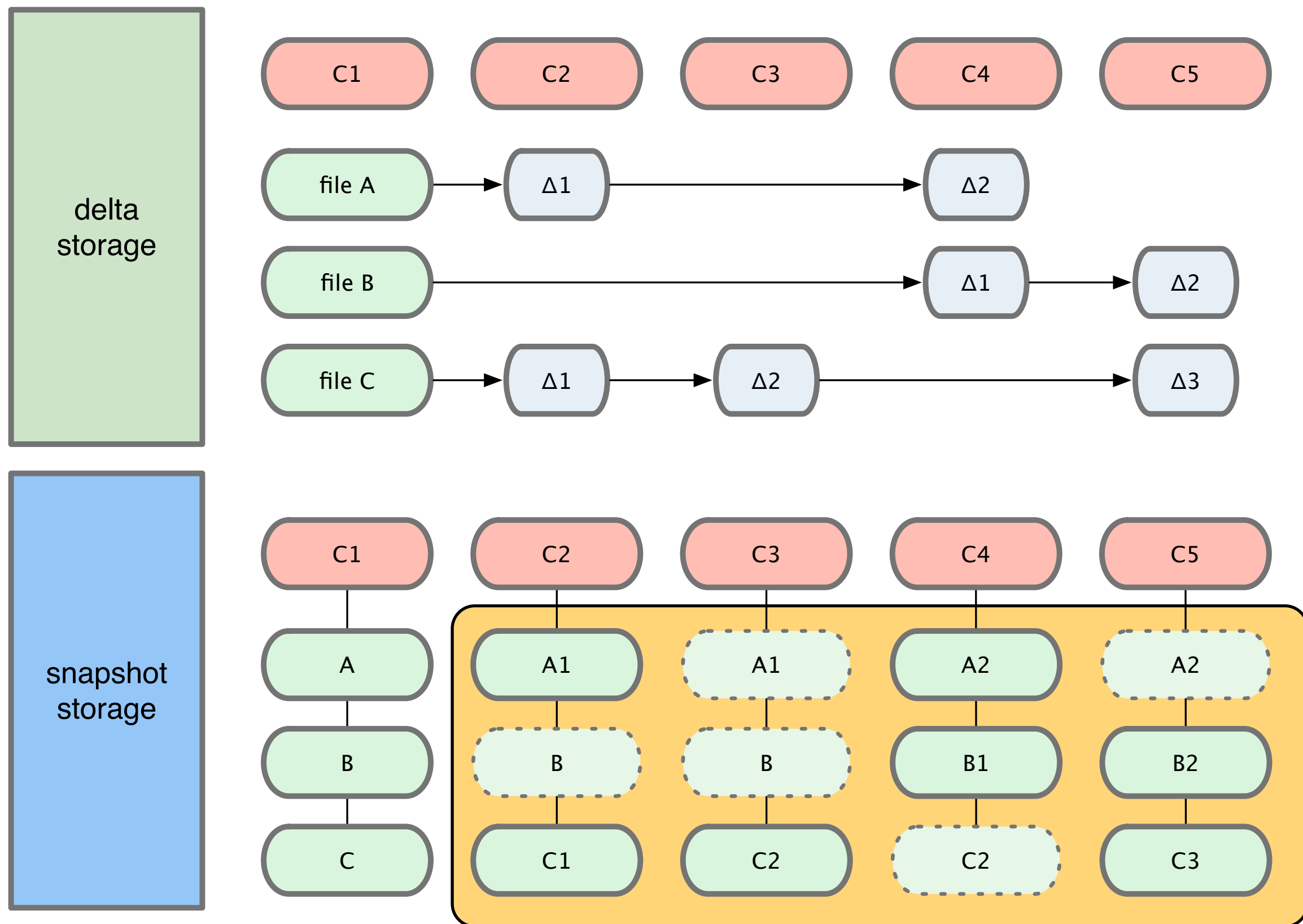






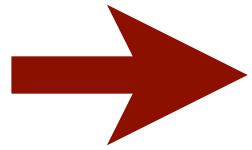






# Where are we?

What is Git?



**Basic Git Usage**

Intermediate Git Usage

Git Branching

Working with Repo & Gerrit

Advanced Git Usage

Git Internals

Git Maintenance

Running a Git Server

# 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**

**“staging”**

**repository**

# 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

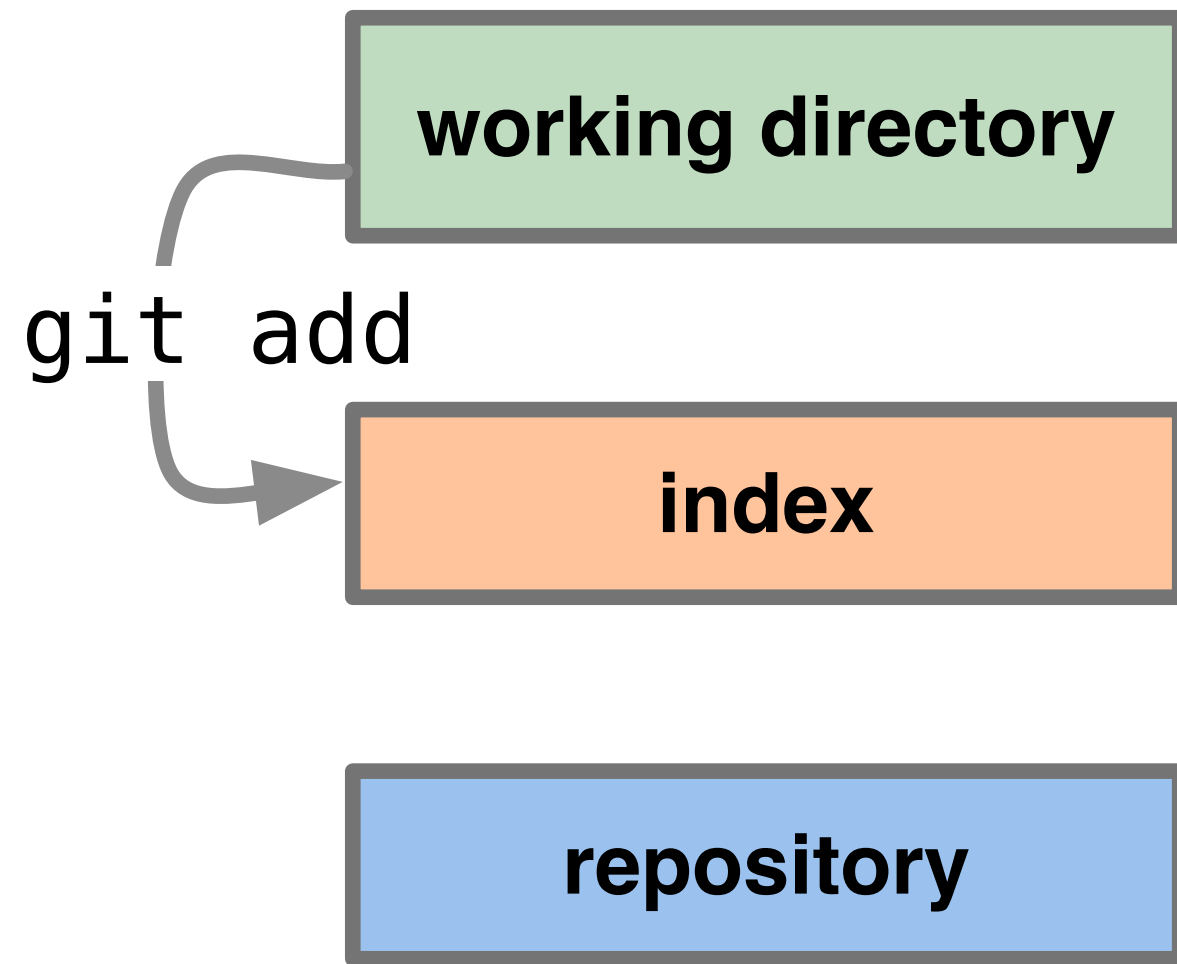
Commit the changes

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

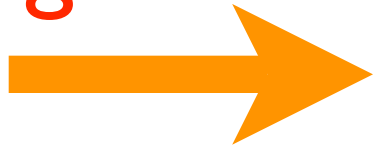
#

# 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 com

#

# modified: app.yaml



# modified: main.py

#

# Staged

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

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

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

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

wsgiref.handlers.CGIHandler().run(applicat

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

# In Working Directory

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

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

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

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

wsgiref.handlers.CGIHandler().run(applicat

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

# Staged

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

# this program prints out 'hello world'

class MainHandler(webapp.RequestHandler):

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

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

wsgiref.handlers.CGIHandler().run(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

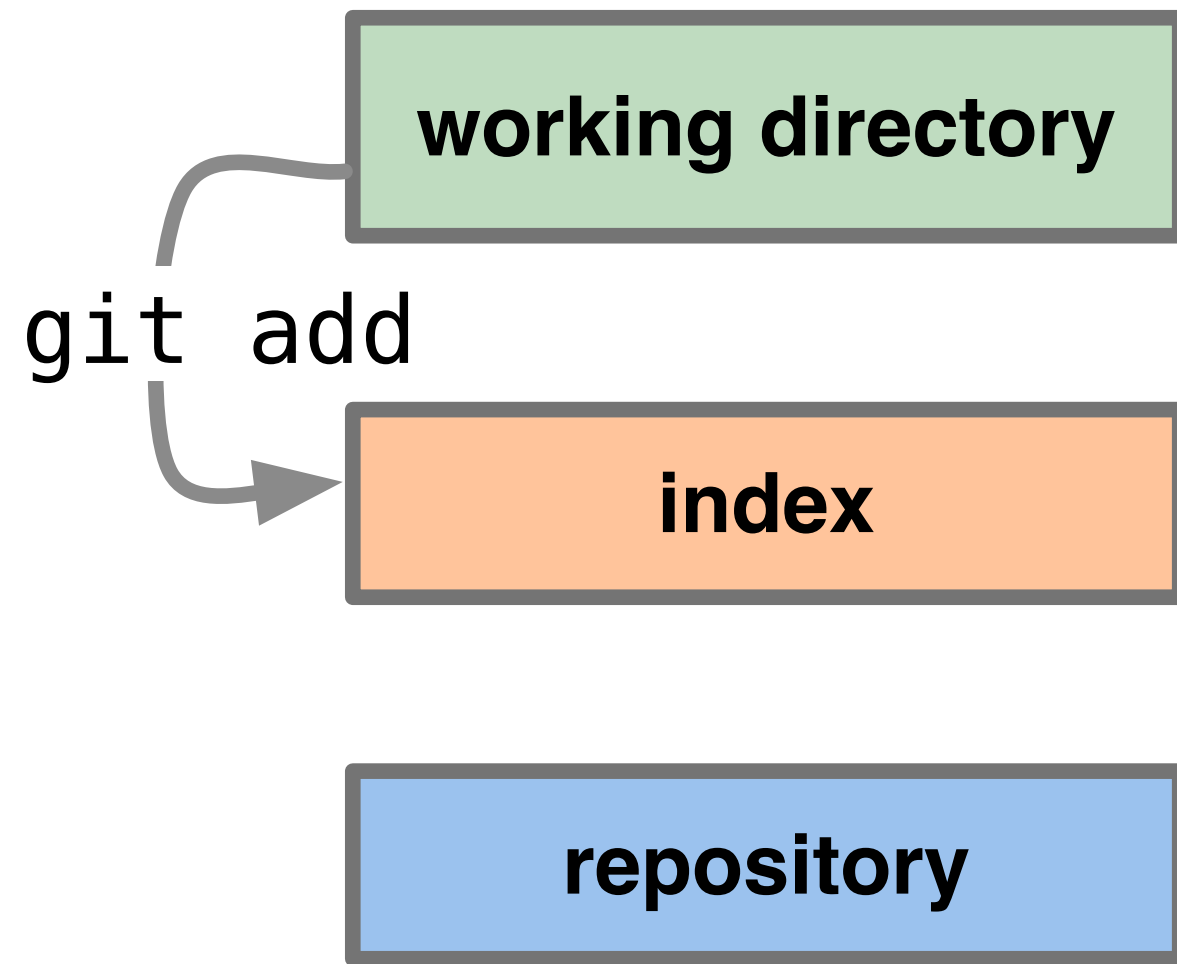
Edit files

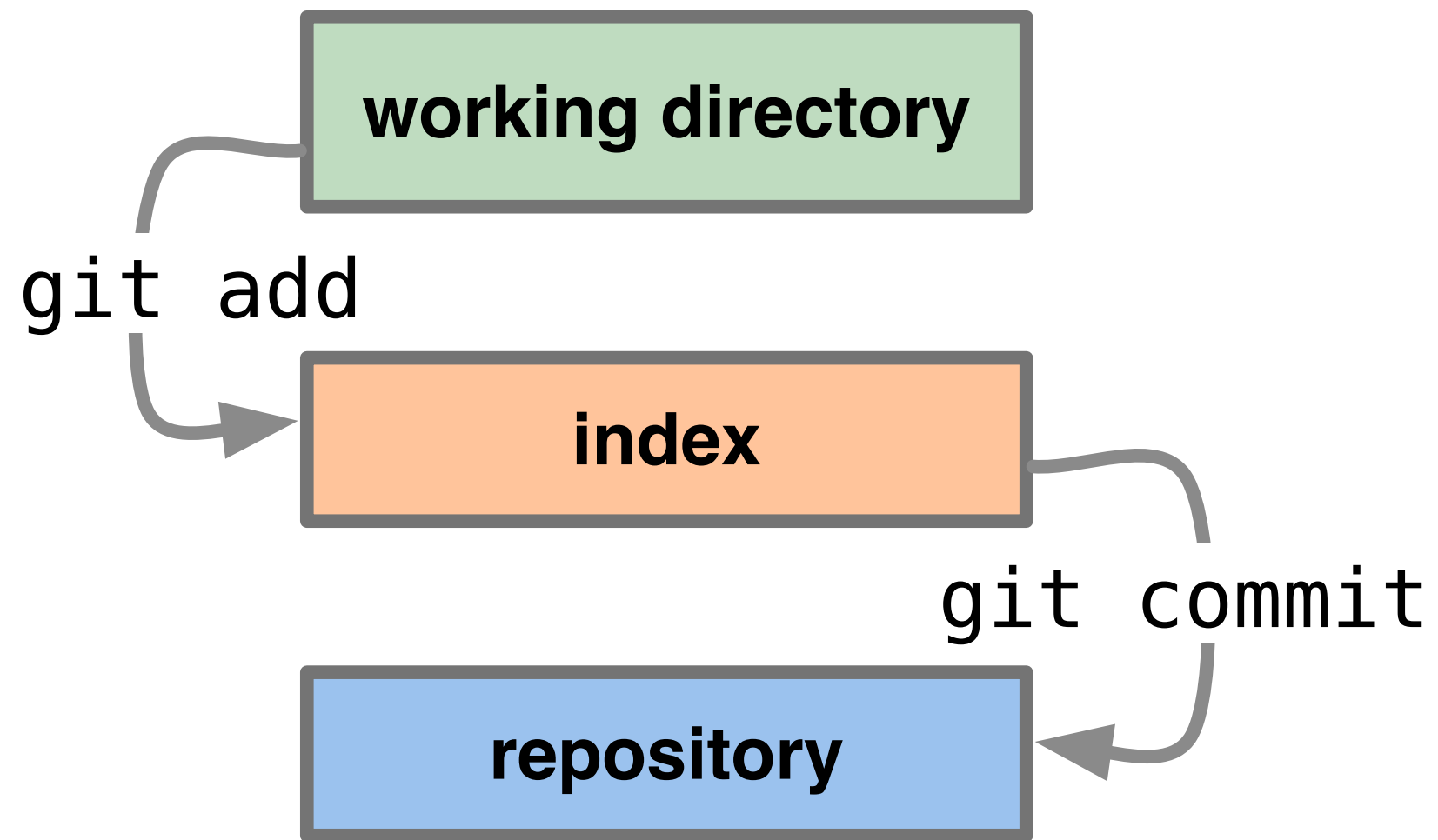
Stage the changes

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



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
```

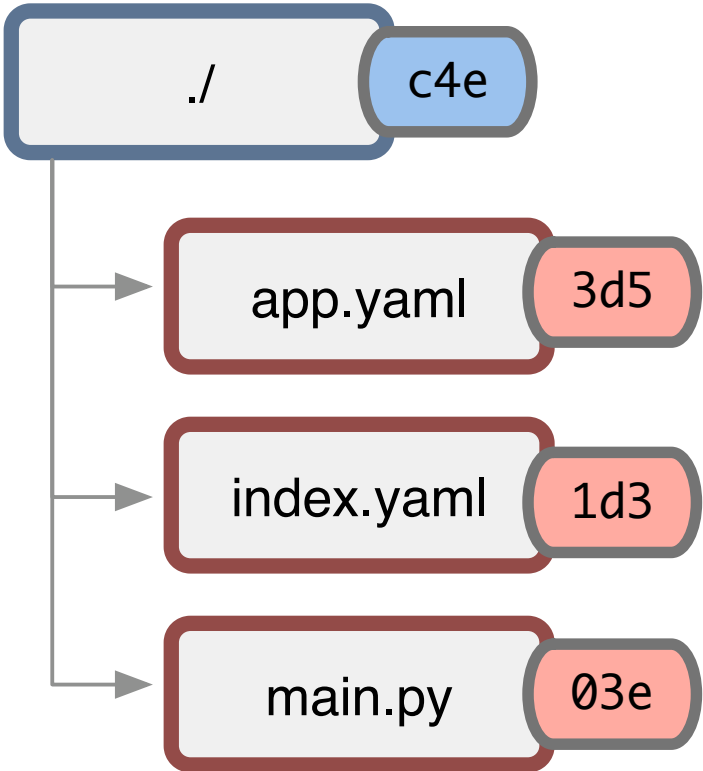


77d3001a1de6bf8f5e431972fe4d25b01e595c0b

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

77d3001a1de6bf8f5e431972fe4d25b01e595c0b

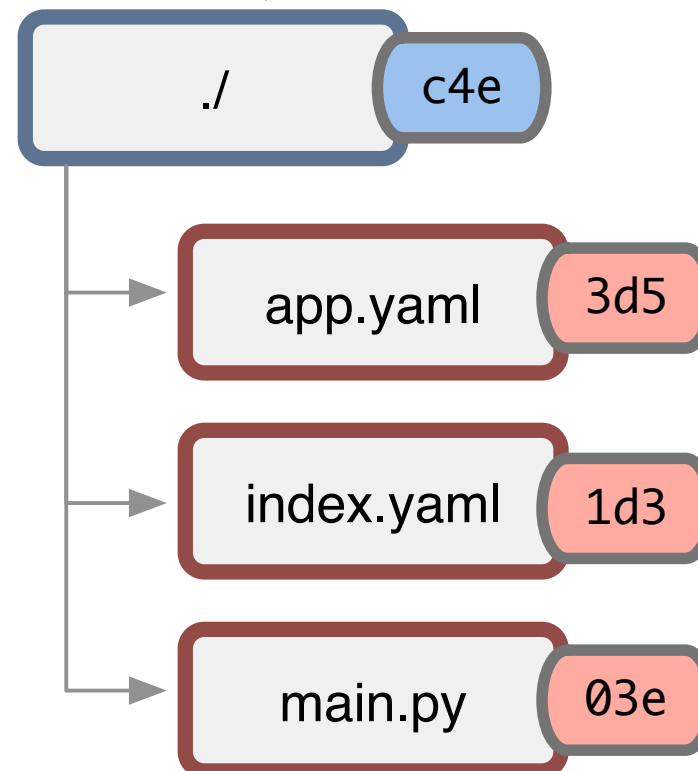
commit		size
tree	c4ec5	
parent	a149e	
author	Scott	
committer	Scott	
my commit message goes here and it is really, really cool		



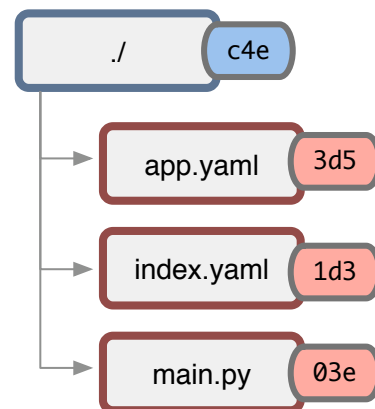
commit	size
tree	2de54
parent	38def
author	Scott
committer	Scott
this is the previous commit and I am very proud of it	



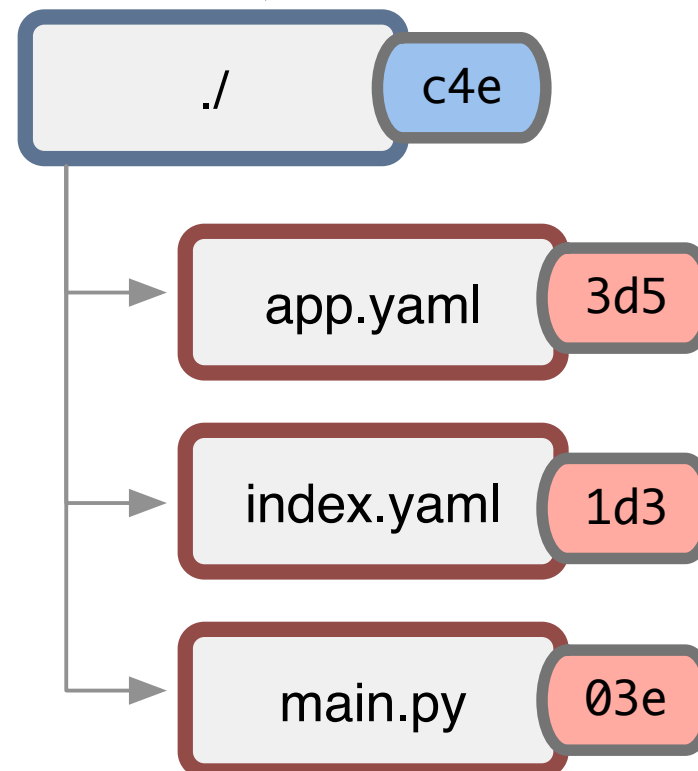
commit	size
tree	c4ec5
parent	a149e
author	Scott
committer	Scott
my commit message goes here and it is really, really cool	



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



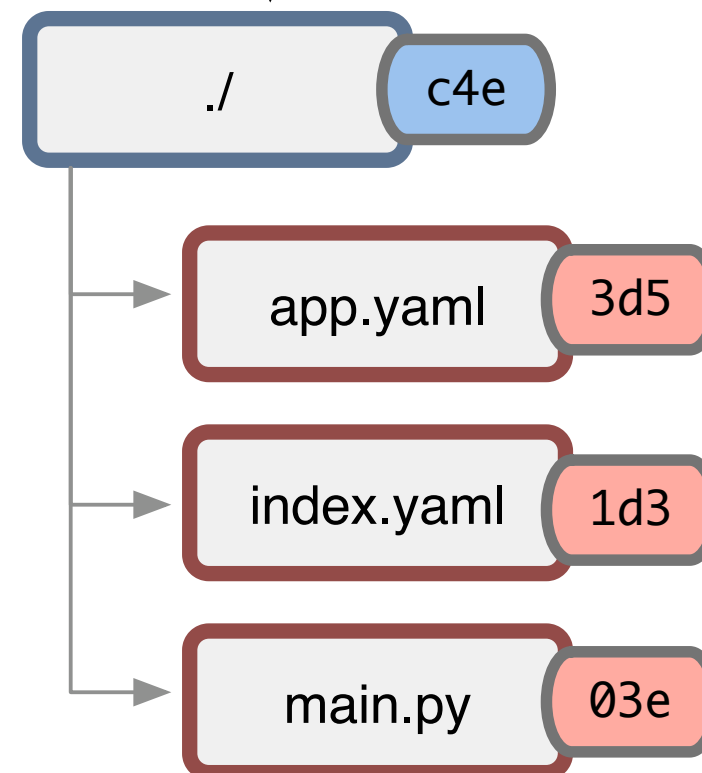
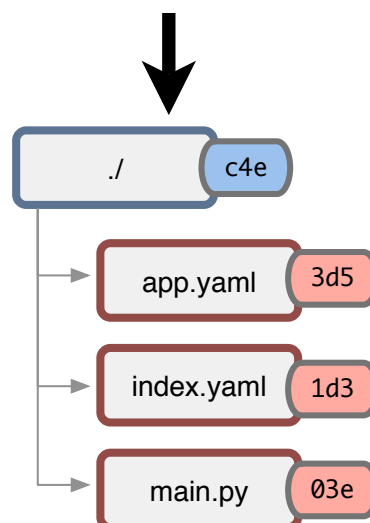
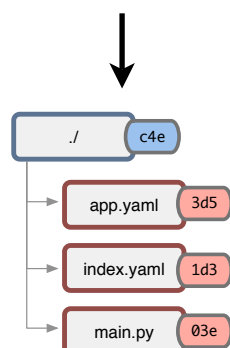
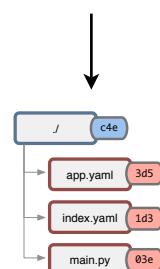
commit	size
tree	2fe65
parent	90ecd
author	Scott
committer	Scott
this is the commit before that and I'm not sure why	



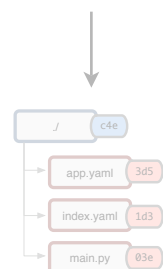
commit	size
tree	2de54
parent	38def
author	Scott
committer	Scott
this is the previous commit and I am very proud of it	



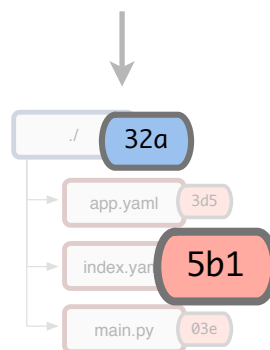
commit	size
tree	c4ec5
parent	a149e
author	Scott
committer	Scott
my commit message goes here and it is really, really cool	



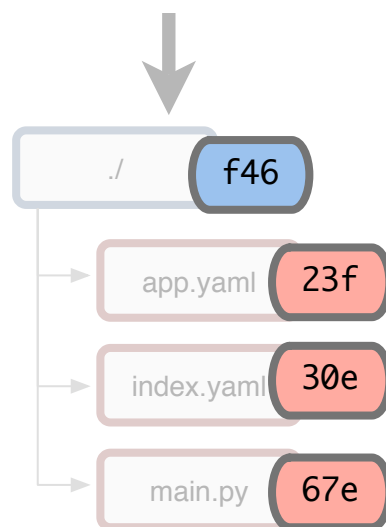
commit	size
tree 48e	
parent	
author	
committer Scott	
this is the commit before that and I'm not sure why	



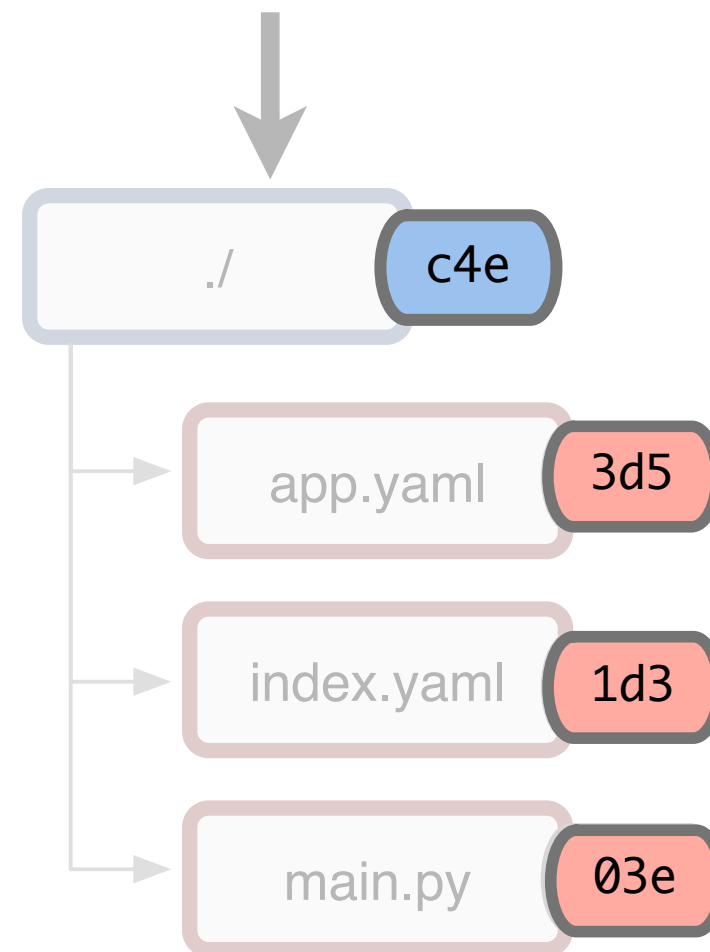
commit	size
tree 2fe65	
parent 38d	d
author Scott	
committer Scott	
this is the commit before that and I'm not sure why	



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



commit	size
tree c4ec5	
parent 77d	49e
author Scott	
committer Scott	
my commit message goes here and it is really, really cool	



# Repository

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

# Repository

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

```
git checkout ae635
```



# Repository

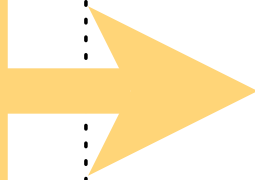
5b1	32a	c36
1d3	f46	3d4
ffe	6fe	ae6
38d	23f	03e
254	30e	5b1
a14	67e	1d3
3d5	735	d23
c4e	c4e	48e
77d	de3	2d3

git checkout ae635

# Repository

# Index

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

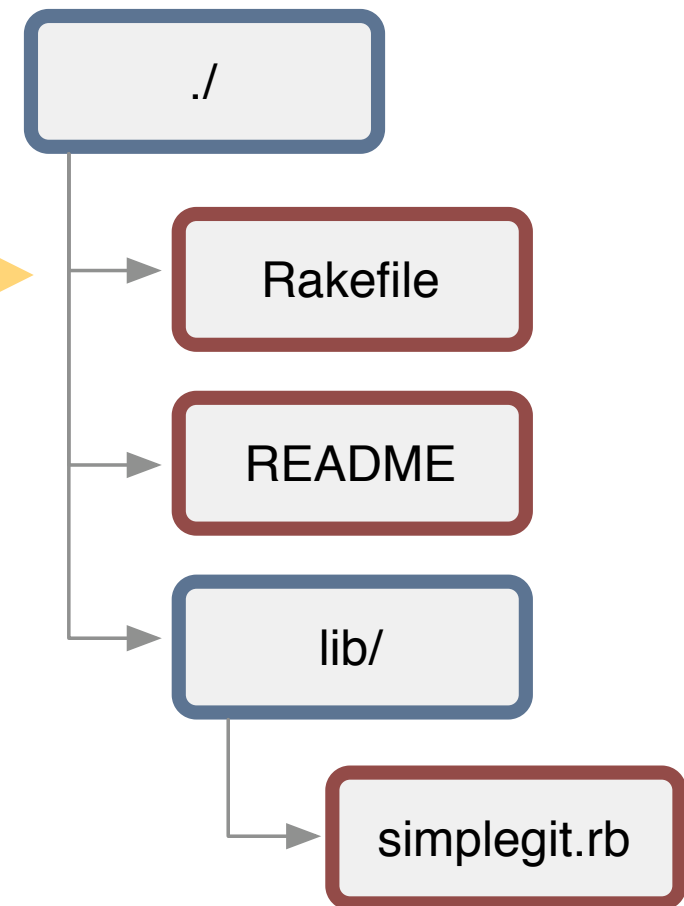
# Repository

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



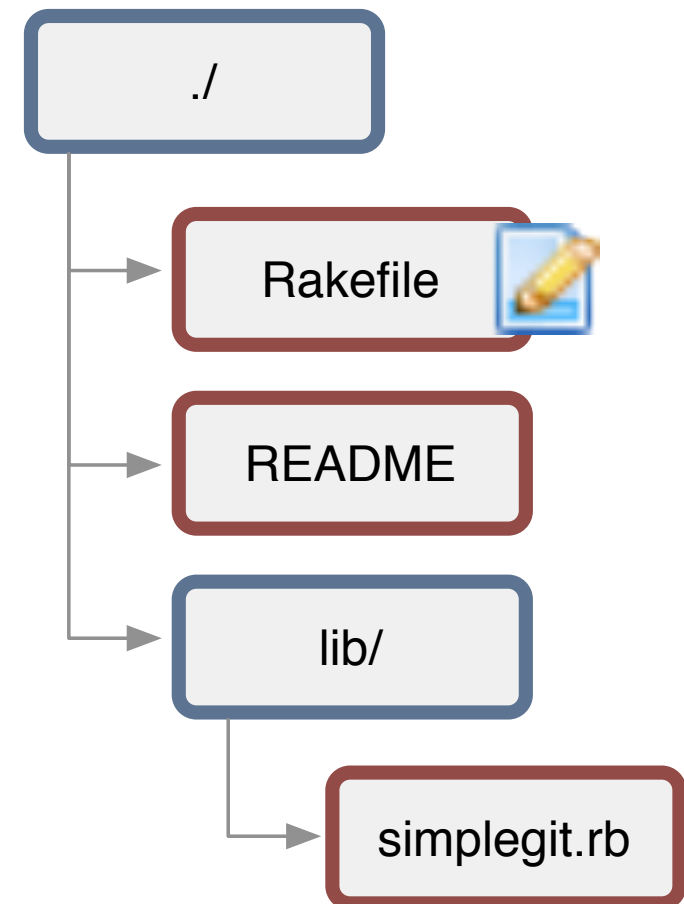
# Repository

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



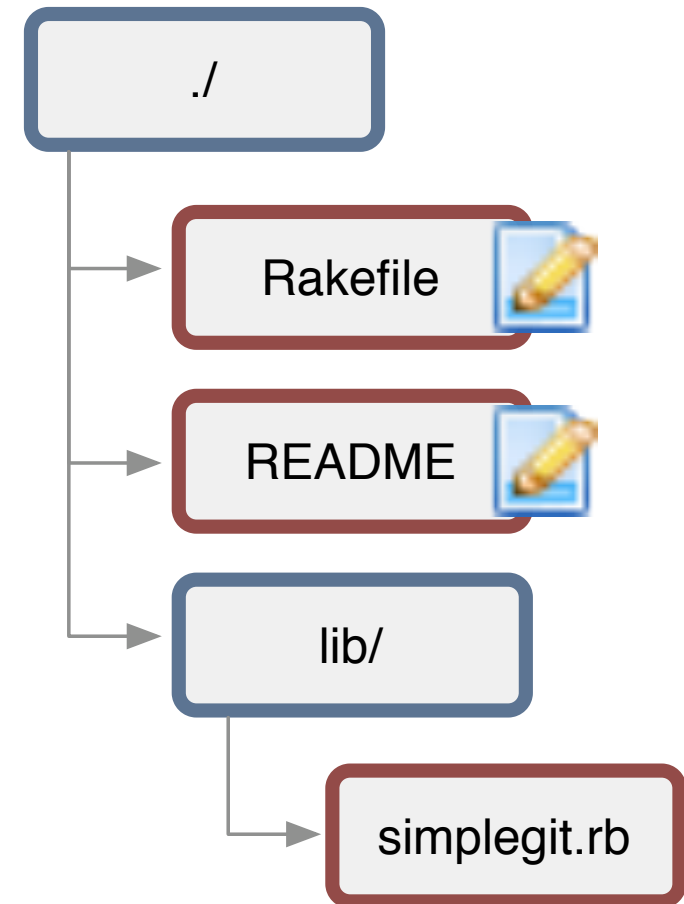
## Repository

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



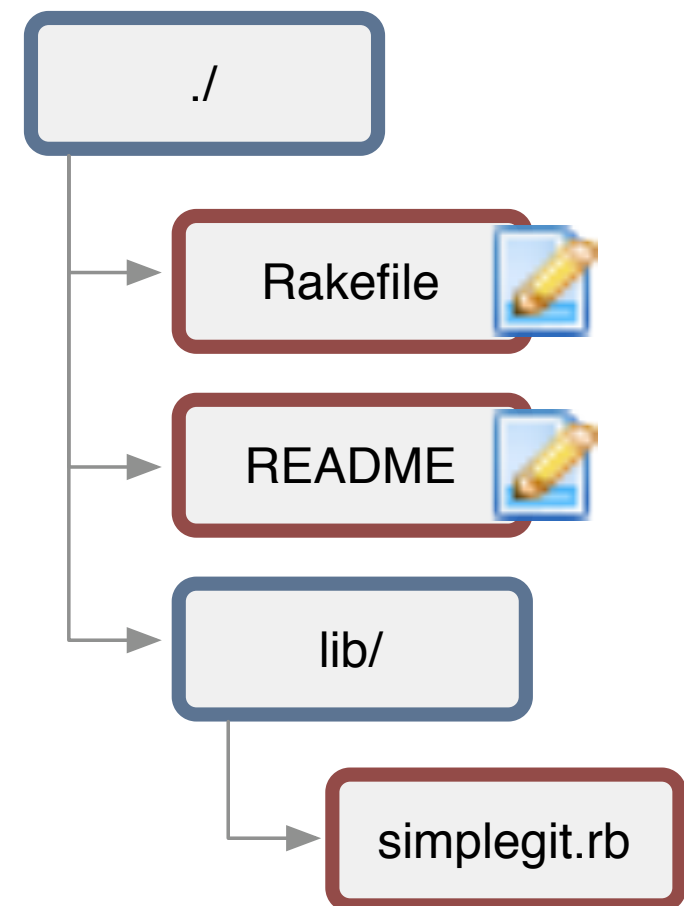
## Repository

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



git add

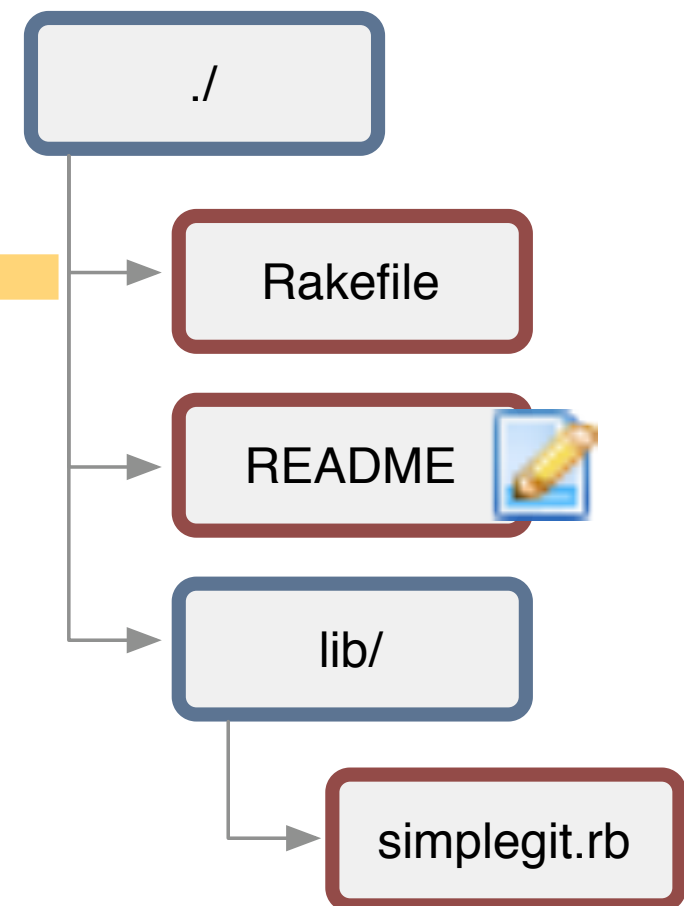
## Repository

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

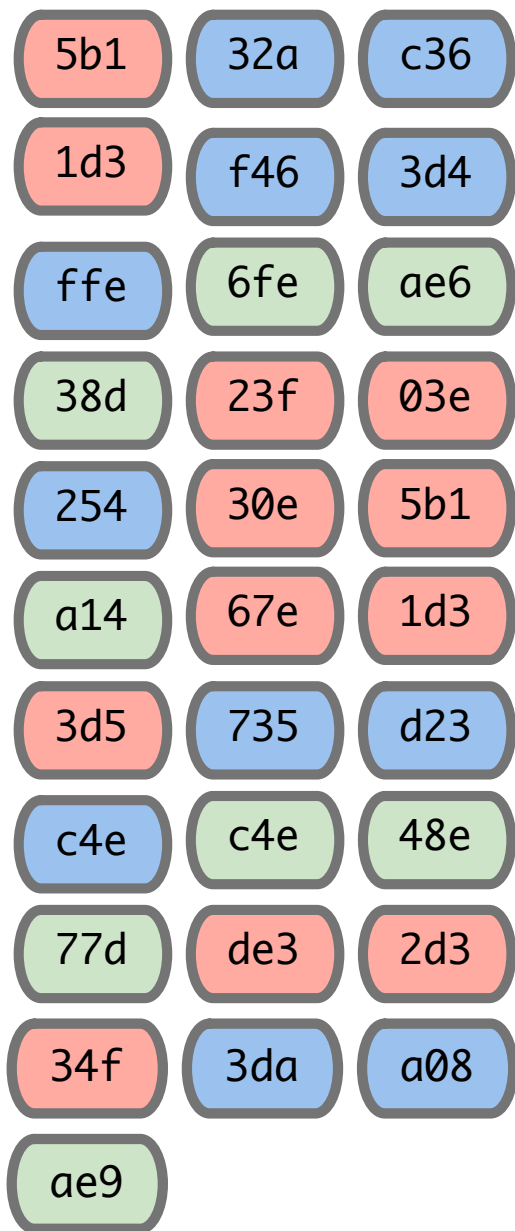
./	3d4
./Rakefile	1d3 34f
./README	03e
./lib/	c36
./lib/simplegit.rb	5b1

## Working Directory

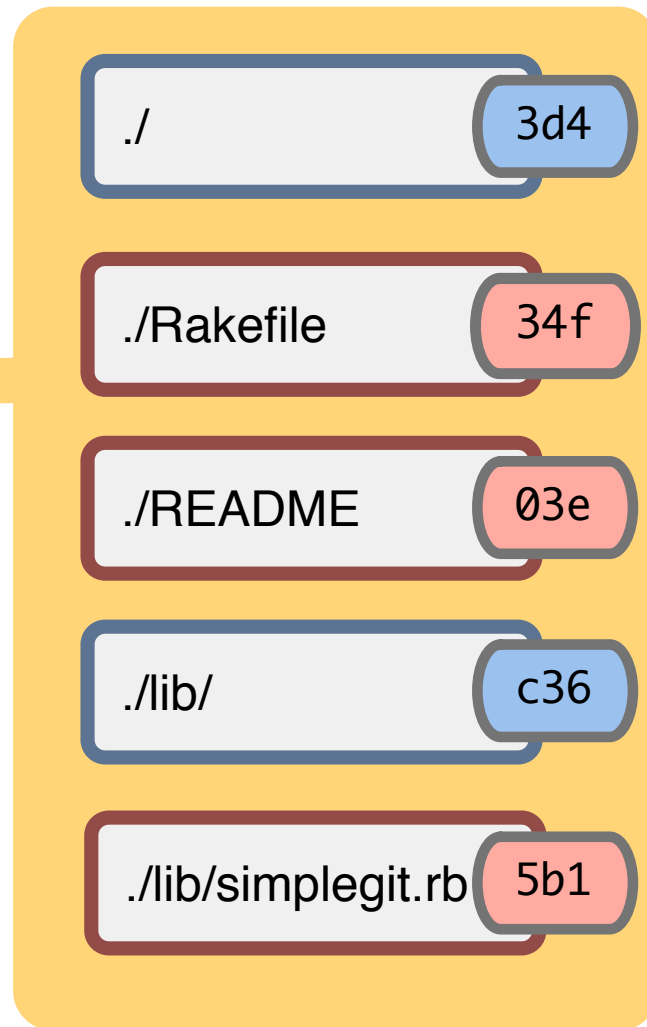


git add

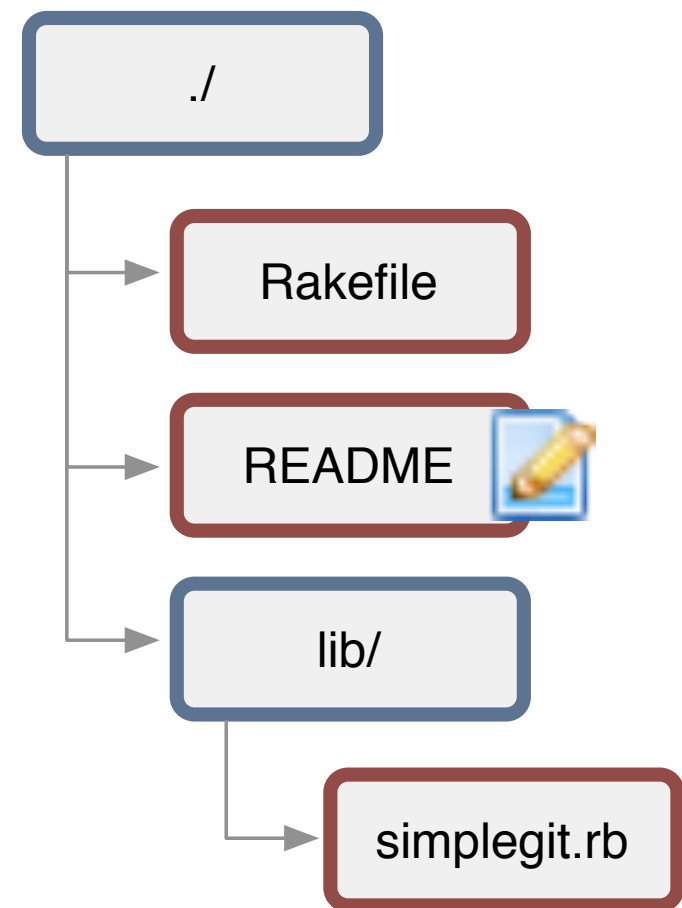
## Repository



## Index



## Working Directory



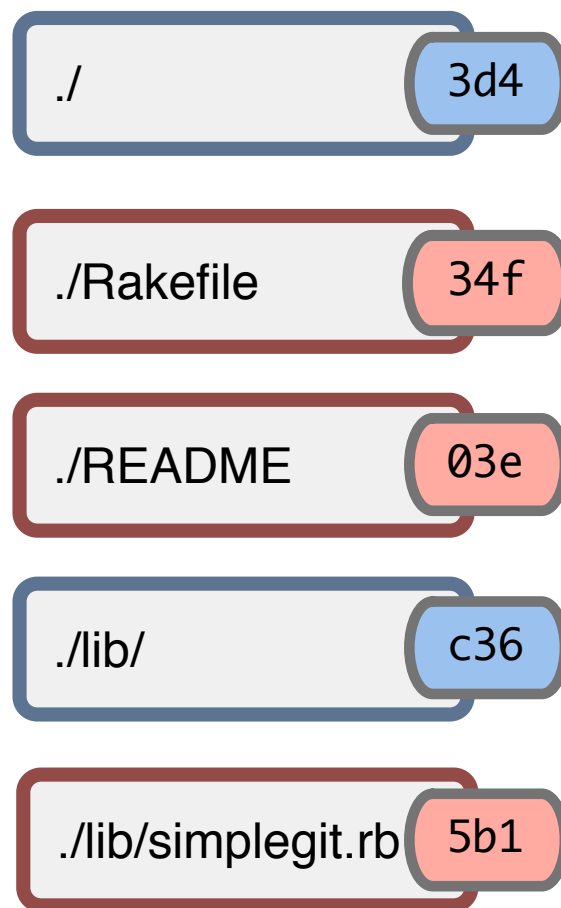
git commit



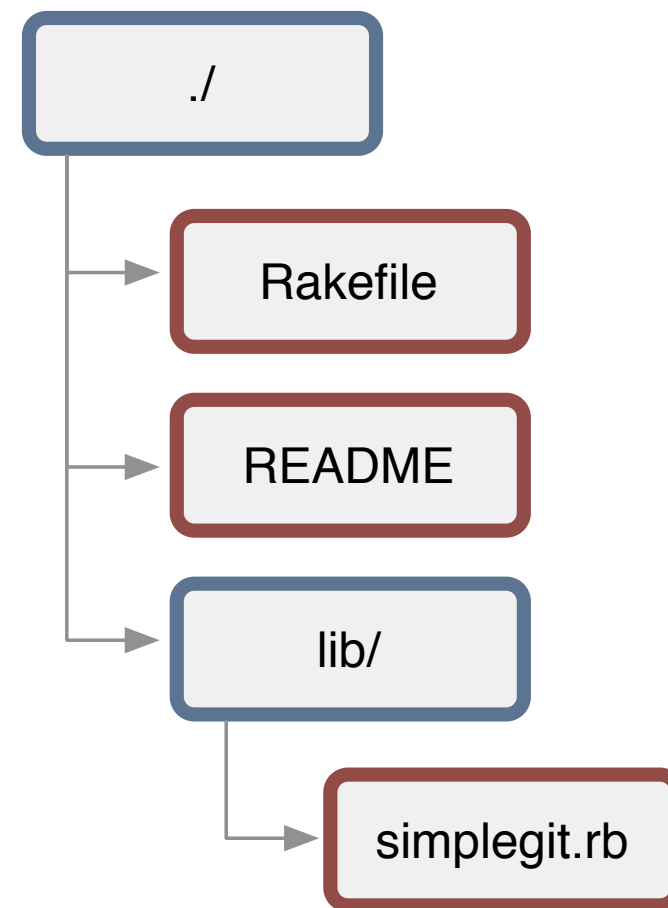
# Repository



# Index



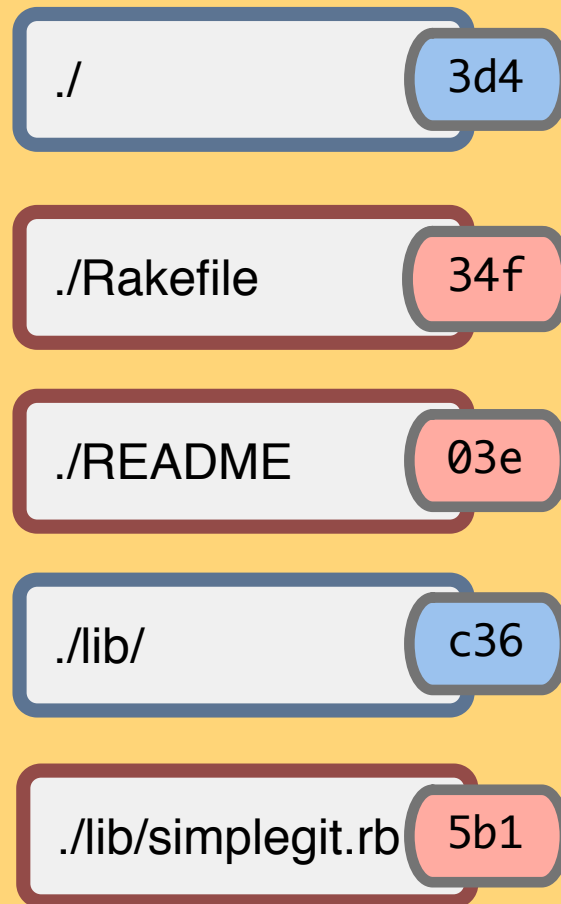
# Working Directory



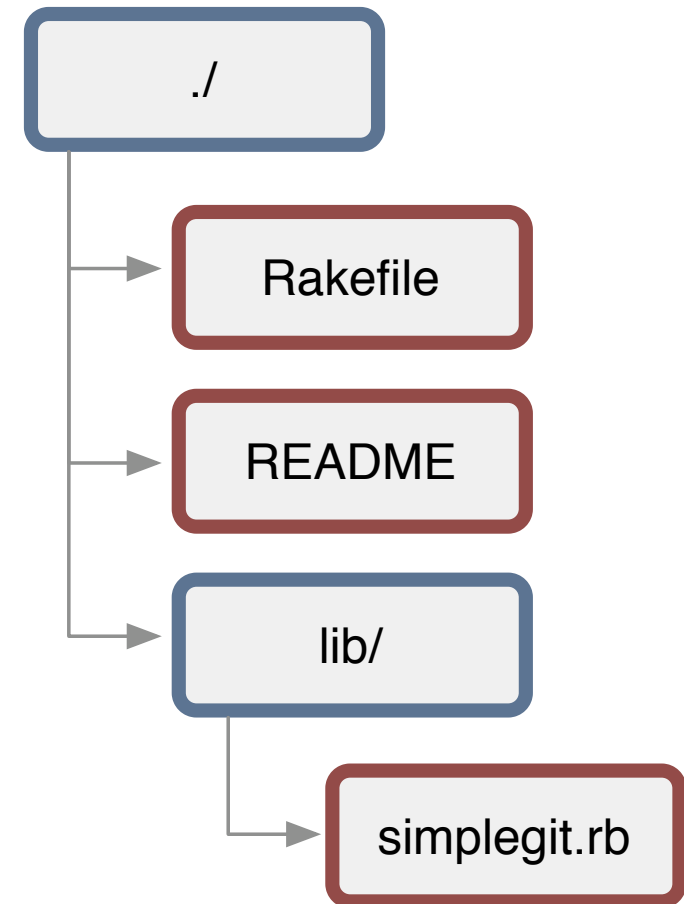
# Repository



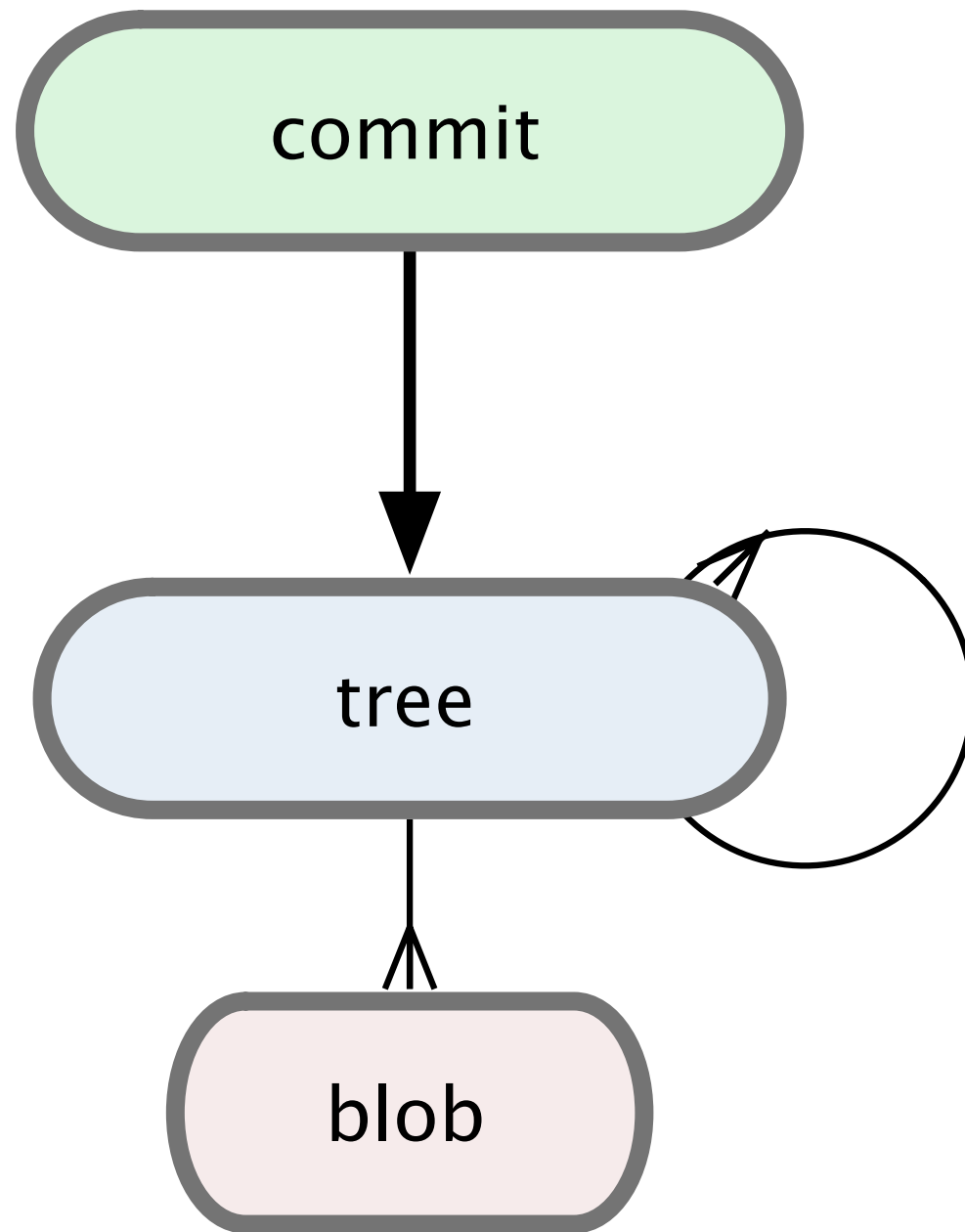
# Index



# Working Directory



# object model

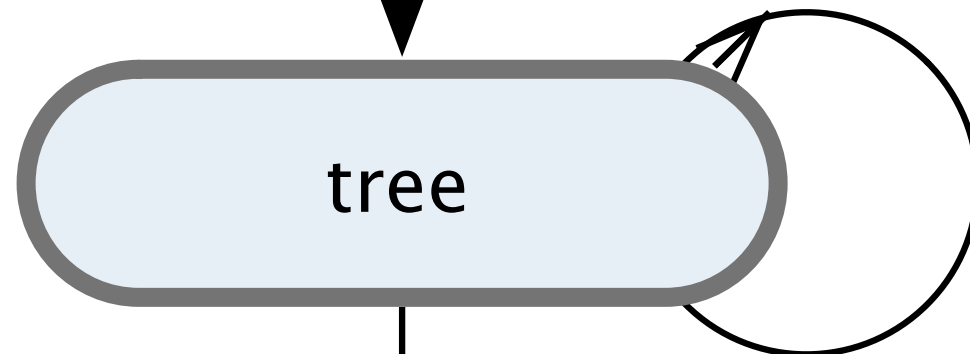


# object model

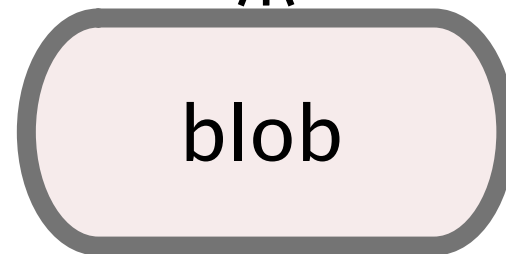
pointer to a  
snapshot



directory list



file contents



# 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](#)

[facebox](#)

[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](#)

[facebox](#)

[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](#) / [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: 0000000000000000000000000000000000000000 -> 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
$>_
```



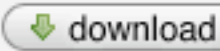
[schacon](#) / [simplegit2](#)



[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
<a href="#">README</a>	about 14 hours ago	updated README [Magnus Chacon]	
<a href="#">Rakefile</a>	39 minutes ago	changes [Magnus Chacon]	
<a href="#">TODO</a>	39 minutes ago	changes [Magnus Chacon]	
<a href="#">lib/</a>	16 minutes ago	added file_size [Magnus Chacon]	





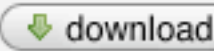
**schacon** / **simplegit2**



[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
<a href="#">README</a>	about 14 hours ago	updated README [Magnus Chacon]	
<a href="#">Rakefile</a>	39 minutes ago	changes [Magnus Chacon]	
<a href="#">TODO</a>	39 minutes ago	changes [Magnus Chacon]	
<a href="#">lib/</a>	16 minutes ago	added file_size [Magnus Chacon]	



Source Browser

Commits

Wiki

Network (1)

Admin

master

all branches

all tags

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

add



Magnus Chacon (author)

17 minutes ago

parent [f40831985e265eb8cca2d0e66c11ac48ccfe170d](#)

simplegit2 /

name	age	message	history
<a href="#">README</a>	about 14 hours ago	updated README [Magnus Chacon]	
<a href="#">Rakefile</a>	39 minutes ago	changes [Magnus Chacon]	
<a href="#">TODO</a>	39 minutes ago	changes [Magnus Chacon]	
<a href="#">lib/</a>	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
$>
```

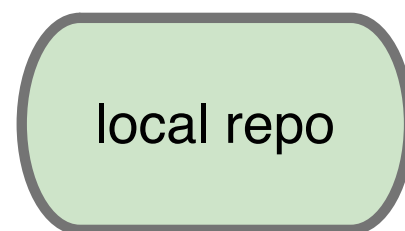
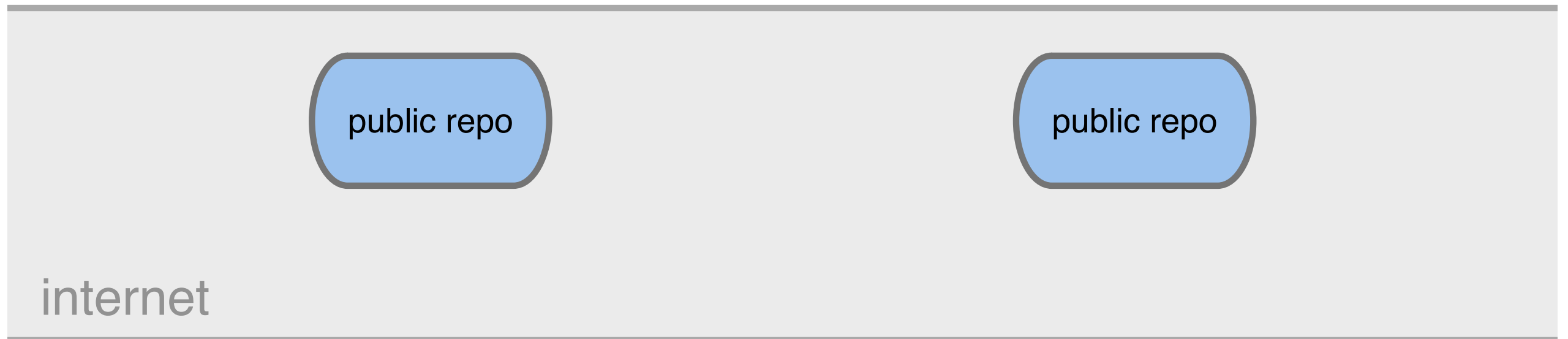
```
$>git remote
public
$>git remote show public
* remote public
  URL: git@github.com:schacon/simplegit.git
  Tracked remote branches
    master
$>git push public master
Counting objects: 7, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 365 bytes, done.
Total 4 (delta 2),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

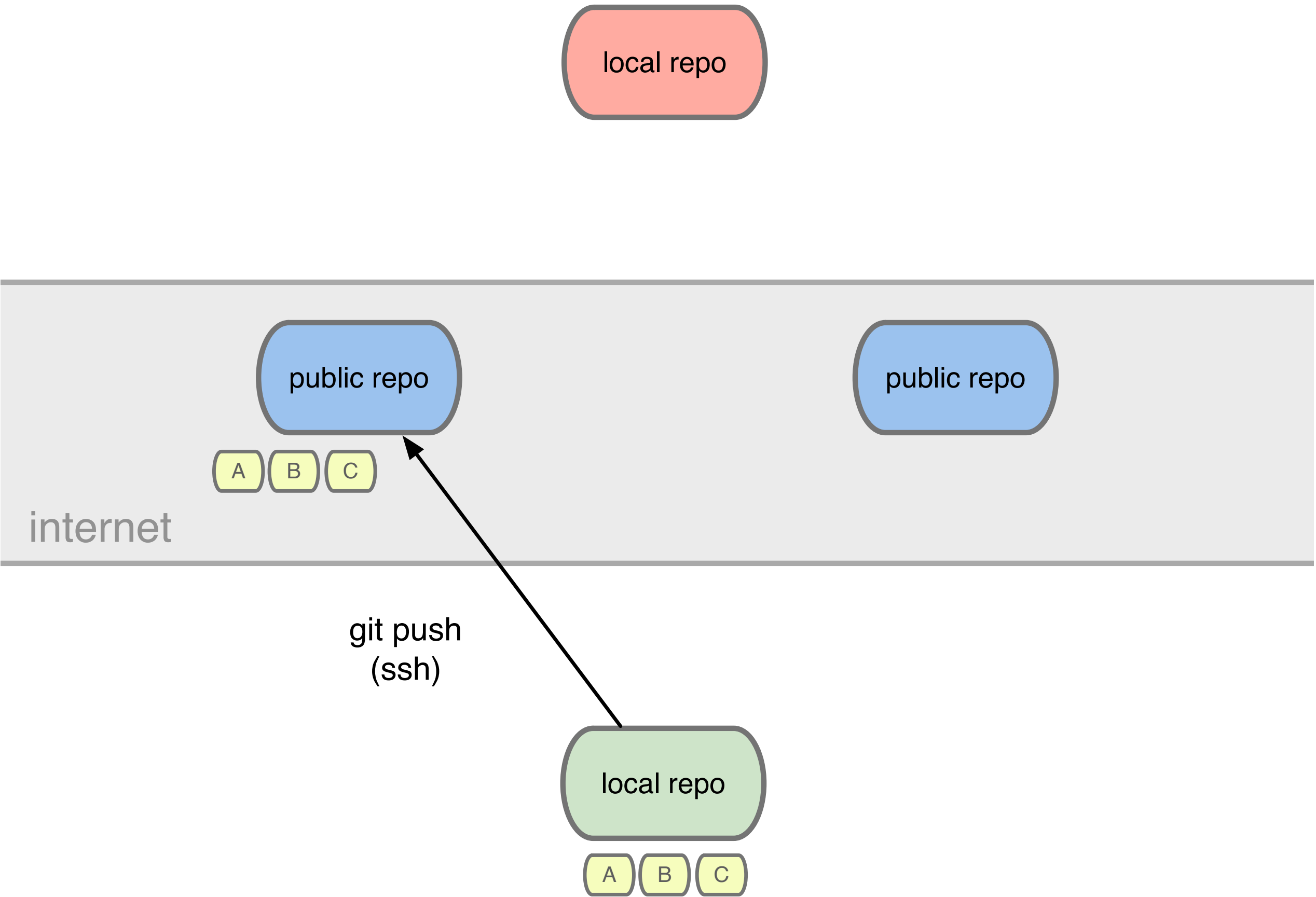
# git push

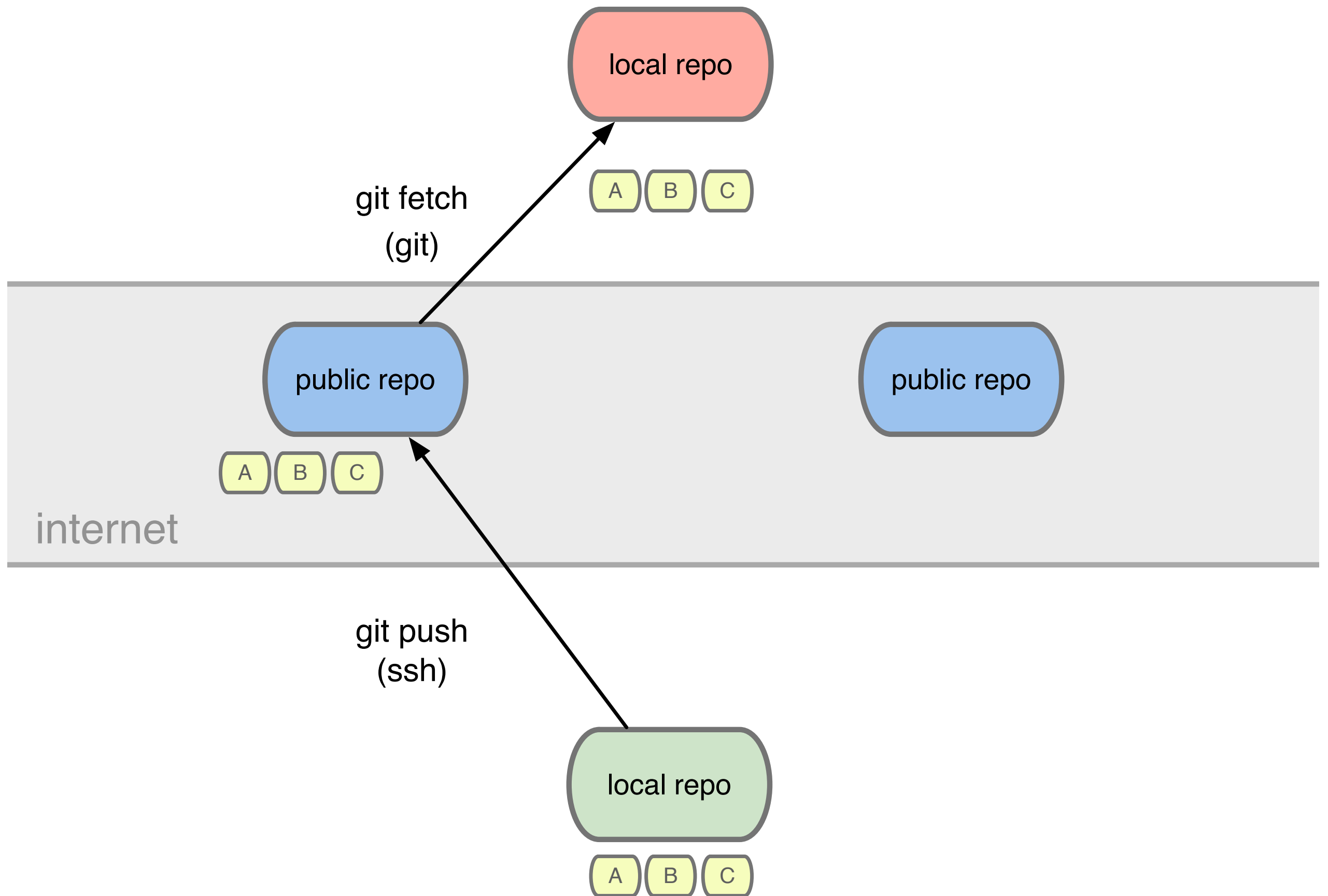


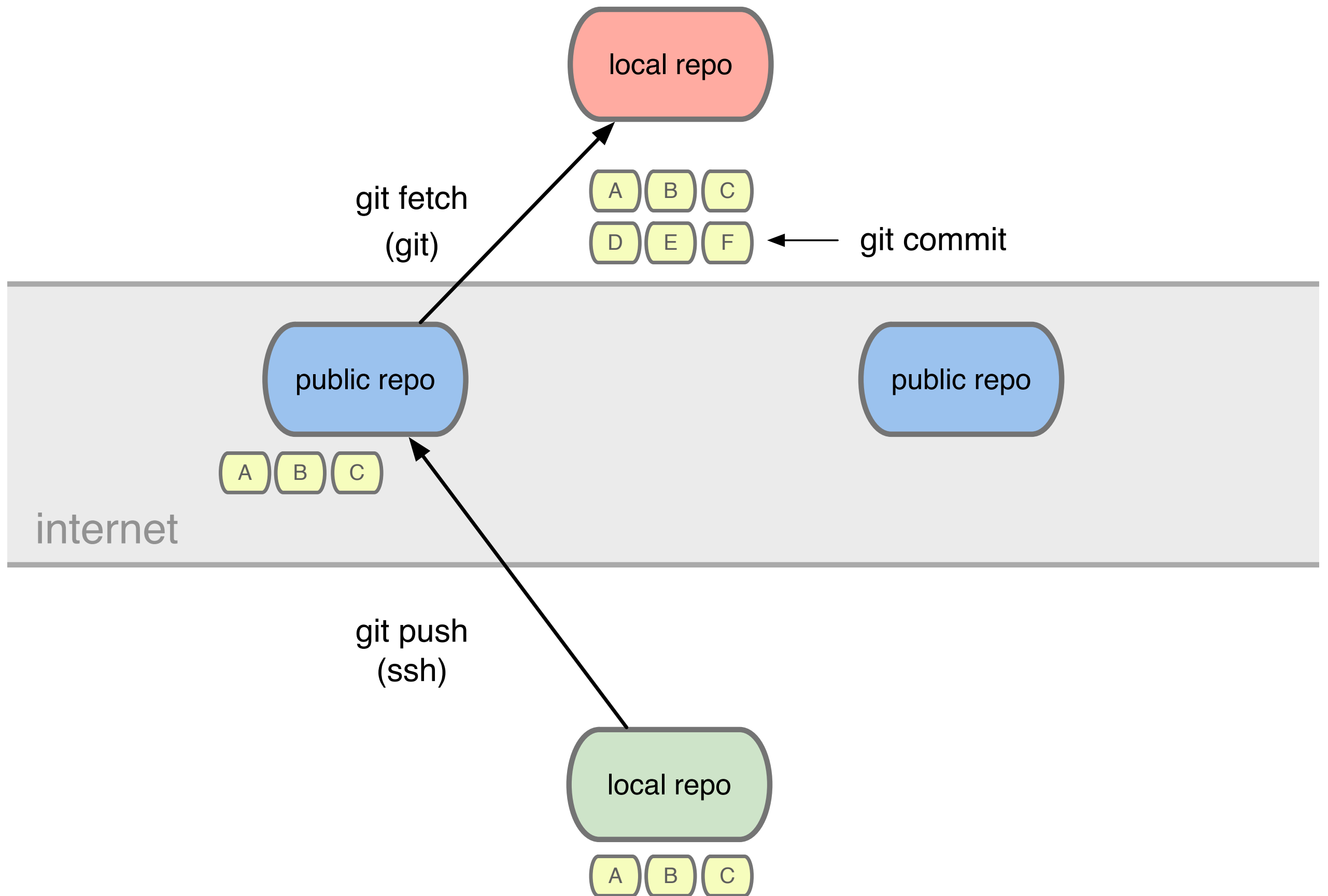
```
$>git remote
public
$>git remote show public
* remote public
  URL: git@github.com:schacon/simplegit.git
  Tracked remote branches
    master
$>git push public master
Counting objects: 7, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 365 bytes, done.
Total 4 (delta 2),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

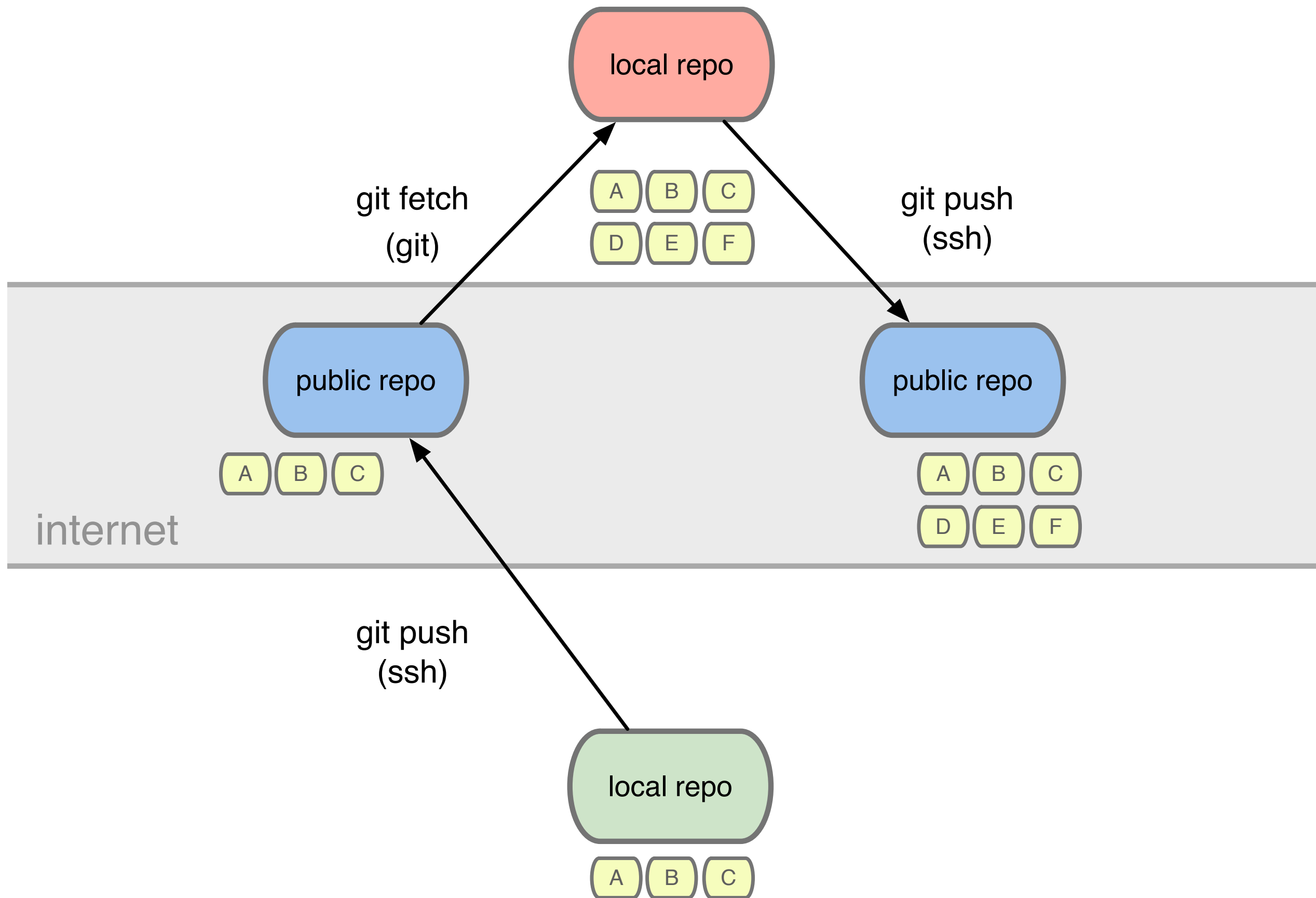
**a who in the what now?**

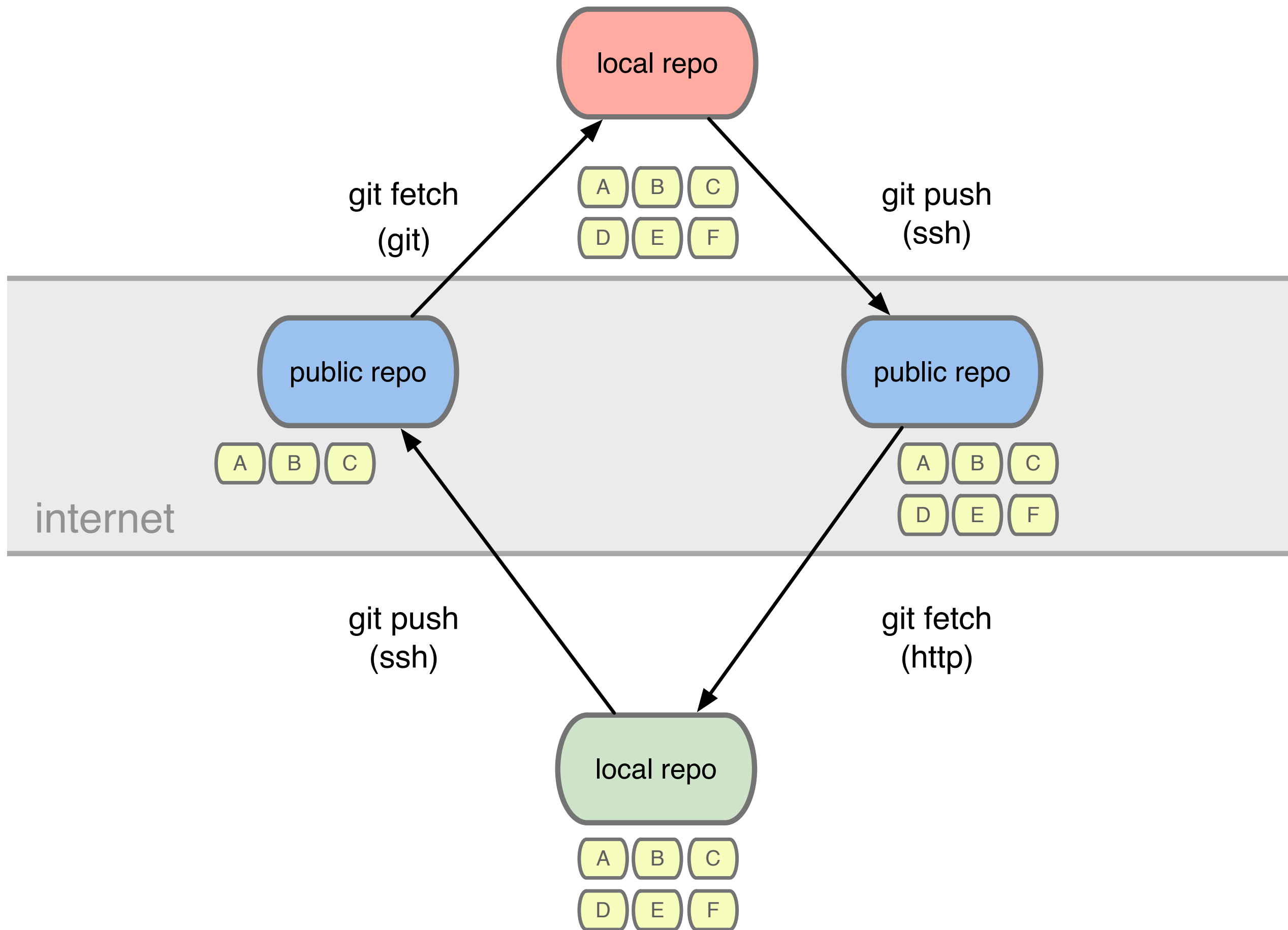




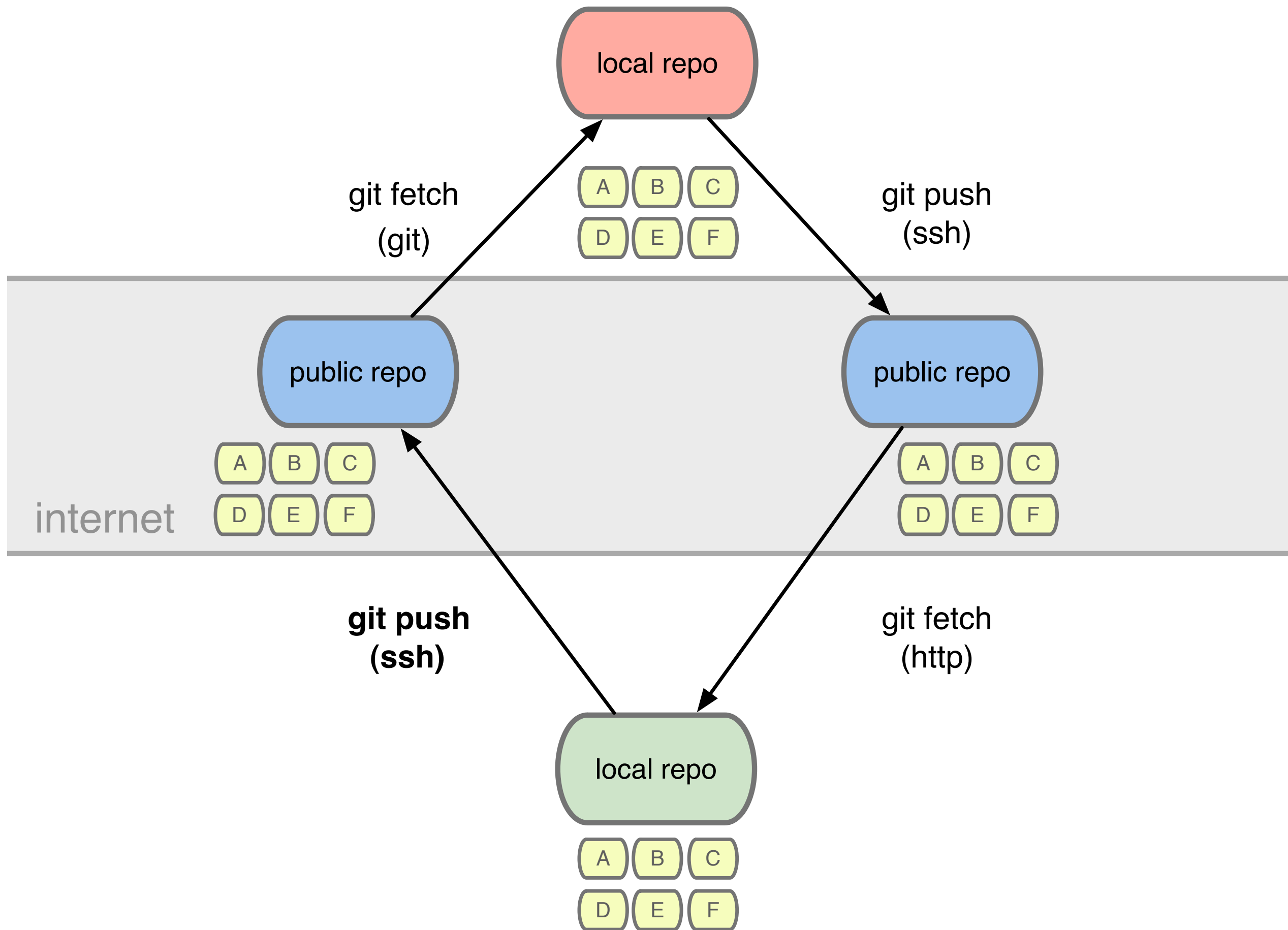








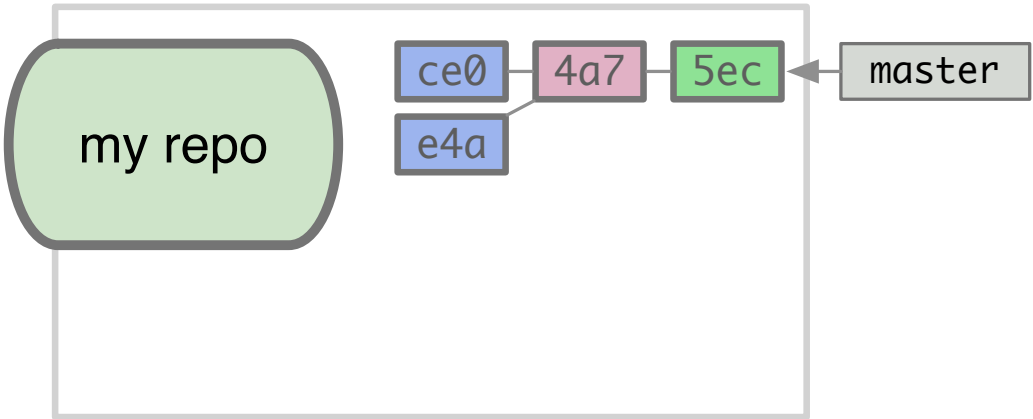
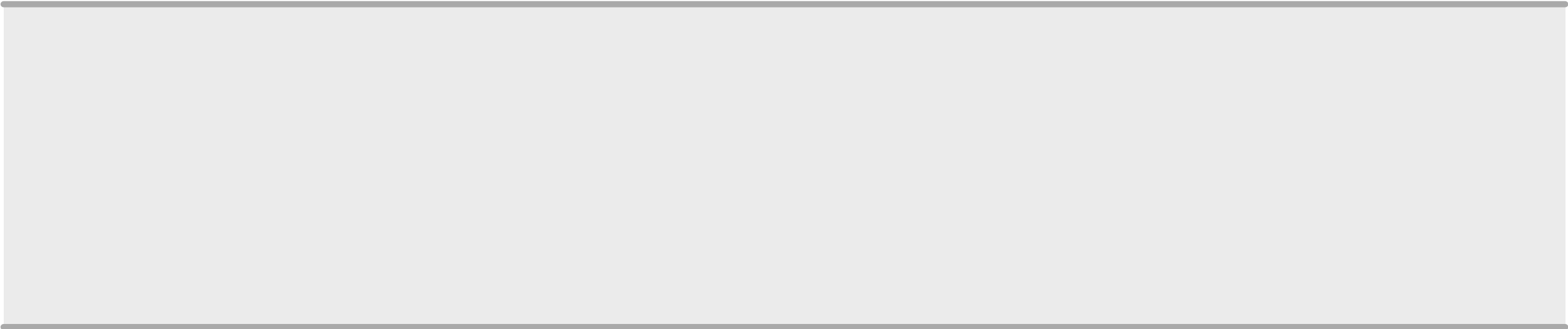




multiple remotes

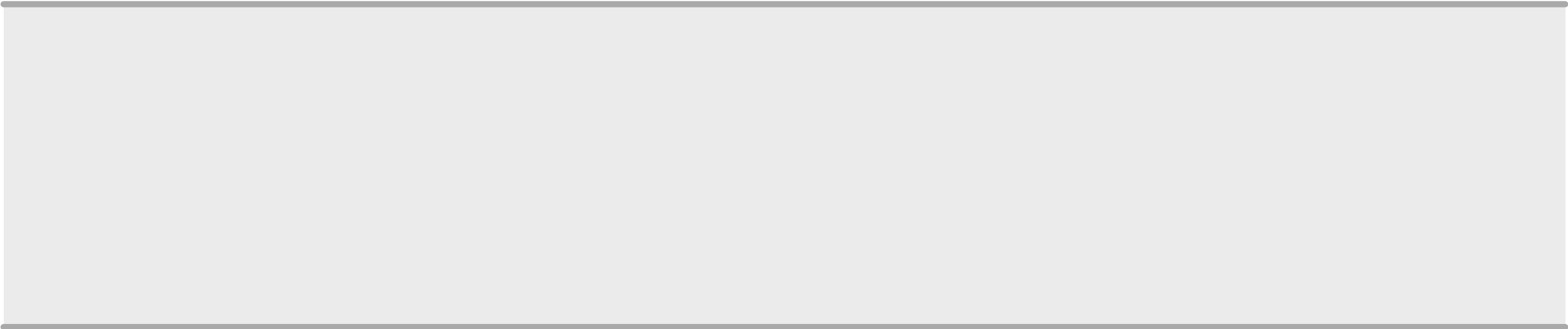
developer  
nick

developer  
jessica

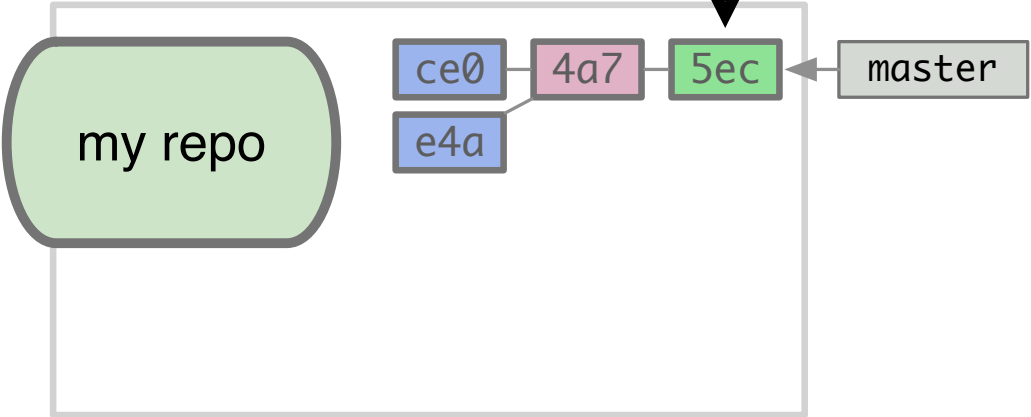


developer  
nick

developer  
jessica

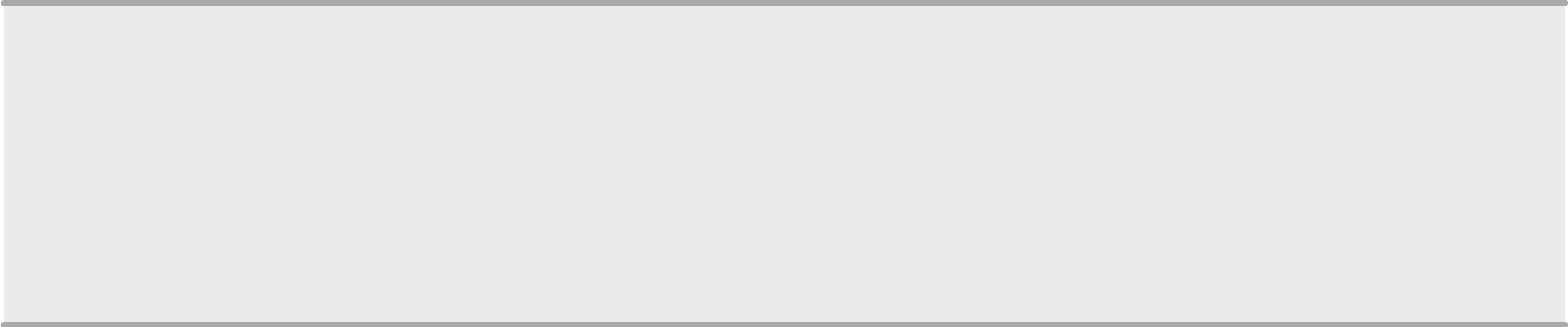


commit

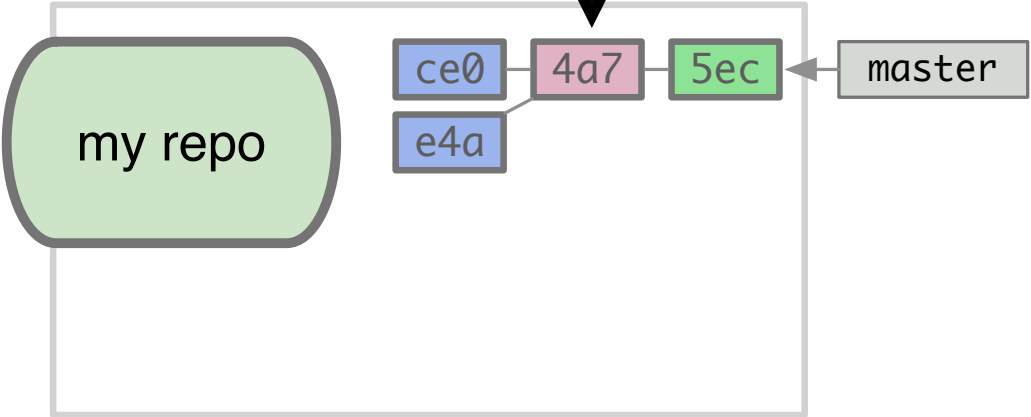


developer  
nick

developer  
jessica

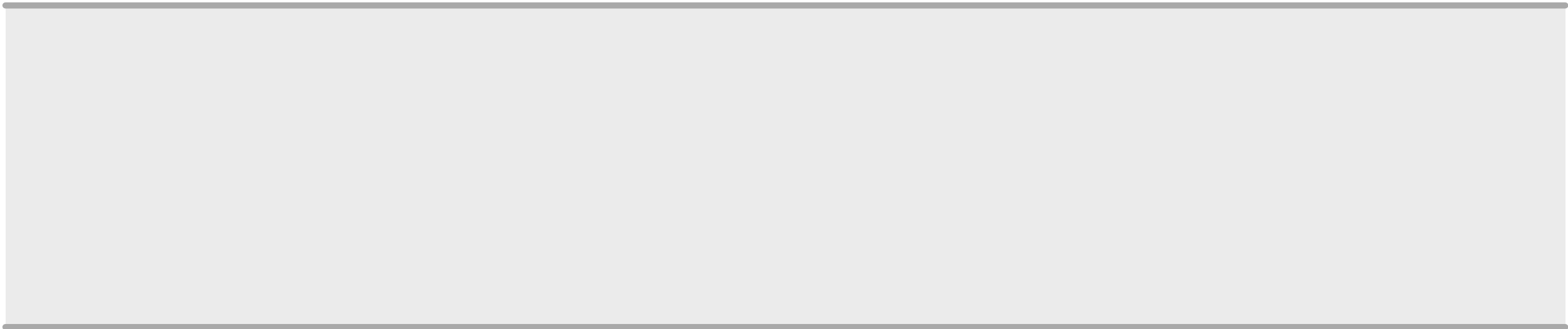


tree

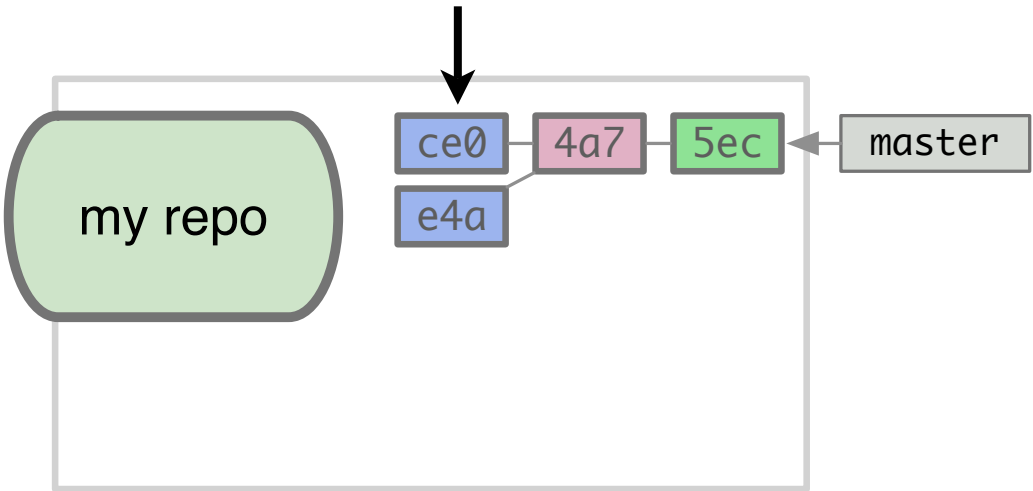


developer  
nick

developer  
jessica

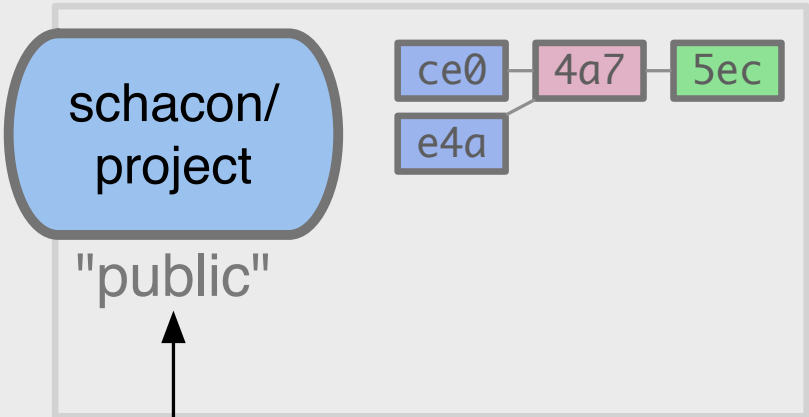


blobs

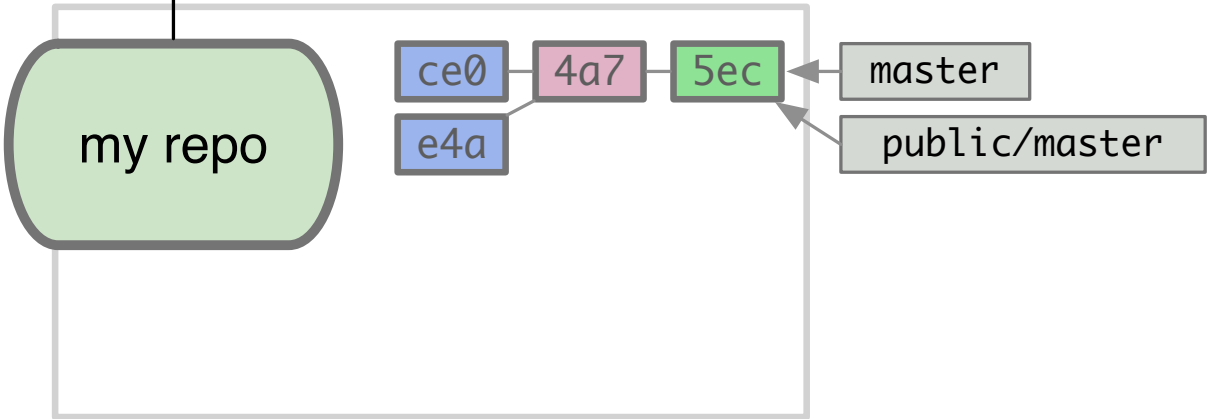


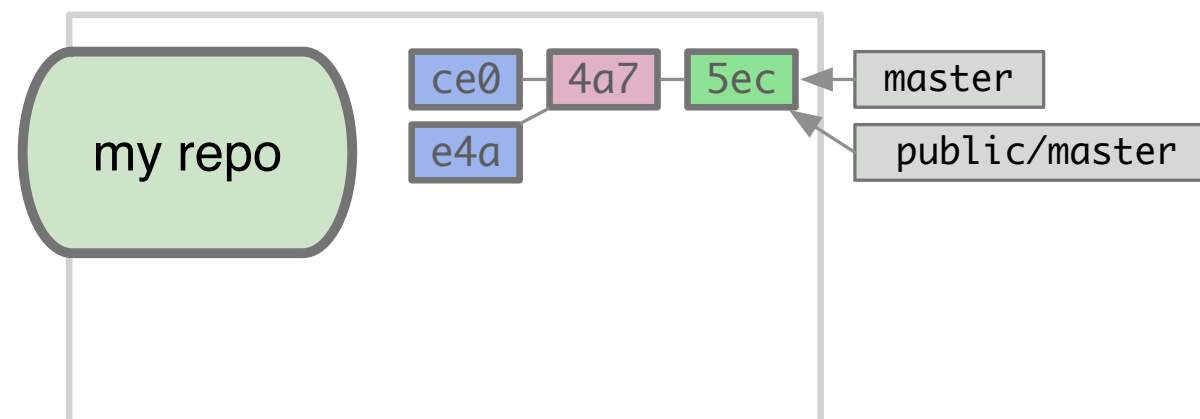
developer  
nick

developer  
jessica

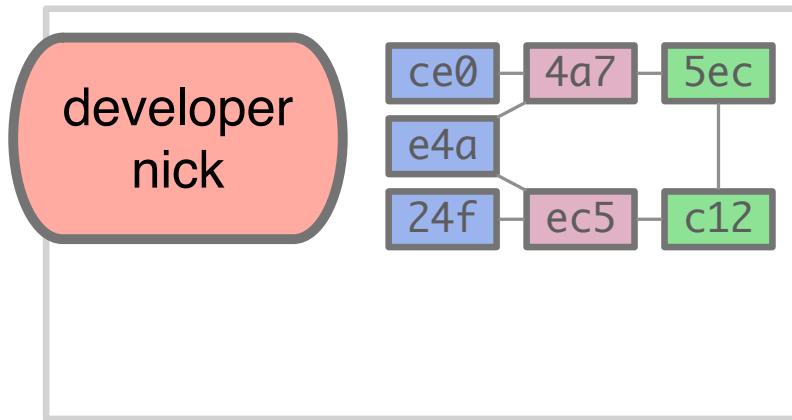


git push public

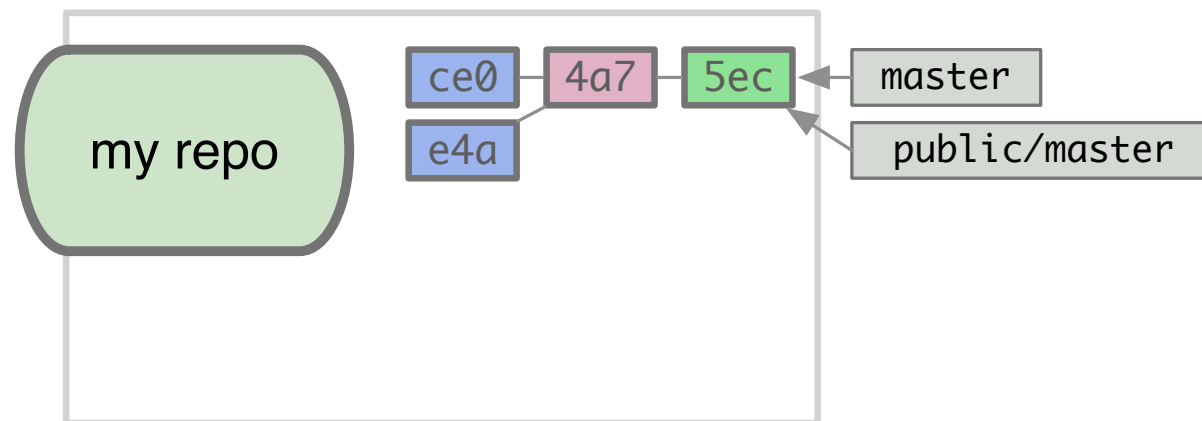
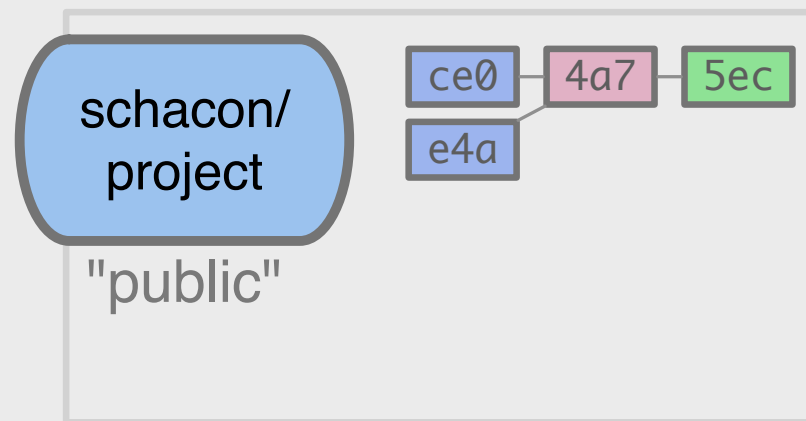


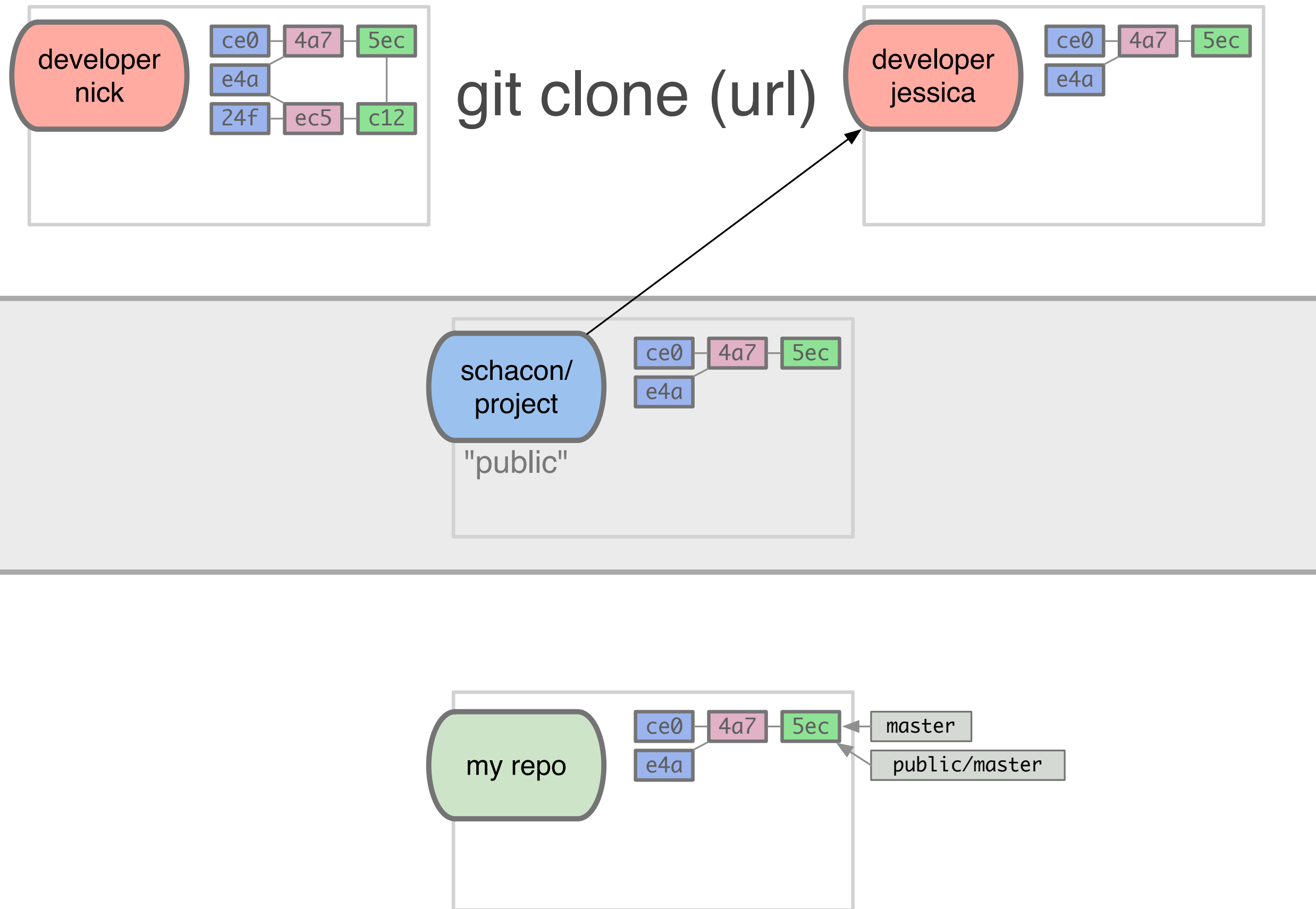


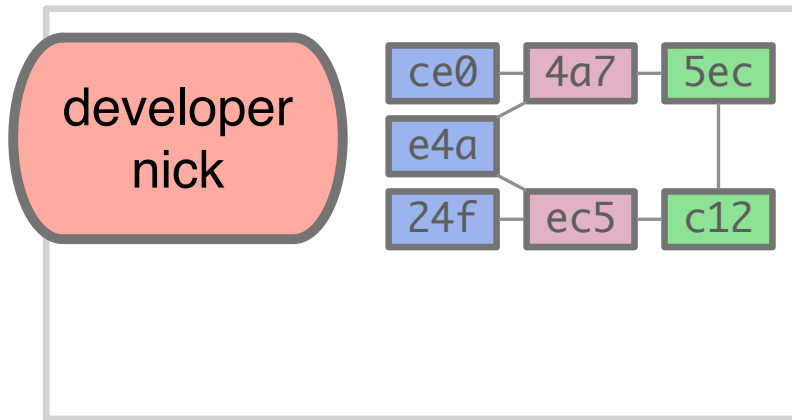




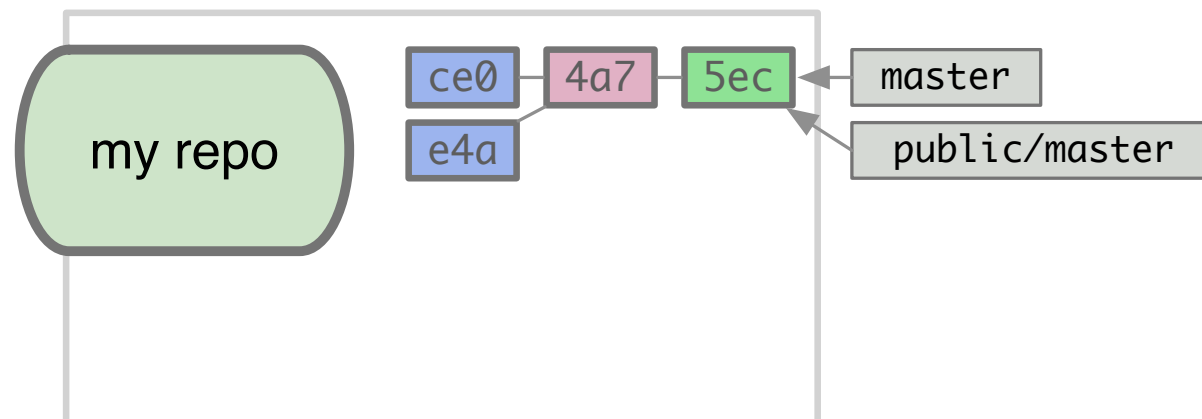
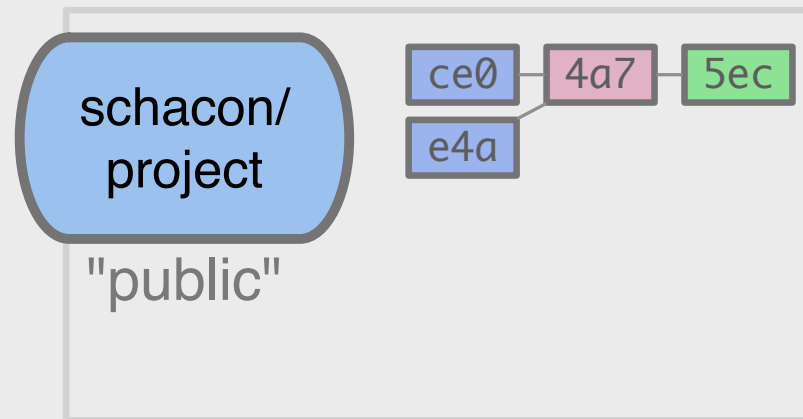
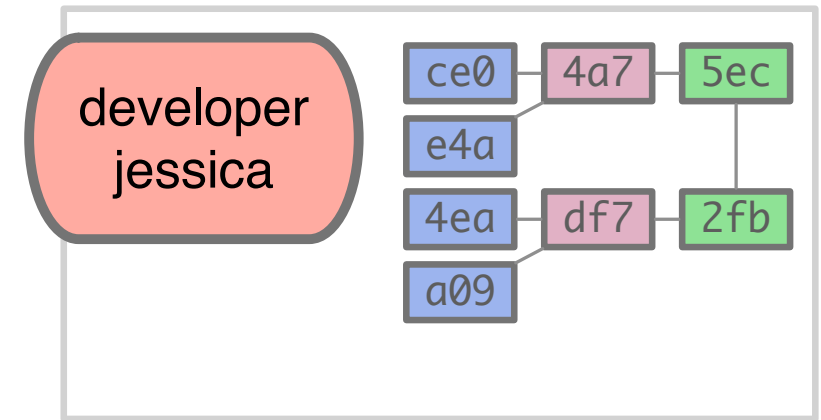
git commit

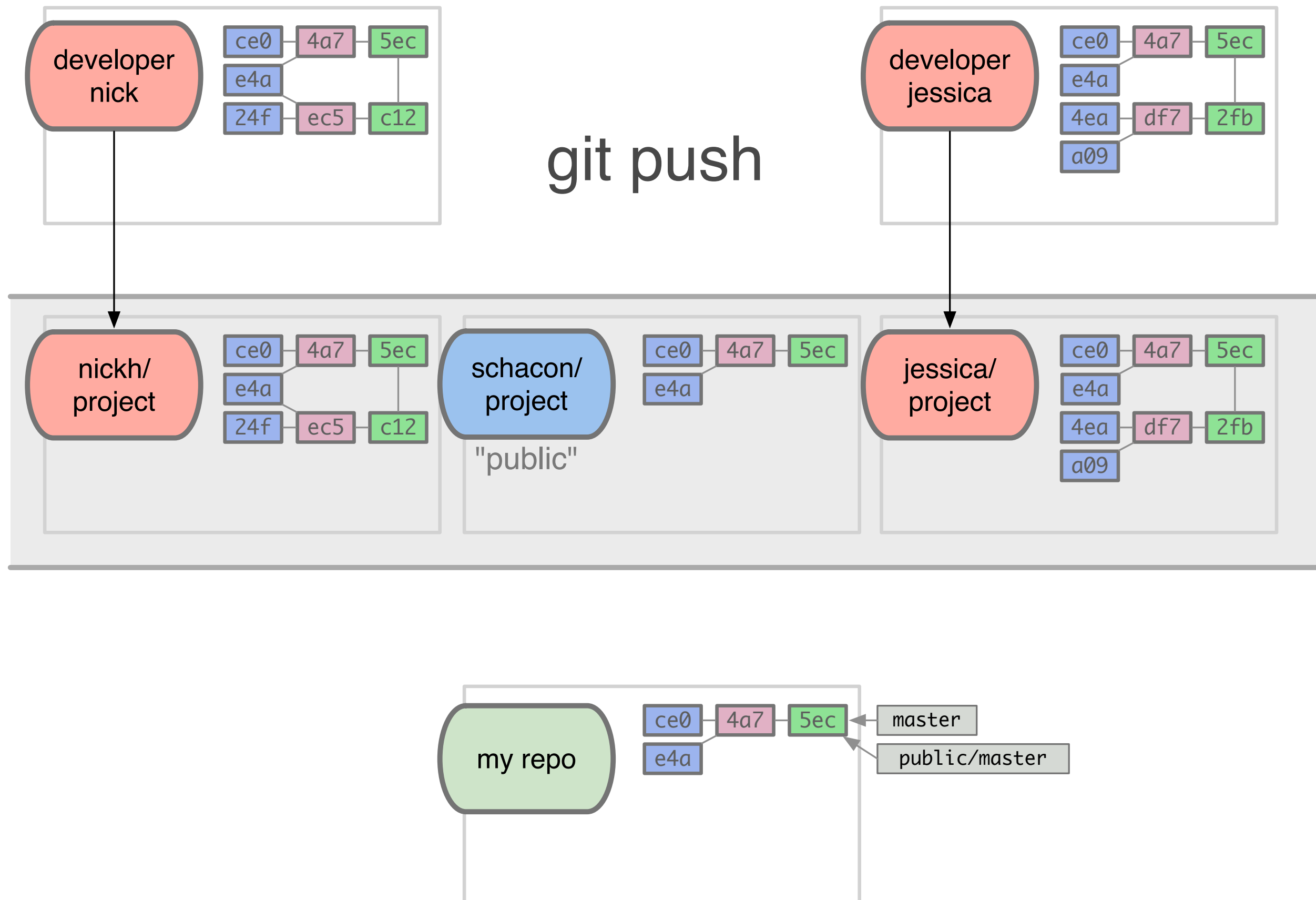


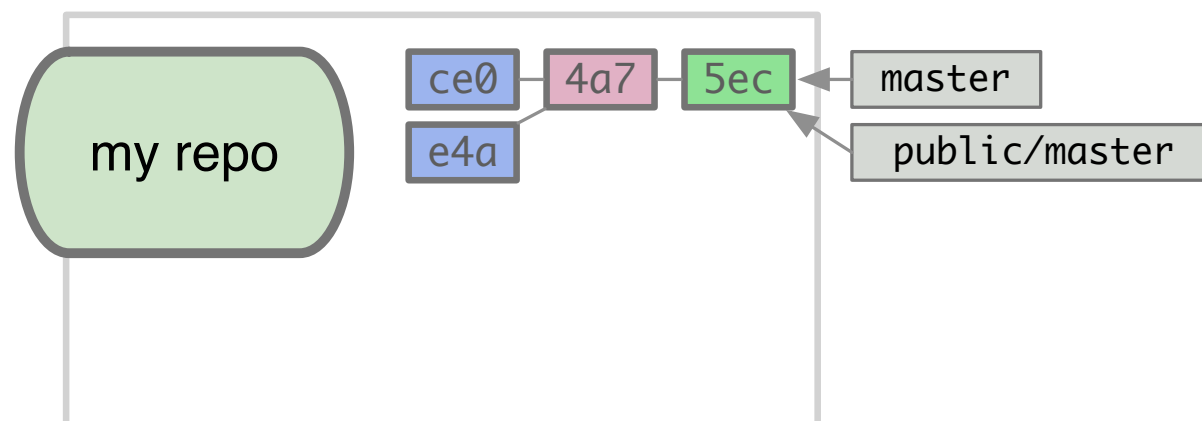
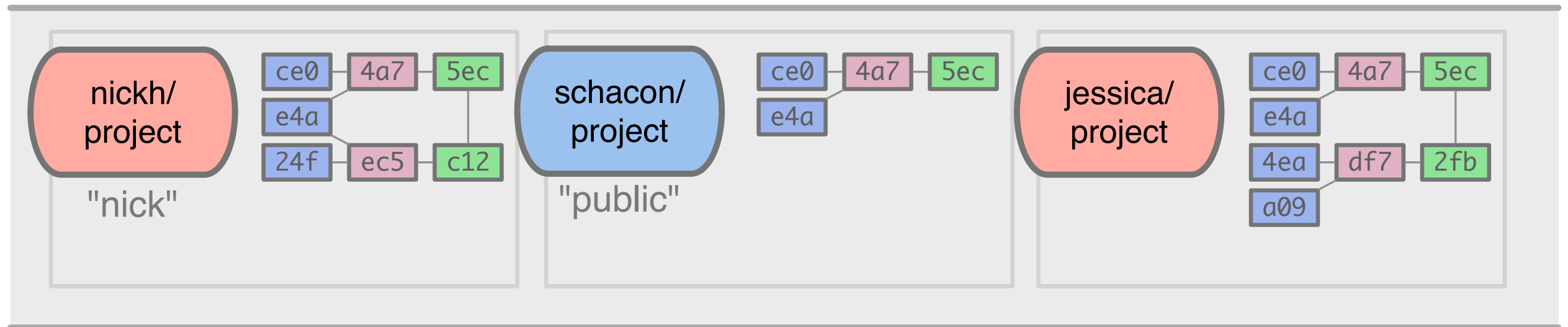
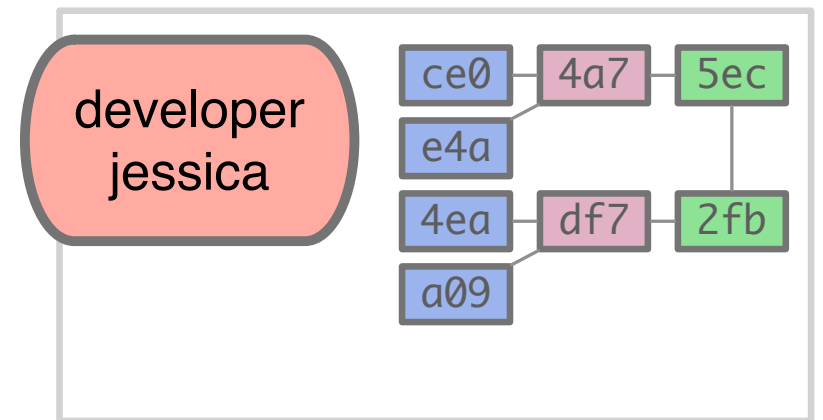
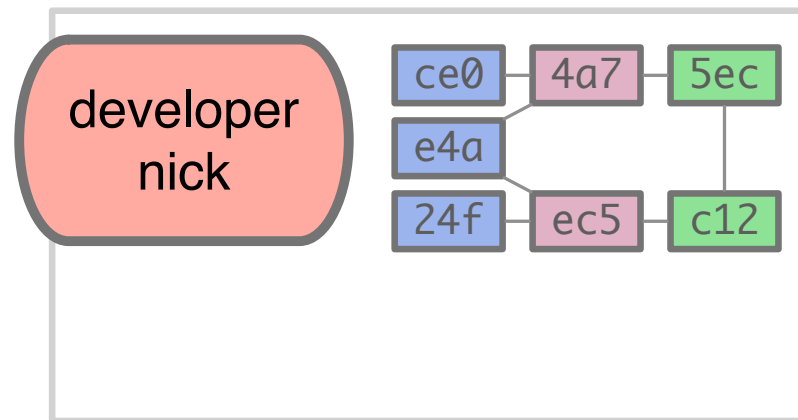




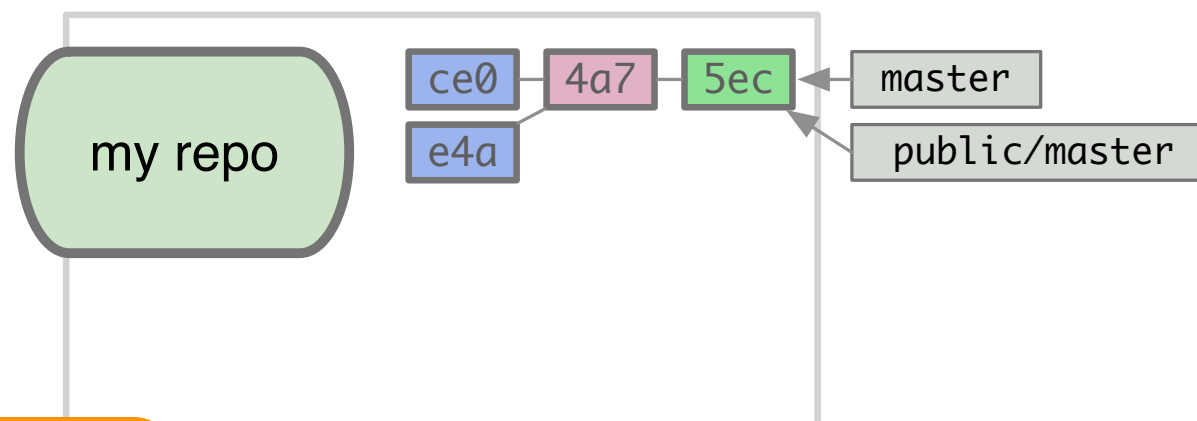
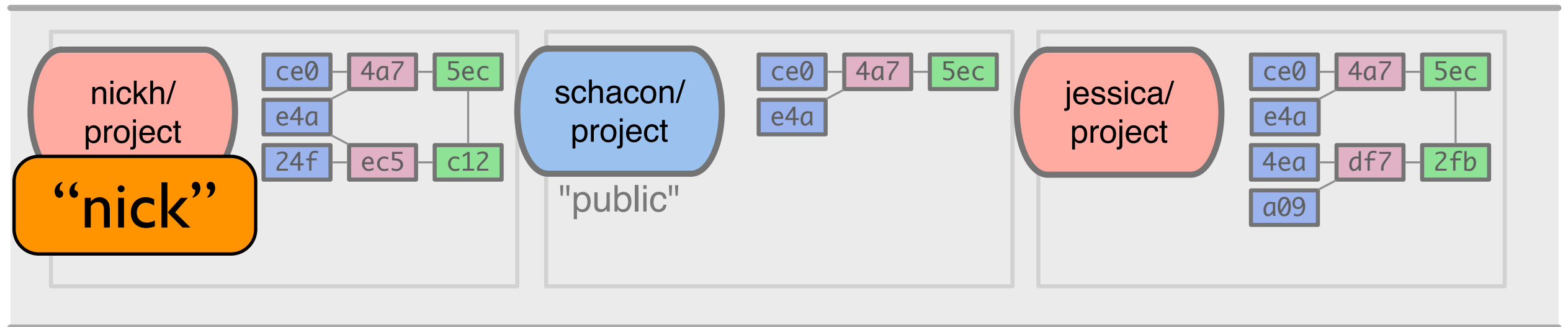
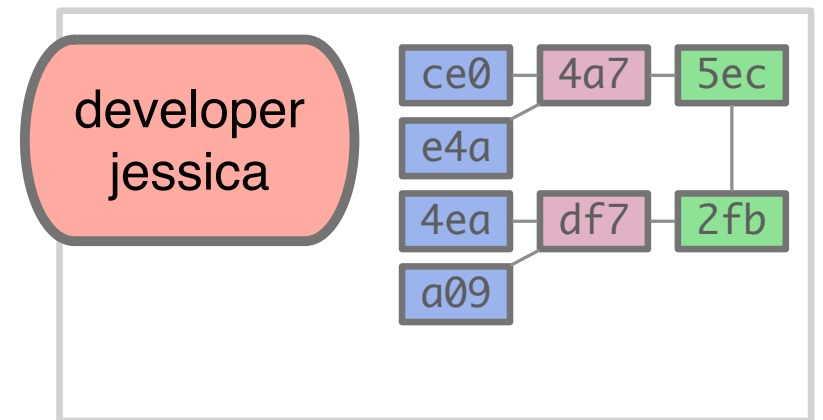
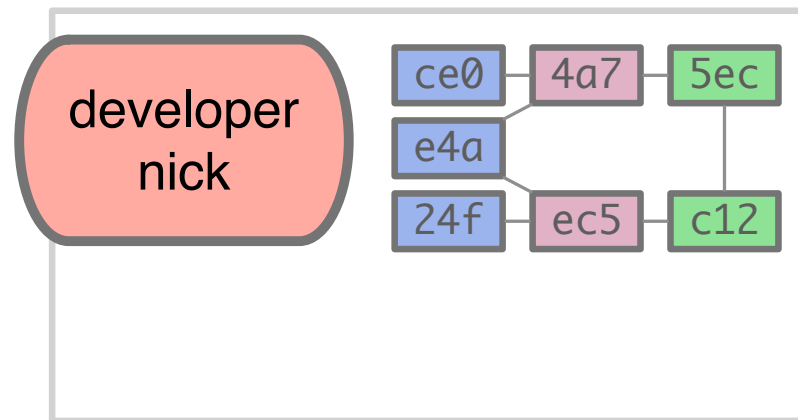
git commit



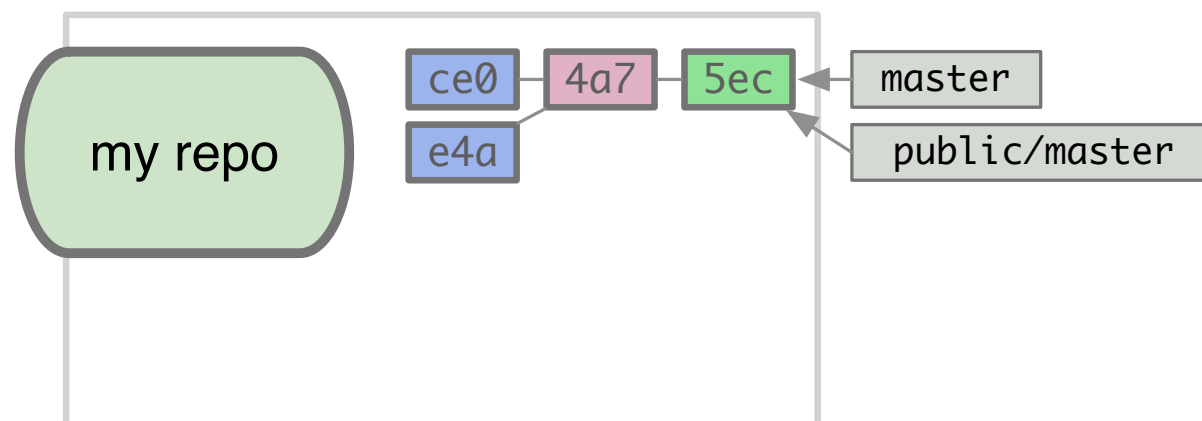
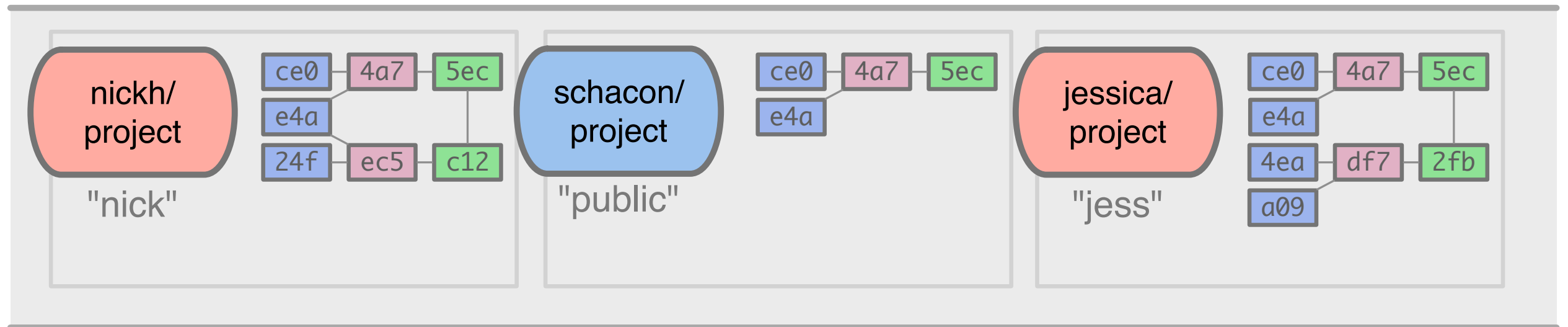
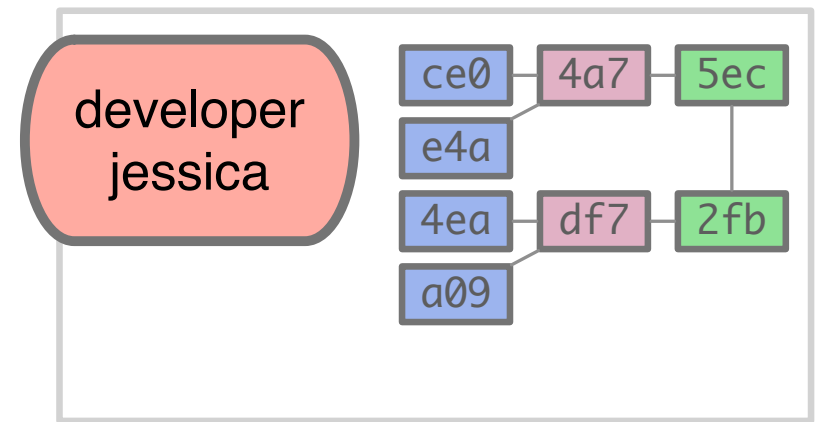
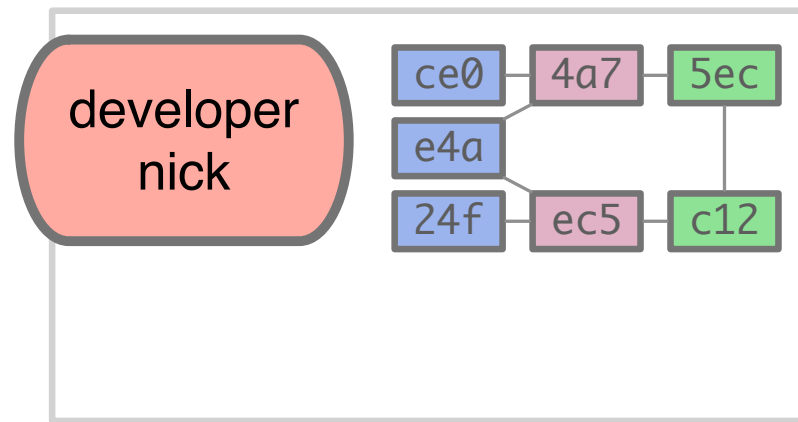




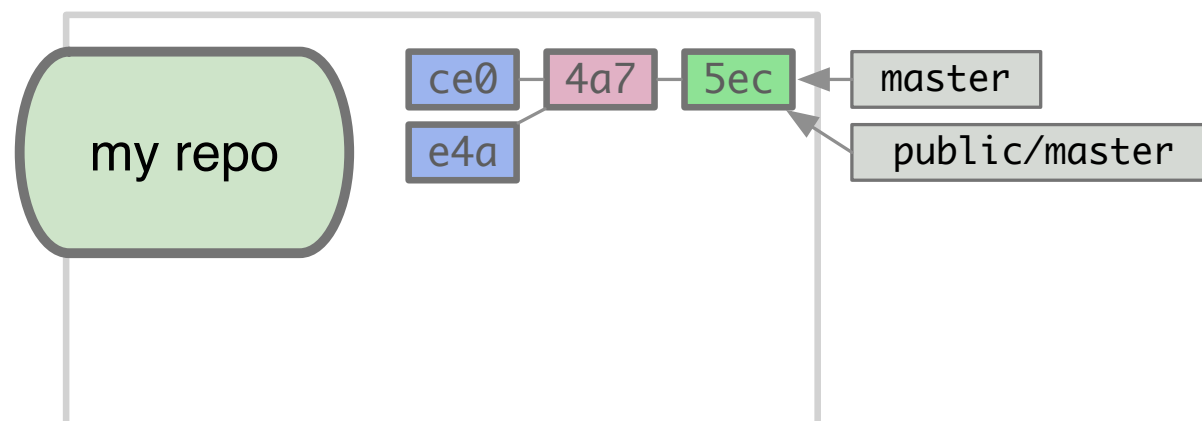
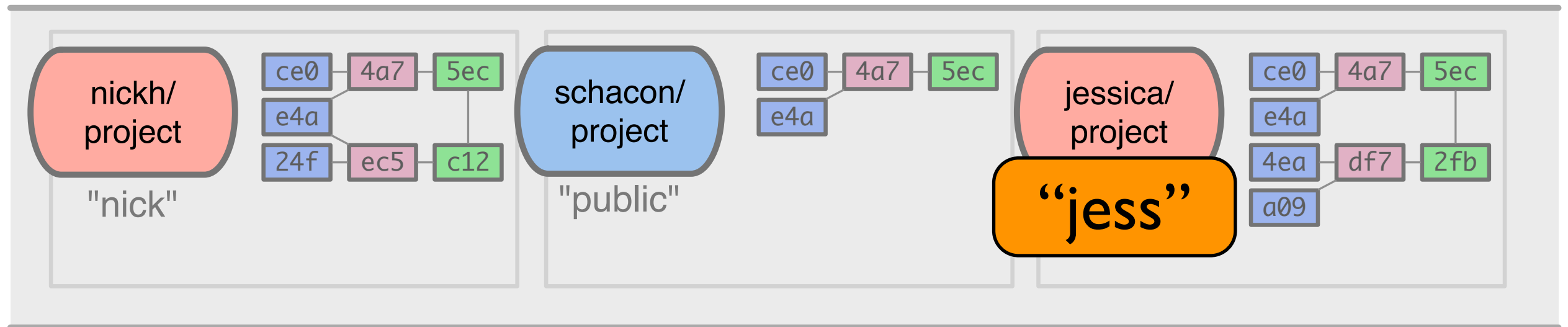
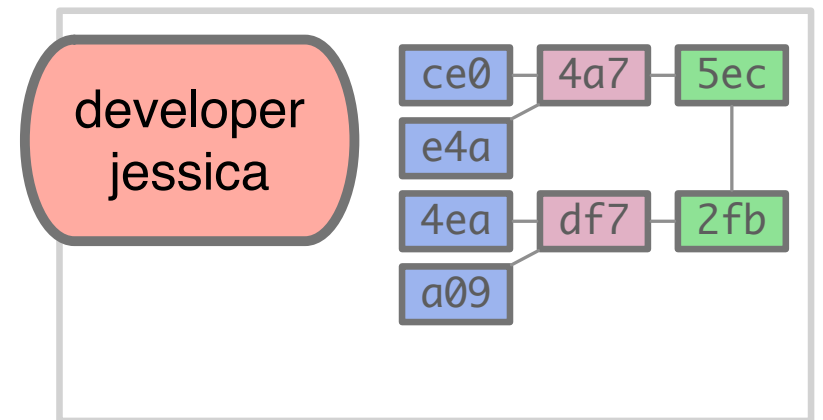
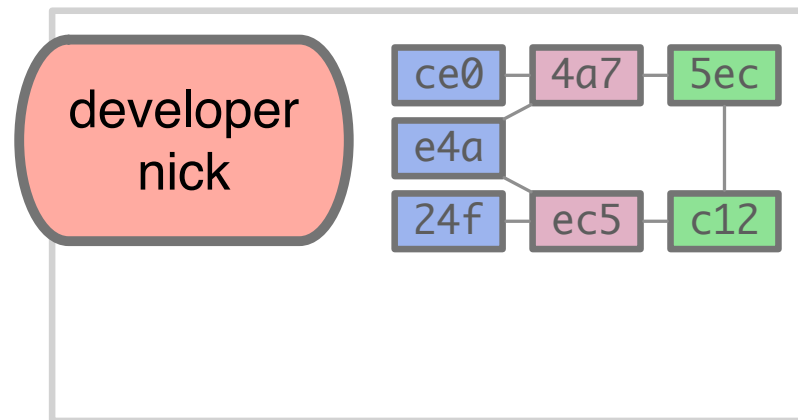
git remote add nick git://github.com/nickh/project.git



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

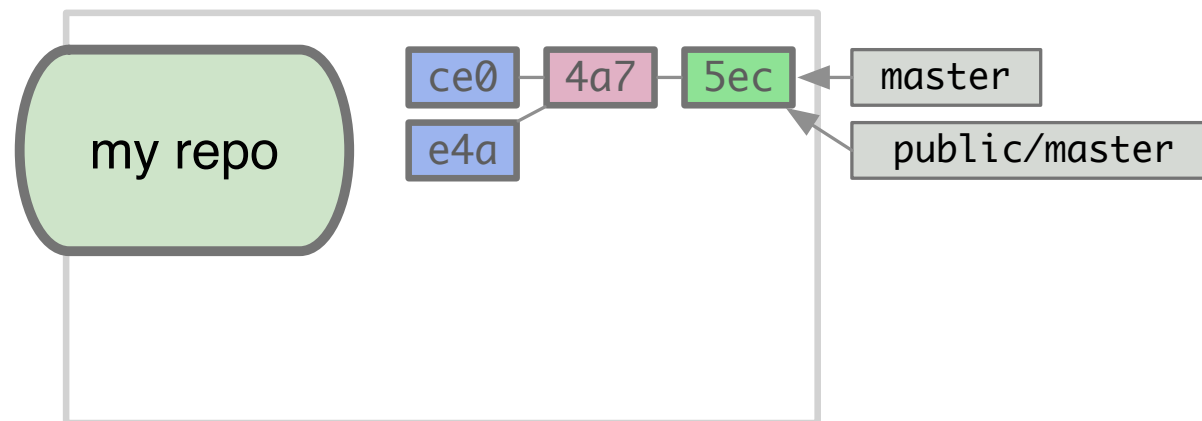
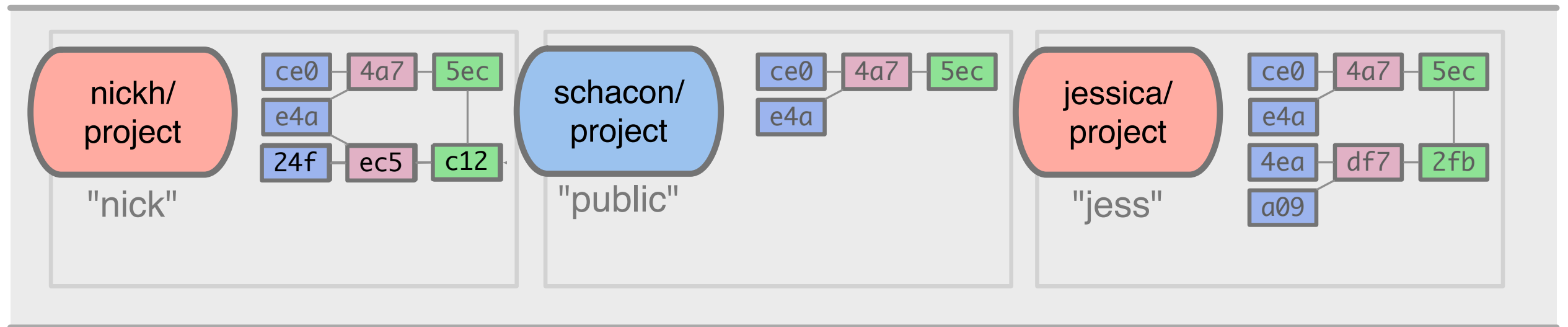
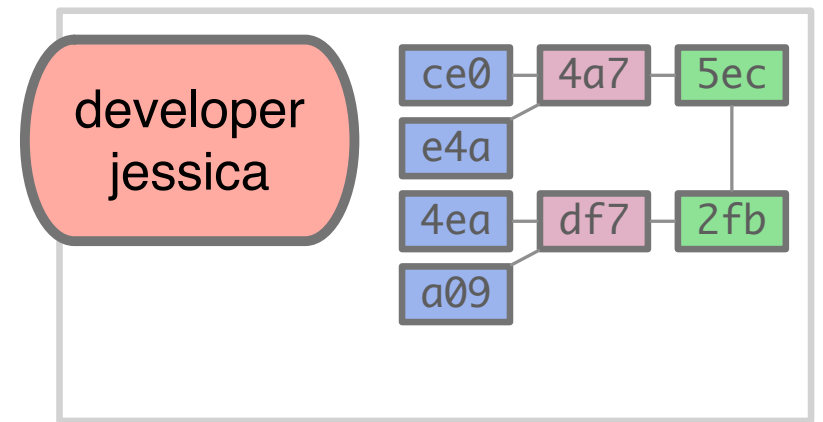
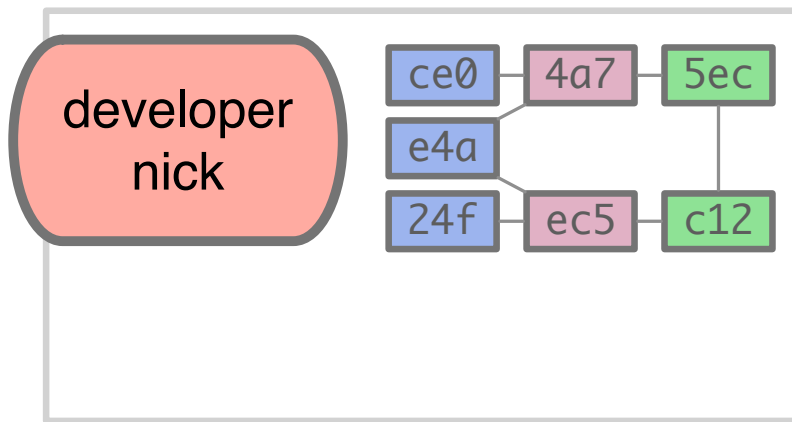


git remote add jess git://github.com/jessica/project.git

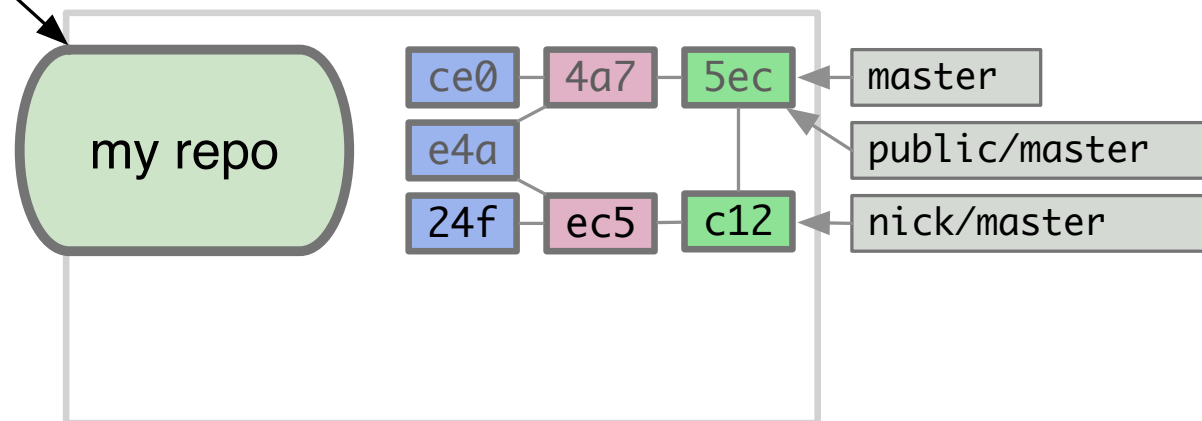
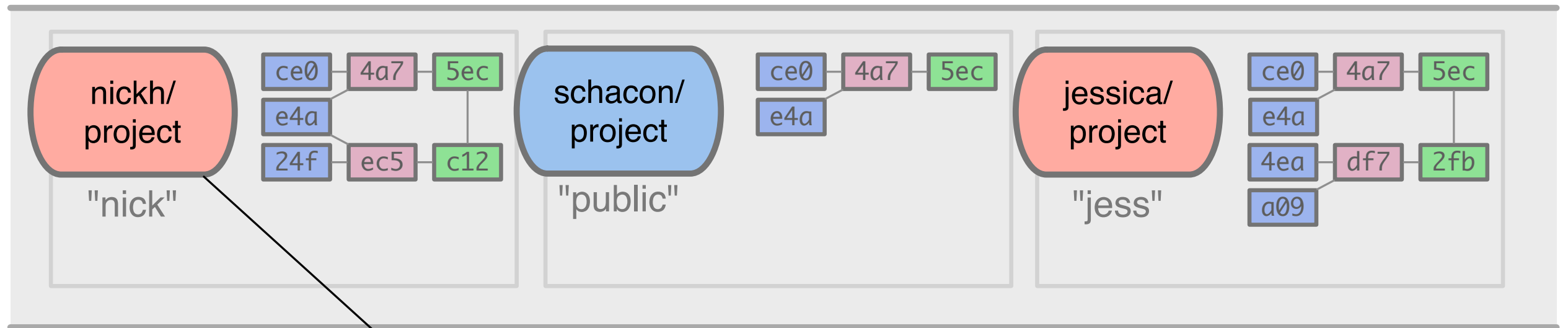
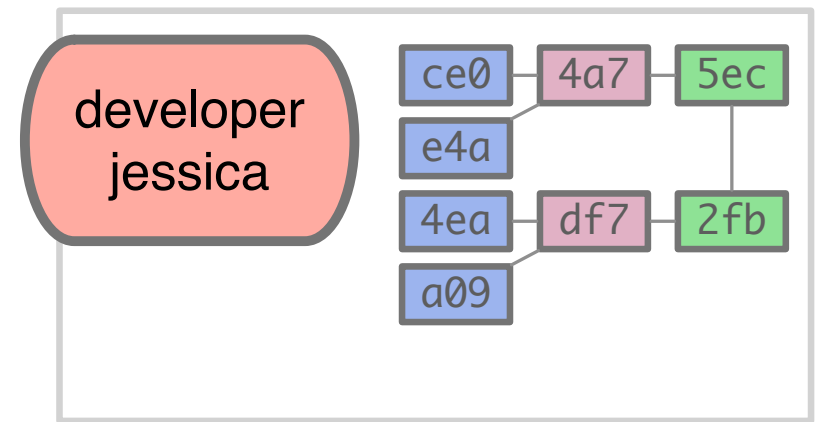
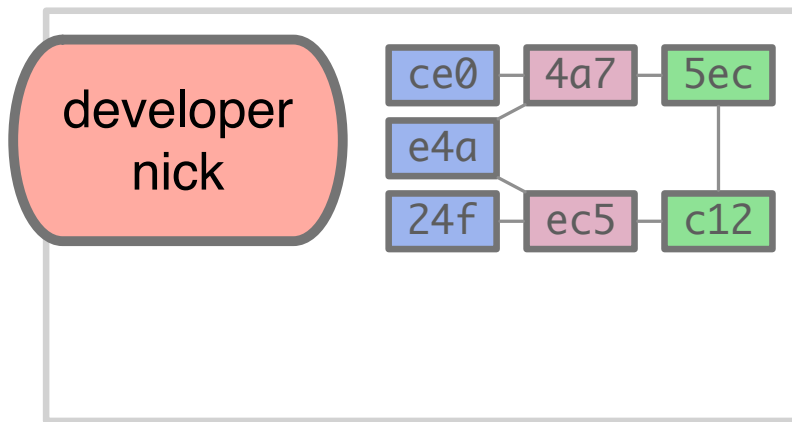


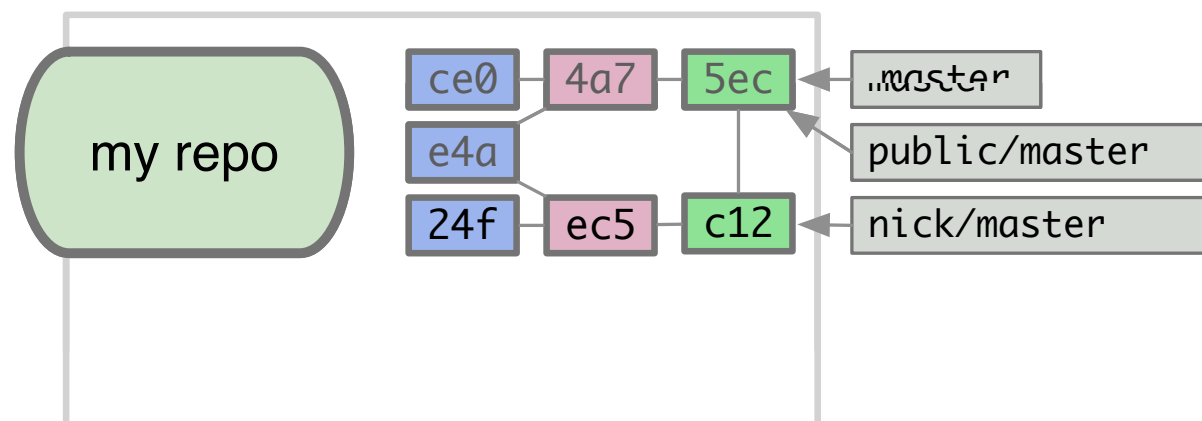
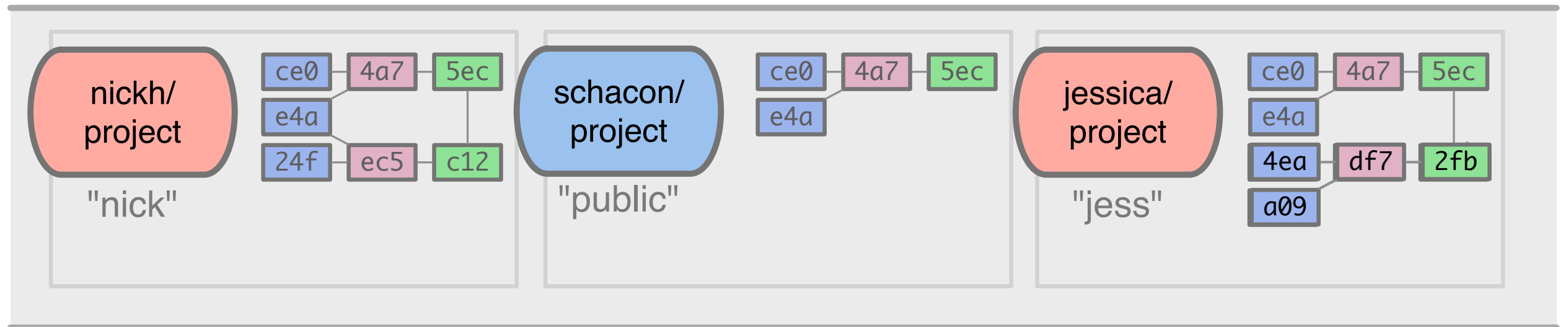
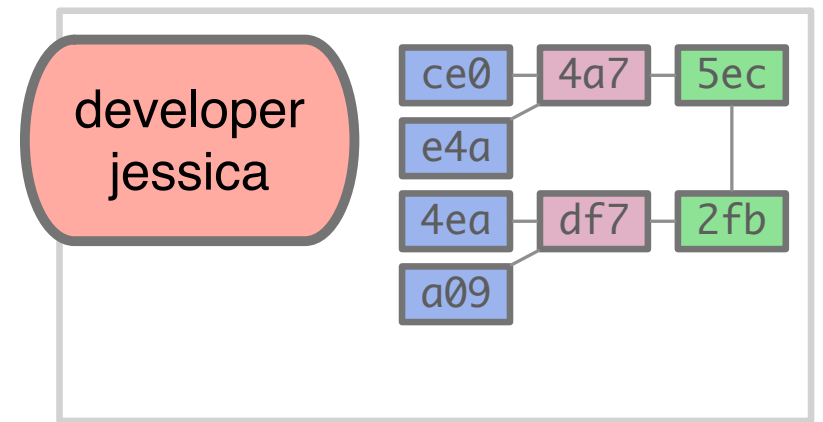
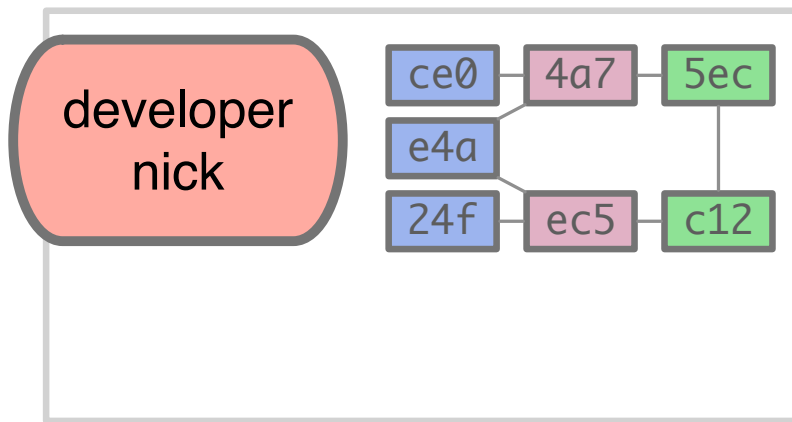
git remote add **jess** git://github.com/jessica/project.git



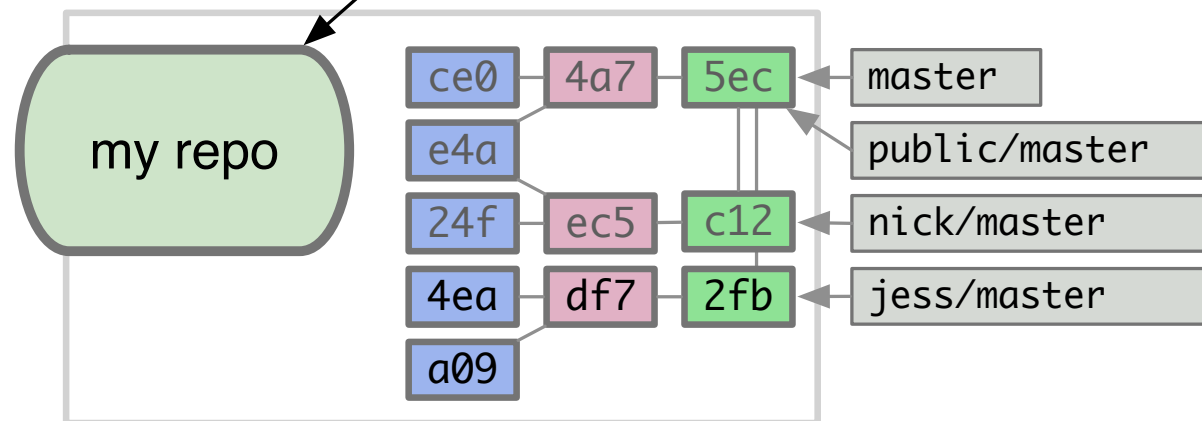
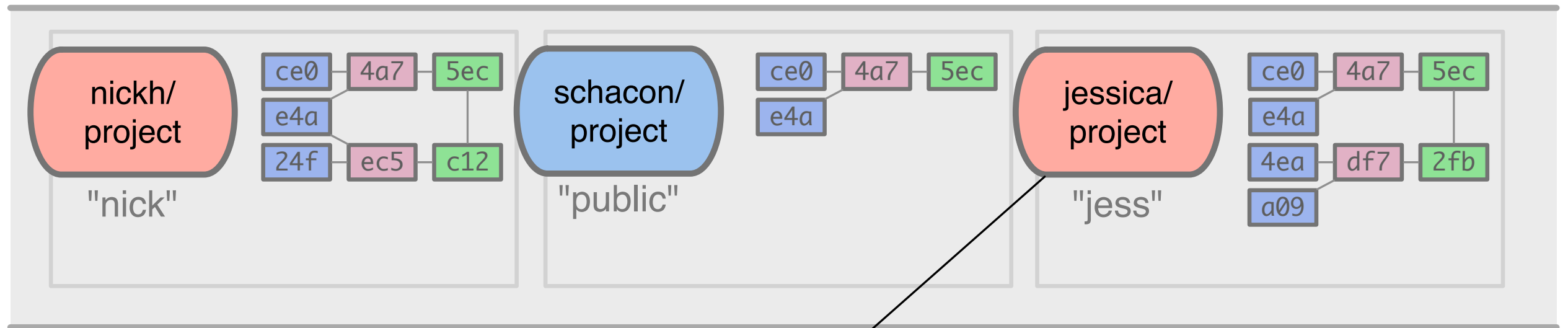
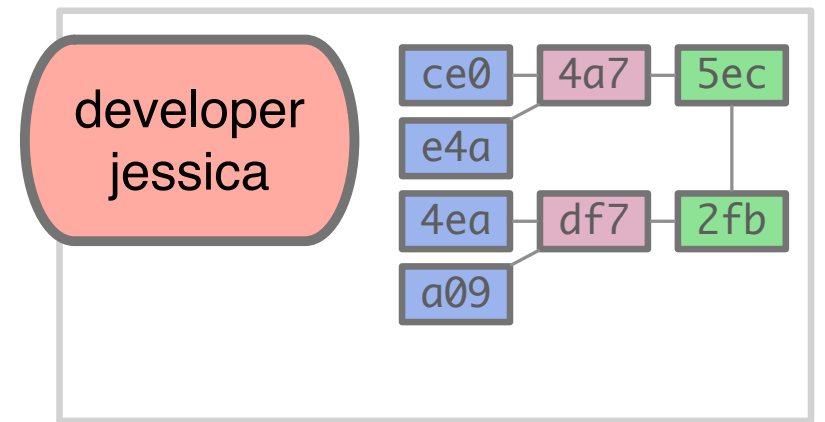
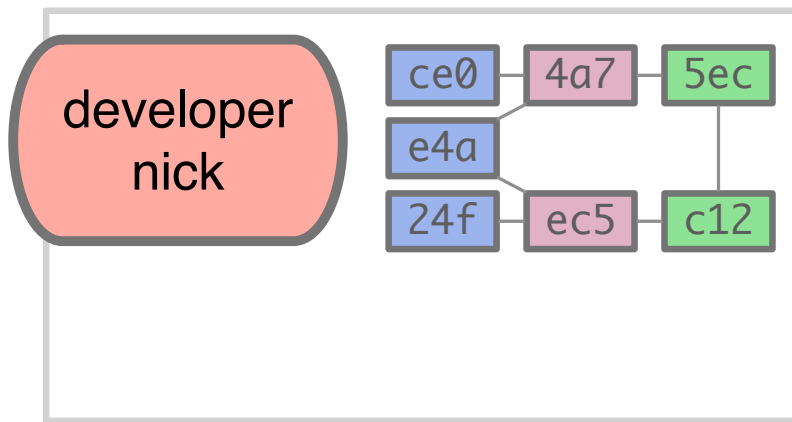


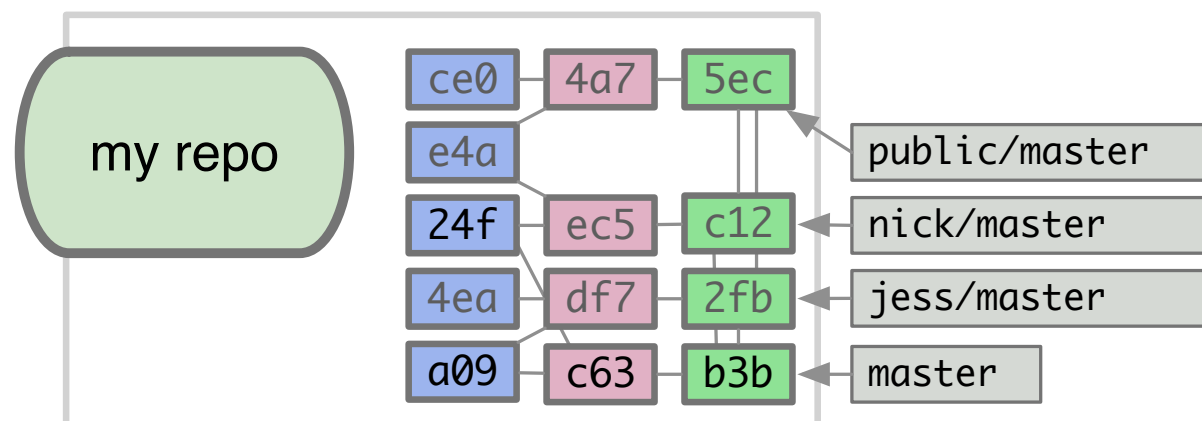
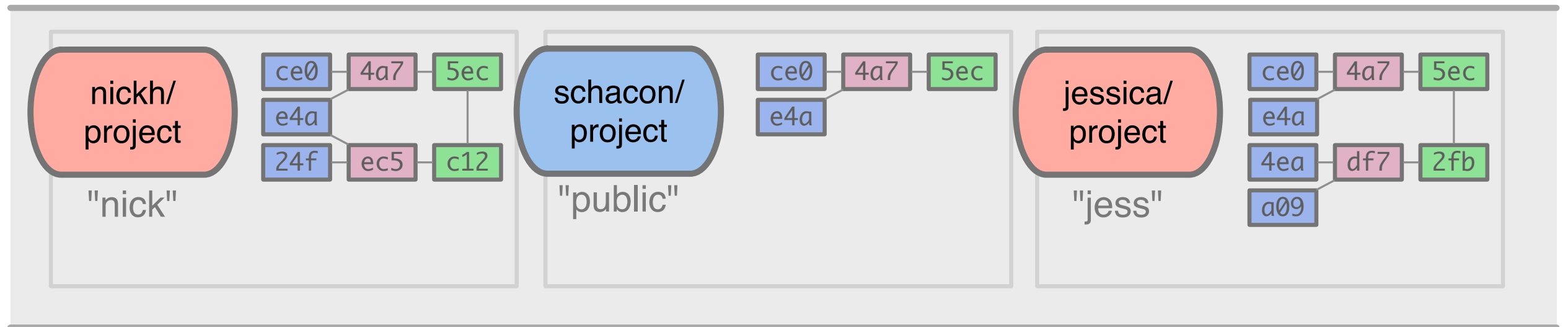
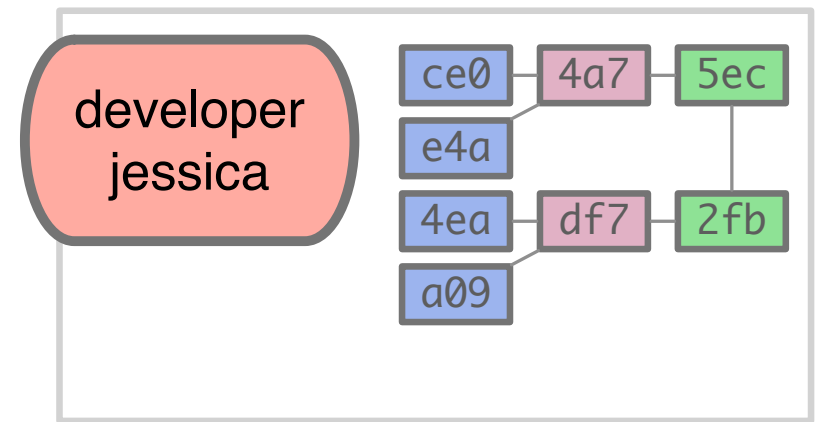
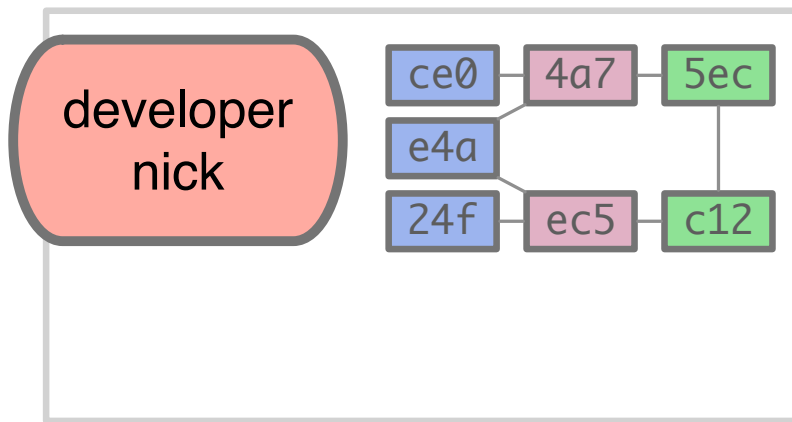
git fetch nick



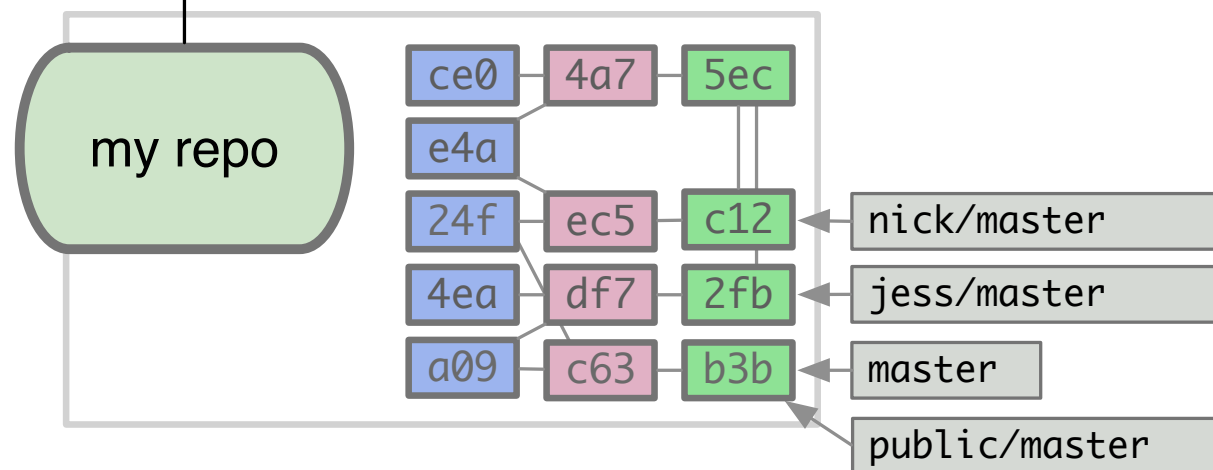
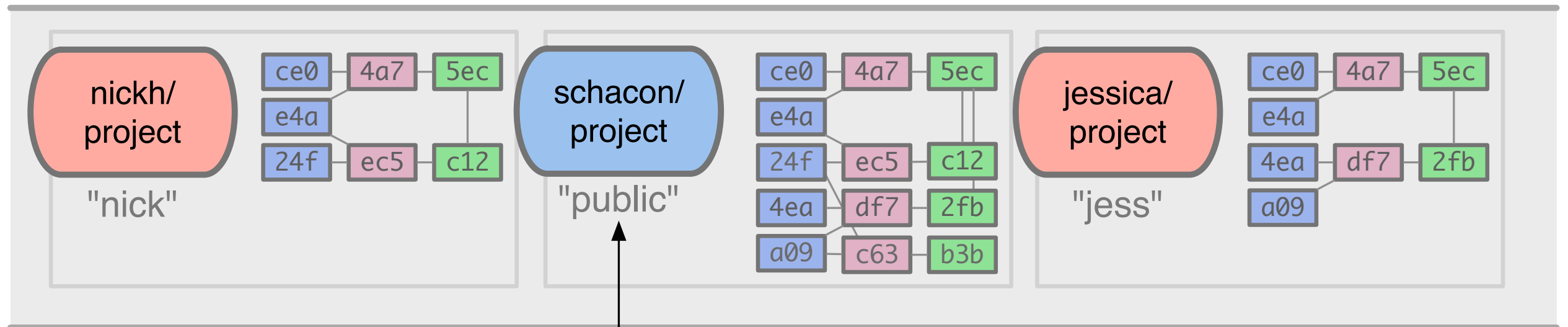
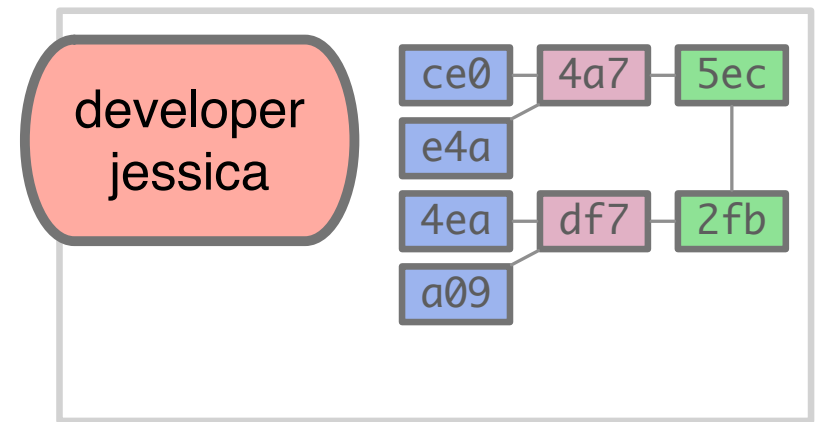
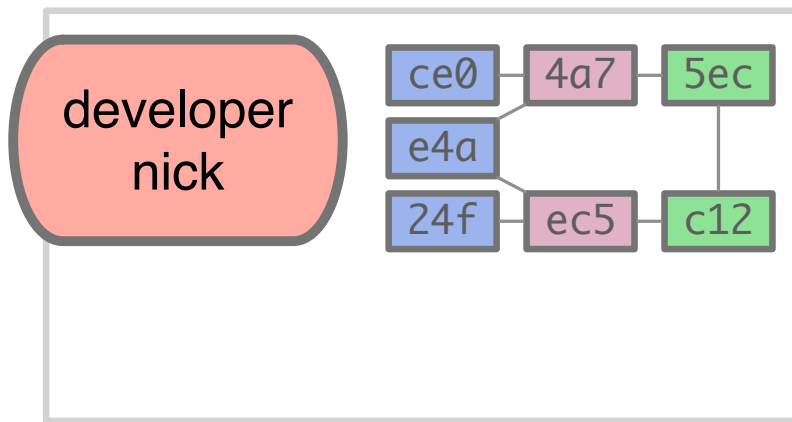


git fetch jess

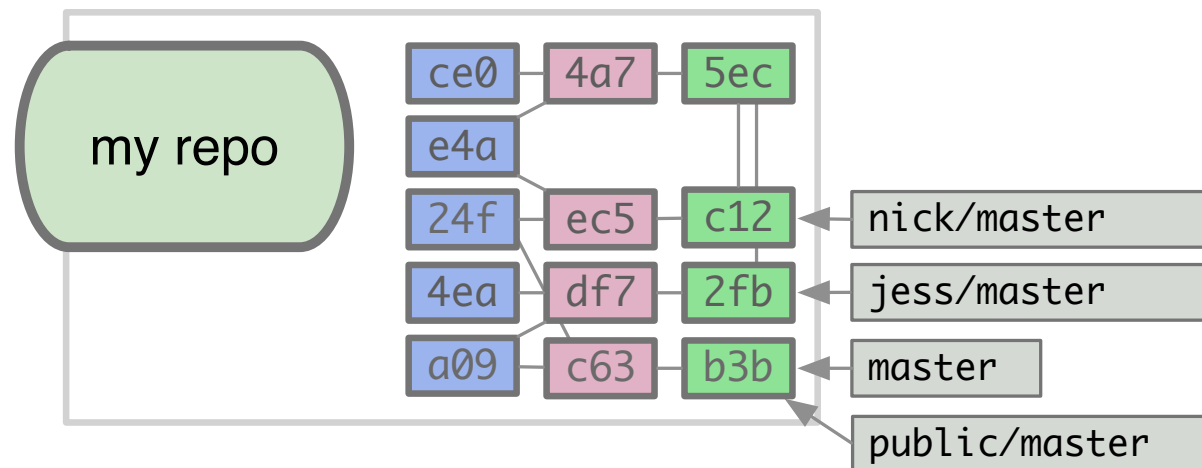
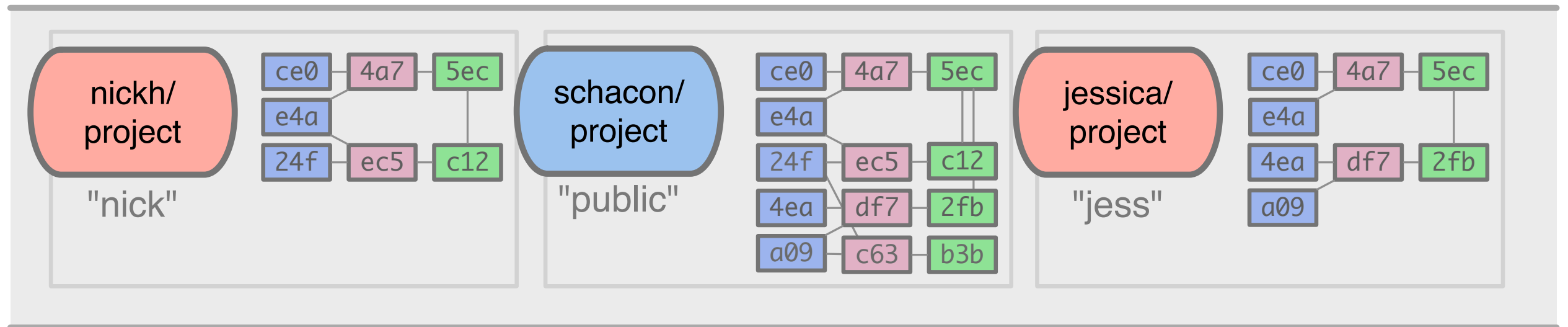
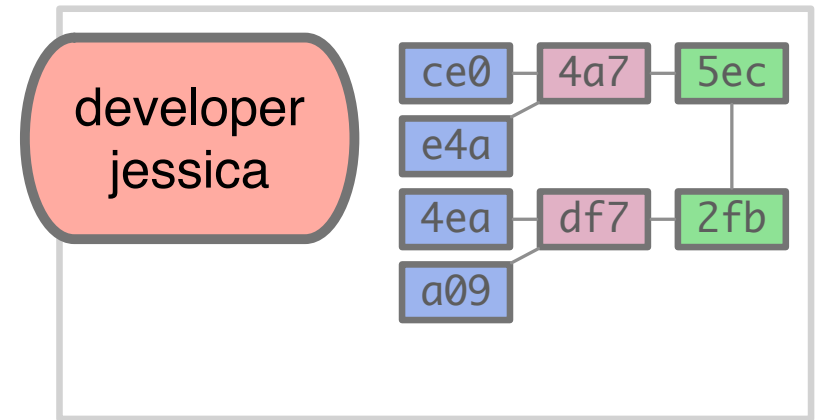
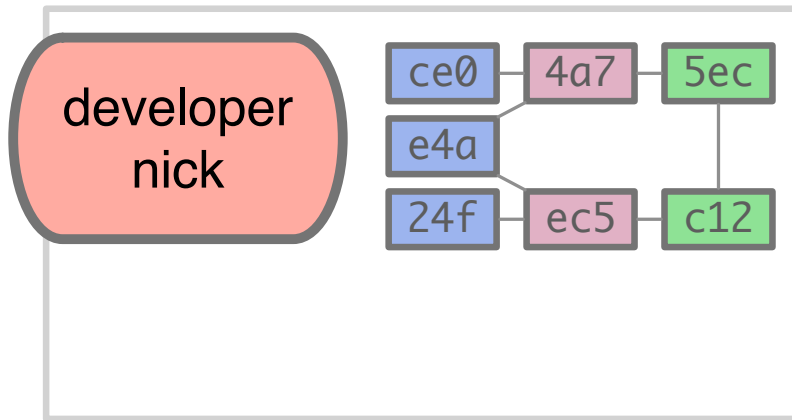


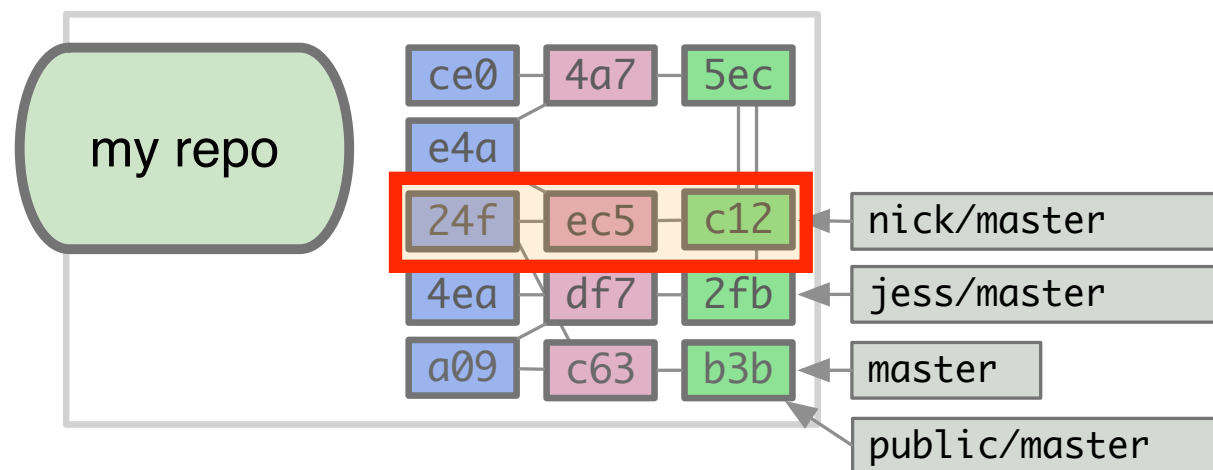
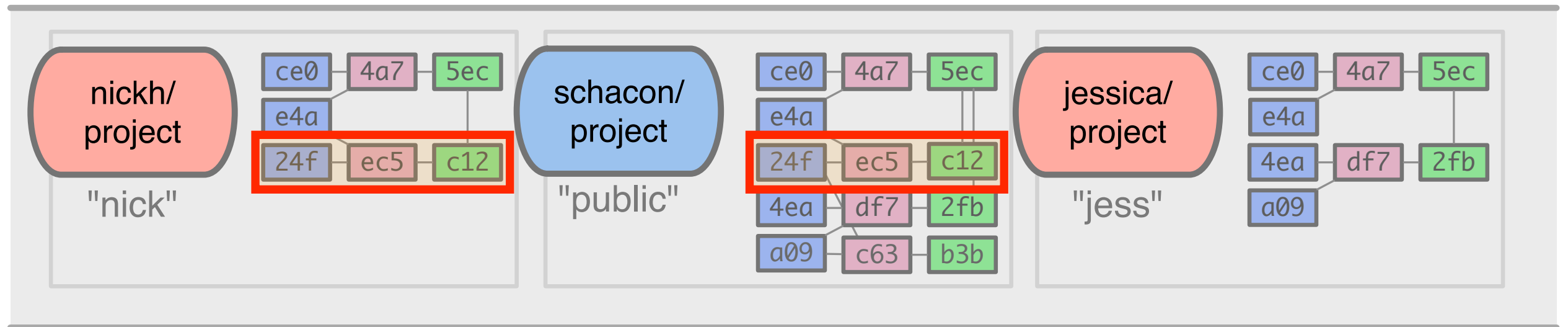
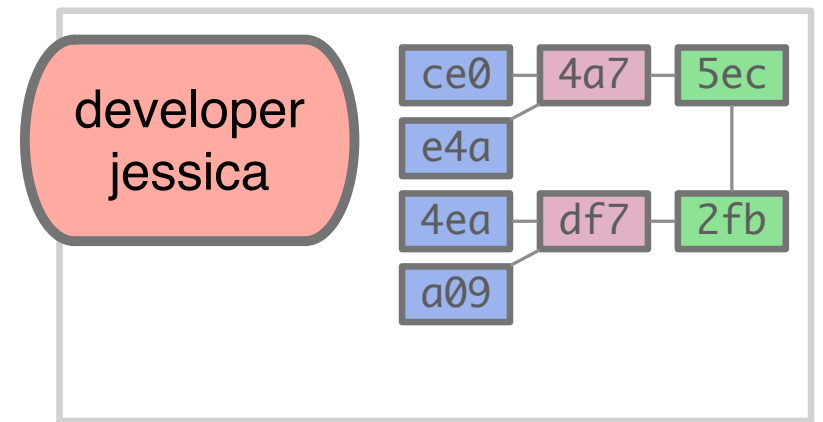
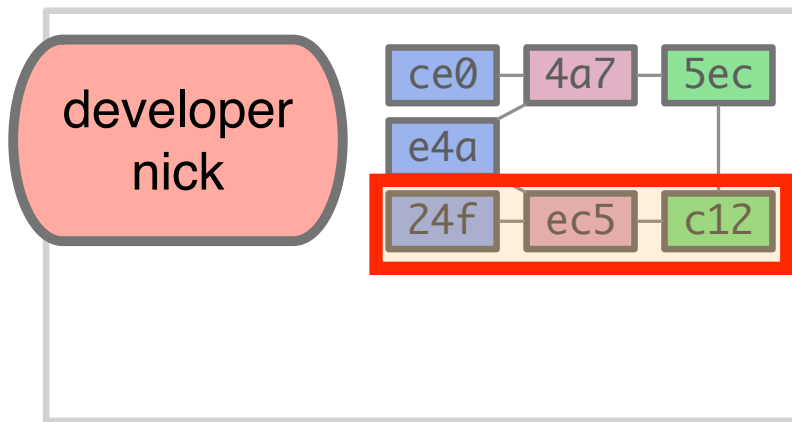


git merge nick jess

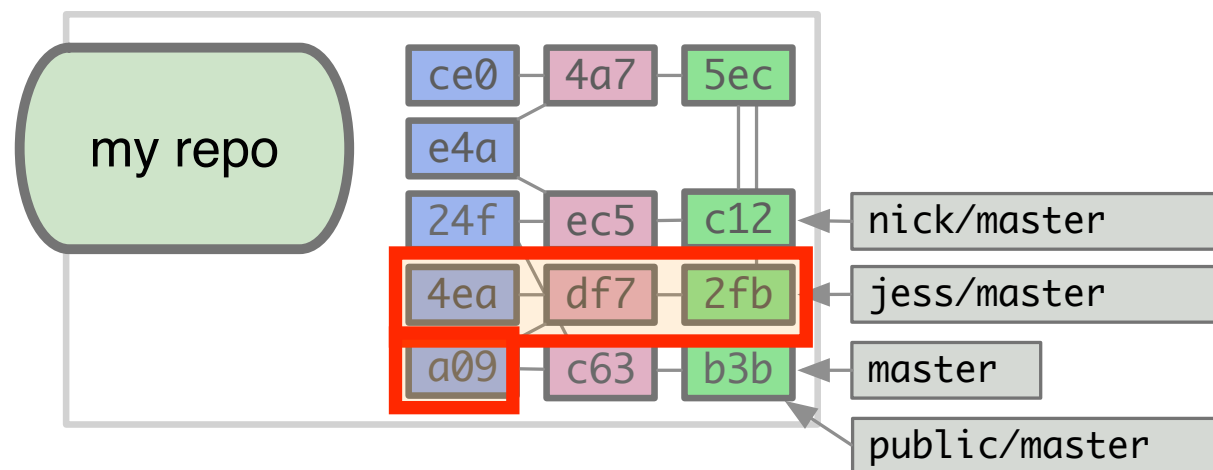
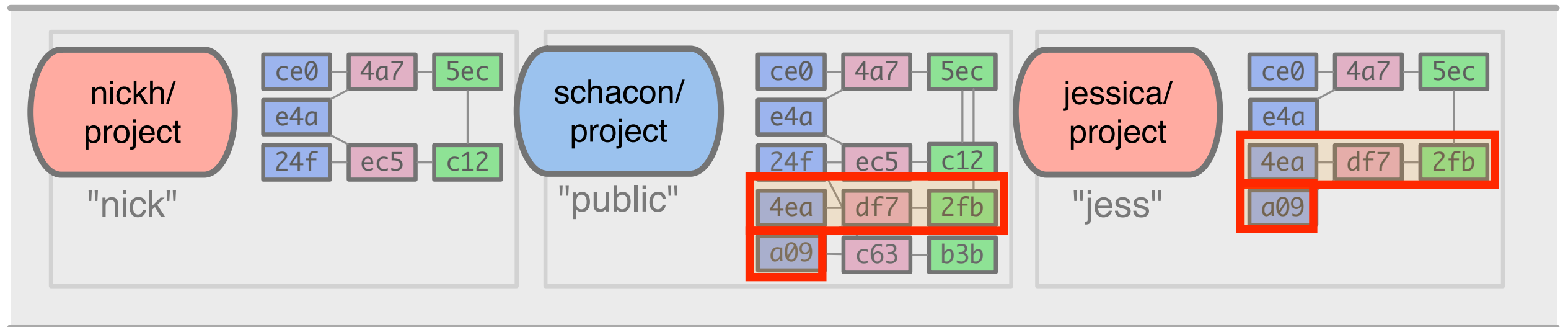
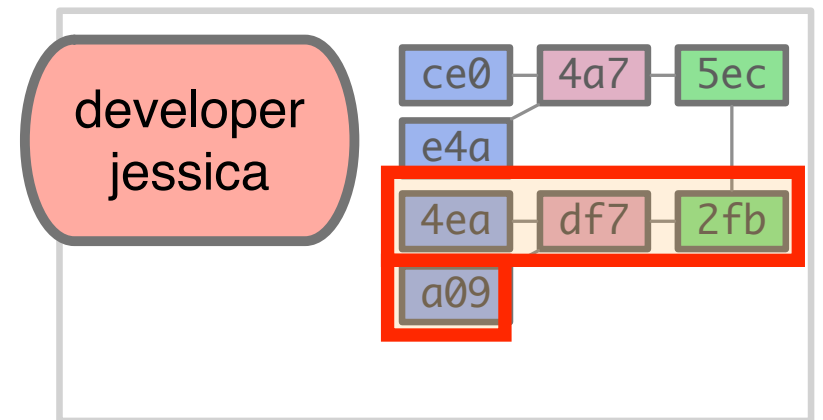
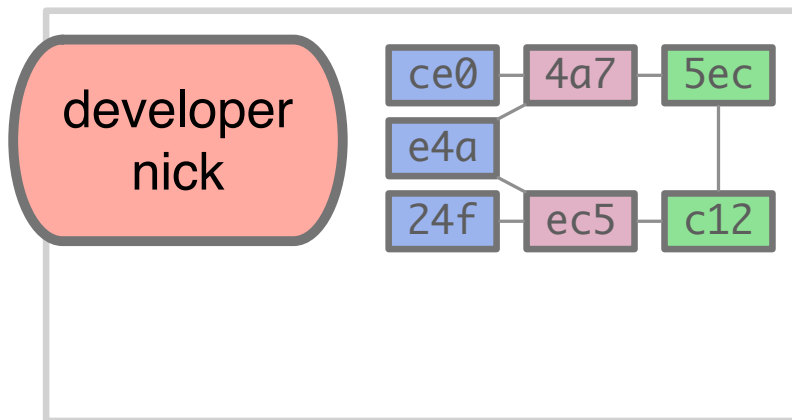


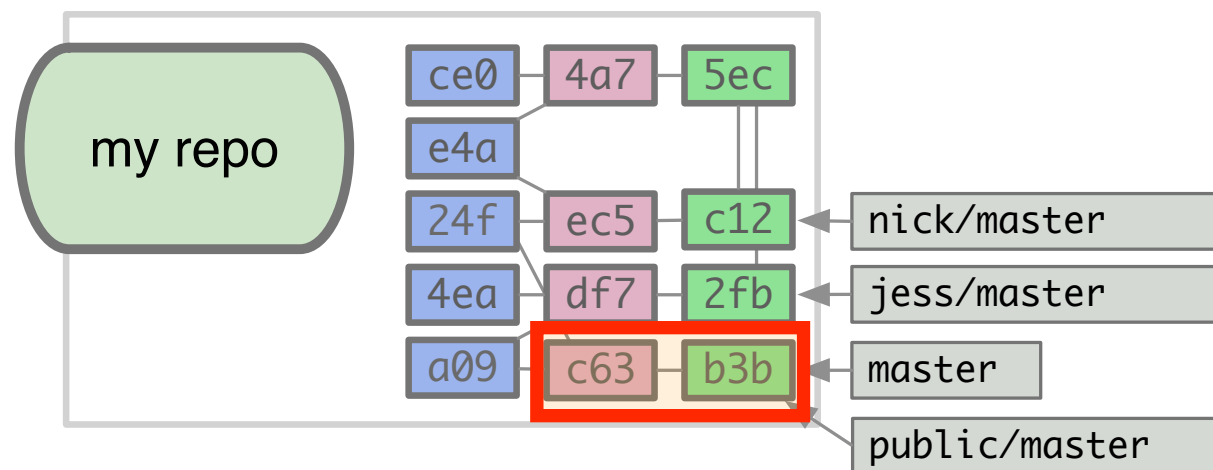
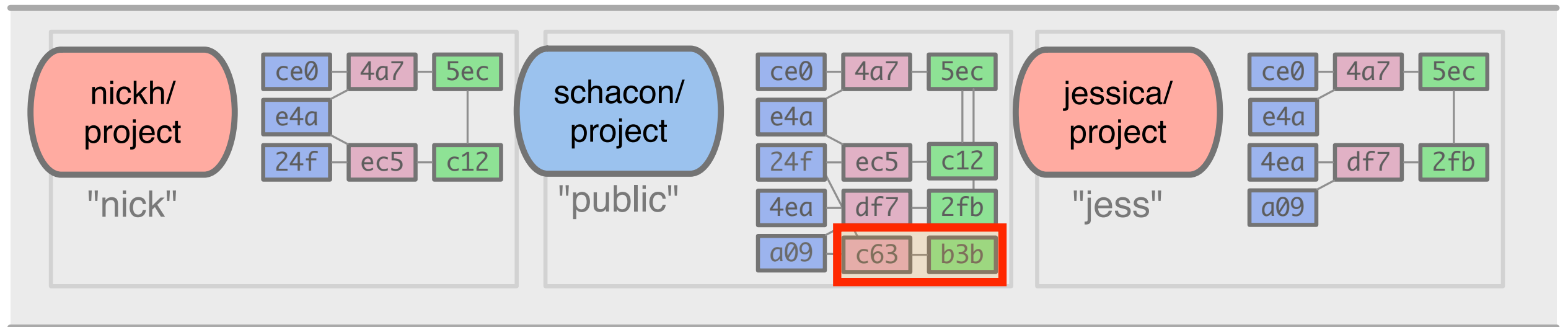
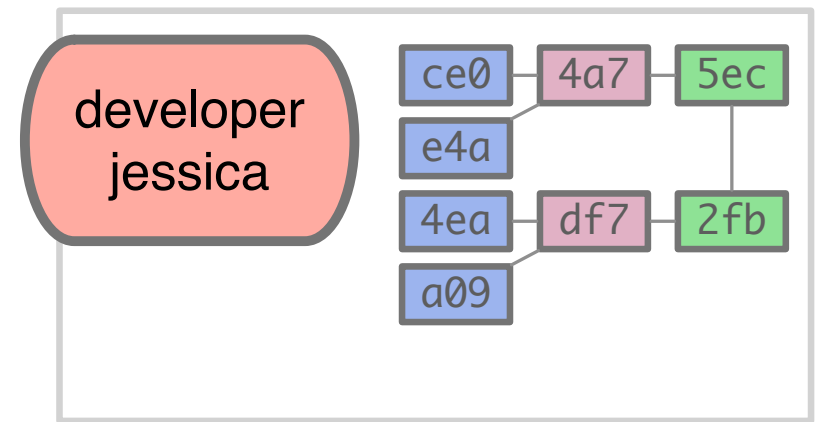
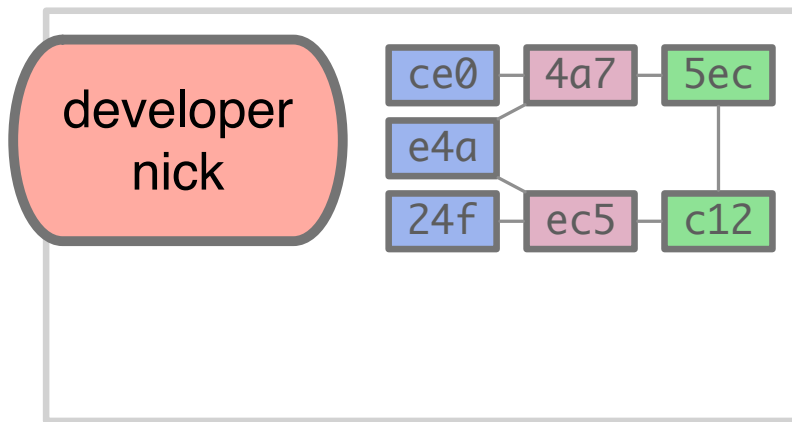
git push public





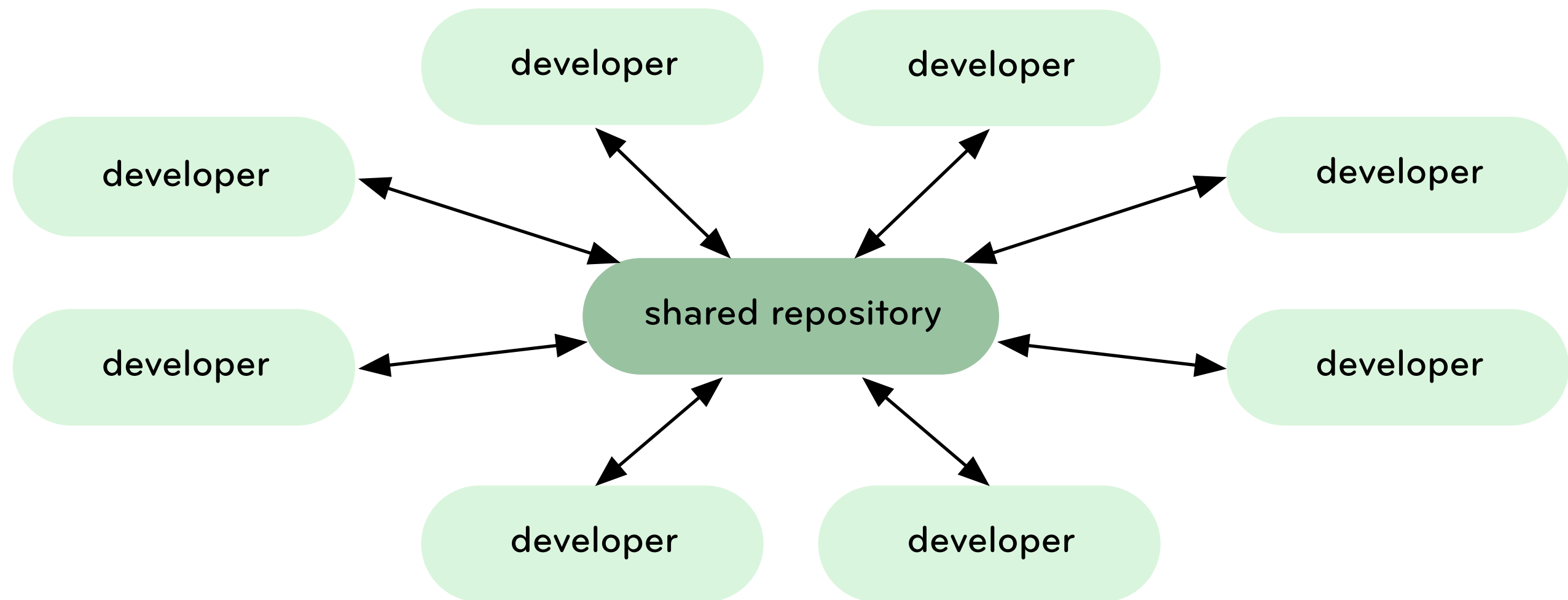


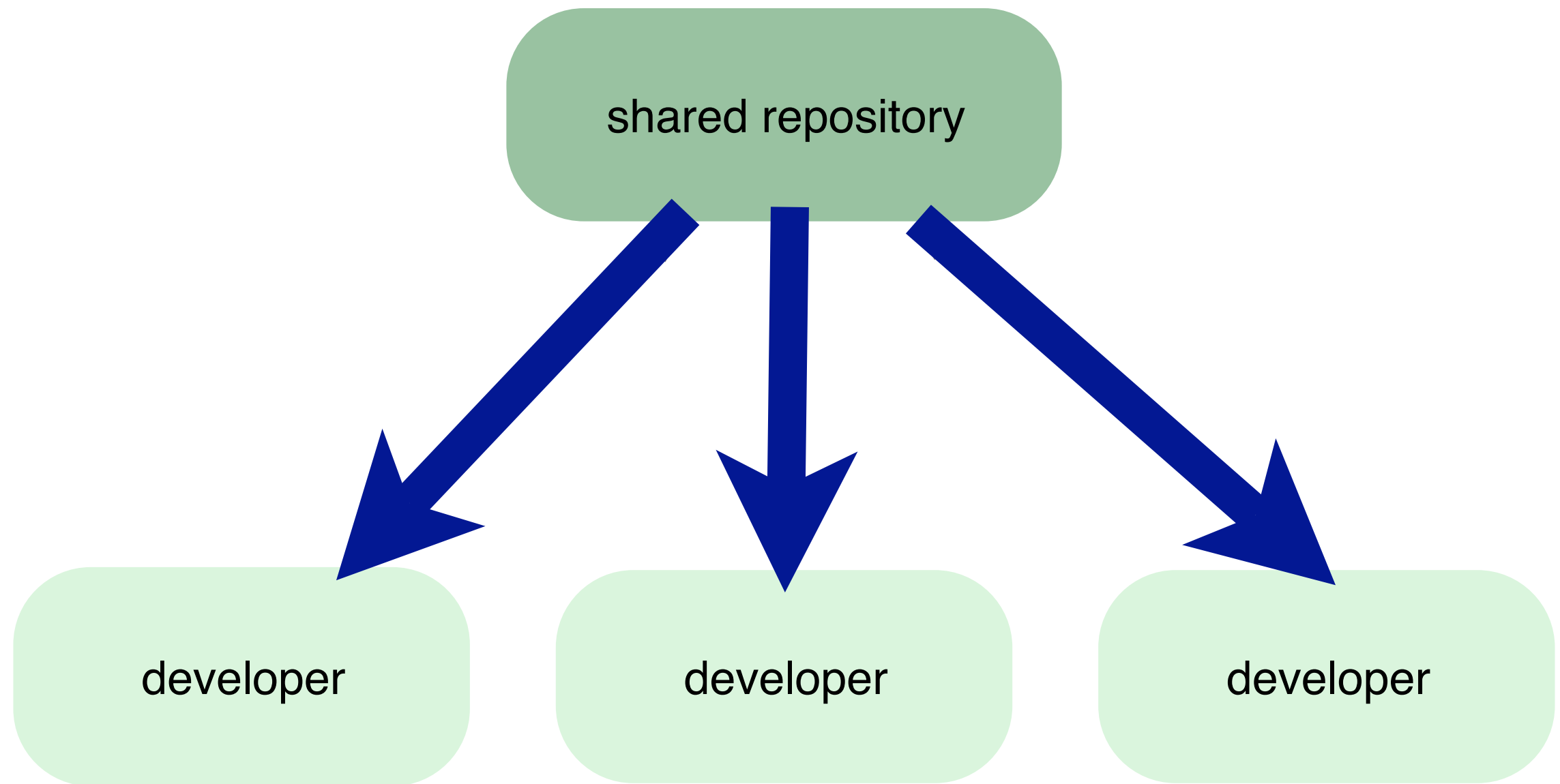




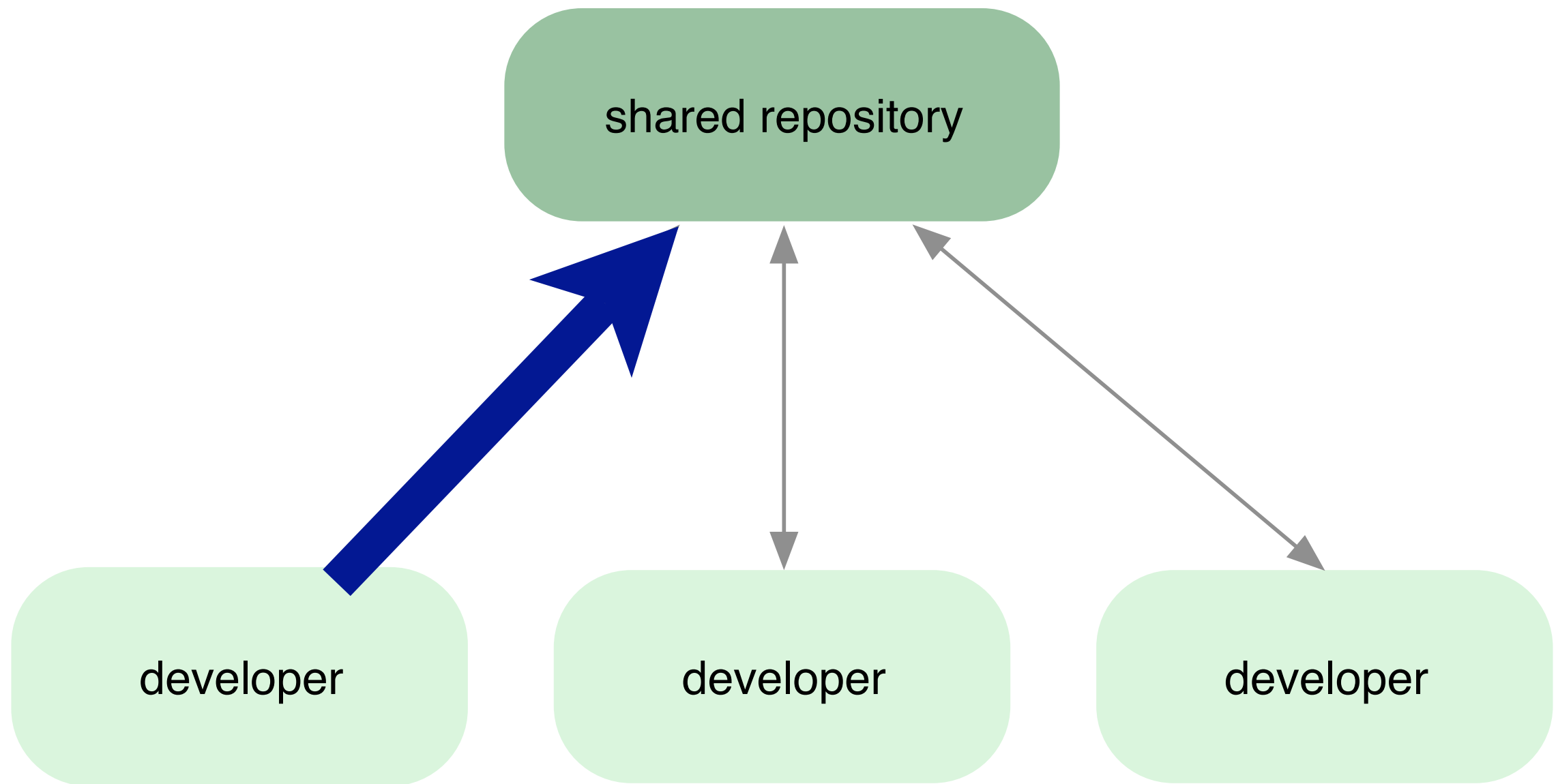
popular workflows

central repository  
model

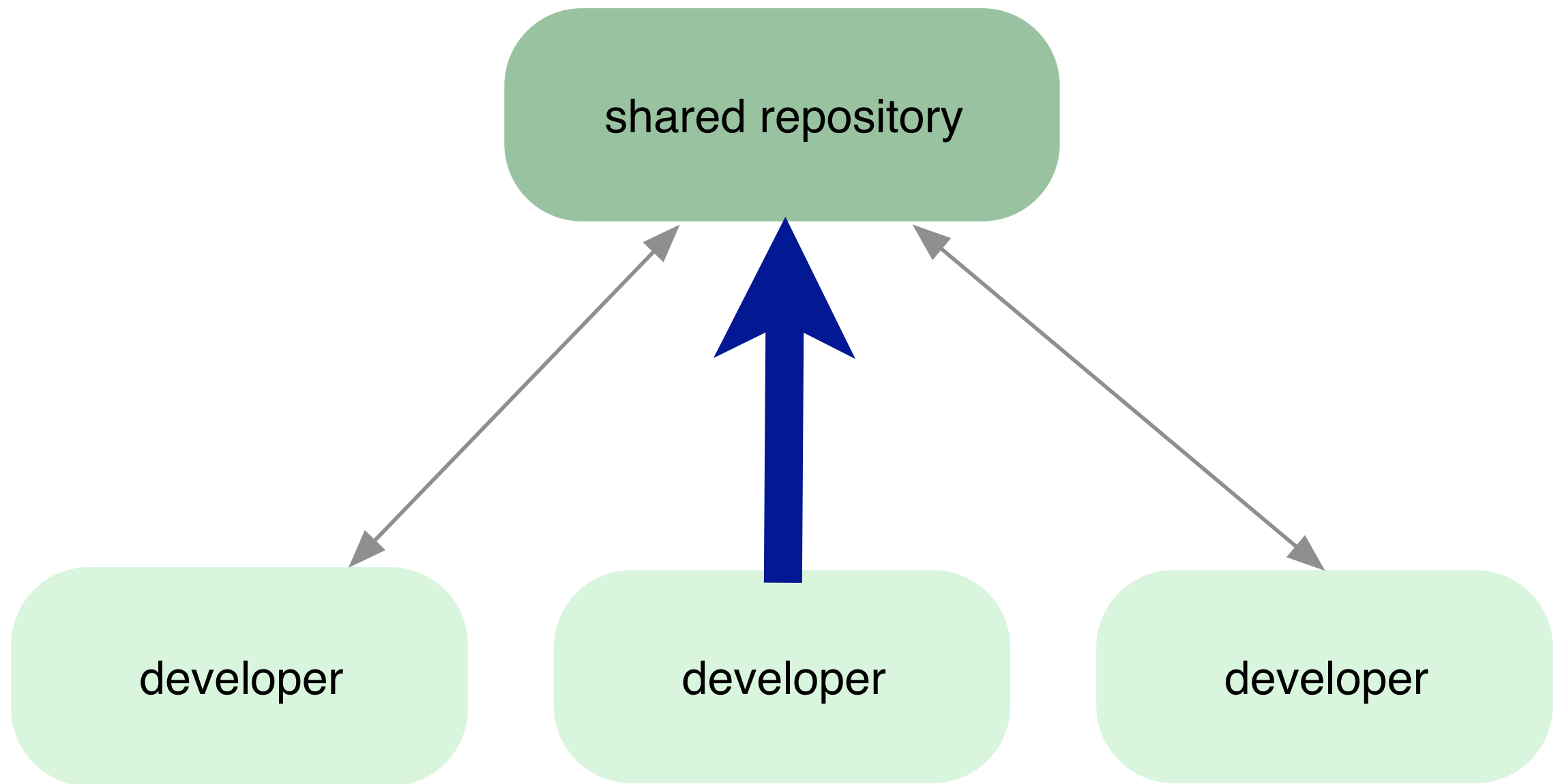




git clone

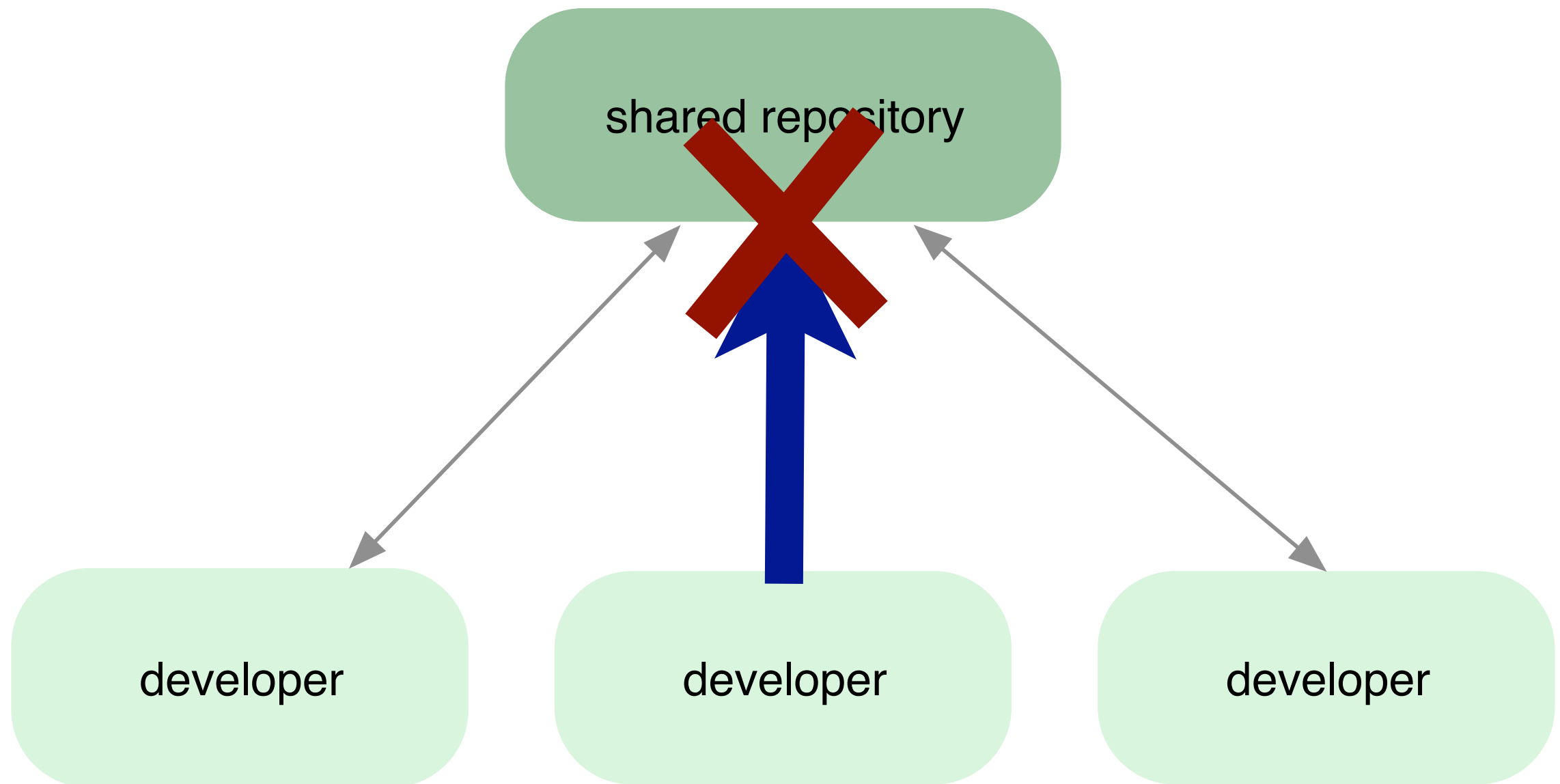


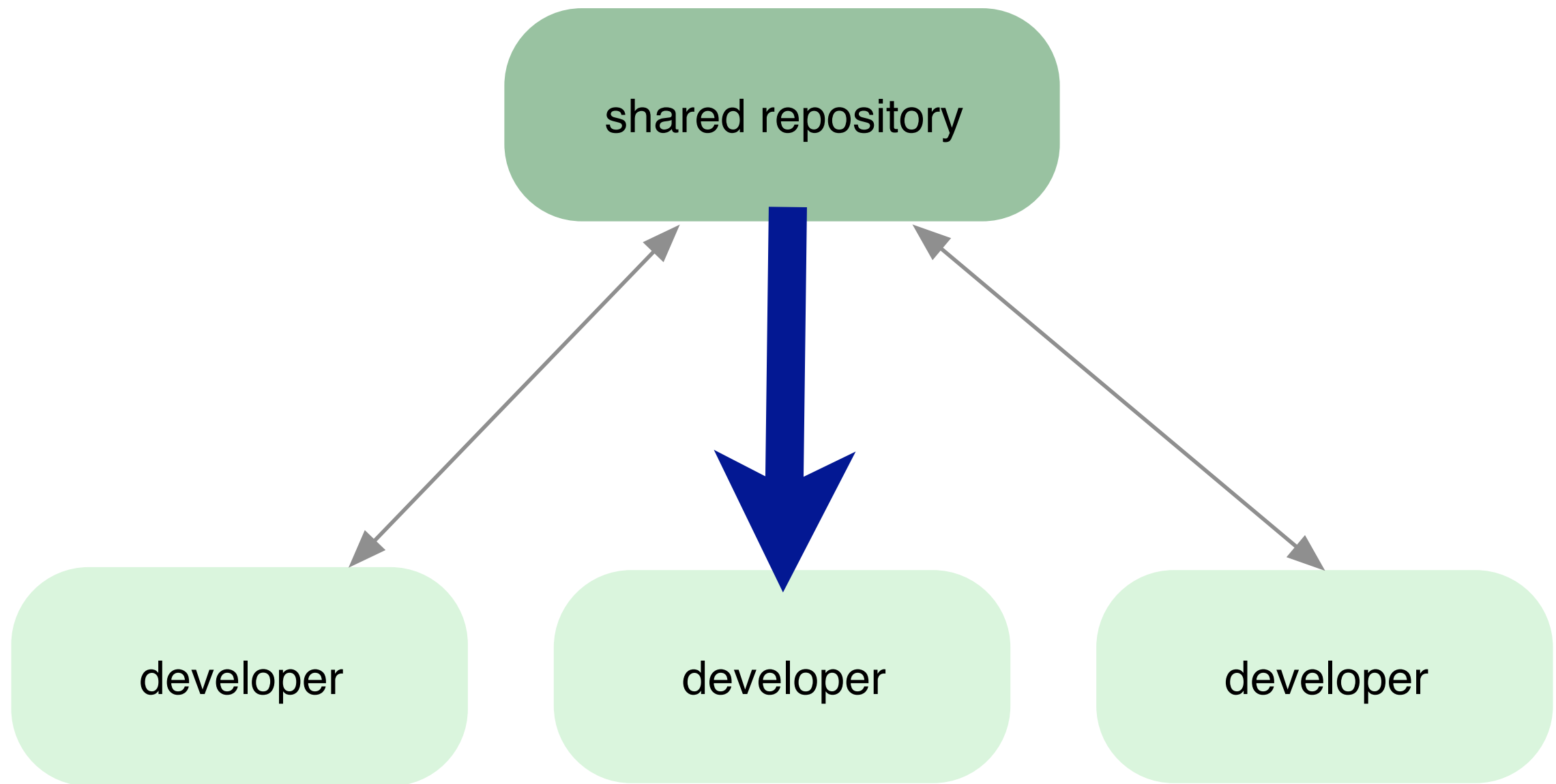
git push



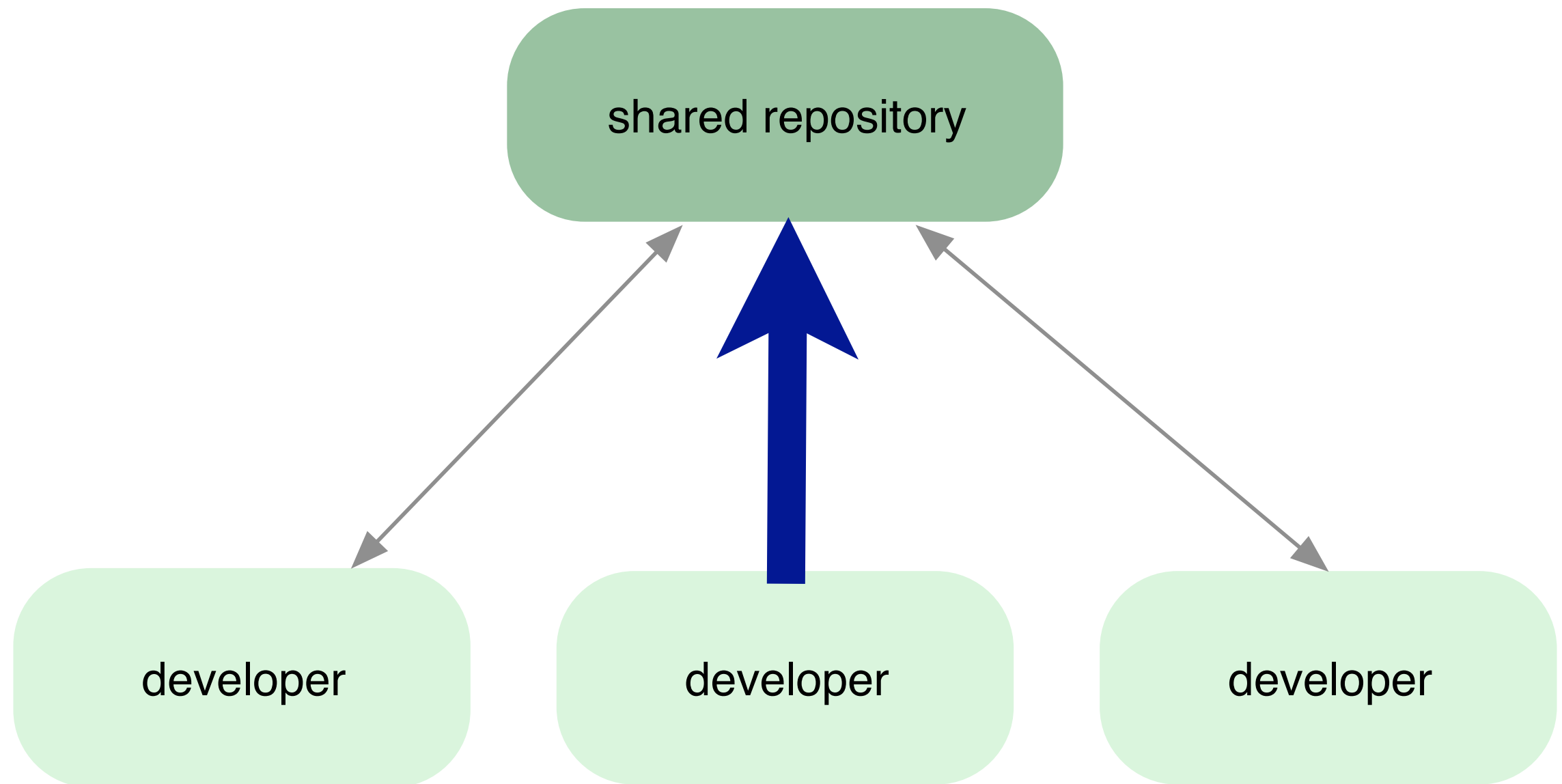
git push





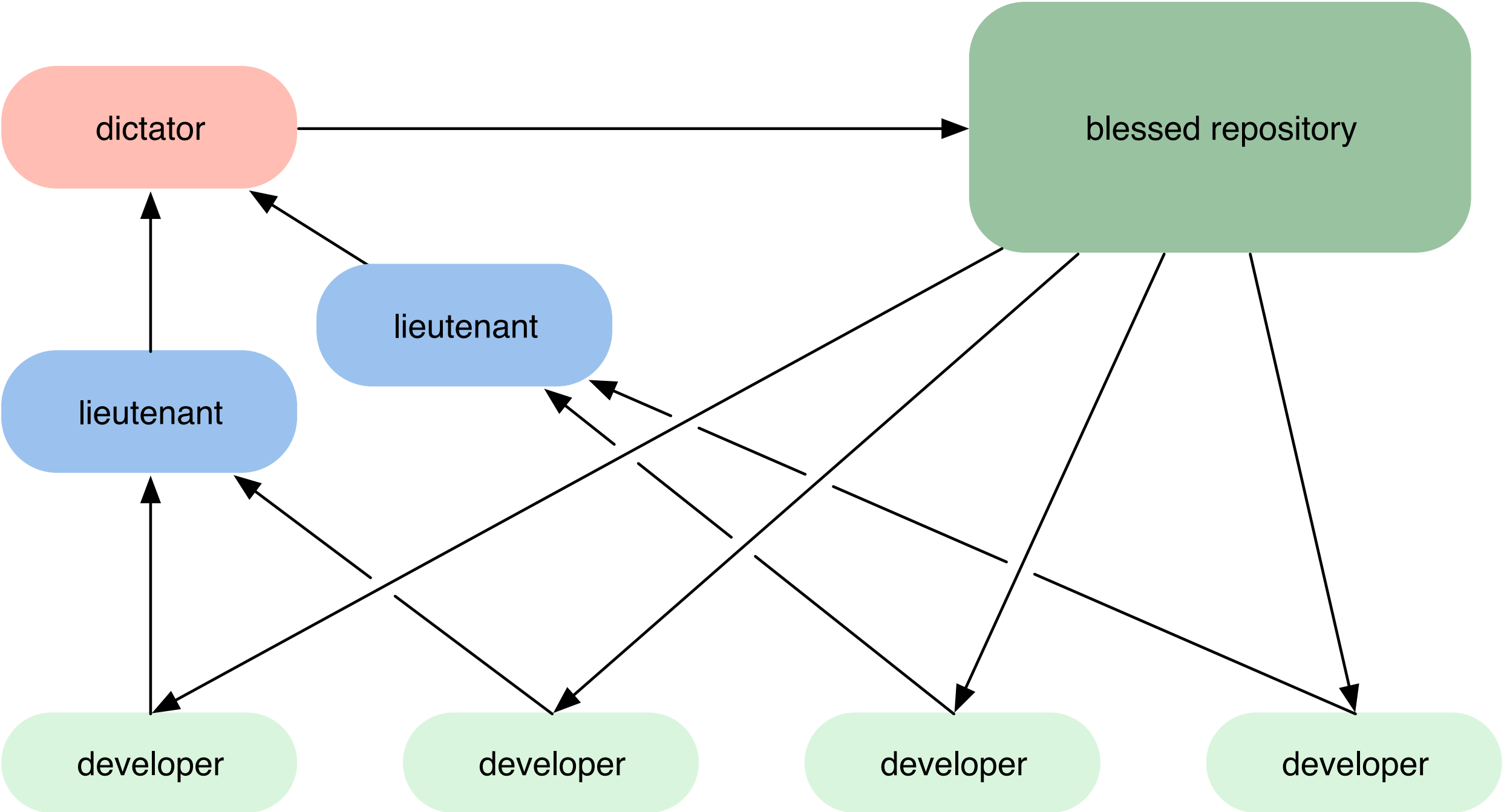


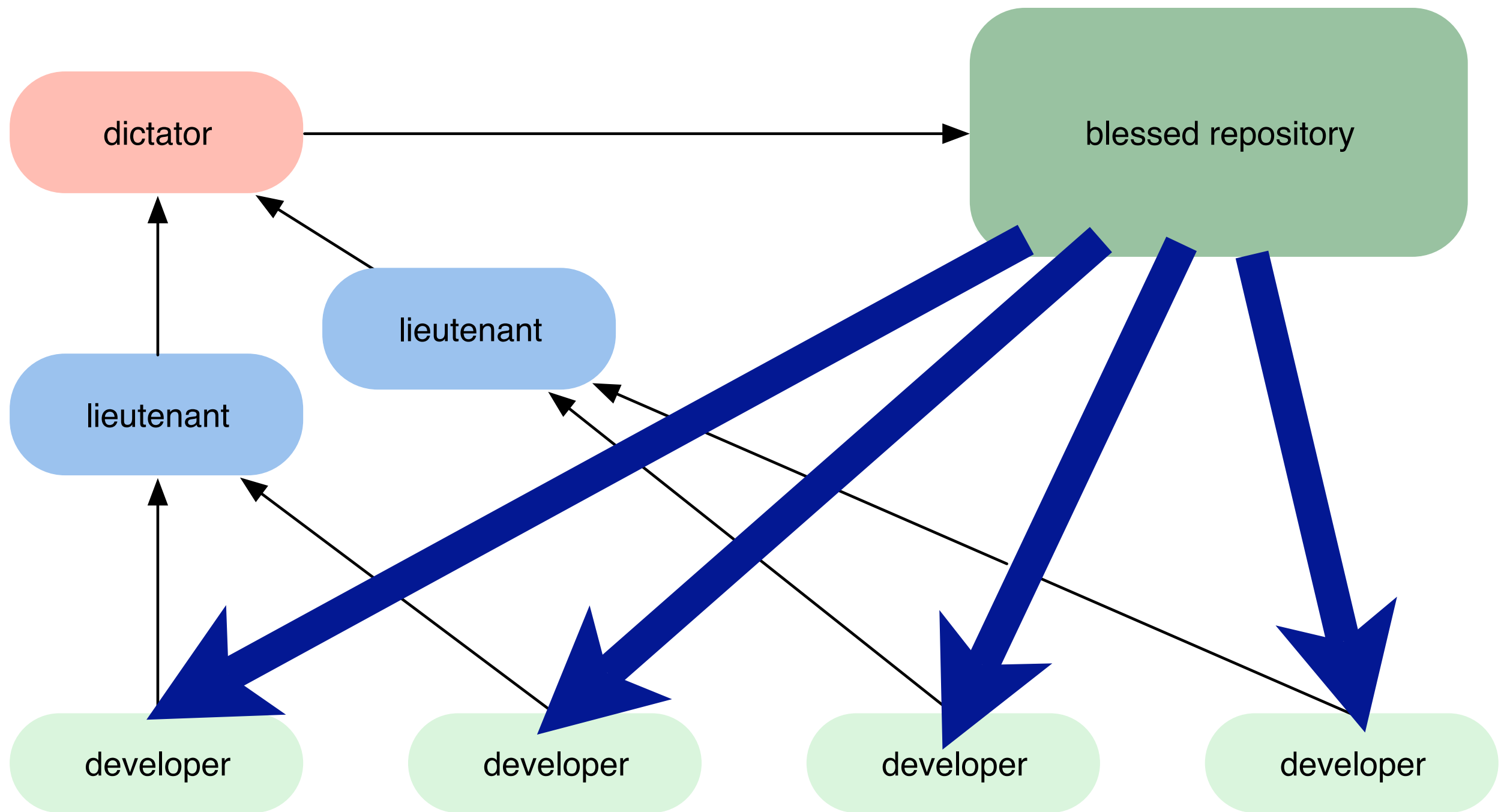
git fetch  
git merge



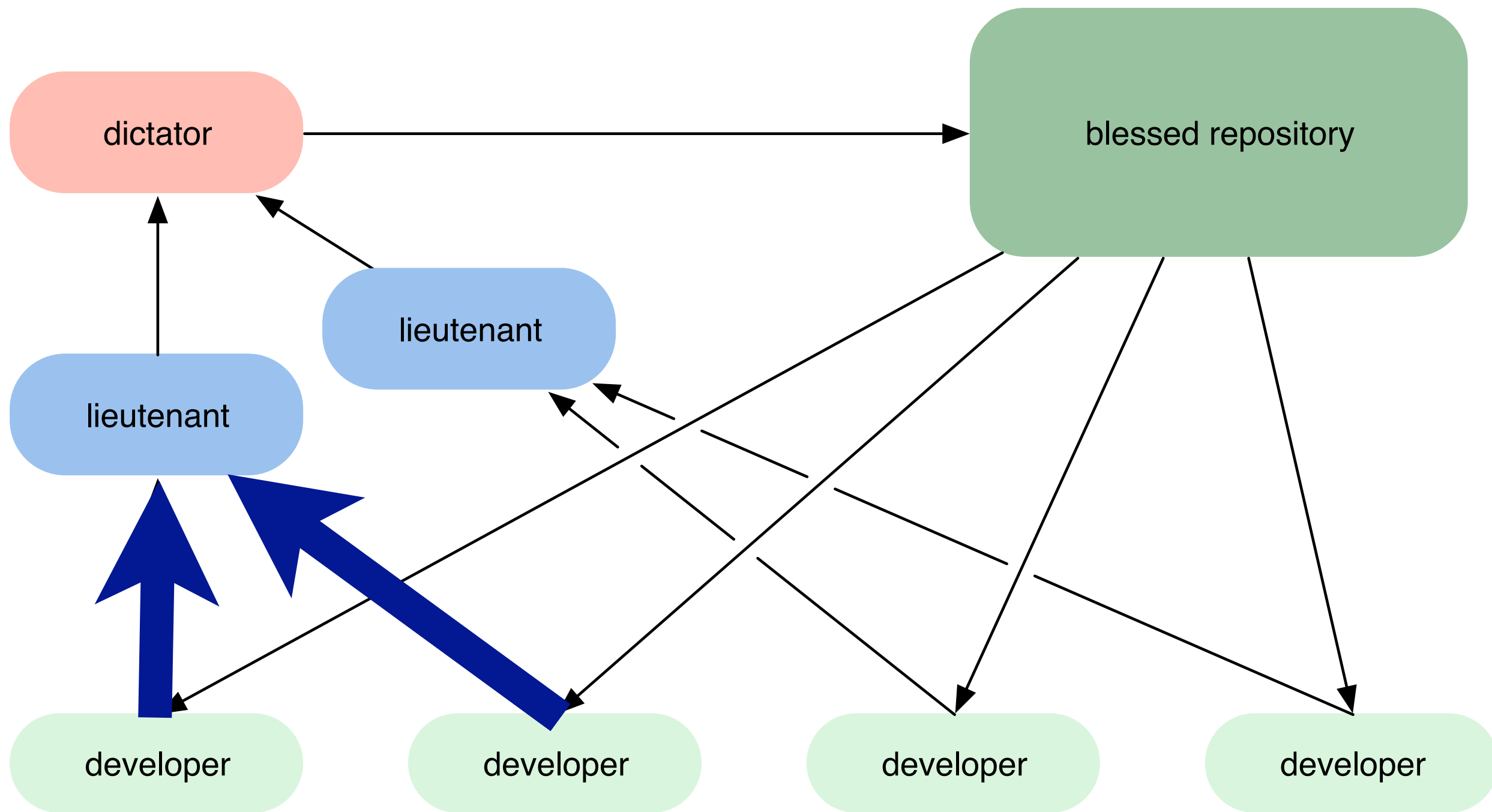
**git push**

# dictator and lieutenants model

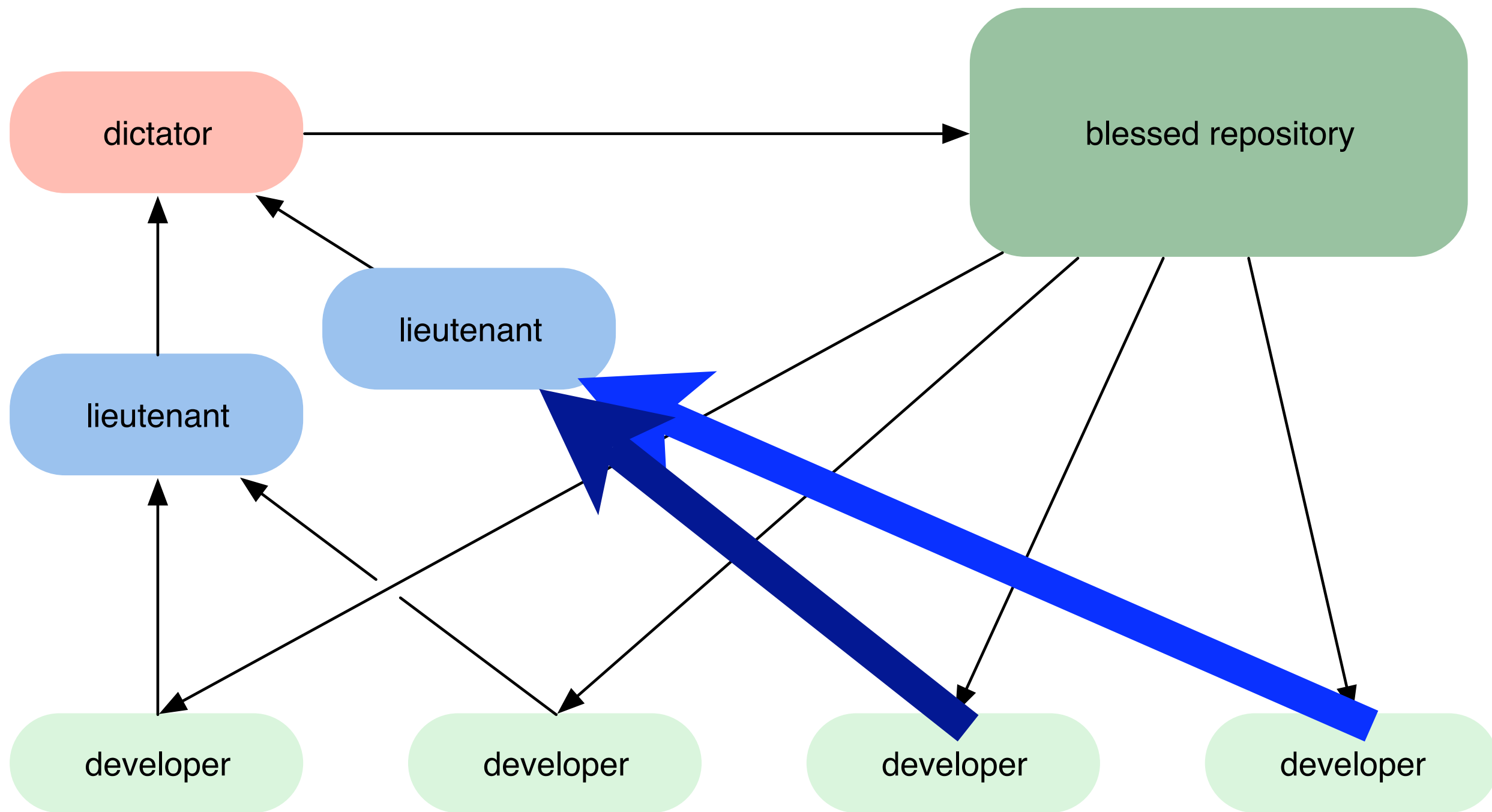




git clone



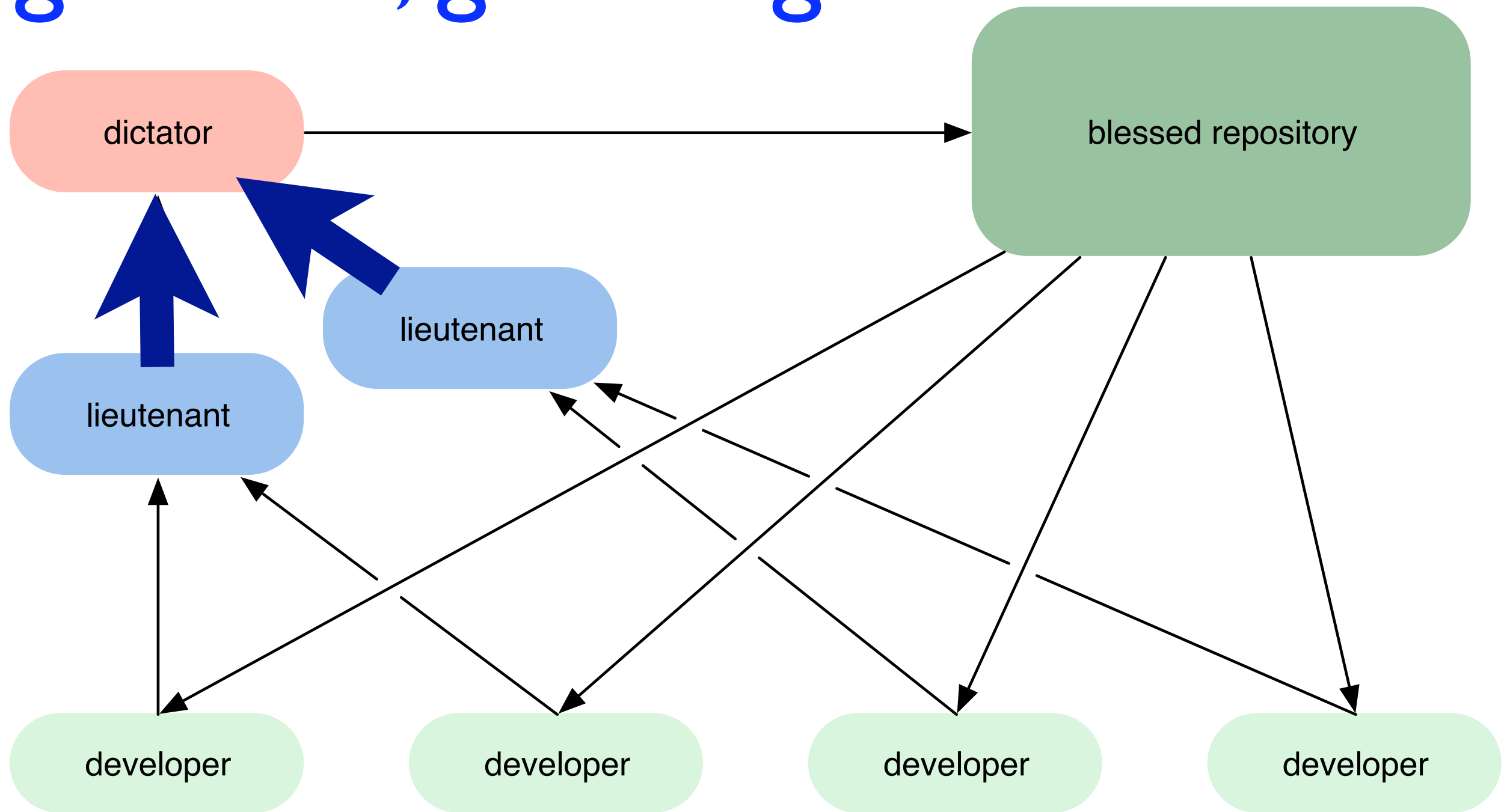
git fetch  
git merge



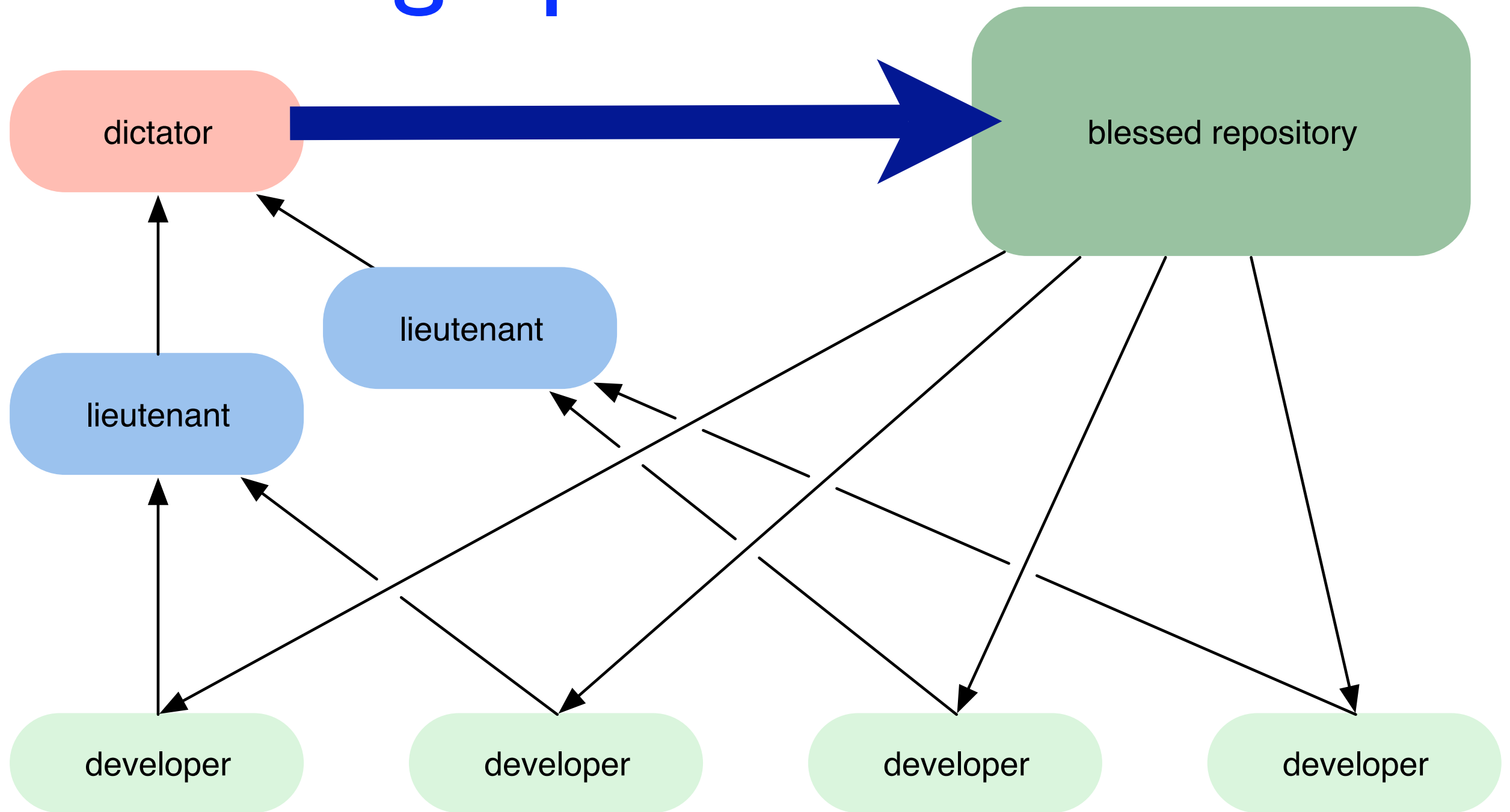
git fetch  
git merge

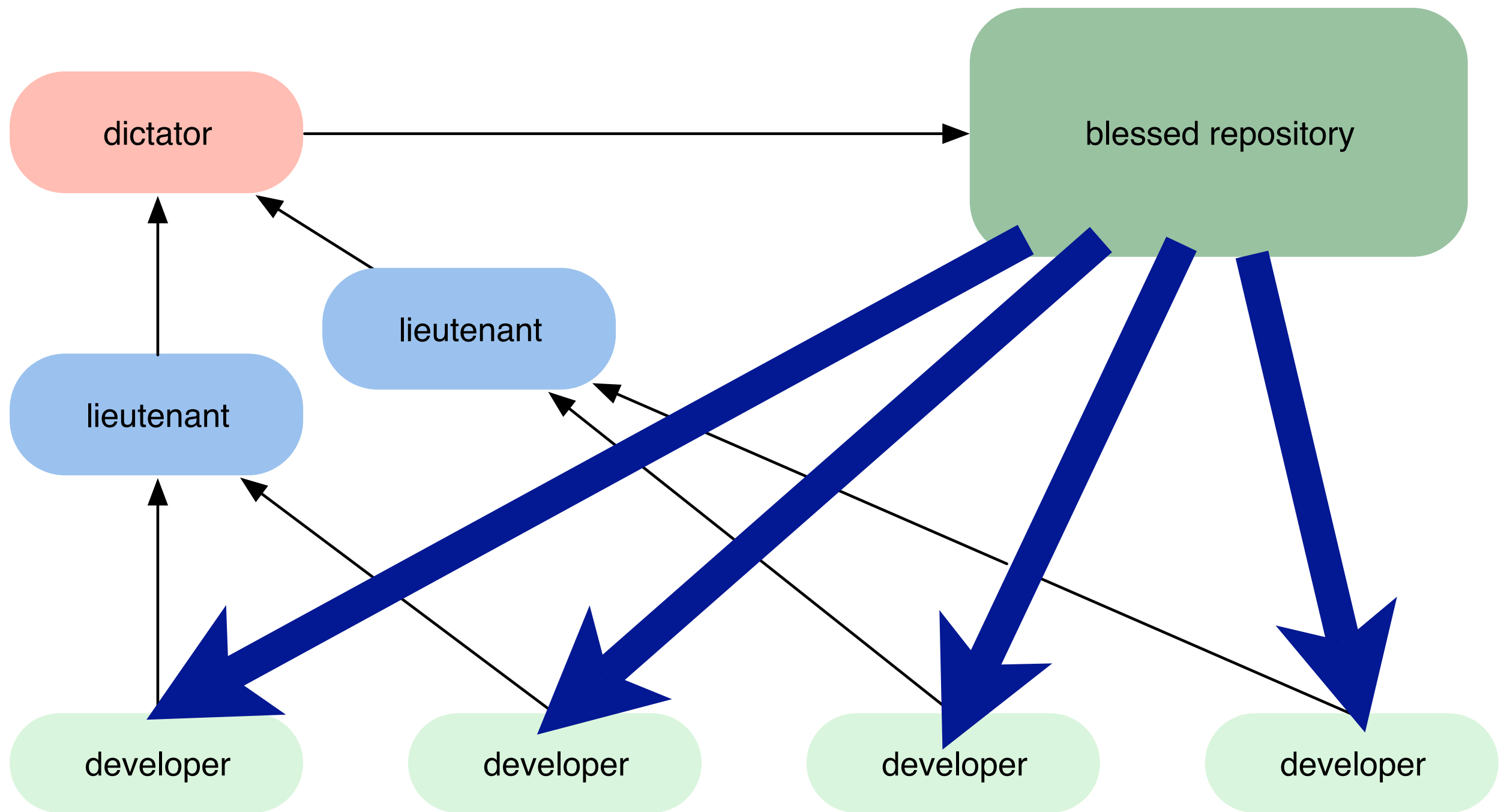


# git fetch; git merge



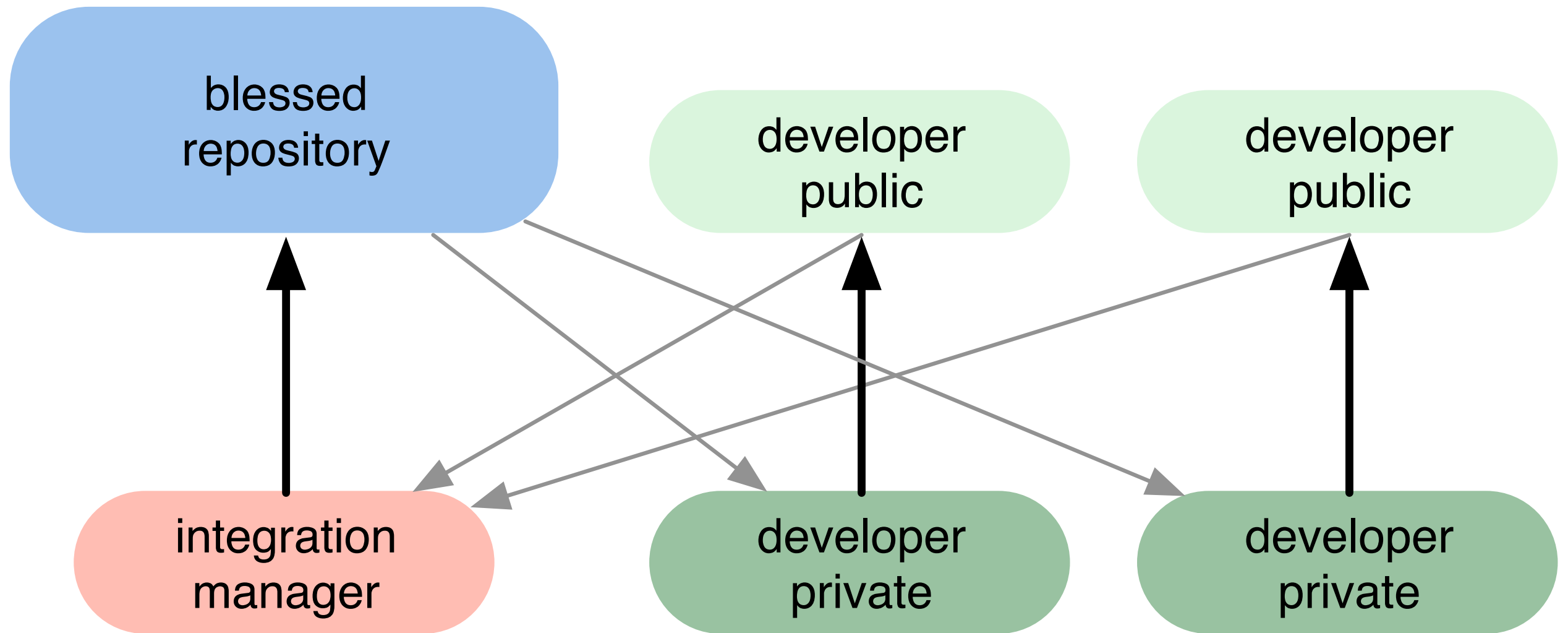
# git push





git fetch

# integration manager model



# github.com

/schacon/ticgit

blessed  
repository

/yob/ticgit

developer  
public

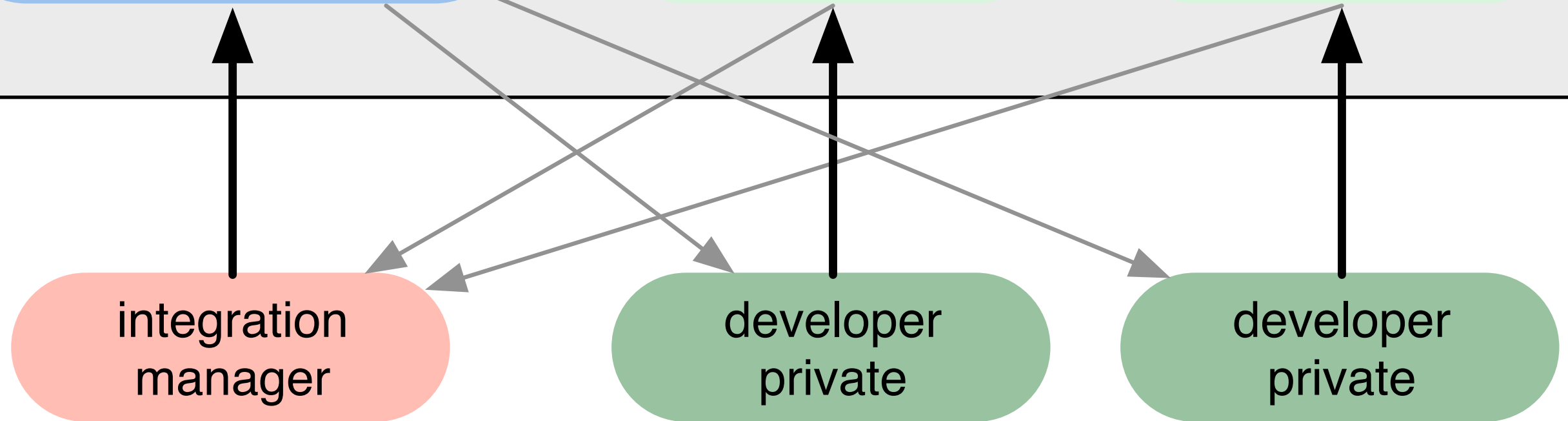
/pope/ticgit

developer  
public

integration  
manager

developer  
private

developer  
private



# github.com

/schacon/ticgit

blessed  
repository

“forks”

/yob/ticgit

developer  
public

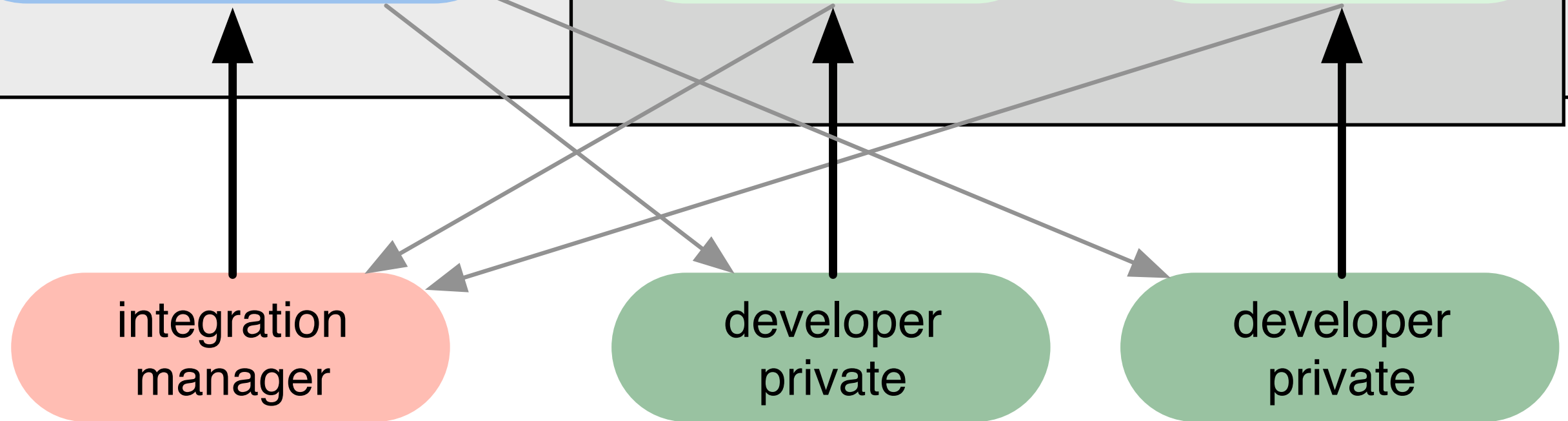
/pope/ticgit

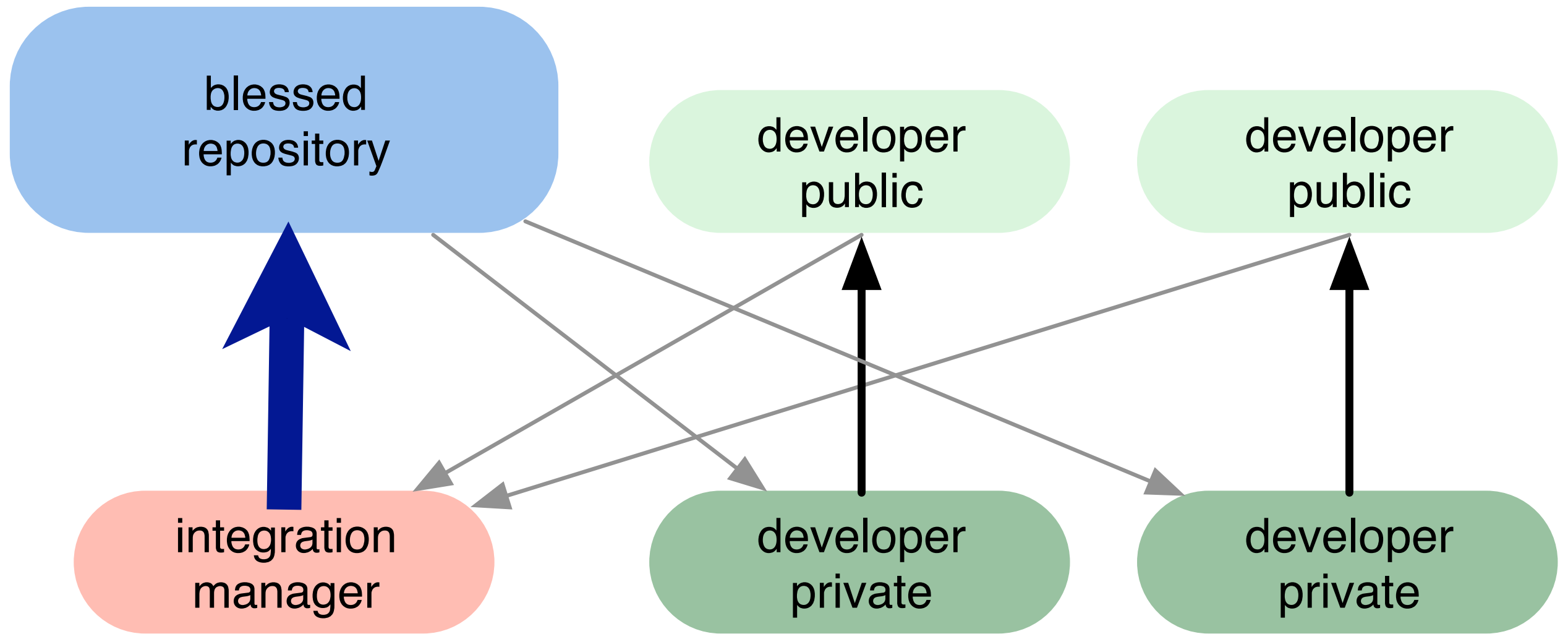
developer  
public

integration  
manager

developer  
private

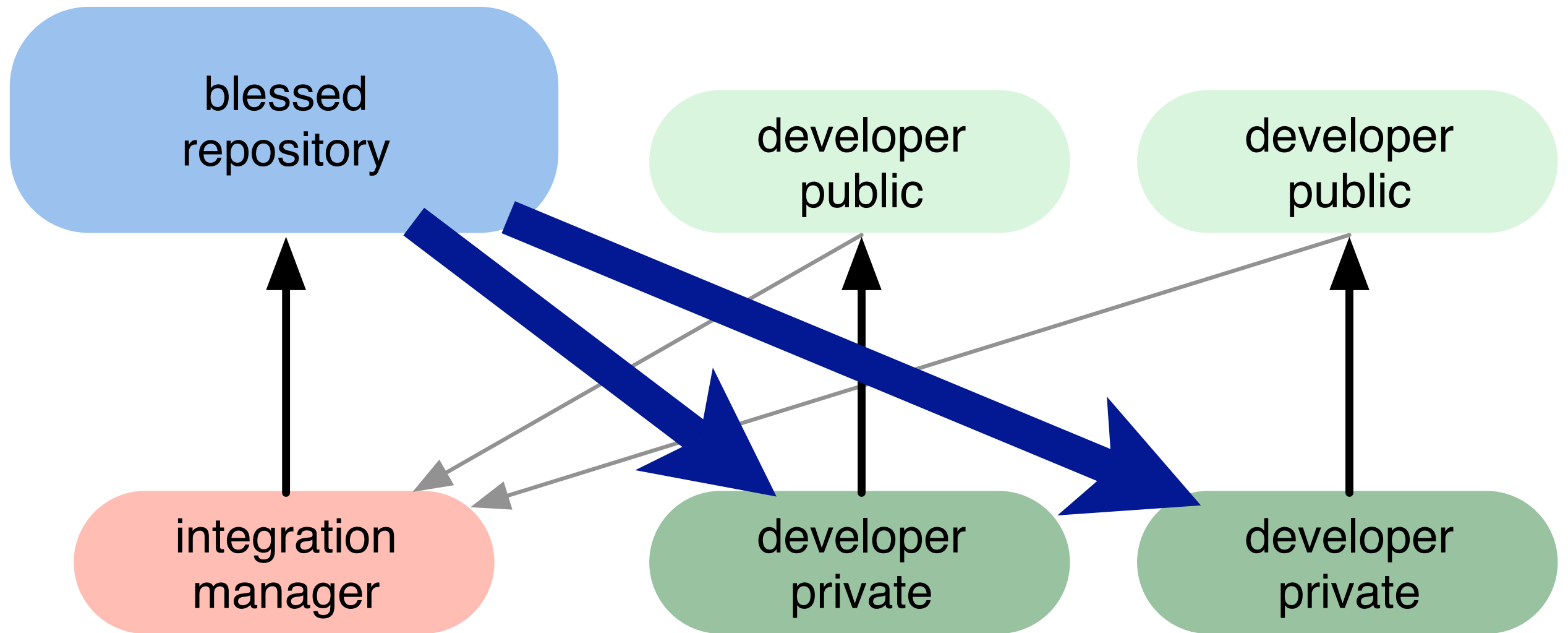
developer  
private



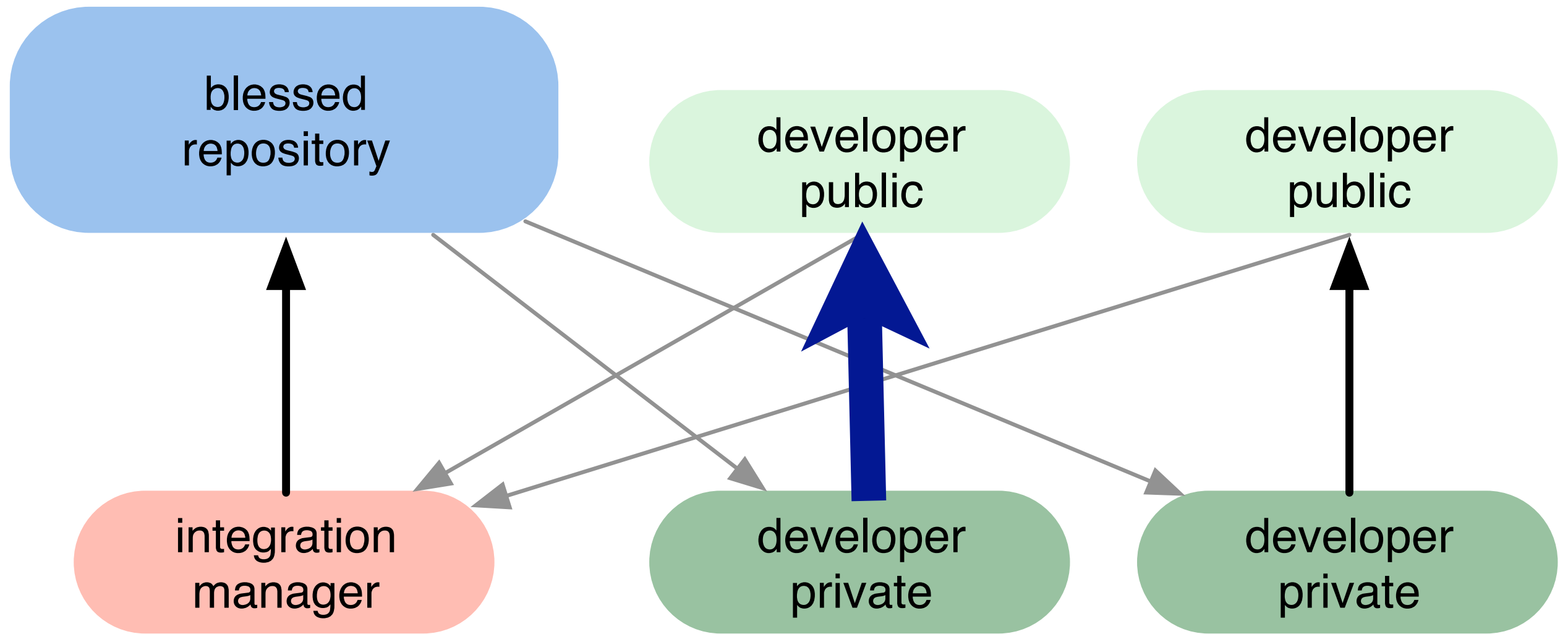


git push

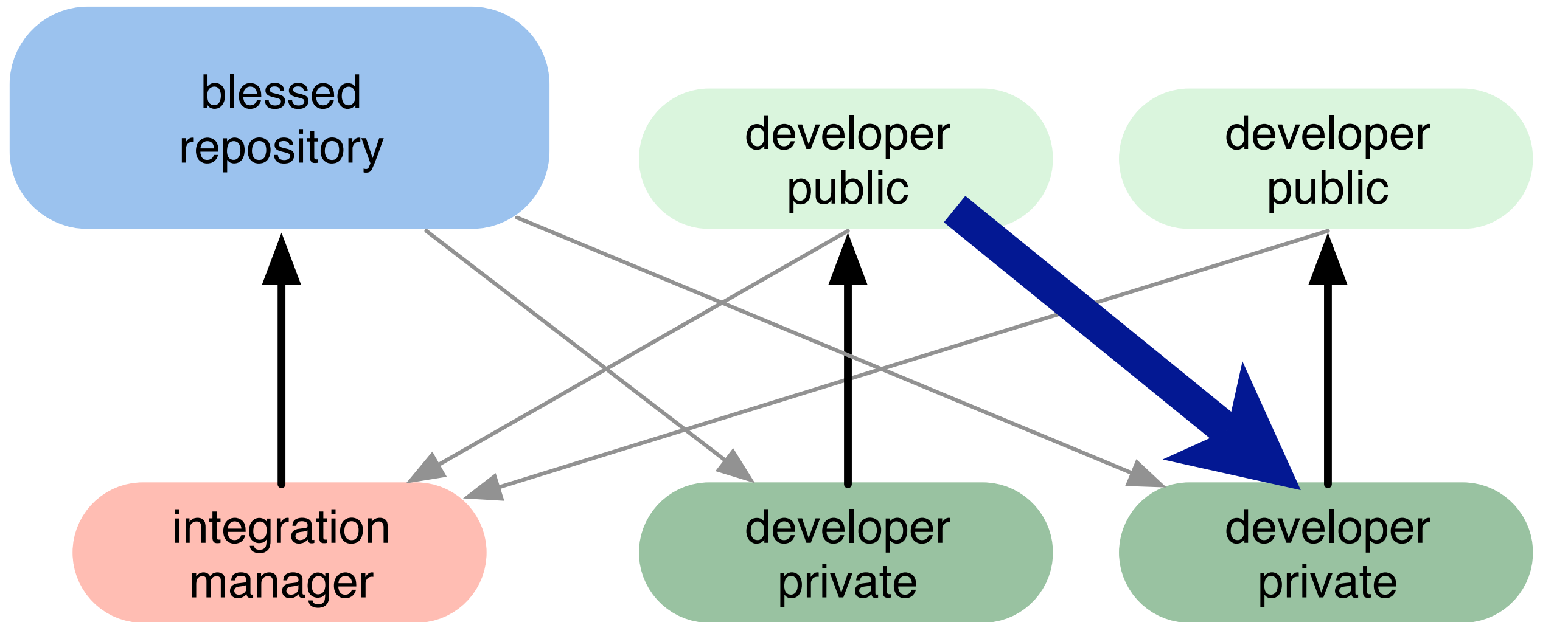




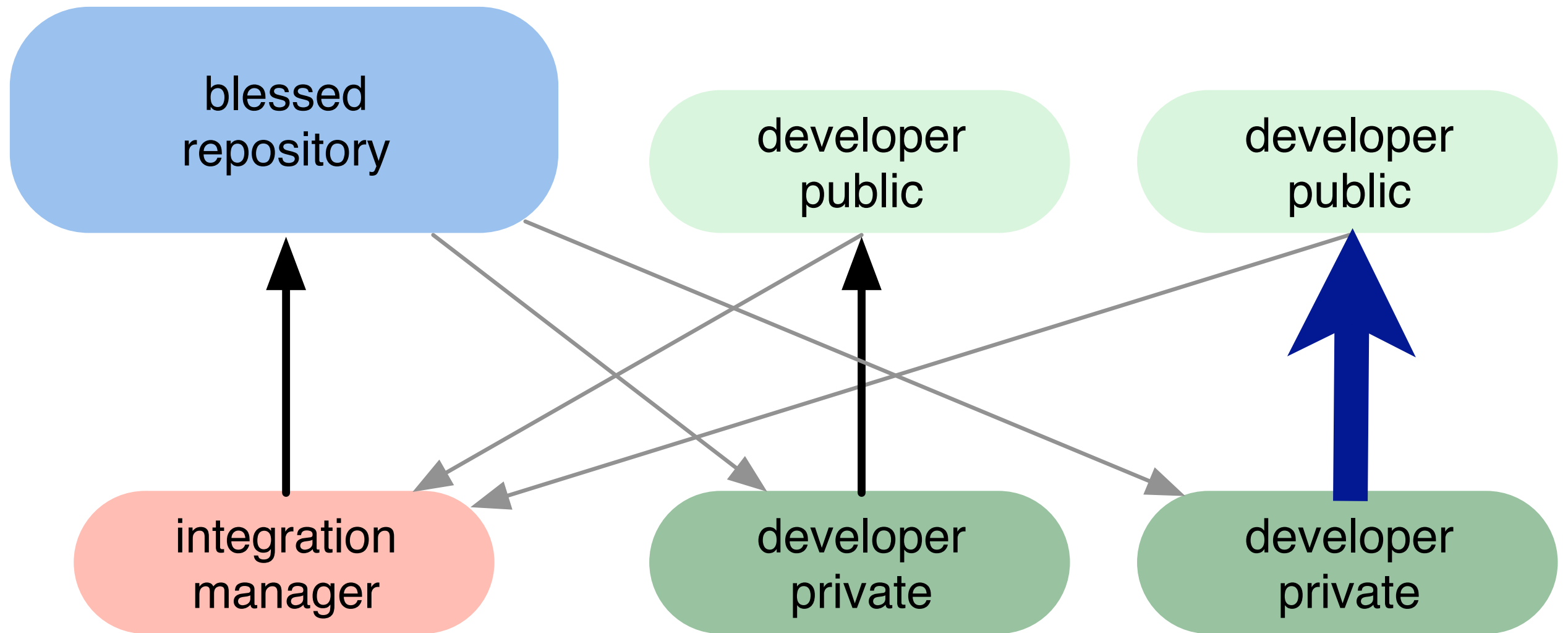
git clone



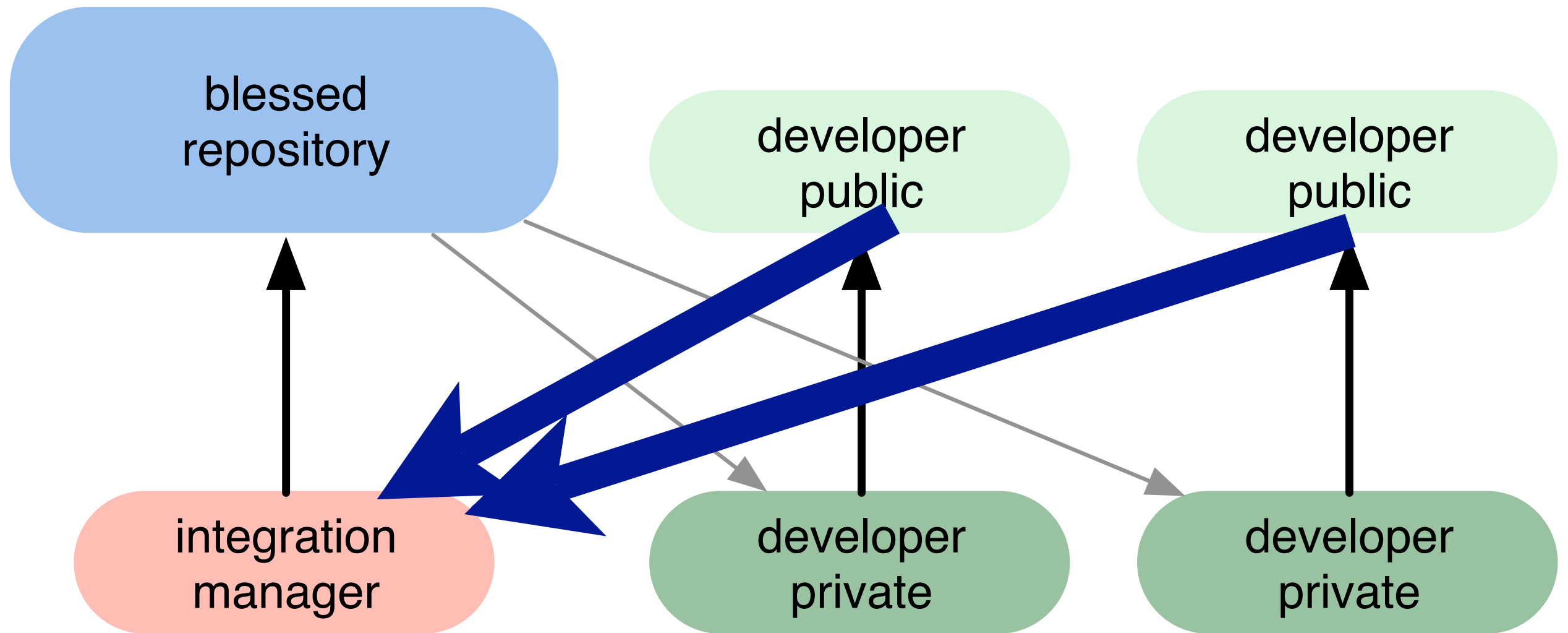
git push



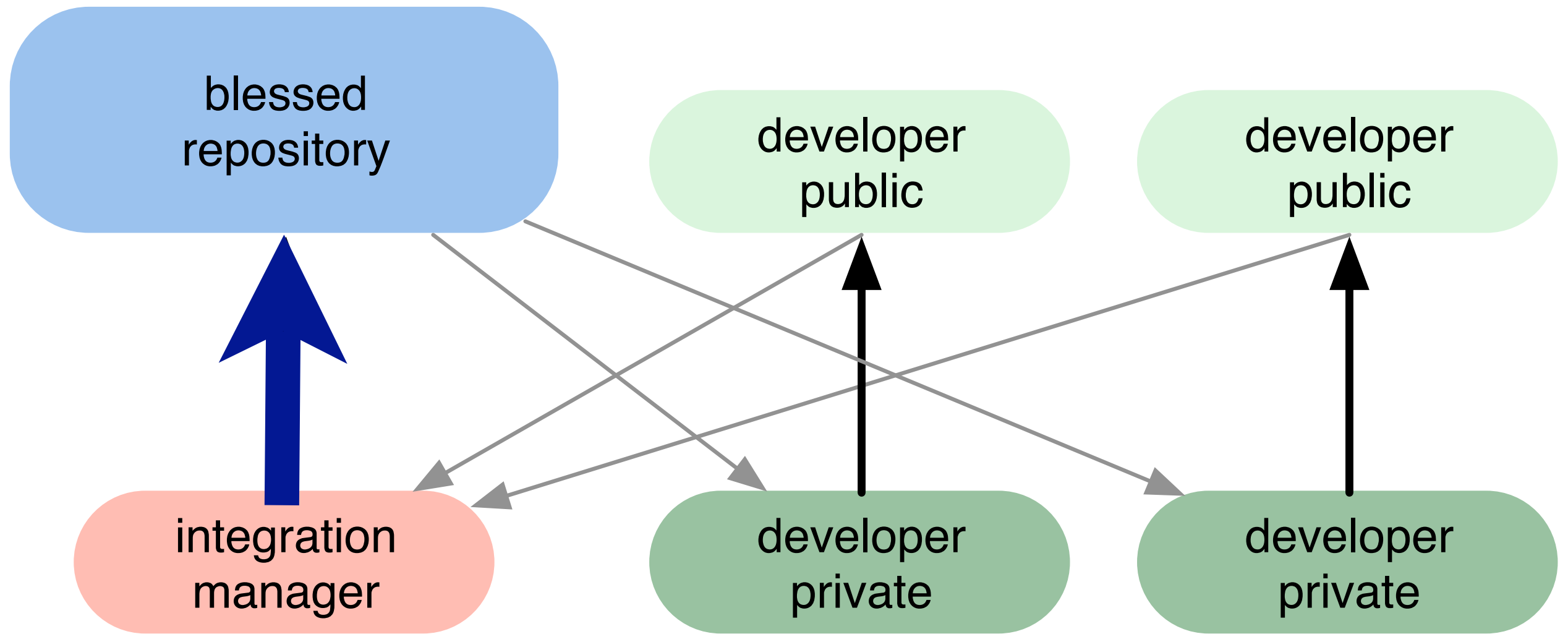
git fetch  
git merge



git push



git fetch  
git merge

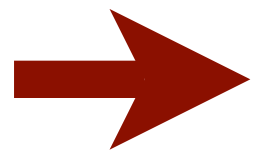


git push

# Where are we?

What is Git?

Basic Git Usage



**Intermediate Git Usage**

Git Branching

Working with Repo & Gerrit

Advanced Git Usage

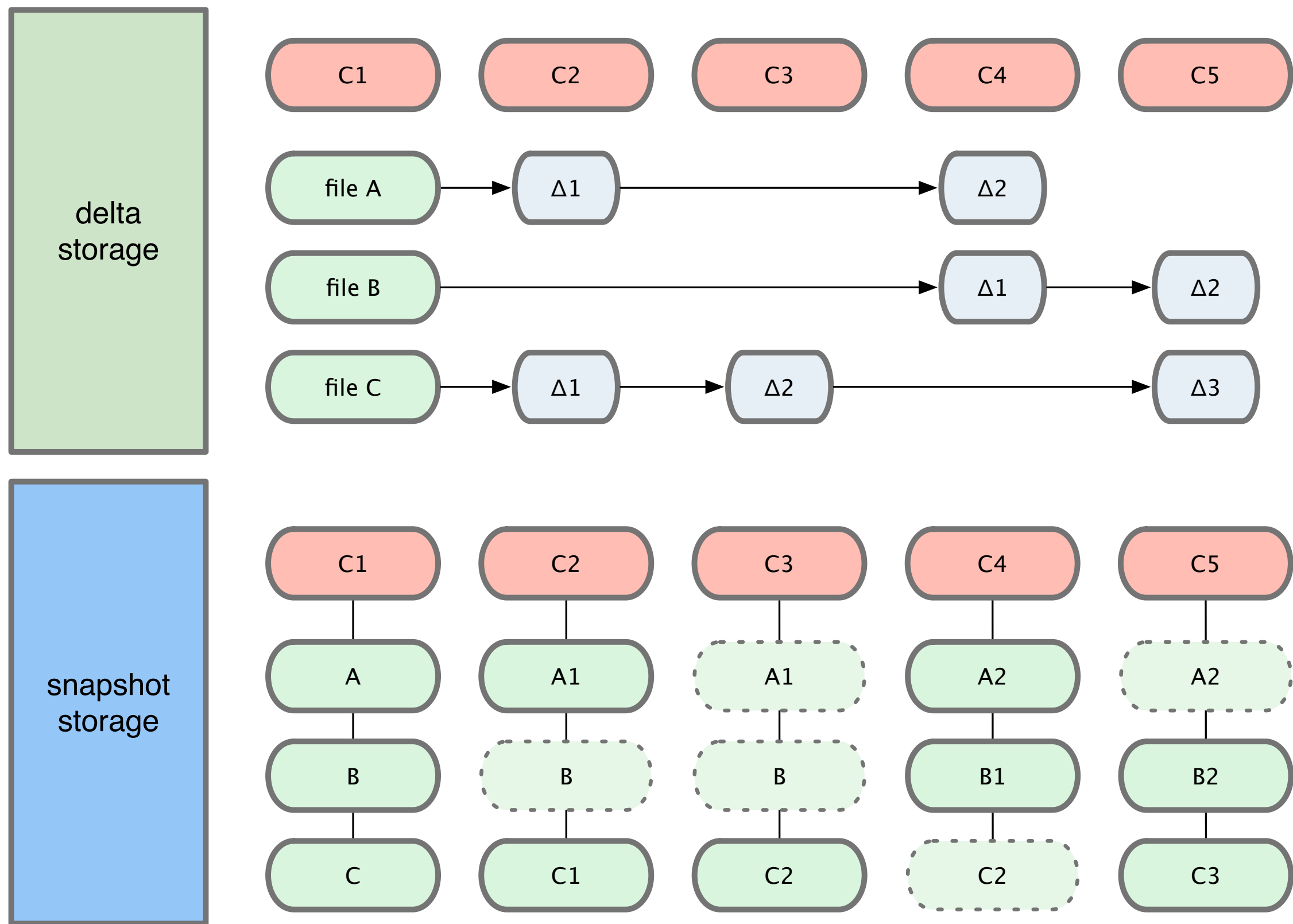
Git Internals

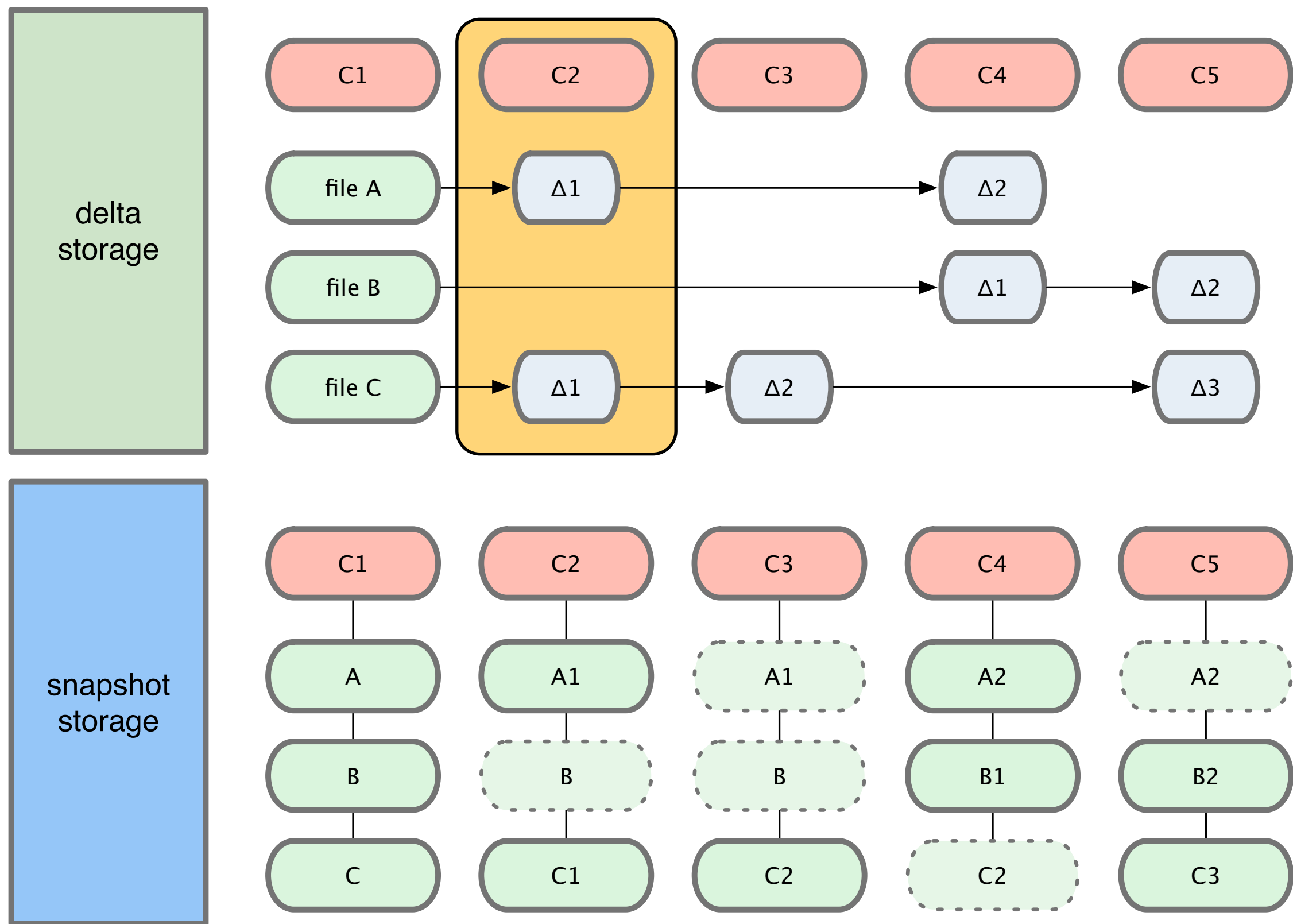
Git Maintenance

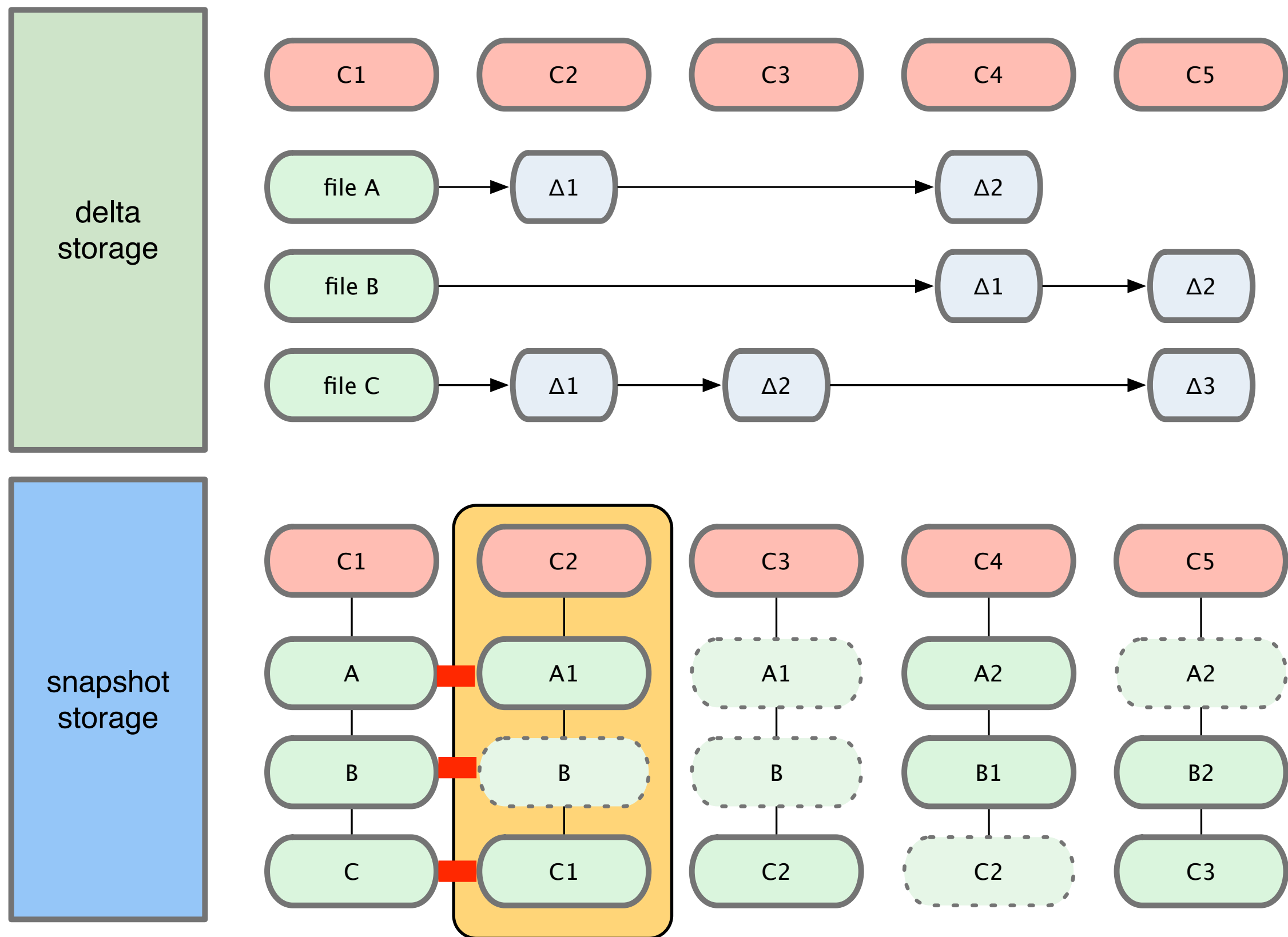
Running a Git Server

# 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):

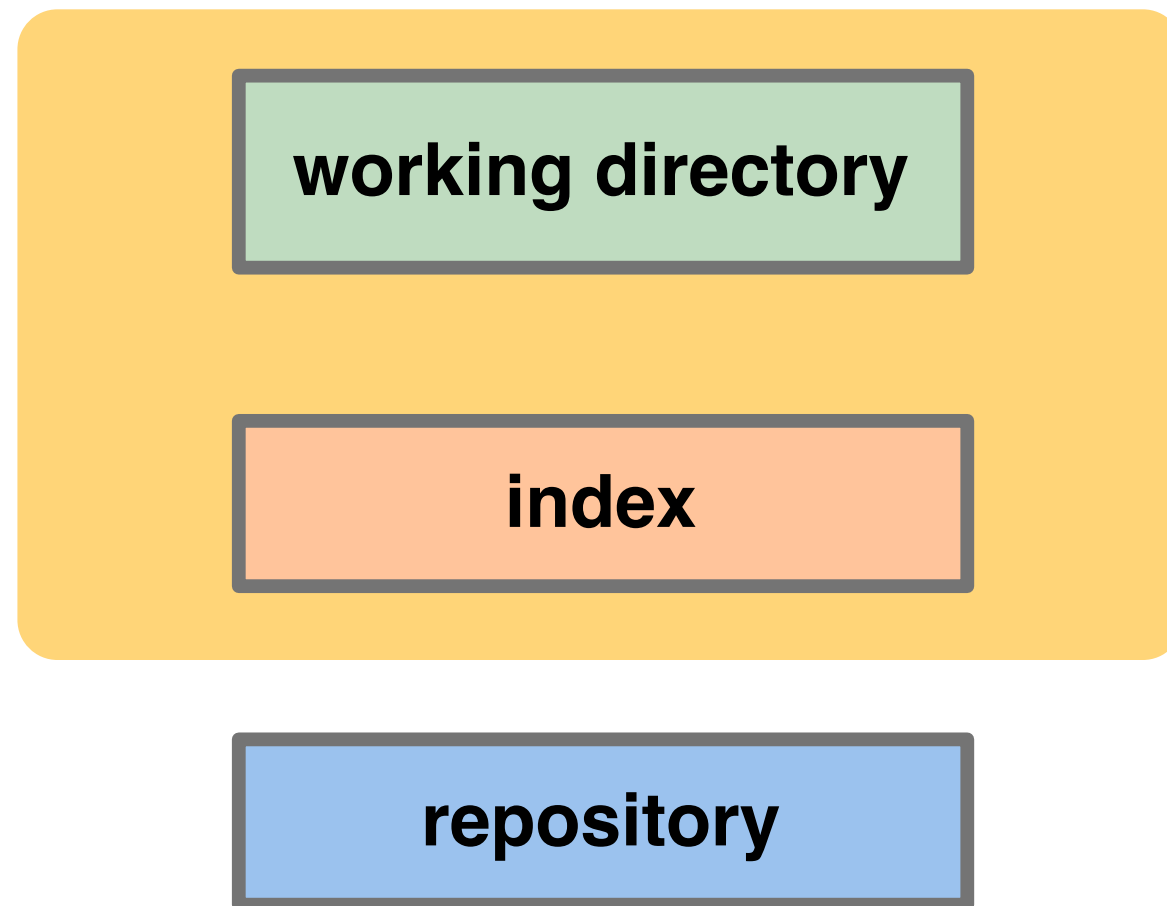
     def get(self):
-        self.response.out.write('Hello world!')
+        self.response.out.write('Hola mundo!')

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

**What is not yet staged?**

**git diff**

# git diff

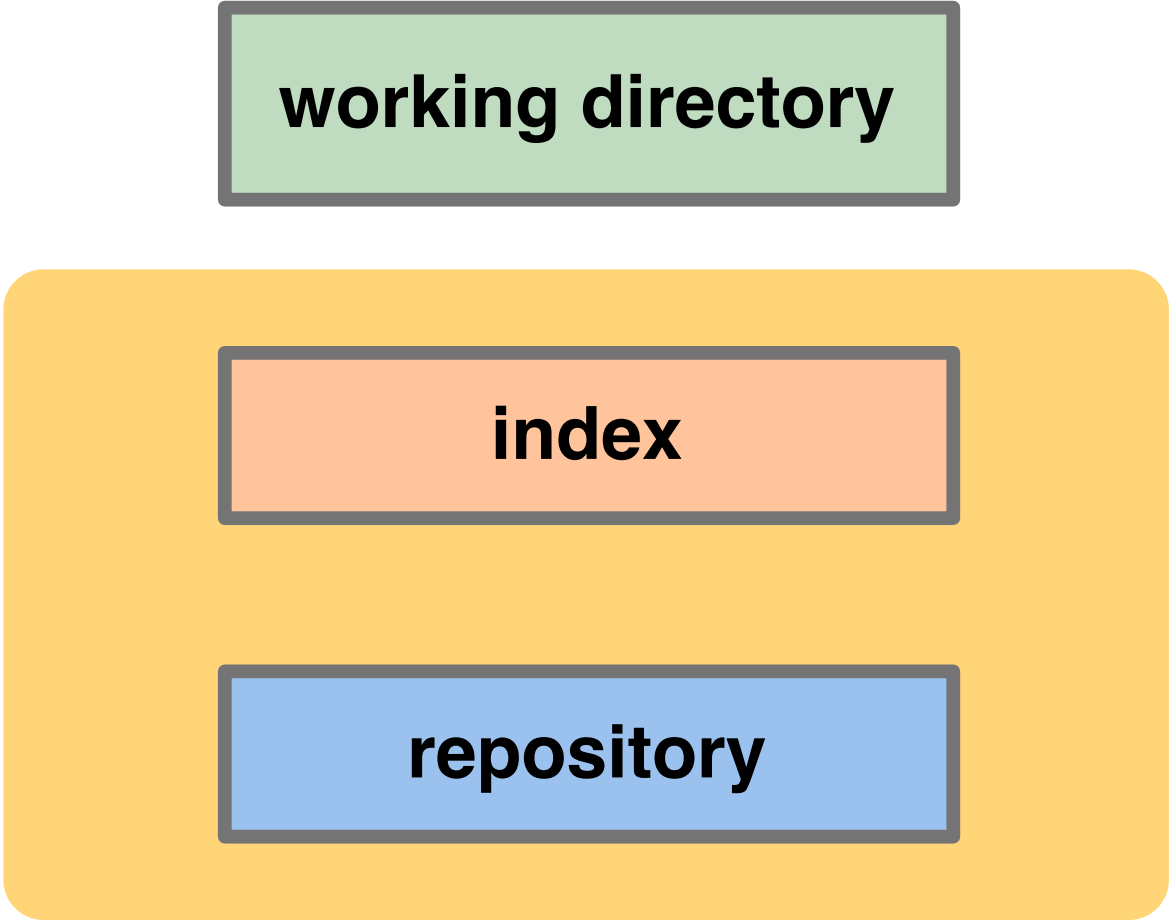




**What is staged?**

```
git diff --cached
```

# `git diff --cached`



A diagram illustrating the components of a Git repository. It features a large yellow rounded rectangle containing two smaller rectangles: an orange one labeled 'index' and a blue one labeled 'repository'. Above this yellow rectangle is a green rectangle labeled 'working directory'.

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 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 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
```

**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

# Merging

# Merge Conflicts

C1

README :

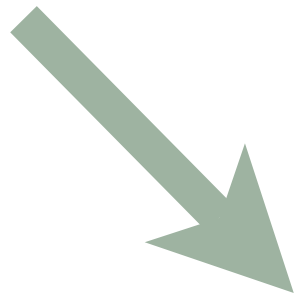
Pets are cute.



C2

README:

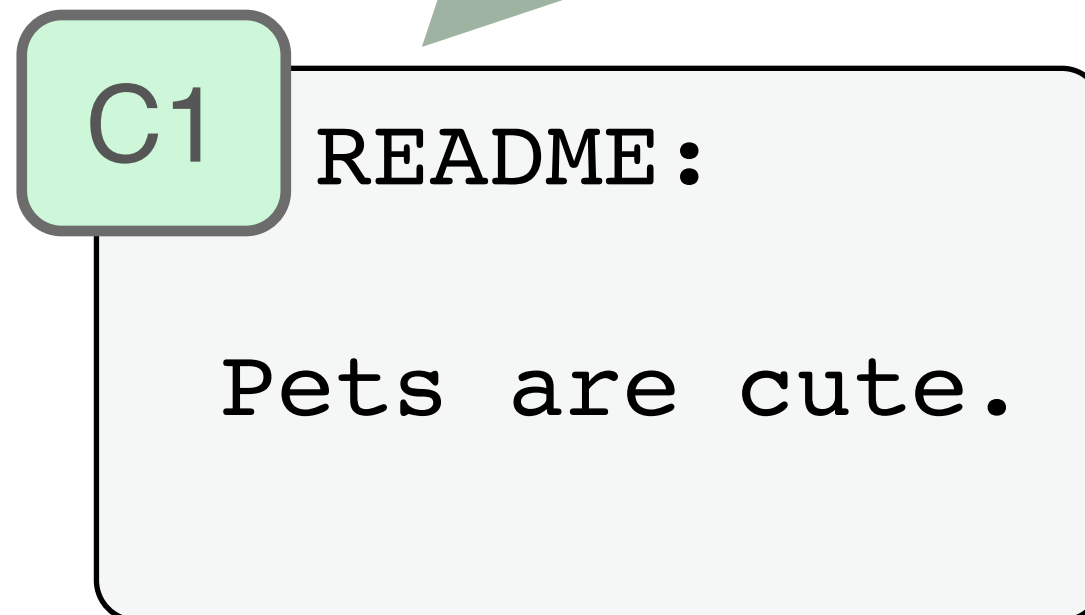
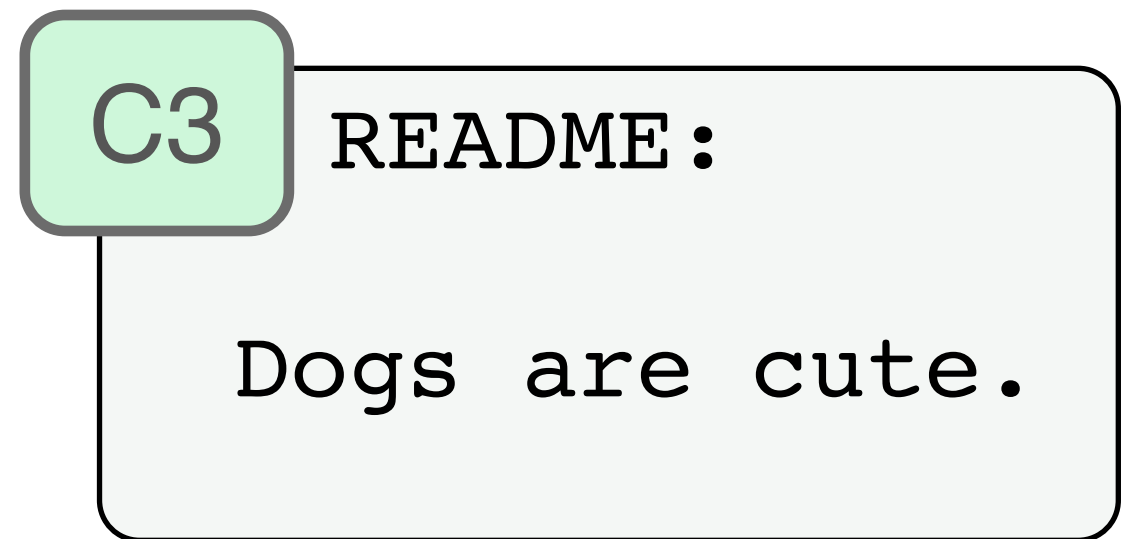
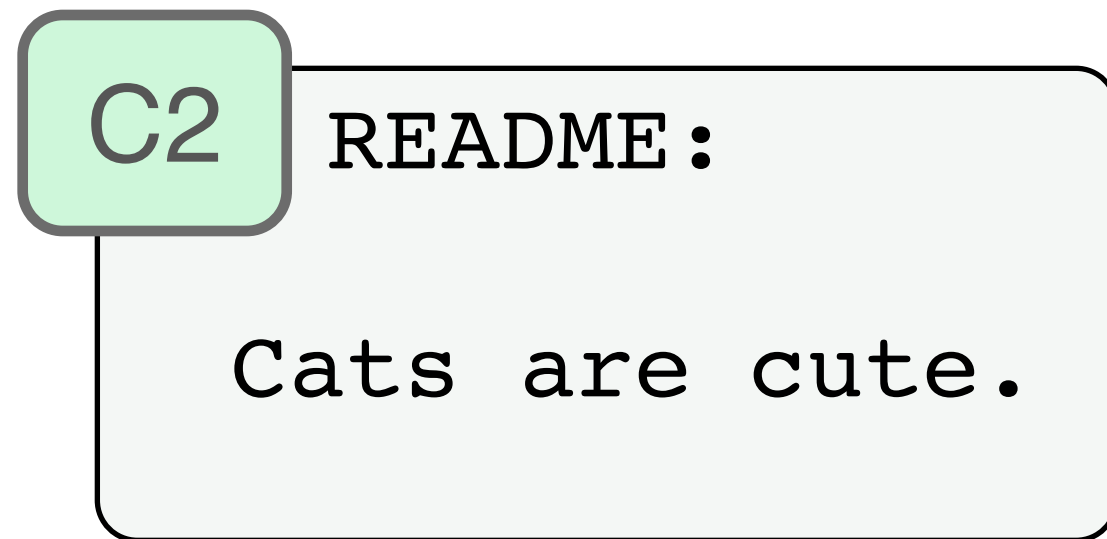
Cats are cute.



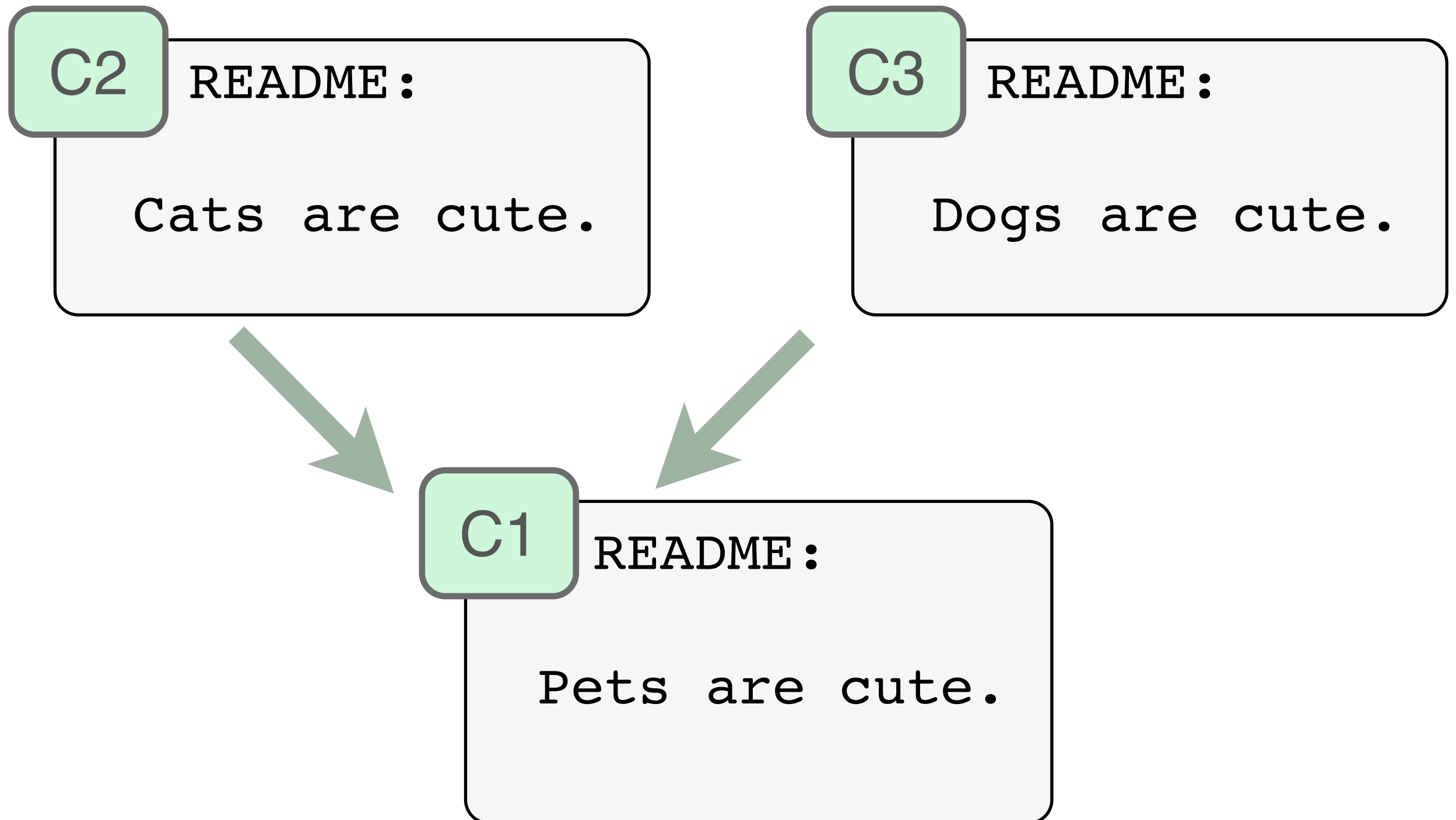
C1

README:

Pets are cute.



# git merge



# git merge

```
$ git merge C2
Auto-merged README
CONFLICT (content): Merge conflict in README
Automatic merge failed; fix conflicts and then commit the
result.
```

```
Pets are cute.
```

# git merge

```
$ git merge C2
```

```
Auto-merged README
```

```
CONFLICT (content): Merge conflict in README
```

```
Automatic merge failed; fix conflicts and then commit the  
result.
```

```
Pets are cute.
```

# git merge

```
$ git diff --merge
```

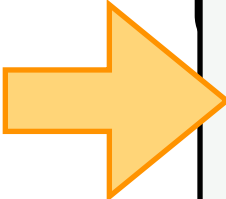

```
Pets are cute.
```

# git merge

```
$ git diff --merge  
diff --cc README  
index ea13f01,997fd23..bd61923  
--- a/README  
+++ b/README  
@@@ -1,1 -1,1 +1,1 @@@  
- Cats are cute.  
-Pets are cute.  
++Dogs are cute.
```

Pets are cute.

# git merge

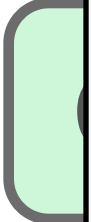


```
$ git diff --merge
diff --cc README
index ea13f01,997fd23..bd61923
--- a/README
+++ b/README
@@ -1,1 -1,1 +1,1 @@
- Cats are cute.
-Pets are cute.
++ Dogs are cute.
```

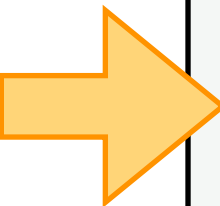
Pets are cute.



# git merge



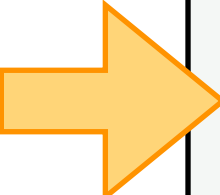
```
$ git diff --merge
diff --cc README
index ea13f01,997fd23..bd61923
--- a/README
+++ b/README
@@ -1,1 -1,1 +1,1 @@
- Cats are cute.
- Pets are cute.
++ Dogs are cute.
```



Pets are cute.

# git merge

```
$ git diff --merge  
diff --cc README  
index ea13f01,997fd23..bd61923  
--- a/README  
+++ b/README  
@@@ -1,1 -1,1 +1,1 @@@  
- Cats are cute.  
-Pets are cute.  
++Dogs are cute.
```



Pets are cute.

```
$ cat README
```

```
<<<<<<< HEAD:README
```

```
Dogs are cute
```

```
=====
```

```
Cats are cute
```

```
>>>>>>> one:README
```

# git merge

```
$ git diff --merge
diff --cc README
index ea13f01,997fd23..bd61923
--- a/README
+++ b/README
@@@ -1,1 -1,1 +1,1 @@@
- Cats are cute.
- Pets are cute.
++Dogs are cute.

$ echo 'Cats are cuter than dogs' > README
```

Pets are cute.

# git merge

```
$ git diff --merge
diff --cc README
index ea13f01,997fd23..bd61923
--- a/README
+++ b/README
@@@ -1,1 -1,1 +1,1 @@@
- Cats are cute.
- Pets are cute.
++Dogs are cute.

$ echo 'Cats are cuter than dogs' > README
$ git add README
```

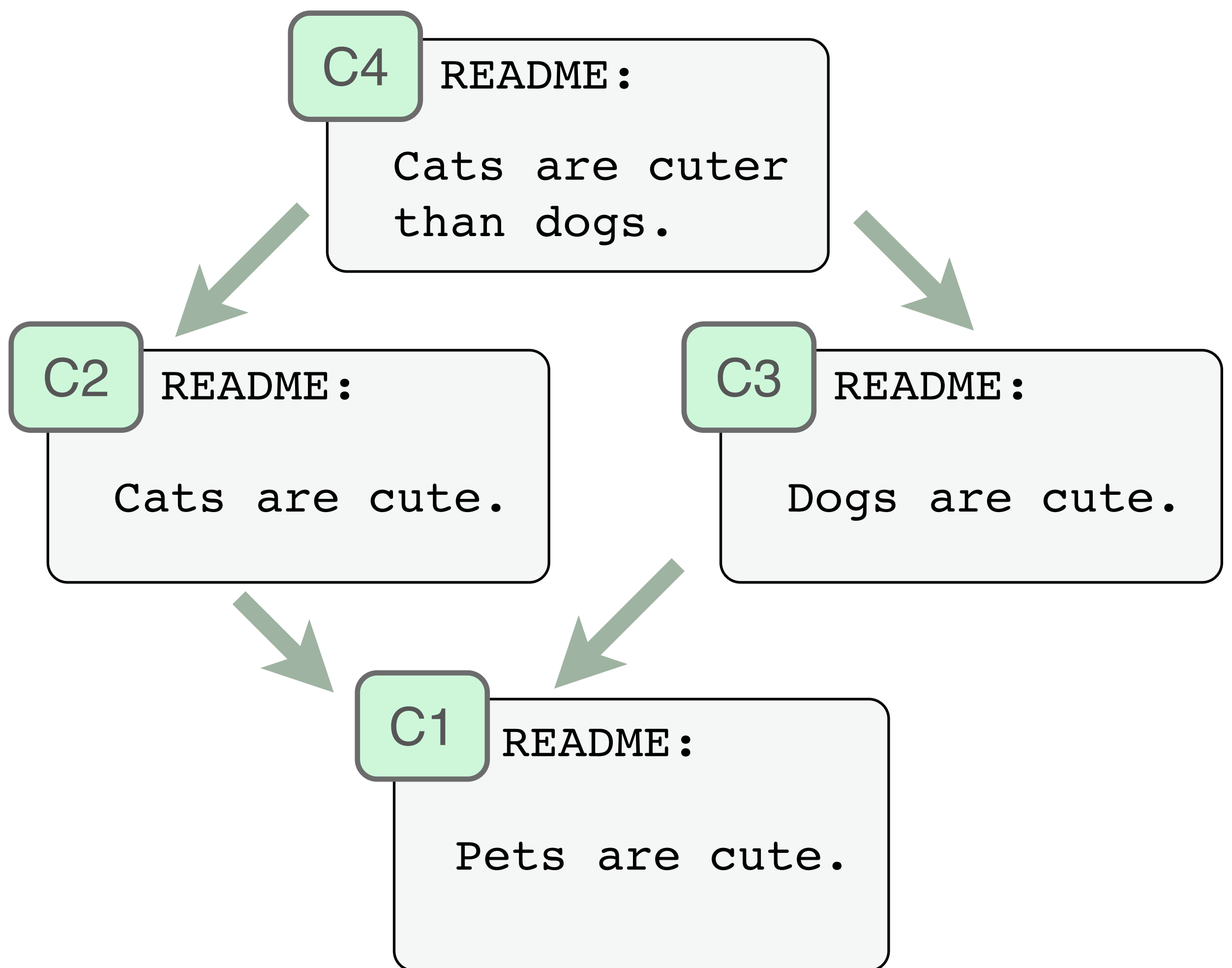
Pets are cute.

# git merge

```
$ git diff --merge
diff --cc README
index ea13f01,997fd23..bd61923
--- a/README
+++ b/README
@@@ -1,1 -1,1 +1,1 @@@
- Cats are cute.
- Pets are cute.
++Dogs are cute.

$ echo 'Cats are cuter than dogs' > README
$ git add README
$ git commit -m 'the truth'
Created commit 274b6ad: the truth
```

Pets are cute.



C4

README :

data one out on

```
$ git show C4
```

C2

e.



C4

## README:

~~Dogs are cute.~~

```
$ git show C4
```

```
commit 274b6ad2d72fe6cc33458662f10b2baad281c215
```

```
Merge: 88b7d11... b752ab5...
```

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

```
Date: Tue Oct 14 21:53:27 2008 -0700
```

```
the truth
```

```
diff --cc README
```

```
index 997fd23,ea13f01..3640d08
```

```
--- a/README
```

```
+++ b/README
```

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

```
- Dogs are cute.
```

```
-Cats are cute.
```

```
++Cats are cuter than dogs.
```

C2

e.

C4

README:

\$ git show C4

commit 274b6ad2d72fe6cc33458662f10b2baad281c215

Merge: 88b7d11... b752ab5...

Author: Scott Chacon <schacon@gmail.com>

Date: Tue Oct 14 21:53:27 2008 -0700

the truth

diff --cc README

index 997fd23,ea13f01..3640d08

--- a/README

+++ b/README

@@@ -1,1 -1,1 +1,1 @@@

- Dogs are cute.

-Cats are cute.

++Cats are cuter than dogs.

C2

e.

C4

README:

Dogs are cuter

```
$ git show C4
```

```
commit 274b6ad2d72fe6cc33458662f10b2baad281c215
```

```
Merge: 88b7d11... b752ab5...
```

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

```
Date: Tue Oct 14 21:53:27 2008 -0700
```

```
the truth
```

```
diff --cc README
```

```
index 997fd23,ea13f01..3640d08
```

```
--- a/README
```

```
+++ b/README
```

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

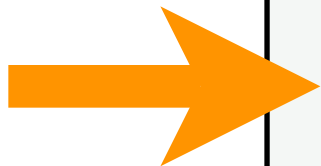
```
- Dogs are cute.
```

```
-Cats are cute.
```

```
++Cats are cuter than dogs.
```

C2

e.



C4

README:

\$ git show C4

commit 274b6ad2d72fe6cc33458662f10b2baad281c215

Merge: 88b7d11... b752ab5...

Author: Scott Chacon <schacon@gmail.com>

Date: Tue Oct 14 21:53:27 2008 -0700

the truth

diff --cc README

index 997fd23,ea13f01..3640d08

--- a/README

+++ b/README

@@@ -1,1 -1,1 +1,1 @@@

- Dogs are cute.

-Cats are cute.

++Cats are cuter than dogs.

C2

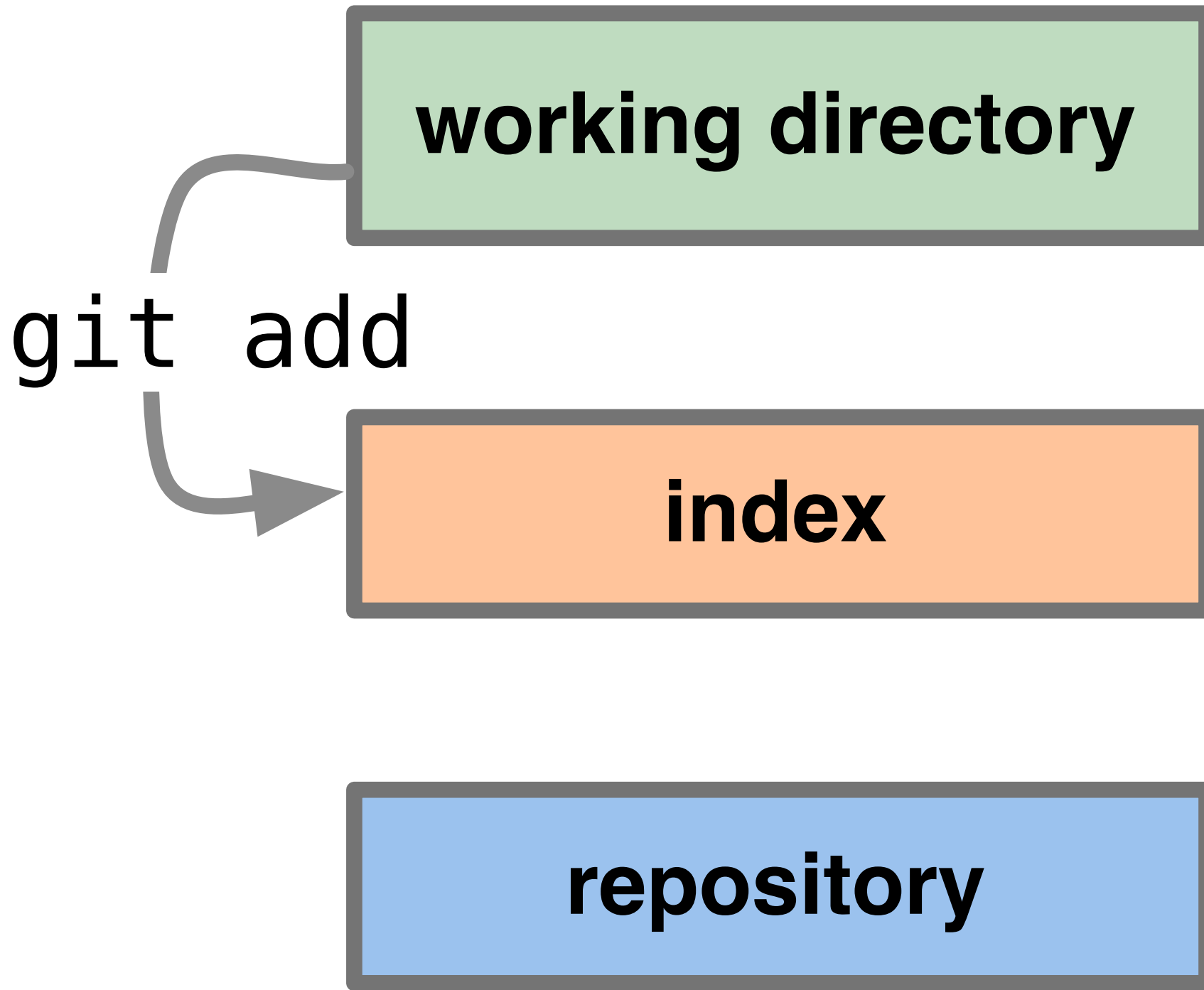
e.

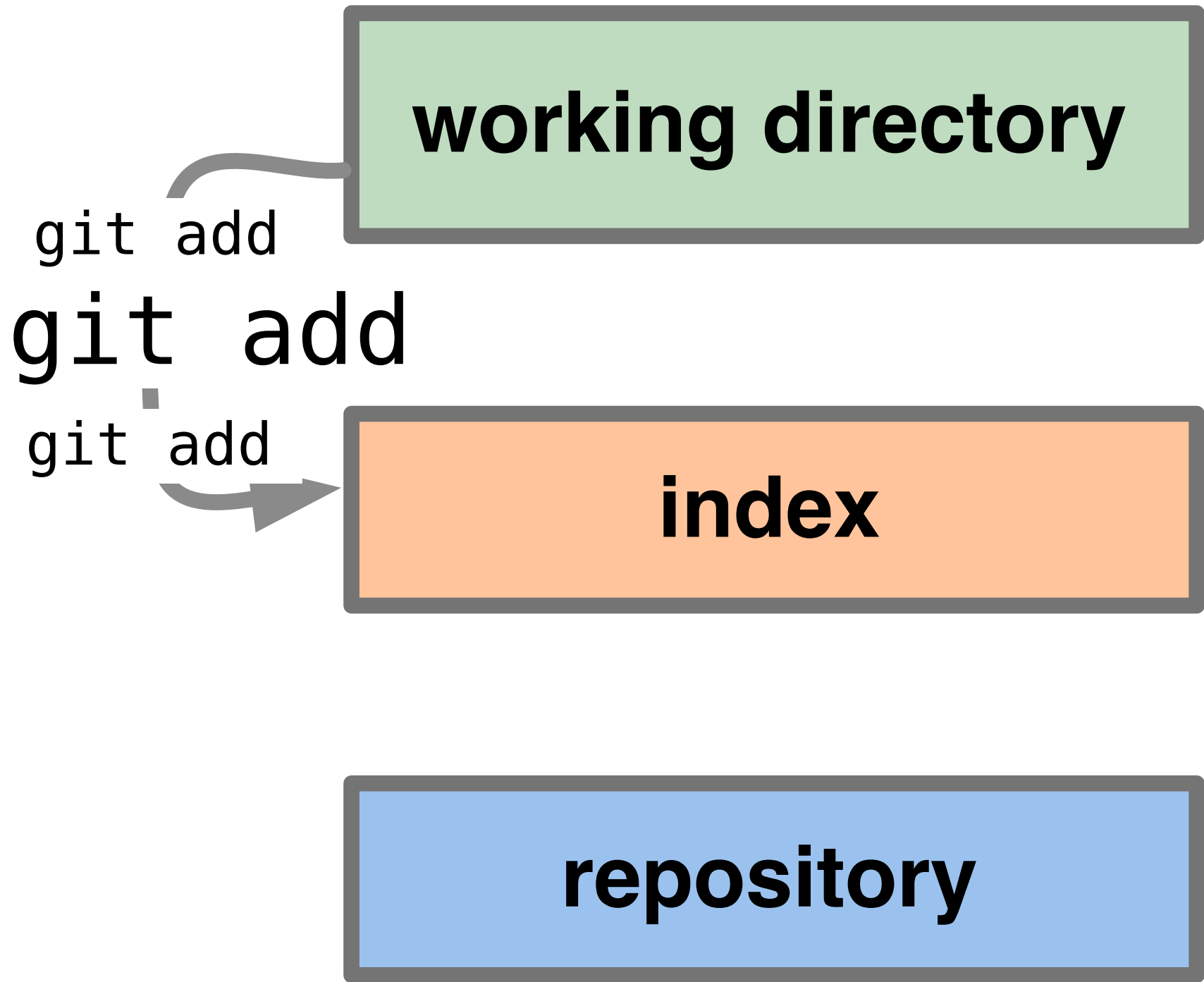
# Bulk Staging

**working directory**

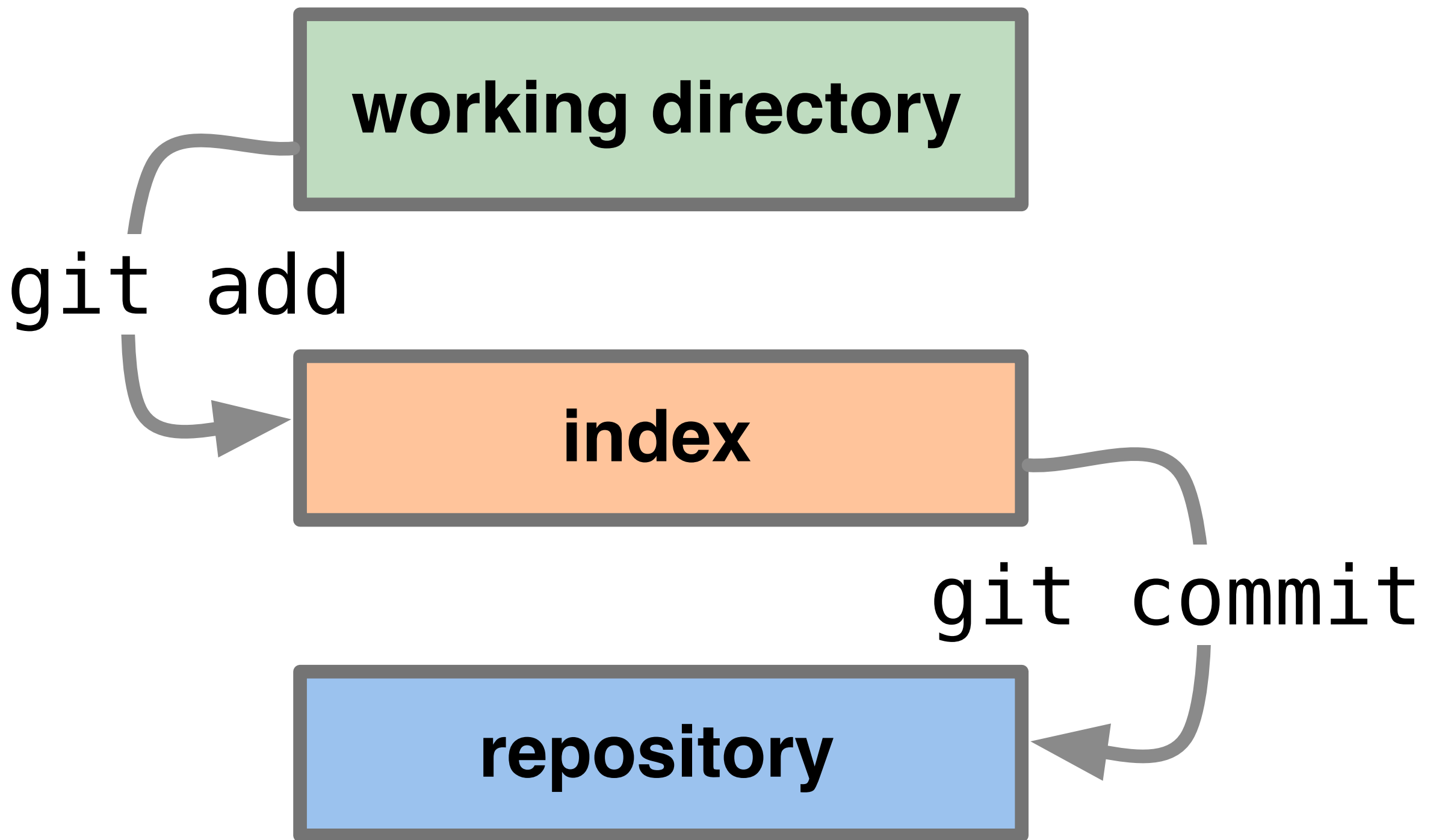
**index**

**repository**









```
git add -u
```

```
git commit -a
```

**working directory**

**index**

**repository**

`git commit -a`

`git commit -a`

`git add -u`  
`git commit`

# Interactive Staging

```
git add -i
```

git add -i

git add --interactive



```
$ git status
```

```
$ git status
# On branch master
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#
#       modified:   commit.c
#       modified:   commit.h
#       modified:   http-push.c
#       modified:   http-walker.c
#       modified:   http.c
#       modified:   imap-send.c
#       modified:   walker.c
#       modified:   wt-status.c
#       modified:   xdiff-interface.c
#       modified:   xdiff-interface.h
#       modified:   xdiff/xdiff.h
#       modified:   xdiff/xmerge.c
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git status
# On branch master
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#
#       modified:   commit.c
#       modified:   commit.h
#       modified:   http-push.c
#       modified:   http-view.c
#       modified:   http.c
#       modified:   image-send.c
#       modified:   walker.c
#       modified:   wt-status.c
#       modified:   xdiff-interface.c
#       modified:   xdiff-interface.h
#       modified:   xdiff/xdiff.h
#       modified:   xdiff/xmerge.c
#
no changes added to commit (use "git add" and/or "git commit -a")
```

**UNSTAGED**

```
$ git add -i
```

```
$ git add -i
```

	staged	unstaged	path
1:	unchanged	+1/-1	commit.c
2:	unchanged	+1/-1	commit.h
3:	unchanged	+6/-6	http-push.c
4:	unchanged	+5/-5	http-walker.c
5:	unchanged	+1/-1	http.c
6:	unchanged	+513/-412	imap-send.c
7:	unchanged	+1/-2	walker.c
8:	unchanged	+23/-22	wt-status.c
9:	unchanged	+77/-32	xdiff-interface.c
10:	unchanged	+6/-9	xdiff-interface.h
11:	unchanged	+6/-0	xdiff/xdiff.h
12:	unchanged	+165/-72	xdiff/xmerge.c

```
*** Commands ***
```

1: status	2: update	3: revert	4: add untracked
5: patch	6: diff	7: quit	8: help

```
What now>
```

```
$ git add -i
```

	staged	unstaged	path
1:	unchanged	+1/-1	commit.c
2:	unchanged	+1/-1	commit.h
3:	unchanged	+6/-6	http-push.c
4:	unchanged	+5/-5	http-walker.c
5:	unchanged	+1/-1	http.c
6:	unchanged	+513/-412	imap-send.c
7:	unchanged	+1/-2	walker.c
8:	unchanged	+23/-22	wt-status.c
9:	unchanged	+77/-32	xdiff-interface.c
10:	unchanged	+6/-9	xdiff-interface.h
11:	unchanged	+6/-0	xdiff/xdiff.h
12:	unchanged	+165/-72	xdiff/xmerge.c

```
*** Commands ***
```

1: status	2: update	3: revert	4: add untracked
5: patch	6: diff	7: quit	8: help

```
What now>
```

```
$ git add -i
```

	staged	unstaged	path
1:	unchanged	+1/-1	commit.c
2:	unchanged	+1/-1	commit.h
3:	unchanged	+6/-6	http-push.c
4:	unchanged	+5/-5	http-walker.c
5:	unchanged	+1/-1	http.c
6:	unchanged	+513/-412	imap-send.c
7:	unchanged	+1/-2	walker.c
8:	unchanged	+23/-22	wt-status.c
9:	unchanged	+77/-32	xdiff-interface.c
10:	unchanged	+6/-9	xdiff-interface.h
11:	unchanged	+6/-0	xdiff/xdiff.h
12:	unchanged	+165/-72	xdiff/xmerge.c

```
*** Commands ***
```

1: status	2: update	3: revert	4: add untracked
5: patch	6: diff	7: quit	8: help

```
What now>
```

```
$ git add -i
```

	staged	unstaged	path
1:	unchanged	+1/-1	commit.c
2:	unchanged	+1/-1	commit.h
3:	unchanged	+6/-6	http-push.c
4:	unchanged	+5/-5	http-walker.c
5:	unchanged	+1/-1	http.c
6:	unchanged	+513/-412	imap-send.c
7:	unchanged	+1/-2	walker.c
8:	unchanged	+23/-22	wt-status.c
9:	unchanged	+77/-32	xdiff-interface.c
10:	unchanged	+6/-9	xdiff-interface.h
11:	unchanged	+6/-0	xdiff/xdiff.h
12:	unchanged	+165/-72	xdiff/xmerge.c

\*\*\* Commands \*\*\*

1: status	2: update	3: revert	4: add untracked
5: patch	6: diff	7: quit	8: help

What now>



```
$ git add -i
```

	staged	unstaged	path
1:	unchanged	+1/-1	commit.c
2:	unchanged	+1/-1	commit.h
3:	unchanged	+6/-6	http-push.c
4:	unchanged	+5/-5	http-walker.c
5:	unchanged	+1/-1	http.c
6:	unchanged	+513/-412	imap-send.c
7:	unchanged	+1/-2	walker.c
8:	unchanged	+23/-22	wt-status.c
9:	unchanged	+77/-32	xdiff-interface.c
10:	unchanged	+6/-9	xdiff-interface.h
11:	unchanged	+6/-0	xdiff/xdiff.h
12:	unchanged	+165/-72	xdiff/xmerge.c

```
*** Commands ***
```

1: status	2: update	3: revert	4: add untracked
5: patch	6: diff	7: quit	8: help

```
What now>
```

\*\*\* Commands \*\*\*

1: status	2: update	3: revert	4: add untracked
5: patch	6: diff	7: quit	8: help

What now> 2

	staged	unstaged	path
1:	unchanged	+1/-1	commit.c
2:	unchanged	+1/-1	commit.h
3:	unchanged	+6/-6	http-push.c
4:	unchanged	+5/-5	http-walker.c
5:	unchanged	+1/-1	http.c
6:	unchanged	+513/-412	imap-send.c
7:	unchanged	+1/-2	walker.c
8:	unchanged	+23/-22	wt-status.c
9:	unchanged	+77/-32	xdiff-interface.c
10:	unchanged	+6/-9	xdiff-interface.h
11:	unchanged	+6/-0	xdiff/xdiff.h
12:	unchanged	+165/-72	xdiff/xmerge.c

Update>>

\*\*\* Commands \*\*\*

1: status	2: update	3: revert	4: add untracked
5: patch	6: diff	7: quit	8: help

What now> 2

	staged	unstaged	path
1:	unchanged	+1/-1	commit.c
2:	unchanged	+1/-1	commit.h
3:	unchanged	+6/-6	http-push.c
4:	unchanged	+5/-5	http-walker.c
5:	unchanged	+1/-1	http.c
6:	unchanged	+513/-412	imap-send.c
7:	unchanged	+1/-2	walker.c
8:	unchanged	+23/-22	wt-status.c
9:	unchanged	+77/-32	xdiff-interface.c
10:	unchanged	+6/-9	xdiff-interface.h
11:	unchanged	+6/-0	xdiff/xdiff.h
12:	unchanged	+165/-72	xdiff/xmerge.c

Update>>

\*\*\* Commands \*\*\*

1: status	2: update	3: revert	4: add untracked
5: patch	6: diff	7: quit	8: help

What now> 2

	staged	unstaged	path
1:	unchanged	+1/-1	commit.c
2:	unchanged	+1/-1	commit.h
3:	unchanged	+6/-6	http-push.c
4:	unchanged	+5/-5	http-walker.c
5:	unchanged	+1/-1	http.c
6:	unchanged	+513/-412	imap-send.c
7:	unchanged	+1/-2	walker.c
8:	unchanged	+23/-22	wt-status.c
9:	unchanged	+77/-32	xdiff-interface.c
10:	unchanged	+6/-9	xdiff-interface.h
11:	unchanged	+6/-0	xdiff/xdiff.h
12:	unchanged	+165/-72	xdiff/xmerge.c

Update>> 9-12

Update>> 9-12

	staged	unstaged	path
1:	unchanged	+1/-1	commit.c
2:	unchanged	+1/-1	commit.h
3:	unchanged	+6/-6	http-push.c
4:	unchanged	+5/-5	http-walker.c
5:	unchanged	+1/-1	http.c
6:	unchanged	+513/-412	imap-send.c
7:	unchanged	+1/-2	walker.c
8:	unchanged	+23/-22	wt-status.c
* 9:	unchanged	+77/-32	xdiff-interface.c
*10:	unchanged	+6/-9	xdiff-interface.h
*11:	unchanged	+6/-0	xdiff/xdiff.h
*12:	unchanged	+165/-72	xdiff/xmerge.c

Update>>

Update>> 9-12

	staged	unstaged	path
1:	unchanged	+1/-1	commit.c
2:	unchanged	+1/-1	commit.h
3:	unchanged	+6/-6	http-push.c
4:	unchanged	+5/-5	http-walker.c
5:	unchanged	+1/-1	http.c
6:	unchanged	+513/-412	imap-send.c
7:	unchanged	+1/-2	walker.c
8:	unchanged	+23/-22	wt-status.c
* 9:	unchanged	+77/-32	xdiff-interface.c
* 10:	unchanged	+6/-9	xdiff-interface.h
* 11:	unchanged	+6/-0	xdiff/xdiff.h
* 12:	unchanged	+165/-72	xdiff/xmerge.c

Update>>

Update>> 9-12

	staged	unstaged	path
1:	unchanged	+1/-1	commit.c
2:	unchanged	+1/-1	commit.h
3:	unchanged	+6/-6	http-push.c
4:	unchanged	+5/-5	http-walker.c
5:	unchanged	+1/-1	http.c
6:	unchanged	+513/-412	imap-send.c
7:	unchanged	+1/-2	walker.c
8:	unchanged	+23/-22	wt-status.c
* 9:	unchanged	+77/-32	xdiff-interface.c
*10:	unchanged	+6/-9	xdiff-interface.h
*11:	unchanged	+6/-0	xdiff/xdiff.h
*12:	unchanged	+165/-72	xdiff/xmerge.c

Update>> 2,5

Update>> 2,5

	staged	unstaged	path
1:	unchanged	+1/-1	commit.c
* 2:	unchanged	+1/-1	commit.h
3:	unchanged	+6/-6	http-push.c
4:	unchanged	+5/-5	http-walker.c
* 5:	unchanged	+1/-1	http.c
6:	unchanged	+513/-412	imap-send.c
7:	unchanged	+1/-2	walker.c
8:	unchanged	+23/-22	wt-status.c
* 9:	unchanged	+77/-32	xdiff-interface.c
*10:	unchanged	+6/-9	xdiff-interface.h
*11:	unchanged	+6/-0	xdiff/xdiff.h
*12:	unchanged	+165/-72	xdiff/xmerge.c

Update>>



Update>> 2,5

	staged	unstaged	path
1:	unchanged	+1/-1	commit.c
* 2:	unchanged	+1/-1	commit.h
3:	unchanged	+6/-6	http-push.c
4:	unchanged	+5/-5	http-walker.c
* 5:	unchanged	+1/-1	http.c
6:	unchanged	+513/-412	imap-send.c
7:	unchanged	+1/-2	walker.c
8:	unchanged	+23/-22	wt-status.c
* 9:	unchanged	+77/-32	xdiff-interface.c
*10:	unchanged	+6/-9	xdiff-interface.h
*11:	unchanged	+6/-0	xdiff/xdiff.h
*12:	unchanged	+165/-72	xdiff/xmerge.c

Update>>

Update>>

updated 6 paths

\*\*\* Commands \*\*\*

1: status	2: update	3: revert	4: add untracked
5: patch	6: diff	7: quit	8: help

What now> 1

	staged	unstaged	path
1:	unchanged	+1/-1	commit.c
2:	+1/-1	nothing	commit.h
3:	unchanged	+6/-6	http-push.c
4:	unchanged	+5/-5	http-walker.c
5:	+1/-1	nothing	http.c
6:	unchanged	+513/-412	imap-send.c
7:	unchanged	+1/-2	walker.c
8:	unchanged	+23/-22	wt-status.c
9:	+77/-32	nothing	xdiff-interface.c
10:	+6/-9	nothing	xdiff-interface.h
11:	+6/-0	nothing	xdiff/xdiff.h
12:	+165/-72	nothing	xdiff/xmerge.c

\*\*\* Commands \*\*\*

1: status	2: update	3: revert	4: add untracked
5: patch	6: diff	7: quit	8: help

What now>

Update>>  
updated 6 paths

\*\*\* Commands \*\*\*

1: status	2: update	3: revert	4: add untracked
5: patch	6: diff	7: quit	8: help

What now> 1

	staged	unstaged	path
1:	unchanged	+1/-1	commit.c
2:	+1/-1	nothing	commit.h
3:	unchanged	+6/-6	http-push.c
4:	unchanged	+5/-5	http-walker.c
5:	+1/-1	nothing	http.c
6:	unchanged	+513/-412	imap-send.c
7:	unchanged	+1/-2	walker.c
8:	unchanged	+23/-22	wt-status.c
9:	+77/-32	nothing	xdiff-interface.c
10:	+6/-9	nothing	xdiff-interface.h
11:	+6/-0	nothing	xdiff/xdiff.h
12:	+165/-72	nothing	xdiff/xmerge.c

\*\*\* Commands \*\*\*

1: status	2: update	3: revert	4: add untracked
5: patch	6: diff	7: quit	8: help

What now>

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:   commit.h
# modified:   http.c
# modified:   xdiff-interface.c
# modified:   xdiff-interface.h
# modified:   xdiff/xdiff.h
# modified:   xdiff/xmerge.c
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#
# modified:   commit.c
# modified:   http-push.c
# modified:   http-walker.c
# modified:   imap-send.c
# modified:   walker.c
# modified:   wt-status.c
#
~
~
".git/COMMIT_EDITMSG" 27L, 757C
```

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
# modified:   commit.h
# modified:   http.c
# modified:   xdiff-interface.c
# modified:   xdiff-interface.h
# modified:   xdiff/xdiff.h
# modified:   xdiff/xmerge.c
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#
# modified:   commit.c
# modified:   http-push.c
# modified:   http-walker.c
# modified:   imap-send.c
# modified:   walker.c
# modified:   wt-status.c
#
~
~
".git/COMMIT_EDITMSG" 27L, 757C
```

# Configuring Git

```
$ 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

**/etc/gitconfig**



`/etc/gitconfig`

`~/.gitconfig`

/etc/gitconfig

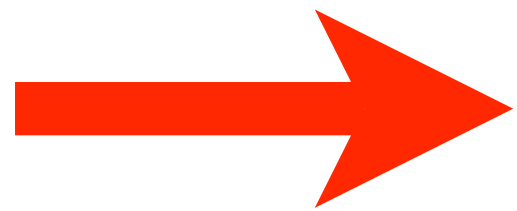
~/.gitconfig

.git/config

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

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

/etc/gitconfig



~/.gitconfig

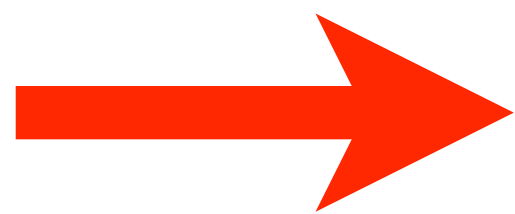
.git/config

```
$ cat ~/.gitconfig  
[user]  
    name = Scott Chacon  
    email = schacon@gmail.com
```

```
$ git config user.email schacon@qualcomm.com
```

/etc/gitconfig

~/.gitconfig



**.git/config**

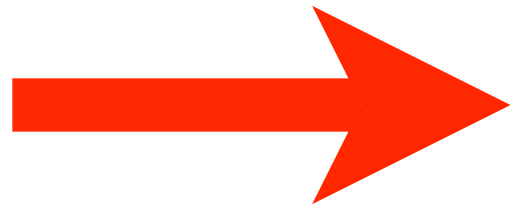
```
$ git config --list
```



```
$ git config --list  
user.name=Scott Chacon  
user.email=schacon@gmail.com  
user.email=schacon@qualcomm.com
```

```
$ git config --list  
user.name=Scott Chacon  
user.email=schacon@gmail.com  
user.email=schacon@qualcomm.com
```

```
$ git config --system user.name "Qualcomm"
```



**/etc/gitconfig**

~/.gitconfig

.git/config

# Review

git init

git clone

git add

git commit

git remote

git config

git pull

git fetch

git push

git diff

git log

11

Demo

# Git Help

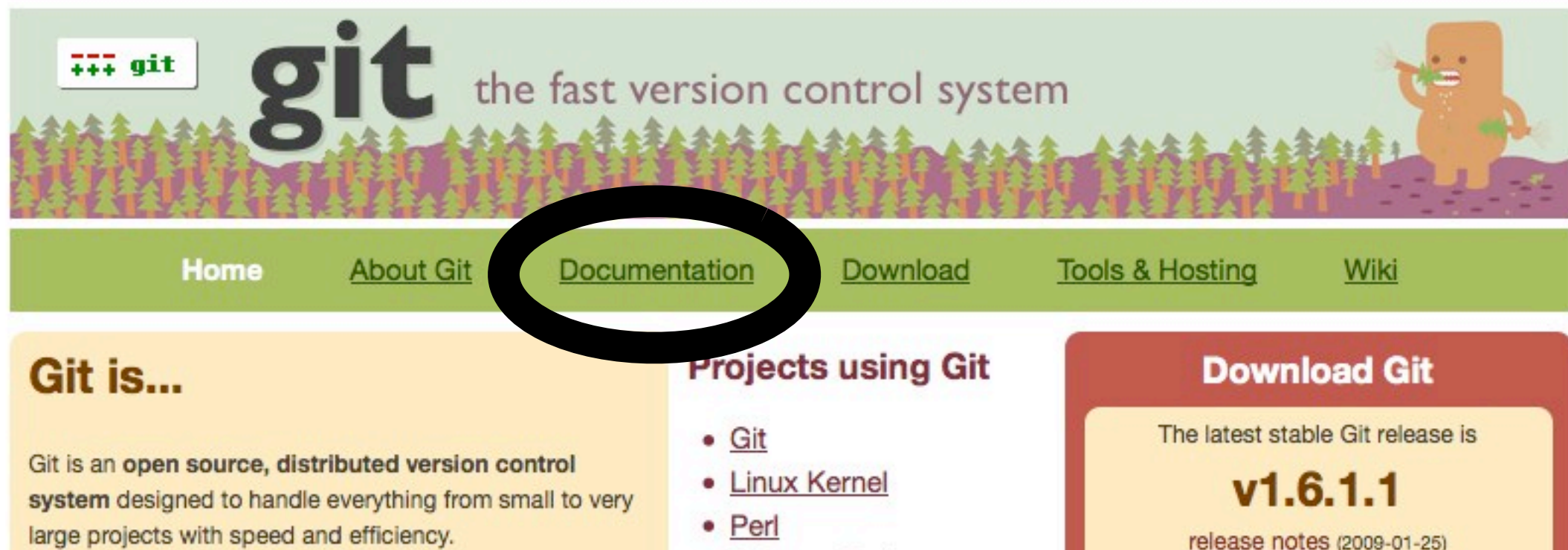



**git [command] --help**

**git help [command]**

**git-scm.com**

# git-scm.com



 [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

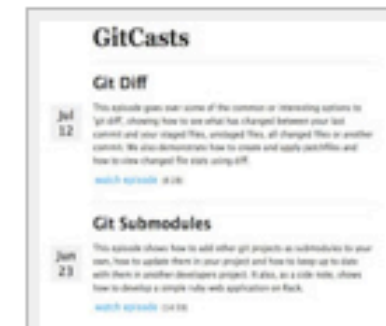
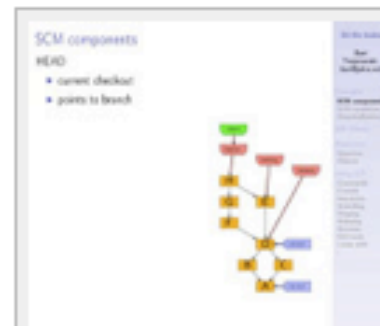


[Git Internals PDF](#)  
by Scott Chacon




[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

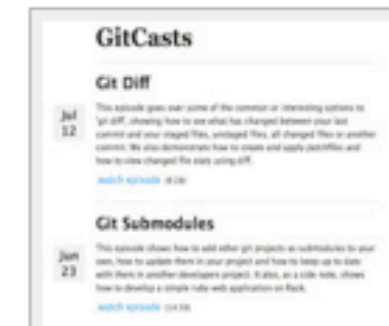
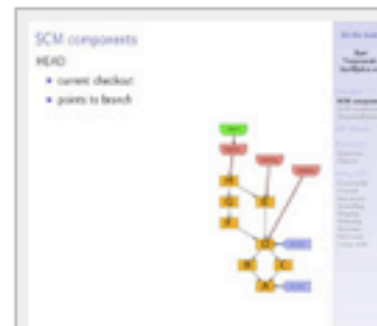



[Git Internals PDF](#)  
by Scott Chacon



[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

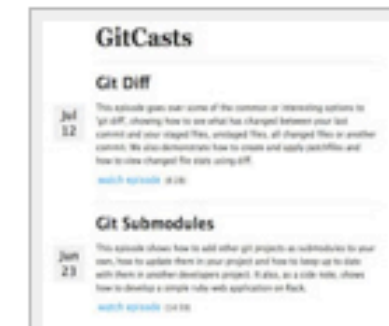
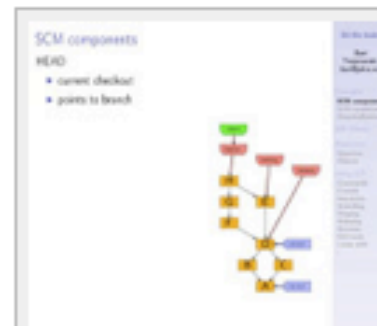


[Git Internals PDF](#)  
by Scott Chacon




[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

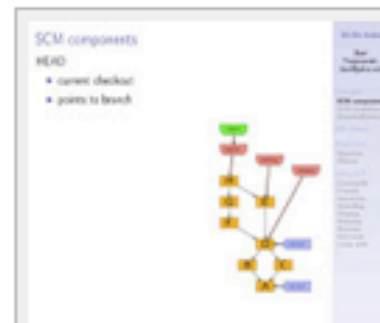



[Git Internals PDF](#)  
by Scott Chacon



[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

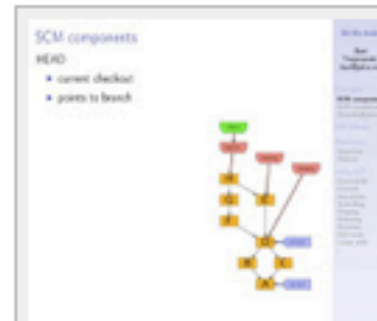



[Git Internals PDF](#)  
by Scott Chacon



[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

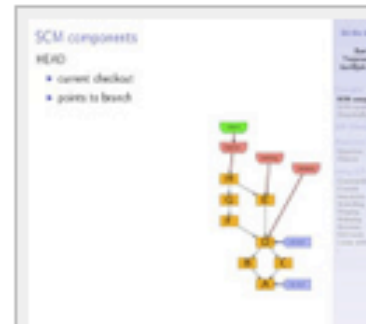


[Git Internals PDF](#)  
by Scott Chacon



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

## Videos



# questions

and possibly answers...

# break

15 minutes